Neurocomputing 127 (2014) 231–246

Contents lists available at ScienceDirect

Neurocomputing

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

A process monitoring method based on noisy independent component analysis ☆

Lianfang Cai [a], Xuemin Tian [a], Sheng Chen [b,c,*]

[a] College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China
[b] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[c] Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 4 January 2013
Received in revised form
23 May 2013
Accepted 22 July 2013
Communicated by J. Zhang
Available online 25 September 2013

Keywords:
Process monitoring
Independent component analysis
Kurtosis
Fourth order cumulant
Fault detection time
Fault detection rate

ABSTRACT

Independent component analysis (ICA) is an effective feature extraction tool for process monitoring. However, the conventional ICA-based process monitoring methods usually adopt noise-free ICA models and thus may perform unsatisfactorily under the adverse effects of the measurement noise. In this paper, a process monitoring method using a new noisy independent component analysis, referred to as NoisyICAn, is proposed. Using the noisy ICA model which considers the measurement noise explicitly, a NoisyICAn algorithm is developed to estimate the mixing matrix by setting up a series of the fourth-order cumulant matrices of the measured data and performing the joint diagonalization of these matrices. The kurtosis relationships of the independent components and measured variables are subsequently obtained based on the estimated mixing matrix, for recursively estimating the kurtosis of independent components. Two monitoring statistics are then built to detect process faults using the obtained recursive estimate of the independent components' kurtosis. The simulation studies are carried out on a simple three-variable system and a continuous stirred tank reactor system, and the results obtained demonstrate that the proposed NoisyICAn-based monitoring method outperforms the conventional noise-free ICA-based monitoring methods as well as the benchmark monitoring methods based on the existing noisy ICA schemes adopted from blind source separation, in terms of the fault detection time and local fault detection rate.

1. Introduction

Efficient and reliable process monitoring plays an extremely important role in ensuring plant safety and product quality. As a large number of process variables are measured in industrial processes, multivariate statistical process monitoring (MSPM) approaches have attracted great attentions from both academic researchers and process engineers [1–3]. Among them, principal component analysis (PCA) is a classical method that has been widely studied in the process monitoring field [4–6]. Classical PCA can handle high-dimensional and highly correlated data by projecting them onto a low-dimensional subspace that contains the most variance of the measured data. However, PCA only considers up to the second-order statistics of the measured data to extract uncorrelated latent variables, and it lacks the ability to provide higher-order representations for the measured data [7]. Moreover, in PCA-based monitoring methods, the control limits of Hotelling's T-squared and the companion squared prediction error (SPE) statistics are determined based on the assumption that the extracted latent variables follow a multivariate Gaussian distribution. However, industrial data often obey non-Gaussian distributions, for which PCA-based monitoring methods are ill-suited [8]. More recently, a MSPM method, known as independent component analysis (ICA), has emerged as a powerful tool in process monitoring. It originates from the blind source separation problem and has found wide-ranging applications in many areas, including signal processing, telecommunications, and audio signal separation [9]. Different from PCA, ICA takes into account the higher-order statistics in recovering the mutually independent latent variables, called independent components (ICs), from the measured variables. The specified merits of ICA endow it with the ability to reveal more useful information from the measured data than PCA [10].

Because of its favorable performance in information extraction, many researchers have implemented ICA for monitoring process behaviours. Kano et al. [11] directly monitored the ICs obtained from an ICA algorithm and demonstrated the superiority of ICA

☆This work is supported by the National Natural Science Foundation of China (Grant no. 61273160), the Natural Science Foundation of Shandong Province of China (Grant no. ZR2011FM014), the Fundamental Research Funds for the Central Universities (Grant no. 12CX06071A), and the Postgraduate Innovation Funds of China University of Petroleum (Grant no. CX2013060).

* Corresponding author at: Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK.
E-mail addresses: cailianfang@163.com (L. Cai), tianxm@upc.edu.cn (X. Tian), sqc@ecs.soton.ac.uk (S. Chen).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.07.029

over PCA. However, in the method of [11], the number of monitoring charts increases with the number of ICs, which increases the computational burden on process monitoring. Lee et al. [12] proposed three Mahalanobis-type monitoring statistics based on the extracted ICs to alleviate the onerous monitoring task of ICs' charts. In the work [13], a local outlier factor method was adopted to build a density-based monitoring statistic based on the obtained ICs, and it was demonstrated that this approach achieves better monitoring performance than the Mahalanobis-type monitoring statistics. The authors of [14] proposed a particle swarm optimization assisted ICA method for process monitoring, which avoids the local minimum solution problem associated with the gradient based FastICA algorithm [15]. In the light of the important influence of the ICs' order on the monitoring performance, Wang et al. [16] sorted the ICs according to the minimum mean square prediction error criterion. The authors of [17] developed a state-space ICA algorithm for process monitoring to take into consideration both the auto-correlation and cross-correlation of the measured data. Based on the fact that the standard kernel density estimation method used for determining the control limit does not perform well with the calculated ICA based monitoring statistic which is usually correlated, Hsu et al. [18] adopted the support vector machine (SVM) to build a two-class classifier for fault detection with the ICA based monitoring statistic as the classifier's input. Since some of the latent variables may be Gaussian, Liu et al. [19] adopted PCA to obtain the Gaussian and non-Gaussian latent variables and then applied ICA to extract the non-Gaussian ICs from the retained principal components. To further account for the process's nonlinear behaviours, the author of [20] integrated the Kernel ICA with Kernel PCA to extract both the non-Gaussian feature information and Gaussian feature information for fault detection. Tian et al. [21] considered the nonlinear characteristics of batch processes and proposed a multiway kernel ICA monitoring method based on feature samples. Cai et al. [22] integrated the kernel ICA and local preserving projection (LPP) for nonlinear continuous process monitoring by taking into account both non-Gaussian information extraction and local neighborhood information preservation. In order to characterize the shifting modes and process uncertainty, Rashid and Yu [23] proposed a hidden Markov model based adaptive ICA approach for multimodal process monitoring. Taking both the multimodality and nonlinearity of processes into account, Zhang et al. [24] proposed a nonlinear multimode process monitoring method based on the Kronecker product and modified kernel ICA.

In all the above studies, ICA is utilized as an effective feature extraction tool. However, the commonly used ICA algorithms in the existing process monitoring methods, such as the FastICA based on the maximum negentropy criterion [15], are basically “noise-free” algorithms. Specifically, these algorithms use noise-free ICA models, and they are not very effective to take into account ubiquitous measurement noise. In the ideal case where there exists no measurement noise in the measured data, a noise-free ICA algorithm can effectively estimate the mixing matrix or de-mixing matrix. Once the mixing matrix or de-mixing matrix is obtained, the ICs can be easily calculated for process monitoring. As pointed out by Wang [25], however, the noise corruption always exists in industrial processes. When the measured data are corrupted by the measurement noise, the ICs may not be directly calculated because of the adverse effects of the measurement noise. Specifically, under the adverse disturbance of the measurement noise, the extracted ICs by a noise-free ICA algorithm may not represent the process operation information adequately and, consequently, they may result in unsatisfactory monitoring performance. Currently, in some other fields, such as the blind source separation, there exist some ICA algorithms which can explicitly consider the measurement noise. Specifically, Cichocki et al. [26] extended the existing adaptive algorithm with equivalent properties to reduce the bias in the de-mixing matrix caused by measurement noise and developed a recurrent dynamic neural network for estimating the unknown mixing matrix. The author of [27] proposed a contrast function based on Gaussian moments and developed a modified FastICA algorithm to estimate the mixing matrix. Cao et al. [28] proposed a robust prewhitening technique for reducing the effect of noise and a parametrized *t*-distribution density model which was combined with the light-tailed distribution model for estimating the mixing matrix. Liu et al. [29] explicitly considered the effect of noise by using the criterion of minimizing the normalized mean square prediction error to conduct the mixing matrix estimation. Yang and Guo [30] derived two new Gaussian moments algorithms for estimating the mixing matrix by combining Gaussian moments and likelihood estimation based on the assumption that independent components are the time signals. The authors of [31] combined the recursive least squares adaptive noise cancellation via QR decomposition and the FastICA to reduce the bias in the estimation of mixing matrix. Nevertheless, these “noisy” ICA algorithms were not introduced for the process monitoring purpose. The main reason lies in the fact that these “noisy” ICA algorithms usually require some strict assumptions, such as that the covariance matrix of the noise is a diagonal matrix with the identical diagonal elements or has been obtained by prior knowledge, which may not be satisfied in real industrial processes. Therefore, developing an appropriate “noisy” ICA algorithm which can effectively remove or alleviate the effects of the measurement noise is of great significance for improving the process monitoring performance. More specifically, how to obtain the ICs or ICs-related statistics that are resistant or robust to the measurement noise based on the estimated mixing matrix is an urgent problem to solve for process monitoring.

Motivated by the above analysis, in this contribution, a process monitoring method is proposed based on a new noisy independent component analysis, referred to as the NoisyICAn. First, we consider the measurement noise explicitly in the noisy ICA model, and develop the NoisyICAn algorithm, which does not need the knowledge of the noise's covariance matrix, to estimate the mixing matrix by building a series of the fourth-order cumulant matrices of the measured data and carrying out the joint diagonalization of these matrices with the least-squares based non-orthogonal joint diagonalization algorithm [32]. Furthermore, the estimated mixing matrix is adopted to establish the kurtosis relationships of the ICs and the measured variables, and an effective approach for recursively estimating the kurtosis of ICs is constructed. This further reduces the effect of the noise. Then, two monitoring statistics, the I^2 and SPE statistics, are built using the obtained recursive estimate of ICs' kurtosis to conduct process monitoring. Unlike the existing “noisy” ICA algorithms, such as the two algorithms given in [27,29], respectively, our proposed NoisyICAn algorithm does not require the information of the noise's covariance matrix and thus it can be applied more easily and conveniently to monitor actual industrial processes. In our extensive simulation study involving a simple three-variable system and a continuous stirred tank reactor system, we compare our proposed NoisyICAn-based monitoring method with the four benchmark schemes, the conventional noise-free FastICA-based monitoring method [12] and the kernel FastICA-based monitoring scheme [20,21] as well as the two monitoring methods based on the existing noisy ICA schemes of [27,29], referred to as the NoisyICA1 and NoisyICA2, respectively. The results obtained demonstrate that the proposed NoisyICAn-based monitoring method outperforms the other four benchmark methods in terms of the fault detection time and fault detection rate.

The remainder of the paper is organized as follows. In Section 2, the conventional ICA-based monitoring methods are briefly reviewed.

Then, our proposed NoisyICAn-based process monitoring method is detailed in Section 3. The simulation studies on a simple three-variable system and a continuous stirred tank reactor system are carried out in Section 4, while our conclusions are drawn in Section 5.

2. The conventional ICA-based monitoring methods

The conventional ICA-based monitoring methods usually take the following noise-free ICA model as the basis:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}, \tag{1}$$

where $\boldsymbol{x} = [x_1 x_2 \dots x_m]^{\mathrm{T}} \in \mathbb{R}^m$ denotes the vector of the zero-mean measured variables, $\boldsymbol{s} = [s_1 s_2 \dots s_m]^{\mathrm{T}} \in \mathbb{R}^m$ is the vector of the zero-mean ICs, and $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ is known as the mixing matrix, while $^{\mathrm{T}}$ denotes the vector or matrix transpose operator. The objective of the ICA is to estimate the ICs by finding a de-mixing matrix $\boldsymbol{W} \in \mathbb{R}^{m \times m}$ from the measured variables, such that the estimate of the ICs can be expressed by

$$\widehat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x}, \tag{2}$$

in which the elements of the estimate of the ICs $\widehat{\boldsymbol{s}} = [\widehat{s}_1 \widehat{s}_2 \dots \widehat{s}_m]^{\mathrm{T}}$ are as independent of each other as possible.

Usually, the measured data need to be whitened before applying the ICA algorithm. The whitened variables can be obtained by means of the PCA as

$$\boldsymbol{z} = \boldsymbol{Q}\boldsymbol{x}, \tag{3}$$

where $\boldsymbol{Q}$ denotes the whitening matrix which can be chosen as follows. Let $E[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}] = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$, where $E[\bullet]$ denotes the expectation operator, $\boldsymbol{V}$ is the orthogonal matrix consisting of the eigenvectors of $E[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}]$, and $\boldsymbol{\Lambda}$ is the diagonal matrix with the eigenvalues of $E[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}]$ as its diagonal elements. Then, $\boldsymbol{Q} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^{\mathrm{T}}$. The whitened variables satisfy the condition

$$E[\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}] = \boldsymbol{I}_m, \tag{4}$$

where $\boldsymbol{I}_m$ denotes the $m \times m$ identity matrix.

For the mathematical convenience, all the ICs are made to have the unit variance by appropriately choosing the de-mixing matrix $\boldsymbol{W} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{Q}$. Then Eq. (2) can be reformulated as

$$\widehat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{z}, \tag{5}$$

where $\boldsymbol{U}$ is an orthonormal matrix satisfying $E[\widehat{\boldsymbol{s}}\widehat{\boldsymbol{s}}^{\mathrm{T}}] = E[\boldsymbol{U}^{\mathrm{T}}\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{U}] = \boldsymbol{U}^{\mathrm{T}}E[\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}]\boldsymbol{U} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \boldsymbol{I}_m$. There exist several ways of calculating the matrix $\boldsymbol{U}$, including the measures of non-Gaussian, such as kurtosis and negentropy, as well as the minimization of mutual information and the maximum likelihood estimation [9]. Among them, the FastICA algorithm [15] is most widely used in the conventional ICA-based monitoring methods [7,8,10–13,16–21]. The optimization objective of the FastICA is defined by

$$\max_{\boldsymbol{u}_i} J(\boldsymbol{u}_i) = \max_{\boldsymbol{u}_i}(E[G(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{z})] - E[G(v)])^2,$$
$$\text{s.t.} \quad E[(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{z})^2] = 1, \tag{6}$$

for $1 \le i \le m$, where $\boldsymbol{u}_i$ is the ith column of $\boldsymbol{U}$ and v is a Gaussian variable with zero mean and unit variance, while $G(\bullet)$ is a non-quadratic function which can be chosen as $G(y) = -\exp(-y^2/2)$. The details of the FastICA algorithm can be found in [15] and therefore they are not repeated here.

Once the matrix $\boldsymbol{U}$ is obtained, the ICs can be estimated by using Eq. (5). The estimated ICs can be arranged in the descending order according to their non-Gaussian degrees [21], and the rows of the matrix $\boldsymbol{U}^{\mathrm{T}}$ are also sorted correspondingly. The l extracted dominant ICs, $\widehat{\boldsymbol{s}}_l \in \mathbb{R}^l$, are then expressed as

$$\widehat{\boldsymbol{s}}_l = \boldsymbol{U}_l^{\mathrm{T}}\boldsymbol{z}, \tag{7}$$

where $\boldsymbol{U}_l^{\mathrm{T}}$ denotes the matrix consisting of the l corresponding rows of $\boldsymbol{U}^{\mathrm{T}}$. The I^2 and SPE statistics are next constructed to detect process faults. Let $\boldsymbol{x}(t)$ be the sample value of $\boldsymbol{x}$ at the sample time t, and $\widehat{\boldsymbol{s}}_l(t)$ be the corresponding extracted l dominant ICs. The I^2 and SPE statistics are defined respectively by [7,12,14,16,21]

$$I^2(t) = (\widehat{\boldsymbol{s}}_l(t))^{\mathrm{T}}\widehat{\boldsymbol{s}}_l(t), \tag{8}$$

$$\mathrm{SPE}(t) = (\boldsymbol{x}(t) - \boldsymbol{Q}^{-1}\boldsymbol{U}_l\widehat{\boldsymbol{s}}_l(t))^{\mathrm{T}}(\boldsymbol{x}(t) - \boldsymbol{Q}^{-1}\boldsymbol{U}_l\widehat{\boldsymbol{s}}_l(t)). \tag{9}$$

$I^2(t)$ is used to monitor the systematic part of the process variation, while SPE(t) is used to monitor the non-systematic part of the process variation.

3. The new noisy ICA-based monitoring method

It is well acknowledged that the process data measured by the sensors usually contain the measurement noise. Under the influence of the measurement noise, the process information extracted by the FastICA may not be sufficiently accurate. A natural solution is using the existing noisy ICA algorithms [26–31] to substitute for the FastICA. However, these noisy ICA algorithms generally require the knowledge of the noise's covariance matrix, which may not be realistic in most real-world situations. Moreover, the monitoring statistics calculated using Eqs. (8) and (9) by the process monitoring schemes based on these noisy ICA algorithms are still subject to the effect of the noise. All of these may lead to unsatisfactory monitoring performance. In this section, we explicitly consider the measurement noise in the noisy ICA model and develop a new noisy ICA algorithm, called the NoisyICAn which does not require the noise's covariance matrix. Then we further propose a NoisyICAn-based monitoring method to improve the process monitoring performance.

3.1. The NoisyICAn algorithm

Unlike the noise-free ICA model of Eq. (1), the following noisy ICA model is taken as the basis to develop our NoisyICAn-based monitoring method:

$$\boldsymbol{x} = \tilde{\boldsymbol{x}} + \boldsymbol{\varepsilon} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\varepsilon}, \tag{10}$$

where $\boldsymbol{x} \in \mathbb{R}^m$ are the measured variables which are corrupted by the measurement noise variables $\boldsymbol{\varepsilon} = [\varepsilon_1 \varepsilon_2 \dots \varepsilon_m]^{\mathrm{T}} \in \mathbb{R}^m$, while $\tilde{\boldsymbol{x}} = [\tilde{x}_1 \tilde{x}_2 \dots \tilde{x}_m]^{\mathrm{T}} = \boldsymbol{A}\boldsymbol{s} \in \mathbb{R}^m$ are the effective variables with zero mean which are not subject to the effects of the measurement noise. The noise variables $\boldsymbol{\varepsilon}$ and the ICs $\boldsymbol{s}$ satisfy the conditions: (1) the elements of $\boldsymbol{s}$ are zero-mean and mutually independent; (2) the elements of $\boldsymbol{\varepsilon}$ are zero-mean and uncorrelated Gaussian variables; and (3) the elements of $\boldsymbol{s}$ and the elements of $\boldsymbol{\varepsilon}$ are mutually independent.

Inspired by the work [33] which proposed a signal separation technique using a double referenced system, we construct a series of the fourth-order cumulant matrices of the measured variables $\boldsymbol{x}$, denoted as $\boldsymbol{C}_{\boldsymbol{x}}(j,k)$, for $1 \le j,k \le m$. The i_1th-row and i_2th-column element of each matrix $\boldsymbol{C}_{\boldsymbol{x}}(j,k)$ is defined by

$$\begin{aligned}[\boldsymbol{C}_{\boldsymbol{x}}(j,k)]_{i_1,i_2} &= \mathrm{c}_4(x_{i_1}, x_{i_2}, x_j, x_k) = E[x_{i_1}x_{i_2}x_jx_k] \\ &\quad - E[x_{i_1}x_{i_2}]E[x_jx_k] - E[x_{i_1}x_j]E[x_{i_2}x_k] \\ &\quad - E[x_{i_1}x_k]E[x_{i_2}x_j], \end{aligned} \tag{11}$$

for $1 \le i_1, i_2, \le m$, where $\mathrm{c}_4(\bullet)$ denotes the fourth-order cumulant. According to the conditions (1) to (3) for $\boldsymbol{\varepsilon}$ and $\boldsymbol{s}$ as well as the multi-linearity property of cumulant [33,34], Eq. (11) can further be expressed as

$$\begin{aligned}[\boldsymbol{C}_{\boldsymbol{x}}(j,k)]_{i_1,i_2} &= \mathrm{c}_4(\tilde{x}_{i_1} + \varepsilon_{i_1}, \tilde{x}_{i_2} + \varepsilon_{i_2}, \tilde{x}_j + \varepsilon_j, \tilde{x}_k + \varepsilon_k) \\ &= \mathrm{c}_4(\tilde{x}_{i_1}, \tilde{x}_{i_2}, \tilde{x}_j, \tilde{x}_k)\end{aligned}$$

$$
\begin{aligned}
&= \mathsf{c}_4\left(\sum_{p_1=1}^{m} a_{i_1,p_1}s_{p_1}, \sum_{p_2=1}^{m} a_{i_2,p_2}s_{p_2}, \sum_{p_3=1}^{m} a_{j,p_3}s_{p_3}, \sum_{p_4=1}^{m} a_{k,p_4}s_{p_4}\right)\\
&= \sum_{p_1=1}^{m} a_{i_1,p_1}\sum_{p_2=1}^{m} a_{i_2,p_2}\times \mathsf{c}_4\left(s_{p_1}, s_{p_2}, \sum_{p_3=1}^{m} a_{j,p_3}s_{p_3}, \sum_{p_4=1}^{m} a_{k,p_4}s_{p_4}\right)\\
&= \sum_{p=1}^{m} a_{i_1,p}a_{i_2,p}\mathsf{c}_4\left(s_p, s_p, \sum_{p_3=1}^{m} a_{j,p_3}s_{p_3}, \sum_{p_4=1}^{m} a_{k,p_4}s_{p_4}\right)\\
&= \sum_{p=1}^{m} a_{i_1,p}a_{i_2,p}\sum_{p_3=1}^{m} a_{j,p_3}\sum_{p_4=1}^{m} a_{k,p_4}\mathsf{c}_4(s_p, s_p, s_{p_3}, s_{p_4})\\
&= \sum_{p=1}^{m} a_{i_1,p}a_{i_2,p}a_{j,p}a_{k,p}\mathsf{c}_4(s_p, s_p, s_p, s_p)\\
&= \sum_{p=1}^{m} a_{i_1,p}a_{i_2,p}a_{j,p}a_{k,p}\mathsf{k}_4(s_p), \qquad (12)
\end{aligned}
$$

where $a_{i,j}$ denotes the ith-row and jth-column element of the mixing matrix $\boldsymbol{A}$, while $\mathsf{k}_4(s_p) = \mathsf{c}_4(s_p, s_p, s_p, s_p)$ is known as the kurtosis of the variable s_p. According to Eq. (12), the matrix $\boldsymbol{C}_{\boldsymbol{x}}(j,k)$ can be expressed as

$$\boldsymbol{C}_{\boldsymbol{x}}(j,k) = \boldsymbol{A}\boldsymbol{\Gamma}(j,k)\boldsymbol{A}^{\mathrm{T}}, \qquad (13)$$

with the diagonal matrix $\boldsymbol{\Gamma}(j,k)$ defined by

$$
\begin{aligned}
\boldsymbol{\Gamma}(j,k) = &\operatorname{diag}\{a_{j,1}, a_{j,2}, \ldots, a_{j,m}\}\\
&\times \operatorname{diag}\{\mathsf{k}_4(s_1), \mathsf{k}_4(s_2), \ldots, \mathsf{k}_4(s_m)\}\\
&\times \operatorname{diag}\{a_{k,1}, a_{k,2}, \ldots, a_{k,m}\}, \qquad (14)
\end{aligned}
$$

where $\operatorname{diag}\{y_1, y_2, \ldots, y_m\}$ denotes the $m \times m$ diagonal matrix with $y_1, y_2, \ldots, y_m$ as its diagonal elements.

From Eqs. (13) and (14), it becomes obvious that each matrix $\boldsymbol{C}_{\boldsymbol{x}}(j,k)$ can be diagonalized by the mixing matrix $\boldsymbol{A}$. Therefore, the joint diagonalization degree of the cumulant matrices can be taken as the optimization objective to estimate $\boldsymbol{A}$. Specifically, to estimate $\boldsymbol{A}$ based on the measure of the attainable diagonalization degree, the following weighted least-squares (WLS) criterion is adopted [32]

$$J_{\mathrm{WLS}}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{\Gamma}}) = \sum_{j=1}^{m}\sum_{k=1}^{m} w_{j,k}\| \boldsymbol{C}_{\boldsymbol{x}}(j,k) - \hat{\boldsymbol{A}}\hat{\boldsymbol{\Gamma}}(j,k)\hat{\boldsymbol{A}}^{\mathrm{T}}\|_F^2, \qquad (15)$$

where $w_{j,k}$ are positive weights, $\hat{\boldsymbol{A}}$ is the estimate of the mixing matrix $\boldsymbol{A}$, and $\hat{\boldsymbol{\Gamma}}(j,k)$ is the estimate of the diagonal matrix $\boldsymbol{\Gamma}(j,k)$, while $\hat{\boldsymbol{\Gamma}} = \{\hat{\boldsymbol{\Gamma}}(j,k) : 1 \le j,k \le m\}$ denotes the set of all the estimated diagonal matrices. Typically, equal weightings $w_{j,k}$, $1 \le j,k \le m$, are applied to form the criterion of Eq. (15). The so-called AC-DC algorithm [32] can be utilized to estimate the mixing matrix by minimizing the cost function of Eq. (15), and the algorithm alternates between the following two minimization procedures.

(a) The AC "alternating columns" phase minimizes $J_{\mathrm{WLS}}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{\Gamma}})$ with respect to the ith column of $\hat{\boldsymbol{A}}$, while keeping its other columns as well as $\{\hat{\boldsymbol{\Gamma}}(j,k) : 1 \le j,k \le m\}$ fixed, for $1 \le i \le m$. This is referred to as one sweep. This sweep procedure is repeated q times, and each sweep starts with the initial $\hat{\boldsymbol{A}}$ given from the previous sweep. Typically, $q = 10$ is sufficient.

(b) The DC "diagonal centres" phase minimizes $J_{\mathrm{WLS}}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{\Gamma}})$ with respect to all the diagonal matrices $\hat{\boldsymbol{\Gamma}} = \{\hat{\boldsymbol{\Gamma}}(j,k) : 1 \le j,k \le m\}$, while keeping $\hat{\boldsymbol{A}}$ fixed.

The algorithm iterates between the two phases a number of times, until the reduction in the cost function values of (15) between two consecutive iterations is below a threshold value, e.g. 10^{-5}.

The more detailed descriptions of this AC-DC algorithm can be found in [32]. We now summarize the procedure of the NoisyICAn algorithm, which is developed partially based on the idea of [33].

(i) Construct the fourth-order cumulant matrices of the measured variables, $\boldsymbol{C}_{\boldsymbol{x}}(j,k)$ for $1 \le j,k \le m$, according to Eq. (11).

(ii) Use the calculated cumulant matrices, $\boldsymbol{C}_{\boldsymbol{x}}(j,k)$ for $1 \le j,k \le m$, to form the measure Eq. (15) of the joint diagonalization degree.

(iii) Optimize this objective function by the AC-DC algorithm to estimate the mixing matrix $\boldsymbol{A}$.

The computational complexities of the AC phase and the DC phase are respectively $\mathcal{O}(q \cdot m^5)$ and $\mathcal{O}(m^5)$. Thus, the overall computational complexity of this NoisyICAn algorithm can be shown to be in the order of $\mathcal{O}(q \cdot m^5)$. In comparison, the computational complexity of the FastICA algorithm is in the order of $\mathcal{O}(m \cdot N_1)$ [35], where N_1 is the sample size of the training data set. Moreover, the well-known noisy ICA algorithm of [27], denoted as the NoisyICA1, which is an improved version of the FastICA, as well as the noisy ICA algorithm of [29], denoted as NoisyICA2, both have the computational complexity of $\mathcal{O}(m \cdot N_1)$. The number of the measured variables m and the sample size N_1 of the training data are determined by the specific application. The training data size N_1 is typically large. Thus, for the cases of modest m, the complexity of the NoisyICAn algorithm may be comparable to those of the FastICA, NoisyICA1 and NoisyICA2 algorithms. Note that the operations of estimating the mixing matrix $\boldsymbol{A}$ are carried out off-line and, therefore, the complexity of this modeling stage is not too critical. The significant advantage of our NoisyICAn algorithm over the FastICA algorithm is that it effectively takes into account the ubiquitous measurement noise in the modeling. Furthermore, unlike the NoisyICA1 algorithm [27] and the NoisyICA2 algorithm [29], our NoisyICAn algorithm does not require the information of the noise's covariance matrix and thus it is more practical for real industrial processes.

3.2. The recursive kurtosis estimation of ICs

After obtaining an estimated mixing matrix $\hat{\boldsymbol{A}}$, the usual practice is to apply the inverse $\hat{\boldsymbol{A}}^{-1}$ to the measured variables $\boldsymbol{x}$ to obtain an estimate $\hat{\boldsymbol{s}} = \hat{\boldsymbol{A}}^{-1}\boldsymbol{x}$ of the ICs $\boldsymbol{s}$. However, it can be easily seen that the estimates of the ICs so obtained are contaminated with the noise variables $\hat{\boldsymbol{A}}^{-1}\boldsymbol{\varepsilon}$. In order to alleviate the influence of the measurement noise, we construct the kurtosis relationships of the ICs and the measured variables using the estimated mixing matrix by exploiting the fact that the kurtosis statistic of a Gaussian variable with zero mean is zero [34].

The kurtosis of the ith measured variable x_i is given by

$$\mathsf{k}_4(x_i) = \mathsf{c}_4(x_i, x_i, x_i, x_i) = E[x_i^4] - 3(E[x_i^2])^2. \qquad (16)$$

Noting the property of Eq. (12), Eq. (16) can be expressed as

$$
\begin{aligned}
\mathsf{k}_4(x_i) &= \mathsf{c}_4(\tilde{x}_i + \varepsilon_i, \tilde{x}_i + \varepsilon_i, \tilde{x}_i + \varepsilon_i, \tilde{x}_i + \varepsilon_i)\\
&= \mathsf{c}_4(\tilde{x}_i, \tilde{x}_i, \tilde{x}_i, \tilde{x}_i) = \sum_{p=1}^{m} a_{i,p}^4 \mathsf{c}_4(s_p, s_p, s_p, s_p)\\
&= \sum_{p=1}^{m} a_{i,p}^4 \mathsf{k}_4(s_p). \qquad (17)
\end{aligned}
$$

According to Eq. (17), we can establish the kurtosis relationships between the ICs and the measured variables as

$$\begin{bmatrix} \mathrm{k}_4(s_1) \\ \mathrm{k}_4(s_2) \\ \vdots \\ \mathrm{k}_4(s_m) \end{bmatrix} = \begin{bmatrix} a_{1,1}^4 & a_{1,2}^4 & \cdots & a_{1,m}^4 \\ a_{2,1}^4 & a_{2,2}^4 & \cdots & a_{2,m}^4 \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}^4 & a_{m,2}^4 & \cdots & a_{m,m}^4 \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{k}_4(x_1) \\ \mathrm{k}_4(x_2) \\ \vdots \\ \mathrm{k}_4(x_m) \end{bmatrix}. \tag{18}$$

From Eq. (18), it can be observed that the mapping between the kurtosis of the ICs and the kurtosis of the measured variables can be built with the estimated mixing matrix $\widehat{\boldsymbol{A}}$, and the higher-order kurtosis statistics have removed the influence of the Gaussian measurement noise $\boldsymbol{\varepsilon}$.

The kurtosis of the ith measured variable x_i with zero mean and unit variance at the sample number t can be estimated by the following recursive expression [36]:

$$\mathrm{k}_4(x_i|t) = \mathrm{k}_4(x_i|t-1) - \mu(\mathrm{k}_4(x_i|t-1) - x_i^4(t) + 3), \tag{19}$$

for $1 \le i \le m$, where $\mathrm{k}_4(x_i|t)$ denotes the kurtosis estimation of the measured variable x_i at the sample number t, and μ is a learning rate, while $x_i(t)$ denotes the sample value of x_i at the sample number t. By substituting Eq. (19) into Eq. (18), we construct a recursive estimation of the ICs' kurtosis as follows:

$$\begin{bmatrix} \mathrm{k}_4(s_1|t) \\ \mathrm{k}_4(s_2|t) \\ \vdots \\ \mathrm{k}_4(s_m|t) \end{bmatrix} = \begin{bmatrix} a_{1,1}^4 & a_{1,2}^4 & \cdots & a_{1,m}^4 \\ a_{2,1}^4 & a_{2,2}^4 & \cdots & a_{2,m}^4 \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}^4 & a_{m,2}^4 & \cdots & a_{m,m}^4 \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{k}_4(x_1|t) \\ \mathrm{k}_4(x_2|t) \\ \vdots \\ \mathrm{k}_4(x_m|t) \end{bmatrix}, \tag{20}$$

where $\mathrm{k}_4(s_i|t)$, $1 \le i \le m$, denotes the kurtosis estimation of the ith IC at the sample number t. By defining

$$\boldsymbol{M} = \begin{bmatrix} a_{1,1}^4 & a_{1,2}^4 & \cdots & a_{1,m}^4 \\ a_{2,1}^4 & a_{2,2}^4 & \cdots & a_{2,m}^4 \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}^4 & a_{m,2}^4 & \cdots & a_{m,m}^4 \end{bmatrix}^{-1}, \tag{21}$$

$$\boldsymbol{k4}(\boldsymbol{s}|t) = [\mathrm{k}_4(s_1|t)\mathrm{k}_4(s_2|t)\ldots\mathrm{k}_4(s_m|t)]^{\mathrm{T}}, \tag{22}$$

$$\boldsymbol{k4}(\boldsymbol{x}|t) = [\mathrm{k}_4(x_1|t)\mathrm{k}_4(x_2|t)\ldots\mathrm{k}_4(x_m|t)]^{\mathrm{T}}, \tag{23}$$

Eq. (20) can be expressed in the matrix form as

$$\boldsymbol{k4}(\boldsymbol{s}|t) = \boldsymbol{M} \cdot \boldsymbol{k4}(\boldsymbol{x}|t). \tag{24}$$

The absolute values of the ICs' kurtosis statistics given in Eq. (18) quantitatively represent the non-Gaussian degrees of the ICs. Therefore, the ICs can also be arranged in descending order according to their non-Gaussian degrees, and the rows of $\boldsymbol{M}$ are also sorted accordingly. Specifically, the c recursive kurtosis estimations of the c dominant ICs $\boldsymbol{s}_c$ can be expressed as

$$\boldsymbol{k4}_c(\boldsymbol{s}_c|t) = \boldsymbol{M}_c \cdot \boldsymbol{k4}(\boldsymbol{x}|t). \tag{25}$$

where $\boldsymbol{M}_c \in \mathbb{R}^{c \times m}$ denotes the matrix consisting of the corresponding c rows of the matrix $\boldsymbol{M}$.

3.3. The process monitoring strategy

Assume that the order of the ICs in $\boldsymbol{s}$ has been arranged and the rows of $\boldsymbol{M}$ have been sorted accordingly. In order to conduct process monitoring, the I^2 and SPE monitoring statistics can be calculated respectively using the recursive kurtosis estimate of the dominant ICs as follows:

$$I^2(t) = \boldsymbol{k4}_c(\boldsymbol{s}_c|t)^{\mathrm{T}} \boldsymbol{\Phi}^{-1} \boldsymbol{k4}_c(\boldsymbol{s}_c|t), \tag{26}$$

$$\mathrm{SPE}(t) = (\boldsymbol{k4}(\boldsymbol{x}|t) - \boldsymbol{M}_{1:c}^{-1}\boldsymbol{k4}_c(\boldsymbol{s}_c|t))^{\mathrm{T}} (\boldsymbol{k4}(\boldsymbol{x}|t) - \boldsymbol{M}_{1:c}^{-1}\boldsymbol{k4}_c(\boldsymbol{s}_c|t)), \tag{27}$$

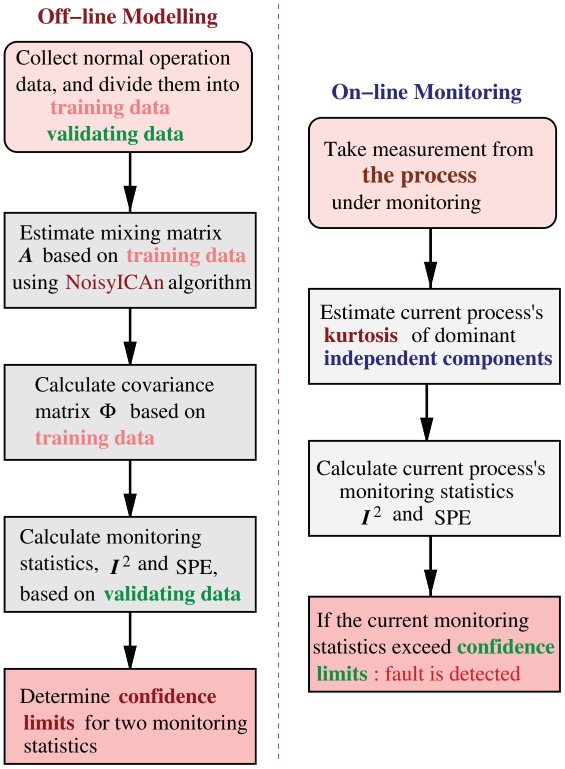
Off–line Modelling
Collect normal operation data, and divide them into training data validating data
Estimate mixing matrix A based on training data using NoisyICAn algorithm
Calculate covariance matrix Φ based on training data
Calculate monitoring statistics, I 2 and SPE, based on validating data
Determine confidence limits for two monitoring statistics
On–line Monitoring
Take measurement from the process under monitoring
Estimate current process's kurtosis of dominant independent components
Calculate current process's monitoring statistics I 2 and SPE
If the current monitoring statistics exceed confidence limits : fault is detected

Fig. 1. The overall procedure of the proposed process monitoring strategy.

where $\boldsymbol{\Phi} = E[\boldsymbol{k4}_c(\boldsymbol{s}_c|t)\boldsymbol{k4}_c(\boldsymbol{s}_c|t)^{\mathrm{T}}]$ is the covariance matrix of $\boldsymbol{k4}_c\ (\boldsymbol{s}_c|t)$ which is estimated using the training data, while $\boldsymbol{M}_{1:c}^{-1}$ denotes the matrix consisting of the first c columns of $\boldsymbol{M}^{-1}$. Again, $I^2(t)$ is used to monitor the systematic part of the process variation, and SPE(t) is used to monitor the non-systematic part of the process variation.

To use these two monitoring statistics for judging whether the process is in control or not, the corresponding confidence limits need to be determined. The δ confidence limits for the built monitoring statistics can be determined as follows. Split the measured data collected from the process under the normal operating conditions into the two parts: the training data with N_1 samples and the validating data with N_2 samples. Based on the training data, both the mixing matrix $\widehat{\boldsymbol{A}}$ and the covariance matrix $\boldsymbol{\Phi}$ of $\boldsymbol{k4}_c(\boldsymbol{s}_c|t)$ are estimated. Based on the validating data, the two monitoring statistics are calculated, which are denoted as $\{I^2(t)\}_{t=1}^{N_2}$ and $\{\mathrm{SPE}(t)\}_{t=1}^{N_2}$, respectively. Then round $N_2(1-\delta)$ towards the nearest integer, which is denoted as r. For each monitoring statistic, the rth highest value is adopted as its confidence limit. Specifically, the rth highest value of $\{I^2(t)\}_{t=1}^{N_2}$ is used as the confidence limit for the I^2 statistic, while the rth highest value of $\{\mathrm{SPE}(t)\}_{t=1}^{N_2}$ is adopted as the confidence limit for the SPE statistic. With this strategy, the false alarm rate can be reduced significantly [37]. Furthermore, the false alarm rates of different monitoring methods can be adjusted to the same level approximately, for fair and convenient comparison of the fault detection times and fault detection rates of different methods.

Referring to Fig. 1, the proposed monitoring procedure based on the NoisyICAn algorithm is now detailed, which consists of the off-line modelling stage and the on-line monitoring stage.

The off-line modelling stage:

(1) Collect the data from the process under normal operation conditions, and divide the measured data into the training data part and the validating data part.

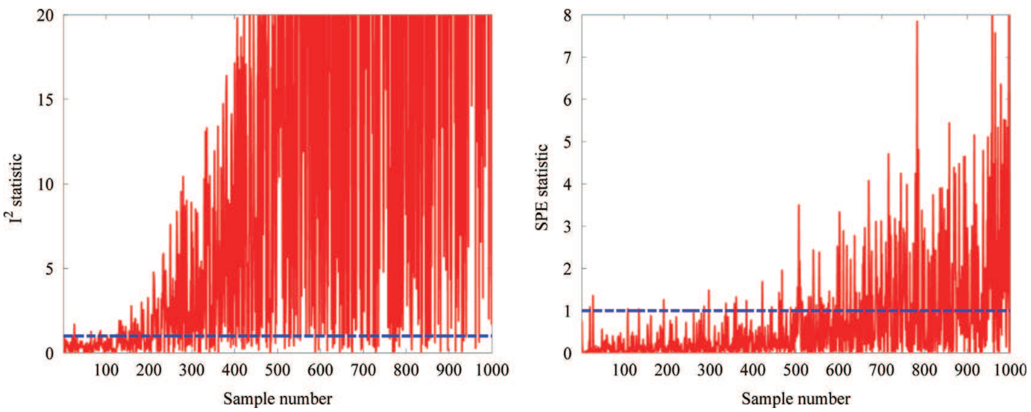

Fig. 2. The monitoring charts of the FastICA-based method for the simple three-variable system, where the fault occurs at the 101th sample.

(2) Based on the training data, estimate the mixing matrix by the NoisyICAn algorithm.
(3) Based on the training data, estimate the kurtosis values of the measured variables recursively using Eq. (19) and, with the obtained kurtosis values of the measured variables and the estimated mixing matrix, estimate the kurtosis values of the dominant ICs using Eq. (25). Calculate the covariance matrix $\mathbf{\Phi}$ of $\boldsymbol{k4}_c(\boldsymbol{s}_c|t)$.
(4) Based on the validating data, estimate the kurtosis values of the measured variables recursively using Eq. (19), and obtain the kurtosis values of the dominant ICs using Eq. (25).
(5) Use the estimated kurtosis values of the dominant ICs obtained in (4) in Eqs. (26) and (27) to calculate the two monitoring statistics. Determine the δ confidence limits of the two monitoring statistics.

The on-line monitoring stage:

(1) Take the measurement from the process under monitoring, and use the current measured data to estimate the current kurtosis values of the measured variables using Eq. (19).
(2) Use the current kurtosis values of the measured variables obtained in (1) in Eq. (25) to calculate the current kurtosis values of the dominant ICs.
(3) Use the current kurtosis values of the dominant ICs obtained in (2) in Eqs. (26) and (27) to calculate the current values of the two monitoring statistics.
(4) Determine whether the current monitoring statistics exceed their respective confidence limits, and give an alarm if a fault is detected.

In the above procedure, the training data, the validating data and the current measured data for monitoring are all normalized with the means and variances of the measured variables in the training data. The kurtosis estimates of the measured variables in the training data, the validating data and the current measured data for monitoring are also all normalized with the means and variances of kurtosis estimates of the measured variables in the training data. How to choose an appropriate learning rate μ for the recursive kurtosis estimation of (19) is discussed in Appendix A, while how to determine an appropriate number of the dominant ICs is summarized in Appendix B.

4. Simulation studies

The proposed NoisyICAn-based monitoring method is evaluated in the two case studies, a simple three-variable system and a continuous stirred tank reactor (CSTR) system. In both the case studies, the fault detection performance of our NoisyICAn-based method is compared with those of the four benchmark schemes, namely, the conventional noise-free FastICA-based monitoring method [12] and the kernel FastICA-based monitoring scheme [20,21] as well as the two monitoring methods based on the existing noisy ICA schemes of [27,29], referred to as the NoisyICA1 and NoisyICA2, respectively.

4.1. A three-variable system

The three-variable system which is a modified version of the system studied by Kano et al. [11] is given by $\boldsymbol{x}=\boldsymbol{As}+\boldsymbol{\varepsilon}$ of Eq. (10), with the mixing matrix defined by

$$\boldsymbol{A}=\begin{bmatrix} -0.433 & 0.287 & 1.190 \\ -1.666 & -1.146 & 0.038 \\ 0.125 & 1.326 & 0.327 \end{bmatrix}. \tag{28}$$

The three ICs $\boldsymbol{s}=[s_1\ s_2\ s_3]^{\mathrm{T}}$ are the uncorrelated random signals following the uniform distributions within $[-1,1]$, $[-1.5,1.5]$ and $[-2,2]$, respectively, while the three outputs $\boldsymbol{x}=[x_1\ x_2\ x_3]^{\mathrm{T}}$ are corrupted by the three zero-mean measurement noises $\boldsymbol{\varepsilon}=[\varepsilon_1\ \varepsilon_2\ \varepsilon_3]^{\mathrm{T}}$ which follow the Gaussian distributions. We define the noise intensity of ε_i as the ratio percentage of ε_i's variance over the output x_i's variance, for $1\le i\le 3$. The noise intensities in the three output signals are set to 20%, 50% and 80%, respectively. A fault case, which is the ramp change of the first-row and the second-column element $a_{1,2}$ in the mixing matrix $\boldsymbol{A}$, is simulated. The ramp change rate is set to 0.03. In industrial processes, many faults can be attributed to the slow drift of process parameters, and this simulated fault case is used to represent a typical example of this type of “ramp fault”. The normal operation data with 3000 samples and the fault data with 1000 samples are respectively generated by simulating the three-variable system. The normal operation data are divided into the training set of $N_1=1500$ samples and the validating set of $N_2=1500$ samples. For the fault data used in process monitoring, the fault is introduced at the 101th sample.

The number of the dominant ICs is set to 2 for all the five monitoring methods. The learning rate μ is empirically chosen to be 0.4 for our proposed NoisyICAn-based method. As this is a linear process, the linear kernel is chosen for the kernel FastICA-based monitoring method. The fault detection performance is evaluated in terms of the fault detection rate, which is defined as the percentage of the fault samples whose monitoring statistic values exceed the related confidence limit in all the fault samples, and the fault detection time. In order to decrease the false alarm,

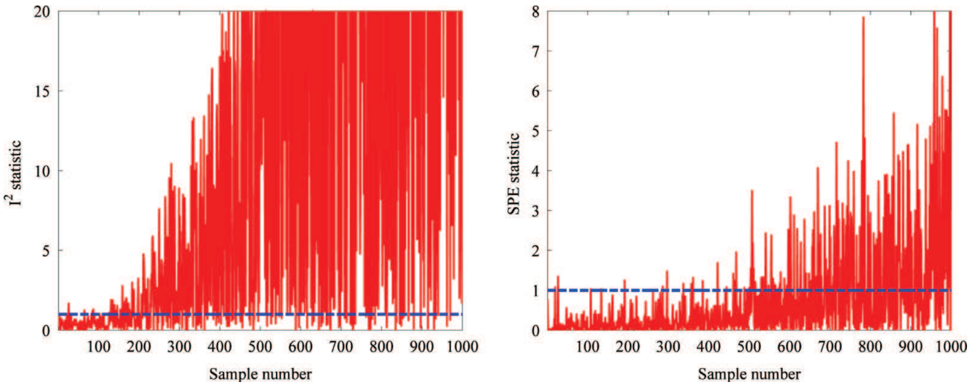

Fig. 3. The monitoring charts of the kernel FastICA-based method for the simple three-variable system, where the fault occurs at the 101th sample.

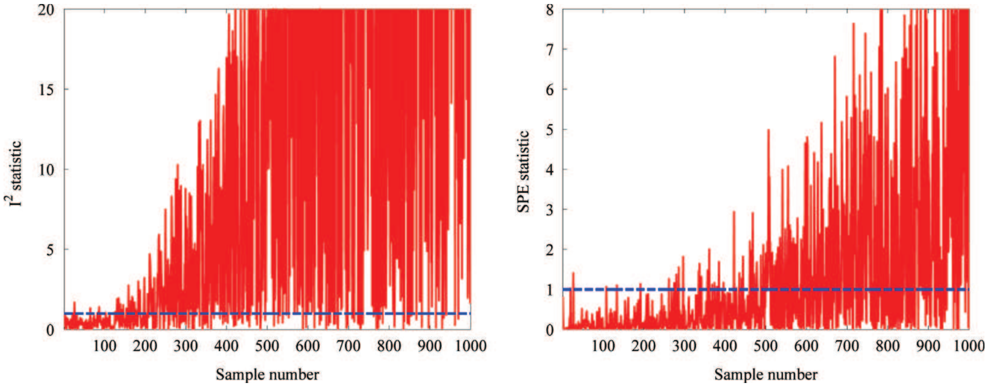

Fig. 4. The monitoring charts of the NoisyICA1-based method for the simple three-variable system, where the fault occurs at the 101th sample.

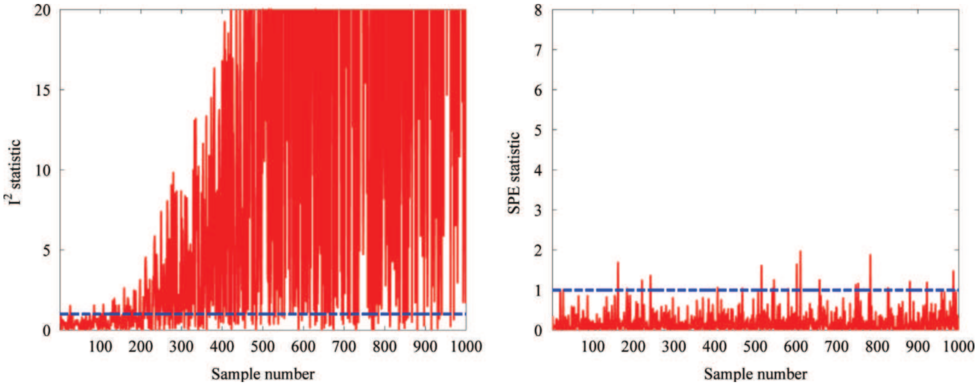

Fig. 5. The monitoring charts of the NoisyICA2-based method for the simple three-variable system, where the fault occurs at the 101th sample.

a fault is indicated only when six consecutive monitoring statistic values exceed the confidence limit and the fault detection time is then defined as the corresponding first sample at which the confidence limit is exceeded [38]. The 99% confidence limit is adopted as the alarming threshold. The monitoring charts of the five methods are depicted in Figs. 2–6, respectively. In order to facilitate the comparison, in each monitoring chart, the monitoring statistic values are normalized by the corresponding confidence limit so that the confidence limit is equal to 1. The monitoring statistic values are plotted as the solid line, while the related confidence limit is indicated by the dashed line.

It can be seen from the FastICA-based, kernel FastICA-based, NoisyICA1-based and NoisyICA2-based monitoring charts of Figs. 2–5 that after the occurrence of the fault at the 101th sample, the I^2 and SPE monitoring statistics' values of many fault samples are still below the corresponding confidence limits, resulting in late detection of the fault. By contrast, from the NoisyICAn-based monitoring charts of Fig. 6, we observe that the two monitoring statistics' values of the fault samples quickly exceed the respective confidence limits, leading to a much earlier detection of the fault. As expected, with the linear kernel, the kernel FastICA-based method is equivalent to the conventional FastICA-based method and, therefore, the both methods achieve the same performance. Table 1 compares the fault detection times (the sample number) and the fault detection rates of the five methods evaluated. It can be seen from Figs. 2–6 as well as Table 1 that the proposed NoisyICAn-based method has a faster fault detection time and achieves a higher fault detection rate than the benchmark FastICA-based, kernel FastICA-based, NoisyICA1-based and NoisyICA2-based methods.

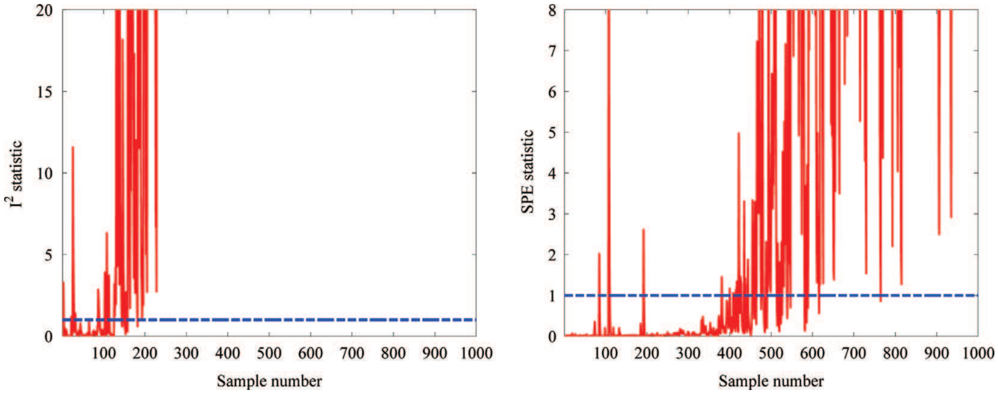

Fig. 6. The monitoring charts of the proposed NoisyICAn-based method for the simple three-variable system, where the fault occurs at the 101th sample.

Table 1
Comparison of the fault detection performance for the simple three-variable system.

Monitoring method	Monitoring statistic	Fault detection time	Fault detection rate (%)
FastICA-based	T^2	210	77.44
	SPE	750	23.89
Kernel FastICA-based	T^2	210	77.44
	SPE	750	23.89
NoisyICA1-based	T^2	210	77.78
	SPE	503	37.00
NoisyICA2-based	T^2	210	77.00
	SPE	Failed	2.44
NoisyICAn-based	T^2	126	96.78
	SPE	422	59.44

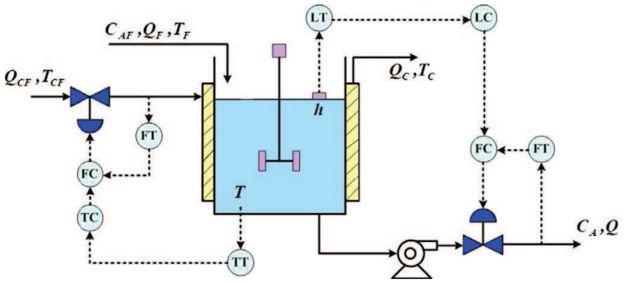

Fig. 7. The continuous stirred tank reactor with cascade control.

4.2. A continuous stirred tank reactor system

The CSTR system is widely used for testing the process monitoring methods [11,39]. The schematic diagram of this CSTR with the cascade control system is shown in Fig. 7. In this CSTR, the reactant A flows into the reactor, and the first-order irreversible reaction A→B happens. The single component B is produced as an outlet stream. Heat from the exothermic reaction is taken away by the cooling flow of the jacket, and the temperature of the reactor is controlled by manipulating the coolant flow, while the level is controlled by manipulating the outlet flow.

Based on the mass, energy and component balances, the dynamic model of this CSTR system can be derived as follows:

$$\frac{dC_A}{dt} = -k_0 e^{-E_0/RT} C_A + \frac{Q_F(C_{AF}-C_A)}{Ah}, \tag{29}$$

Table 2
The measured variables and the noise intensity in each measured variable for the CSTR.

Measured variable	Variable description	Noise intensity (%)
C_A	Concentration of species A in reactor	53.8
T	Reactor temperature	47.8
Q_F	Reactor feed flow rate	23.7
C_{AF}	Concentration of species A in reactor feed stream	10.0
h	Reactor liquid level	11.9
T_F	Temperature of reactor feed stream	23.7
T_C	Temperature of coolant in cooling jacket	80.6
Q_C	Coolant flow rate	5.5
T_{CF}	Temperature of coolant feed	9.6
Q	Reactor outlet flow rate	71.8

Table 3
The parameters of the CSTR.

Parameter	Description
k_0	Preexponential factor
E_0/R	Activation energy
A	Reactor cross-sectional area
ΔH	Reaction heat
ρ	Density of reactor contents
C_p	Heat capacity of reactor contents
UA_C	Heat-transfer coefficient
V_C	Capacity of cooling jacket
C_{pC}	Heat capacity of coolant
ρ_C	Density of coolant

$$\frac{dT}{dt} = \frac{k_0 e^{-E_0/RT} C_A(-\Delta H)}{\rho C_p} + \frac{Q_F(T_F-T)}{Ah} + \frac{UA_C(T_C-T)}{\rho C_p A h}, \tag{30}$$

$$\frac{dT_C}{dt} = \frac{Q_C(T_{CF}-T_C)}{V_C} + \frac{UA_C(T-T_C)}{\rho_C C_{pC} V_C}, \tag{31}$$

$$\frac{dh}{dt} = \frac{Q_F-Q}{A}. \tag{32}$$

The ten measured process variables of this CSTR system are described in Table 2, while the model parameters in Eqs. (29)–(32) are defined in Table 3. The detailed explanation of these parameters and the nominal operating conditions of the CSTR can be found in [39].

The Gaussian measurement noises are added to all the measured data in the simulation procedure, and the noise intensity in each measured variable is also given in Table 2. The simulation data are sampled every 2 s, and 2000 samples are generated by simulating the CSTR system under the normal operation conditions. The measured data under the normal operation conditions are divided into the training set of $N_1 = 1000$ samples and the validating set of $N_2 = 1000$. Eight fault patterns as shown in the Table 4 are simulated, which can be divided into two types according to fault characteristics: the step fault type and the ramp fault type. Among the simulated fault patterns, the faults 1 to 3 belong to the step fault type, while the faults 4 to 8 belong to the ramp fault type. More specifically, the faults 1, 4, 5 and 8 are caused by the internal or external disturbances. The fault 2 signifies that the setpoint for the reactor temperature is changed, while the fault 3 is caused by the temperature sensor for the reactor having a bias, which is an instrumentation fault. The fault 6 is due to the heat-exchanger fouling, while the fault 7 is caused by the catalyst deactivation. Both the faults 6 and 7 are equipment faults. The data for each fault pattern are recorded with 900 samples and the fault is introduced at the 190th sample.

Table 4
The simulated fault patterns for the CSTR system.

Fault	Description	Value
1	Step change in the reactor feed flow rate Q_F	-7 L/min
2	Setpoint change for the reactor temperature T	10 K
3	Bias in the measurement of the reactor temperature T	4 K
4	The reactor feed stream temperature T_F ramps up with the ramp rate	0.3 K/min
5	The feed concentration C_{AF} ramps up with the ramp rate	7×10^{-4} (mol/L)/min
6	The heat-transfer coefficient UA_C ramps down with the ramp rate	-125 (J/(min K))/Min
7	The catalyst activation energy E_0/R ramps up with the ramp rate	6 K/min
8	The coolant feed temperature T_{CF} ramps down with the ramp rate	-0.2 K/min

For all the five monitoring methods, the number of the dominant ICs is selected as 5. The learning rate μ is set to 0.3 for our NoisyICAn-based method. For the kernel FastICA-based monitoring method, we use the Gaussian kernel with the kernel width determined according to the empirical method given in [40]. Again the 99% confidence limit is also adopted as the alarming threshold.

The monitoring charts obtained by all the five methods under the fault pattern 5 are illustrated in Figs. 8–12, respectively. From the monitoring charts of our proposed NoisyICAn-based method shown in Fig. 12, we observe that when the sample number is greater than 420, almost all the monitoring statistics' values exceed the respective confidence limits and stay well above the confidence limits. By contrast, observe from the monitoring charts of the four benchmark methods depicted in Figs. 8–11, respectively, that only when the sample number is greater than 600, can the monitoring statistics' values exceed the corresponding confidence limits. The results shown in Figs. 8–12 clearly demonstrate that our proposed NoisyICAn-based monitoring method can detect the fault pattern 5 of the CSTR process much faster and much more effectively than the FastICA-based and kernel FastICA-based monitoring methods as well as the NoisyICA1-based and NoisyICA2-based monitoring methods.

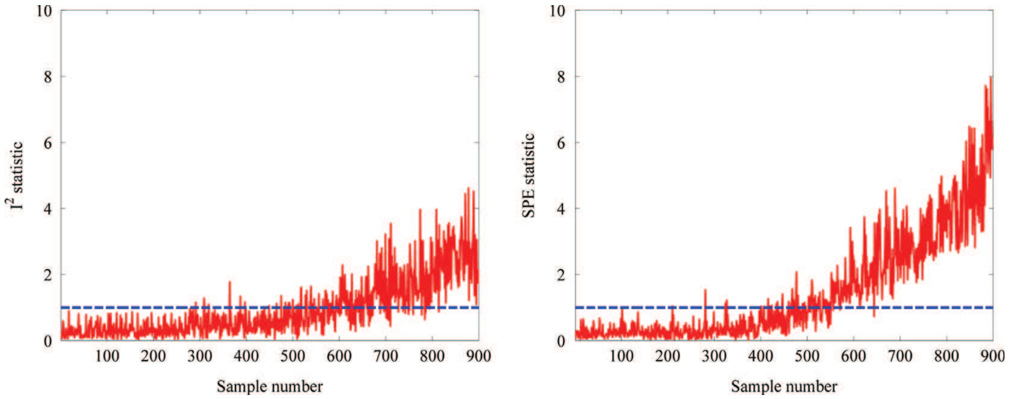

Fig. 8. The monitoring charts of the FastICA-based method for the CSTR under the fault pattern 5 which occurs at the 190th sample.

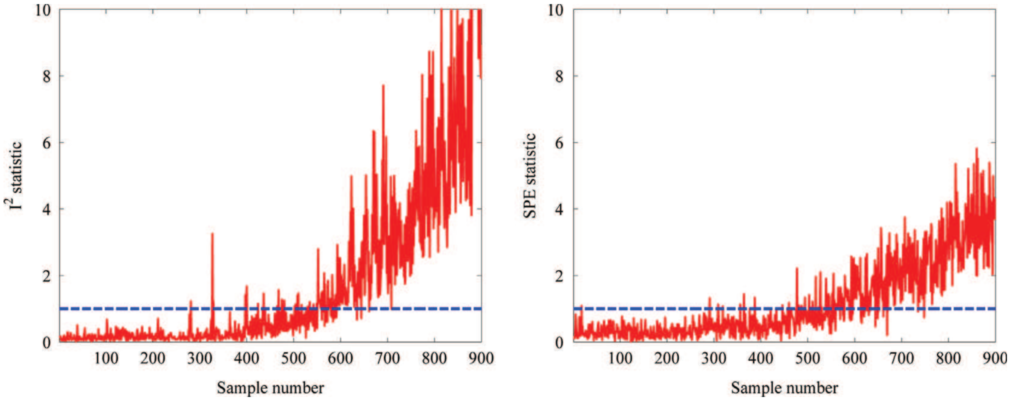

Fig. 9. The monitoring charts of the kernel FastICA-based method for the CSTR under the fault pattern 5 which occurs at the 190th sample.

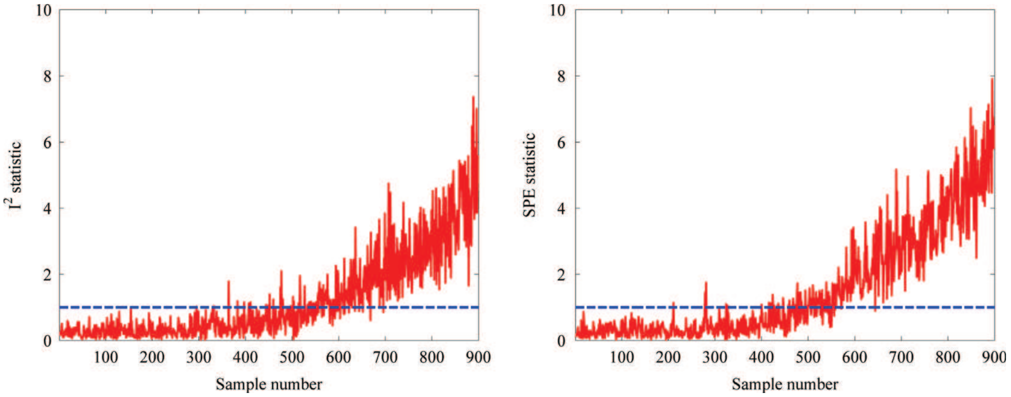

Fig. 10. The monitoring charts of the NoisyICA1-based method for the CSTR under the fault pattern 5 which occurs at the 190th sample.

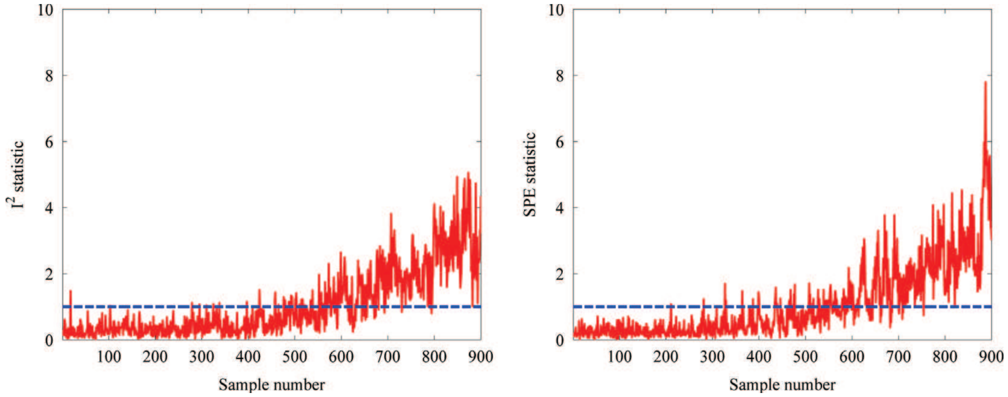

Fig. 11. The monitoring charts of the NoisyICA2-based method for the CSTR under the fault pattern 5 which occurs at the 190th sample.

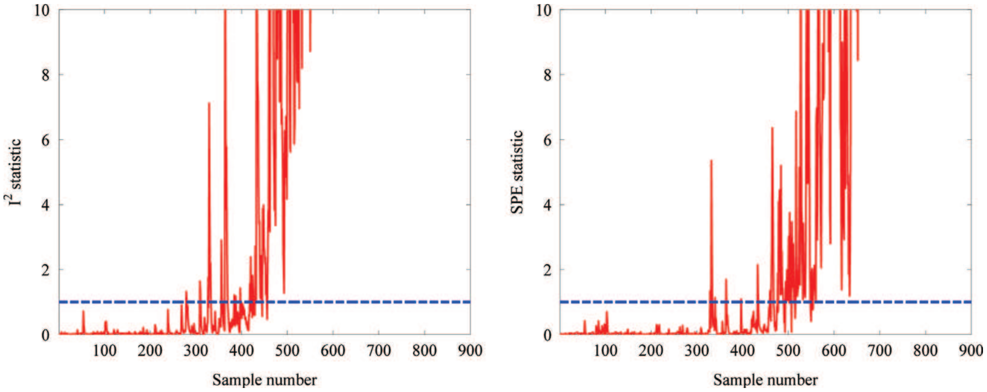

Fig. 12. The monitoring charts of the proposed NoisyICAn-based method for the CSTR under the fault pattern 5 which occurs at the 190th sample.

The monitoring charts obtained by all the five monitoring methods for the fault pattern 8 are examined in Figs. 13–17, respectively. In contrast to the monitoring charts of the FastICA-based, kernel FastICA-based, NoisyICA1-based and NoisyICA2-based methods, the monitoring charts of the NoisyICAn-based method react much more quickly and sharply to the occurring fault. Specifically, in the I^2 monitoring chart of the NoisyICAn-based method shown in Fig. 17, almost all the I^2 statistic's values exceed the corresponding confidence limit after the 400th sample, whereas in the I^2 monitoring charts of the four benchmark methods given in Figs. 13–16, respectively, many I^2 statistic's values from the 400th sample to the 450th sample are still below the related confidence limits. Furthermore, the SPE monitoring chart of the NoisyICAn-based method confidently indicates a fault after the 420th sample, whereas in the SPE monitoring charts of the other four methods, the SPE monitoring statistic's values still fluctuate around the confidence limits from the 420th sample to the 500th sample, and thus cannot give a definite fault indication. The results obtained thus confirm that the NoisyICAn-based method has better fault detection performance than the other four methods for the CSTR system under the fault pattern 8.

We next investigate the achievable monitoring performance of all the five monitoring methods on all the eight fault scenarios of the CSTR system. The fault detection times and the fault detection

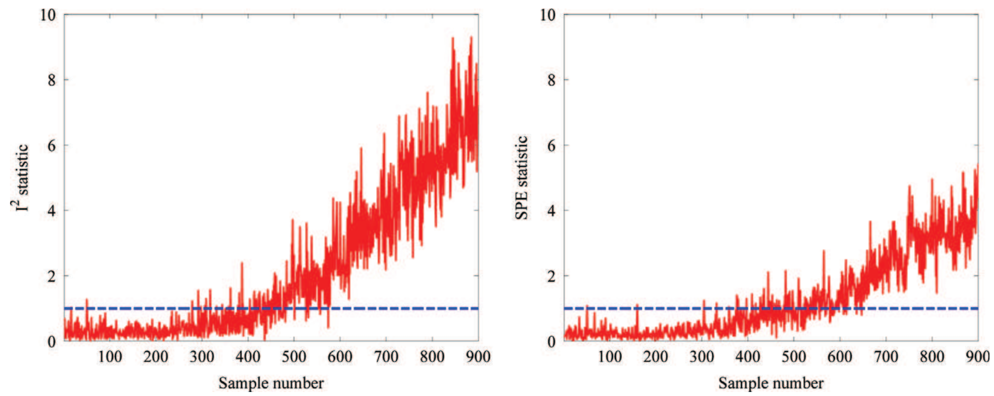

Fig. 13. The monitoring charts of the FastICA-based method for the CSTR under the fault pattern 8 which occurs at the 190th sample.

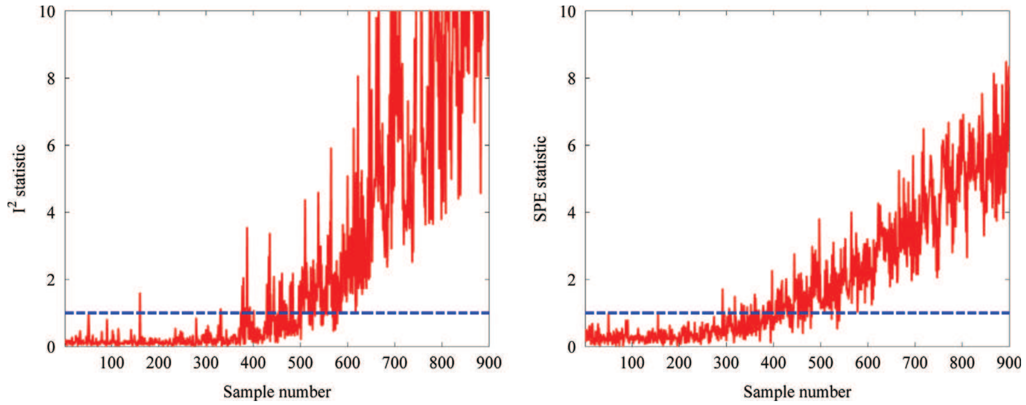

Fig. 14. The monitoring charts of the kernel FastICA-based method for the CSTR under the fault pattern 8 which occurs at the 190th sample.

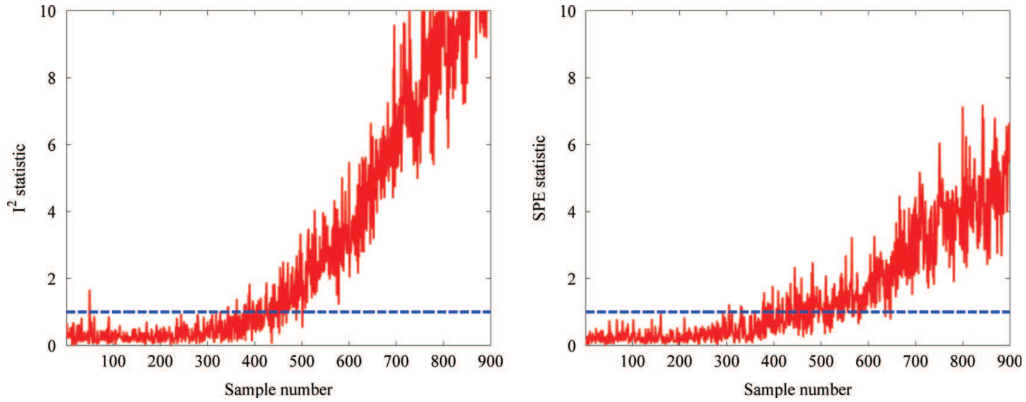

Fig. 15. The monitoring charts of the NoisyICA1-based method for the CSTR under the fault pattern 8 which occurs at the 190th sample.

rates obtained by these five methods are tabulated in Tables 5 and 6, respectively. From Table 5, it can be observed that all the five monitoring methods can detect the occurring fault immediately for the step-change fault patterns 1 to 3. Compared with the step-change faults, the ramp faults are more difficult to detect, as confirmed by the fault detection times for the ramp-change fault patterns 4 to 8. For the challenging ramp fault detection problems, our NoisyICAn-based method achieves much faster fault detection times than the FastICA-based, kernel FastICA-based, NoisyICA1-based and NoisyICA2-based methods, as can be seen from Table 5. Taking the fault pattern 4 as an example, the I^2 statistics of the four benchmark methods detect the fault at the 425th, 405th, 415th and 463th samples, respectively, while the I^2 statistic of the proposed NoisyICAn-based method detects the fault at the 323th sample. The SPE statistics of the FastICA-based, kernel FastICA-based, NoisyICA1-based, NoisyICA2-based and NoisyICAn-based methods on the other hand detect the fault at the 386th, 410th, 386th, 403th and 282th samples, respectively. The results of Table 5 clearly demonstrate the superior ability of the NoisyICAn-based method in shortening the fault detection delay for the challenging fault patterns 4 to 8, in comparison with the other four methods. From Table 6, all the five methods are seen to achieve 100% or close to 100% fault detection rates for the step-change faults 1 to 3. For the challenging ramp faults 4 to 8,

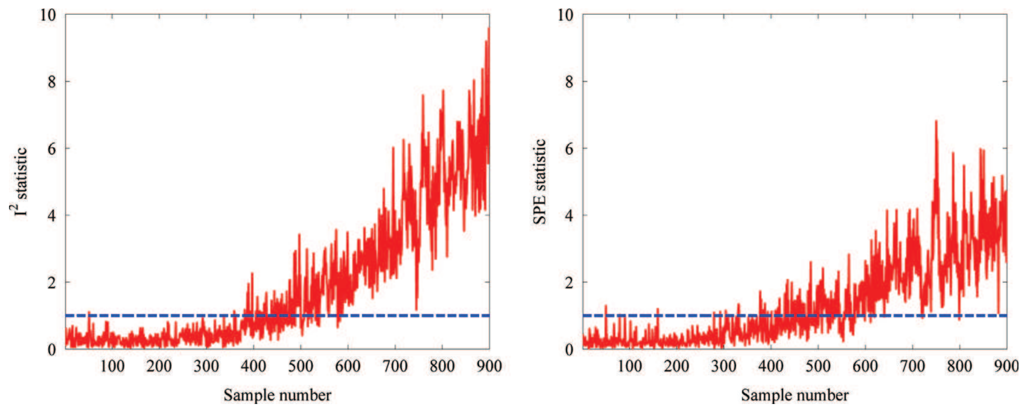

Fig. 16. The monitoring charts of the NoisyICA2-based method for the CSTR under the fault pattern 8 which occurs at the 190th sample.

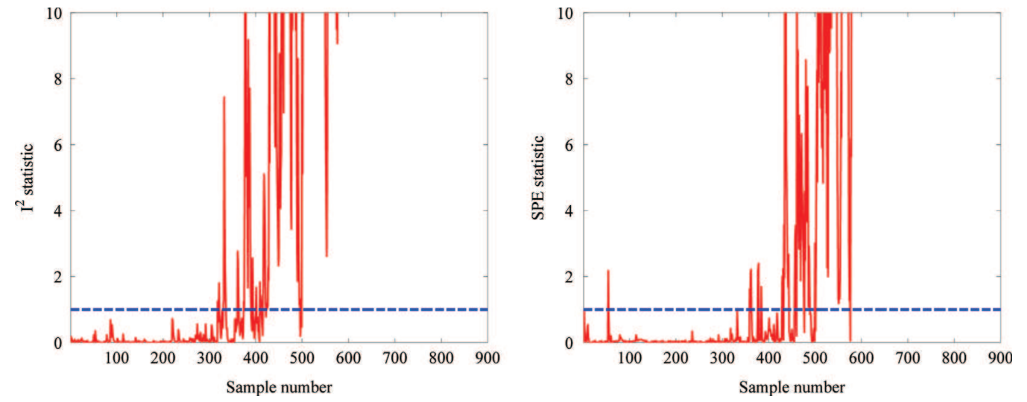

Fig. 17. The monitoring charts of the proposed NoisyICAn-based method for the CSTR under the fault pattern 8 which occurs at the 190th sample.

Table 5
Comparison of the fault detection times (sample number) for the CSTR system.

Fault no.	FastICA-based		Kernel FastICA-based		NoisyICA1-based		NoisyICA2-based		NoisyICAn-based	
	I^2	SPE	I^2	SPE	I^2	SPE	I^2	SPE	I^2	SPE
1	190	190	190	190	190	190	191	190	190	190
2	190	190	190	190	190	190	190	190	190	190
3	190	192	190	190	190	191	190	190	190	191
4	425	386	405	410	415	386	463	403	323	282
5	605	534	551	579	558	500	570	592	364	479
6	470	455	496	462	457	455	500	524	362	369
7	398	368	424	401	403	384	416	499	320	386
8	460	549	502	422	468	486	489	502	330	433

however, our NoisyICAn-based method is observed to attain higher fault detection rates than the other four methods.

The average fault detection rate of each monitoring statistic over the ramp faults 4 to 8 is illustrated in Fig. 18 for a more intuitive comparison. The results of Fig. 18 further confirm the superior monitoring performance of the NoisyICAn-based method over the conventional FastICA-based, kernel FastICA-based, NoisyICA1-based and NoisyICA2-based methods, especially for the challenging ramp faults with changing process parameters. We can also observe that the FastICA-based method exhibits better monitoring performance than the NoisyICA2-based method, whereas the NoisyICA1-based method performs better than the FastICA-based method. This can be attributed to the following reasons: (1) the NoisyICA2 algorithm assumes that the covariance matrix of the noise is a diagonal matrix with the same diagonal elements, which is not the case with the simulated situations; (2) the NoisyICA1 algorithm requires the covariance matrix of the noise, and in the simulation, the true noise's covariance matrix is provided to the NoisyICA1 algorithm. Obviously, in real industrial applications, the noise's covariance matrix must be estimated. As the estimate of the noise's covariance matrix is subject to error, the performance of the NoisyICA1-based method is likely to degrade. Both the noise-free FastICA-based and kernel FastICA-based methods achieve a similar performance. But the latter has a higher complexity than the former, as the latter conducts the high-dimensional nonlinear mapping using kernel function. Our NoisyICAn algorithm can effectively deal with the noise and it does not require the noise's covariance matrix. Based on this NoisyICAn method, the constructed monitoring statistics of Eqs. (26) and (27) can reduce the effect of the noise to a larger extent, compared with the calculated monitoring statistics of Eqs. (8) and (9) using the other four methods. Consequently, our NoisyICAn-based monitoring method can detect faults more quickly and reliably than the other four methods.

5. Conclusions

A NoisyICAn-based process monitoring method has been proposed. Our contribution is two-fold. First, we have developed the NoisyICAn algorithm to estimate the mixing matrix, which explicitly takes the measurement noise into consideration. Second, we have utilised the kurtosis relationship between the ICs and the measured variables to recursively estimate the ICs' kurtosis based

Table 6
Comparison of the fault detection rates for the CSTR system.

Fault no.	FastICA-based		Kernel FastICA-based		NoisyICA1-based		NoisyICA2-based		NoisyICAn-based	
	I^2 (%)	SPE (%)	I^2 (%)	SPE (%)	I^2 (%)	SPE (%)	I^2 (%)	SPE (%)	I^2 (%)	SPE (%)
1	100.0	100.0	100.0	100.0	99.86	100.0	100.0	100.0	100.0	100.0
2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3	100.0	99.72	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
4	70.28	70.00	69.58	70.70	71.41	74.37	66.48	67.46	81.55	84.08
5	41.55	56.76	50.85	48.31	49.58	57.46	47.61	46.67	70.14	60.28
6	64.51	66.06	61.41	67.18	65.07	67.46	62.68	60.14	76.20	74.79
7	76.34	72.25	70.28	71.41	76.76	68.31	74.37	60.00	82.96	70.00
8	64.79	54.93	59.15	69.72	66.62	59.86	61.83	58.45	72.39	63.38

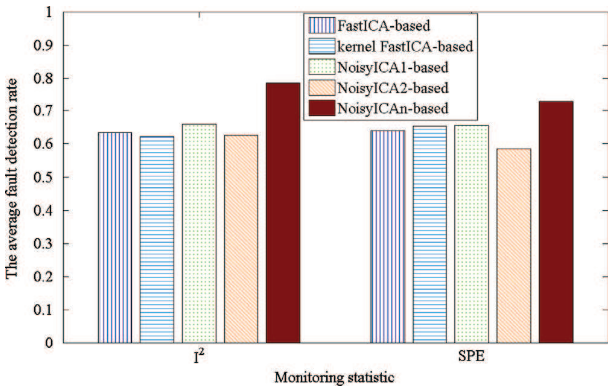

Fig. 18. Comparison of the average fault detection rates over the ramp-change fault patterns 4 to 8 of the CSTR process.

on the estimated mixing matrix, which effectively removes the influence of the Gaussian measurement noise. Two monitoring statistics, the I^2 and SPE statistics, are then built to detect process faults based on the recursive kurtosis estimation of the ICs. Simulation results obtained on the three-variable system and the CSTR system have demonstrated the superior process monitoring performance of the proposed NoisyICAn-based method, in terms of the fault detection time and fault detection rate, over the conventional FastICA-based and kernel FastICA-based monitoring methods as well as the two monitoring methods based on the two existing noisy ICA algorithms referred to as the NoisyICA1 and NoisyICA2, respectively. It is worth emphasizing that the NoisyICA1 algorithm requires the covariance matrix of the noise while the NoisyICA2 algorithm assumes the noise's covariance matrix being a diagonal matrix with the same diagonal elements. By contrast, our NoisyICAn algorithm does not require the knowledge of the noise's covariance matrix, and yet our NoisyICAn-based monitoring method is capable of outperforming both the NoisyICA1-based and NoisyICA2-based monitoring methods.

There are three important issues that should be noted. First, the measurement noises are assumed to follow Gaussian distributions in this paper. For the generic case where the measurement noises have arbitrary distributions, to develop an efficient monitoring method capable of handling non-Gaussian noise will be included in our future work. Second, as industrial processes usually exhibit nonlinear characteristics, further investigation to take into the consideration of nonlinear process behaviors in our proposed method will also be conducted. Third, in this paper, we implicitly assume that the process under monitoring is operating at one mode, as a pure ICA based algorithm without any enhancing preprocessing technique may not deal with the problem of multimode process monitoring well. In our future study, we will investigate how to integrate some appropriate enhancing techniques with our NoisyICAn-based algorithm to extend its application to multimode process monitoring problems.

Appendix A. Empirically choosing the learning rate μ

Currently, there exists no method for determining the optimal learning rate μ. This parameter is usually chosen by trial and error. We develop an empirical method to choose an appropriate value for μ. We now explain the basic idea of this heuristic method. In the off-line modelling stage, the process data measured under the normal operating conditions are used to form the training set and the validation set. Because the samples whose monitoring statistic values exceed the δ confidence limit are considered faulty samples that may associate with some fault, we choose the learning rate μ to make the monitoring statistic values of the validating samples below the confidence limit as much as possible. With this strategy, the region of normal operating conditions may be better preserved.

For the monitoring statistic I^2, we define an index to measure the difference between the δ confidence limit of I^2 and the monitoring statistic values $I^2(1), I^2(2), \dots, I^2(N_2)$ for the validating data, which is expressed as

$$\eta_1 = \frac{R_1}{R_2}, \tag{33}$$

where R_1 denotes the number of the monitoring statistic values between $I^2_{\text{lim},\delta}$ and $I^2_{\text{lim},\delta} - D_1$, and R_2 denotes the number of the monitoring statistic values less than $I^2_{\text{lim},\delta} - D_1$, while $I^2_{\text{lim},\delta}$ is the δ confidence limit of I^2, and D_1 is a predefined constant satisfying the condition $D_1 < I^2_{\text{lim},\delta}$.

For the monitoring statistic SPE, we define a similar index to measure the difference between the δ confidence limit of SPE and the monitoring statistic values $\text{SPE}(1), \text{SPE}(2), \dots, \text{SPE}(N_2)$ for the validating data as

$$\eta_2 = \frac{R_3}{R_4}, \tag{34}$$

where R_3 denotes the number of the monitoring statistic values between $\text{SPE}_{\text{lim},\delta}$ and $\text{SPE}_{\text{lim},\delta} - D_2$, and R_4 denotes the number of the monitoring statistic values less than $\text{SPE}_{\text{lim},\delta} - D_2$, while $\text{SPE}_{\text{lim},\delta}$ is the δ confidence limit of SPE, and D_2 is a predefined constant satisfying the condition $D_2 < \text{SPE}_{\text{lim},\delta}$.

A smaller value of η_1 or η_2 suggests that more monitoring statistic values are far below the confidence limit and, therefore, the region of normal operating conditions is better preserved. As the indexes η_1 and η_2 are directly influenced by the learning rate μ, they can be minimized to determine an appropriate value of μ, and the following search procedure is adopted to choose this learning rate.

(1) Set the search range for μ from $\mu_{\min}$ to $\mu_{\max}$ that covers all the possible choices of the learning rate. Choose the required δ value and set another value $\delta_0 < \delta$.
(2) Start with $\mu = \mu_{\min}$.
(3) Calculate the monitoring statistics' values $\{I^2(t)\}_{t=1}^{N_2}$ and $\{\text{SPE}(t)\}_{t=1}^{N_2}$ for the validating data.
(4) Determine the δ confidence limits $I^2_{\text{lim},\delta}$ and $\text{SPE}_{\text{lim},\delta}$ for I^2 and SPE, respectively.
If $\mu = \mu_{\min}$: also determine the δ_0 confidence limits $I^2_{\text{lim},\delta_0}$ and $\text{SPE}_{\text{lim},\delta_0}$ for I^2 and SPE, respectively, as well as set $D_1 = I^2_{\text{lim},\delta} - I^2_{\text{lim},\delta_0}$ and $D_2 = \text{SPE}_{lim,\delta} - \text{SPE}_{lim,\delta_0}$.
(5) Calculate the index η_1 of Eq. (33) and the index η_2 of Eq. (34).
(6) Set $\mu = \mu + \mu_{\min}$. If $\mu \le \mu_{\max}$ go to step (3); otherwise, stop.

The value μ in the search range $\mu_{\min}$ to $\mu_{\max}$ that approximately minimises η_1 and η_2 may be chosen as the appropriate learning rate.

The three-variable system. The search range for μ is set from $\mu_{\min} = 0.001$ to $\mu_{\max} = 0.5$, while the values of δ and δ_0 are set to 0.99 and 0.96, respectively. The indexes η_1 and η_2 as the functions of the learning rate μ are depicted in Fig. 19. It can be seen from Fig. 19 that a very small μ may result in a large number of normal-operation samples whose monitoring statistics' values are near the confidence limits and this may lead to high false alarming rate for on-line fault detection. Indeed, our simulation experience suggests that a too small μ may result in a large false alarming rate. On the other hand, our empirical experience also suggests that a too large μ may decrease the sensitivity of the algorithm to the occurring fault and, therefore, reduces the fault detection rate. These empirical experience together with the results of Fig. 19 suggest that the learning rate $\mu = 0.4$ is an appropriate choice which offers a good compromise between the false alarming rate and the fault detection rate.

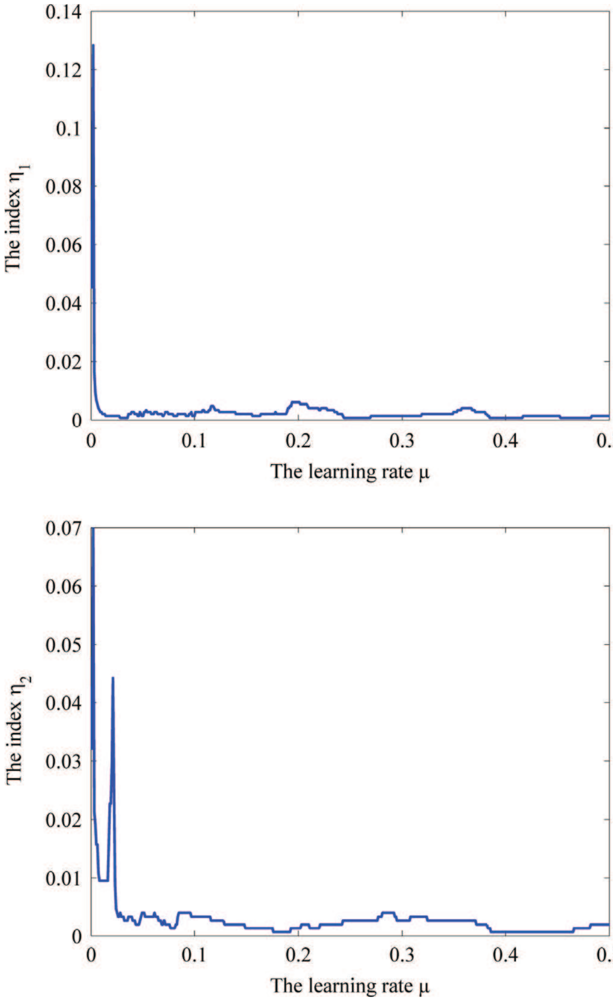

Fig. 19. The relationships between the indexes η_1 and η_2 and the learning rate μ for the three-variable system.

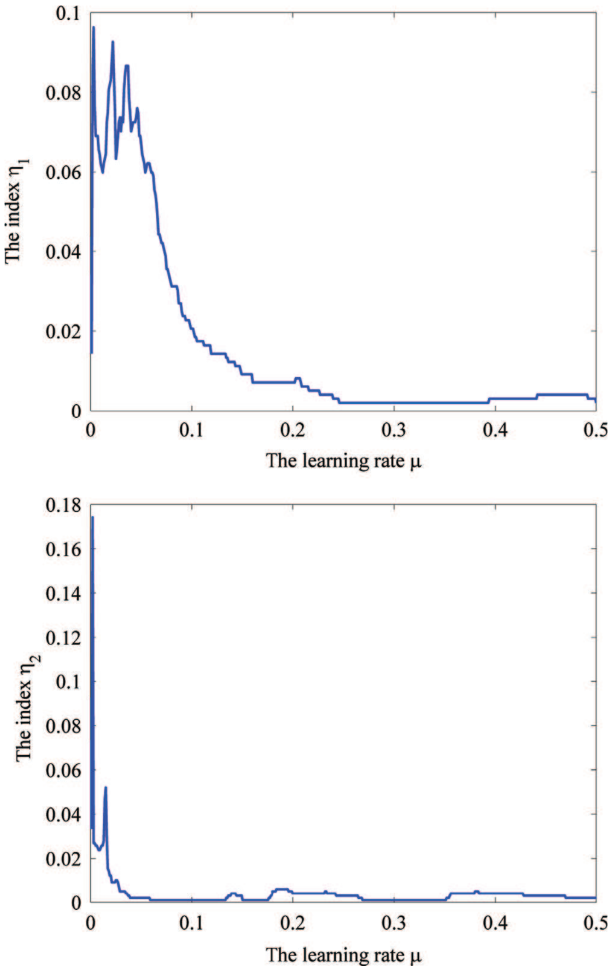

Fig. 20. The relationships between the indexes η_1 and η_2 and the learning rate μ for the CSTR system.

The CSTR system. The search range for μ is also set from $\mu_{\min} = 0.001$ to $\mu_{\max} = 0.5$, while the values of δ and δ_0 are also set to 0.99 and 0.96, respectively. The indexes η_1 and η_2 as the functions of the learning rate μ are plotted in Fig. 20. The results of Fig. 20 indicate that $\mu = 0.3$ is an appropriate choice.

Appendix B. Determining the number of the dominant ICs

We adopt a scheme similar to the one discussed in [21] to determine an appropriate number c of the dominant ICs. The cumulative percent variance (CPV) criterion is commonly applied in the ICA-based fault detection methods [14,16]. We construct a CPV criterion using the absolute values of the ICs' kurtosis statistics given in Eq. (18), to help determining an appropriate number c of the dominant ICs. Assume that the ICs $\boldsymbol{s}=[s_1 s_2 \ldots s_m]^T$ obtained are arranged in the descending order according to their non-Gaussian degrees which are measured by the absolute values of the ICs' kurtosis statistics given in Eq. (18). The CPV criterion can then be defined as

$$\mathrm{CPV}(c)=\frac{\sum_{i=1}^{c}|\mathrm{k}_4(s_i)|}{\sum_{i=1}^{m}|\mathrm{k}_4(s_i)|}\times 100\%. \tag{35}$$

CPV(c) can be set to 90%, 95% or 99% to determine the corresponding value of c, as described in [14].

For our both case studies, CPV$(c)=90\%$ is chosen to determine the number of the dominant ICs. For the simple three-variable system, this leads to $c=2$, while for the CSTR system, an appropriate number of the dominant ICs is found to be $c=5$. In order to conduct a fair comparison, we use the same c value for all the five ICA-based monitoring methods evaluated.

References

[1] J. Yu, A particle filter driven dynamic Gaussian mixture model approach for complex process monitoring and fault diagnosis, Journal of Process Control 22 (April (4)) (2012) 778–788.

[2] S. Stubbs, J. Zhang, J. Morris, Fault detection in dynamic processes using a simplified monitoring-specific CVA state space modelling approach, Computers & Chemical Engineering 41 (June (11)) (2012) 77–87.

[3] C.J. Lu, Y.E. Shao, P.H. Li, Mixture control chart patterns recognition using independent component analysis and support vector machine, Neurocomputing 74 (May (11)) (2011) 1908–1914.

[4] Y.W. Zhang, C. Ma, Fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS, Chemical Engineering Science 66 (January (1)) (2011) 64–72.

[5] I.B. Khediri, M. Limam, C. Weihs, Variable window adaptive kernel principal component analysis for nonlinear nonstationary process monitoring, Computers and Industrial Engineering 61 (October (3)) (2011) 437–446.

[6] C.Y. Cheng, C.C. Hsu, M.C. Chen, Adaptive kernel principal component analysis for monitoring small disturbances of nonlinear processes, Industrial and Engineering Chemistry Research 49 (January (5)) (2010) 2254–2262.

[7] J.M. Lee, S.J. Qin, I.B. Lee, Fault detection and diagnosis based on modified independent component analysis, AIChE Journal 52 (October (10)) (2006) 3501–3514.

[8] M.M. Rashid, J. Yu, A new dissimilarity method integrating multidimensional mutual information and independent component analysis for non-Gaussian dynamic process monitoring, Chemometrics and Intelligent Laboratory Systems 115 (June (15)) (2012) 44–58.

[9] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (4–5) (2000) 411–430.

[10] C.C. Hsu, M.C. Chen, L.S. Chen, Integrating independent component analysis and support vector machine for multivariate process monitoring, Computers and Industrial Engineering 59 (August (1)) (2010) 145–156.

[11] M. Kano, S. Tanaka, S. Hsebe, I. Hashimoto, H. Ohno, Monitoring independent components for fault detection, AIChE Journal 49 (April (4)) (2003) 969–976.

[12] J.M. Lee, C.K. Yoo, I.B. Lee, Statistical process monitoring with independent component analysis, Journal of Process Control 14 (August (5)) (2004) 467–485.

[13] J. Lee, B. Kang, S.H. Kang, Integrating independent component analysis and local outlier factor for plant-wide process monitoring, Journal of Process Control 21 (August (7)) (2011) 1011–1021.

[14] Y.W. Zhang, Y. Zhang, Fault detection of non-Gaussian processes based on modified independent component analysis, Chemical Engineering Science 65 (August (16)) (2010) 4630–4639.

[15] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Transactions on Neural Networks 10 (May (3)) (1999) 626–634.

[16] J.C. Wang, Y.B. Zhang, H. Cao, W.Z. Zhu, Dimension reduction method of independent component analysis for process monitoring based on minimum mean square error, Journal of Process Control 22 (February (2)) (2012) 477–487.

[17] P.P. Odiowei, Y. Cao, State-space independent component analysis for nonlinear dynamic process monitoring, Chemometrics and Intelligent Laboratory Systems 103 (August (1)) (2010) 59–65.

[18] C.C. Hsu, M.C. Chen, L.S. Chen, Intelligent ICA-SVM fault detector for non-Gaussian multivariate process monitoring, Expert Systems with Applications 37 (April (4)) (2010) 3264–3273.

[19] X. Liu, L. Xie, U. Kruger, T. Littler, S. Wang, Statistical-based monitoring of multivariate non-Gaussian systems, AIChE Journal 54 (September (9)) (2008) 2379–2391.

[20] Y.W. Zhang, Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM, Chemical Engineering Science 64 (March (5)) (2009) 801–811.

[21] X.M. Tian, X.L. Zhang, X.G. Deng, S. Chen, Multiway kernel independent component analysis based on feature samples for batch process monitoring, Neurocomputing 72 (March (7–9)) (2009) 1584–1596.

[22] L.F. Cai, X.M. Tian, N. Zhang, Non-Gaussian process fault detection method based on modified KICA, CIESC Journal 63 (9) (2012) 2864–2868. (in Chinese).

[23] M.M. Rashid, J. Yu, Hidden Markov model based adaptive independent component analysis approach for complex chemical process monitoring and fault detection, Industrial & Engineering Chemistry Research 51 (March (15)) (2012) 5506–5514.

[24] Y.W. Zhang, J.Y. An, H.L. Zhang, Monitoring of time-varying processes using kernel independent component analysis, Chemical Engineering Science 88 (January) (2013) 23–32.

[25] D. Wang, Robust data-driven modeling approach for real-time final product quality prediction in batch process operation, IEEE Transactions on Industrial Informatics 7 (May (2)) (2011) 371–377.

[26] A. Cichocki, S.C. Douglas, S. Amari, Robust techniques for independent component analysis (ICA) with noisy data, Neurocomputing 22 (November (1–3)) (1998) 113–129.

[27] A. Hyvärinen, Gaussian moments for noisy independent component analysis, IEEE Signal Processing Letters 6 (June (6)) (1999) 145–147.

[28] J.T. Cao, N. Murata, S. Amari, A. Cichocki, T. Takeda, A robust approach to independent component analysis of signals with high-level noise measurements, IEEE Transactions on Neural Networks 14 (May (3)) (2003) 631–645.

[29] W. Liu, D.P. Mandic, A. Cichocki, Blind second-order source extraction of instantaneous noisy mixtures, IEEE Transactions on Circuits and Systems II: Express Briefs 53 (September (9)) (2006) 931–935.

[30] Y.M. Yang, C.H. Guo, Gaussian moments for noisy unifying model, Neurocomputing 71 (October (16–18)) (2008) 3656–3659.

[31] M.L. Zhang, S.Z. Wang, X.Y. Jia, RLS adaptive noise cancellation via QR decomposition for noisy ICA, Advanced Materials Research 403–408 (November) (2012) 1291–1296.

[32] A. Yeredor, Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation, IEEE Transactions on Signal Processing 50 (July (7)) (2002) 1545–1553.

[33] A. Jbari, A. Adib, D. Aboutajdine, A new signal separation technique using double referenced system, in: Proceedings of IEEE Symposium on Computers and Communications 2008 (Marrakech, Morocco), July 6–8, 2008, pp. 670–673.

[34] X.D. Zhang, Time Series Analysis – High Order Statistics Method, Tsinghua University Press, Beijing, 1996, pp. 20-24 (in Chinese).

[35] R. Boscolo, H. Pan, V.P. Roychowdhury, Independent component analysis based on nonparametric density estimation, IEEE Transactions on Neural Networks 15 (January (1)) (2004) 55–65.

[36] H.H. Yang, S.I. Amari, Adaptive on-line learning algorithms for blind separation—maximum entropy and minimum mutual information, Neural Computation 9 (October (7)) (1997) 1457–1482.

[37] E.L. Russell, L.H. Chiang, R.D. Braatz, Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems 51 (May (1)) (2000) 81–93.

[38] S. Mahadevan, S.L. Shah, Fault detection and diagnosis in process data using one-class support vector machines, Journal of Process Control 19 (December (10)) (2009) 1627–1639.

[39] M.C. Johannesmeyer, A. Singhal, D.E. Seborg, Pattern matching in historical data, AIChE Journal 48 (September (9)) (2002) 2022–2038.

[40] J.M. Lee, S.J. Qin, I.B. Lee, Fault detection of non-linear process using kernel independent component analysis, Canadian Journal of Chemical Engineering 85 (August (4)) (2007) 526–536.

Lianfang Cai was born in Shandong, China, on September 30, 1986. He received his Bachelor degree in 2009 from China University of Petroleum. He is currently a Ph.D. candidate in College of Information and Control Engineering, China University of Petroleum, Qingdao, China. His current research interests focus on fault detection and diagnosis in industrial processes.

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 petrolchemical processes as well as fault detection and diagnosis, and process monitoring.

Sheng Chen received his B.Eng. degree from the East China Petroleum Institute, China, in January 1982, and his Ph.D degree from the City University, London, in September 1986, both in control engineering. In 2005, he was awarded the higher doctorate degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK. From 1986 to 1999, he held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with Electronics and Computer Science, the University of Southampton, UK, where he currently holds the post of Professor in Intelligent Systems and Signal Processing. Dr Chen's research interests include adaptive signal processing, wireless communications, modelling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods and optimisation. He has published over 480 research papers. Dr. Chen is a Fellow of IEEE and a Fellow of IET. He is a Distinguished Adjunct Professor at the King Abdulaziz University, Jeddah, Saudi Arabia. He is an ISI highly cited researcher in the engineering category (March 2004).