Computers & Industrial Engineering 59 (2010) 145–156

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Integrating independent component analysis and support vector machine for multivariate process monitoring ☆

Chun-Chin Hsu [a,*], Mu-Chen Chen [b], Long-Sheng Chen [c]

[a] *Department of Industrial Engineering and Management, Chaoyang University of Technology, 168 Jifong E. Rd., Wufong Township Taichung County 41349, Taiwan*
[b] *Institute of Traffic and Transportation, National Chiao Tung University, 114 Chung Hsiao W. Rd., Sec. 1, Taipei 10012, Taiwan*
[c] *Department of Information Management, Chaoyang University of Technology, 168 Jifong E. Rd., Wufong Township Taichung County 41349, Taiwan*

ARTICLE INFO

Article history:
Received 10 November 2008
Received in revised form 8 December 2009
Accepted 30 March 2010
Available online 2 April 2010

Keywords:
ICA
PCA
SVM
TE process
Fault detection rate

ABSTRACT

This study aims to develop an intelligent algorithm by integrating the independent component analysis (ICA) and support vector machine (SVM) for monitoring multivariate processes. For developing a successful SVM-based fault detector, the first step is feature extraction. In real industrial processes, process variables are rarely Gaussian distributed. Thus, this study proposes the application of ICA to extract the hidden information of a non-Gaussian process before conducting SVM. The proposed fault detector will be implemented via two simulated processes and a case study of the Tennessee Eastman process. Results demonstrate that the proposed method possesses superior fault detection when compared to conventional monitoring methods, including PCA, ICA, modified ICA, ICA–PCA and PCA–SVM.

1. Introduction

Quality is an important issue for today's competitive industries, so development of on-line process monitoring methods for maintaining the yield of products is required. Statistical process control (SPC) is a well-recognized tool for on-line monitoring of the process status. However, when the process contains several variables, the SPC will fail to detect process faults due to high correlations between process variables. Multivariate statistical process control (MSPC) provides a way for engineers to judge the process status. In general, two types of MSPC methods are available: extended traditional univariate SPC, and latent variable projection methods. The former uses original process variables to construct monitoring statistics, such as Hotelling's T^2 chart, exponentially weighted moving average (EWMA) and cumulative sum (CUMSUM) charts. Montgomery (2005) presents the details for related methods.

Nowadays, hundreds or thousands of variables can be recorded on-line every day due to the rapid advancement of computer technology. Thus, dimension reduction by using latent variable projection methods is an important pre-processing step before conducting MSPC. Principal Component Analysis (PCA) is primarily used in the area of chemometrics (e.g., Kourti & MacGregor, 1996; Wasterhuis, Gurden, & Smilde, 2000) but it is also very promising in any kind of multivariate analysis. Jackson (1959) initially developed a T^2 control chart based on PCA-extracted components. Furthermore, Jackson and Mudholkar (1979) introduced a residual analysis for PCA-based MSPC. Nomikos and MacGregor (1994, 1995) presented a multiway PCA method for batch process monitoring. Ku, Storer, and Georgakis (1995) proposed a dynamic PCA method by adding time-lagged variables to augment the original data matrix in order to capture the dynamic characteristics. Jia, Martin, and Morris (1998) proposed a non-linear PCA method for non-linear process fault detection. Furthermore, Shao, Jia, and Morris (1999) integrated wavelets and non-linear PCA for non-linear process monitoring. Lee, Yoo, and Lee (2004b) developed multiway kernel PCA for monitoring non-linear batch processes. As mentioned above, PCA has been successfully applied to multivariate process monitoring. However, PCA is limited to dealing with a Gaussian process because the PCA-extracted components are assumed to follow a Gaussian distribution. Martin and Morris (1996) reported that PCA-extracted components rarely conform to a multivariate Gaussian distribution in real industrial processes.

More recently, a newly developed feature extraction technique named independent component analysis (ICA) was proposed to deal with non-Gaussian processes. ICA was originally developed for signal processing applications, including speech signal processing, communications, medical image processing, financial engineering and so forth. ICA can be seen as an extension of PCA. However, the objectives for both algorithms are quite different.

☆ This manuscript was processed by Area Editor E.A. Elsayed.
* Corresponding author.
E-mail addresses: cchsu@cyut.edu.tw (C.-C. Hsu), ittchen@mail.nctu.edu.tw (M.-C. Chen), lschen@cyut.edu.tw (L.-S. Chen).

0360-8352/$ - see front matter Crown Copyright © 2010 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.03.011

PCA can only impose independence up to second order statistics information (i.e. covariance) and hence its objective is to decorrelate variables. Oppositely, ICA imposes statistical independence on the individual components by considering higher-order statistics. Generally speaking, the PCA possesses "weak" independence, whereas the ICA possesses "strong" independence. Thus, ICA can provide more information than PCA (Lee, Yoo, & Lee, 2004a).

Kano, Tanaka, Hasebe, Hashimoto, and Ohno (2003) compared the monitoring results by applying SPC charts to the components of PCA and ICA, respectively. Their results indicated that it is more efficient to monitor ICA components than PCA components when the behavior of process variables follows a non-Gaussian distribution. Although their work showed that the ICA technique can monitor a non-Gaussian process, it may produce false alarms when the components are individually monitored by SPC charts. Thus, Lee et al. (2004a) developed three ICA-based monitoring statistics to monitor the process. Furthermore, Lee, Qin, and Lee (2006) proposed a modified ICA to overcome the drawbacks of the original ICA algorithm such as pre-determination of both the number of extracted independent components and the proper order of independent components. Yoo, Lee, Vanrolleghem, and Lee (2004) developed a multiway ICA-based monitoring scheme for batch process monitoring. Lu, Wu, Keng, and Chiu (2006) applied ICA to integrating SPC and engineering process control (EPC). Ge and Song (2007) proposed a PCA–ICA method to extract Gaussian and non-Gaussian information for fault detection, and developed a similarity factor for fault diagnosis. Hsu, Chen, and Liu (2009) developed a process monitoring scheme based on ICA and adjusted outliers.

For a PCA-based monitoring method, the control limit for the monitored statistic (i.e. T^2) can be determined by an F distribution according to the Gaussian assumption. However, the control limits for ICA-based monitoring statistics cannot be determined by any specific distribution. Traditionally, the kernel density estimation (KDE) is used to determine the control limits for ICA-based monitoring statistics. However, KDE has two main limitations, the requirement of a large dataset and high sensitivity to the choice of smoothing parameter (Yoo et al., 2004). In addition, it does not perform well if the data are autocorrelated. Even though the use of a dynamic ICA method (DICA) (Lee, Yoo, & Lee, 2004c) could eliminate the autocorrelation of data, the pre-processing data matrix should be extended to contain the lagged variables, which will greatly increase the complexity of ICA computation and will require that more ICs be extracted for analysis.

In this study, a novel intelligent fault detector integrating two historical data-driven techniques, ICA and support vector machine (ICA–SVM), will be proposed. SVM is a supervised learning method which requires no assumption of data structure and which has been widely used for classification problems. There are many successful applications of SVM – for example, Shin, Eom, and Kim (2005) applied one-class SVM for machine fault detection and classification. Shieh and Yang (2008) applied fuzzy SVM to build classification model for product form design. Li (2009) used SVM for copper clad laminate defects classification. For developing a successful SVM-based fault detector, the first step is feature extraction. Liu (2009) proposed a novel wavelet feature extraction technique for reducing dimensionality of high-dimensional microarray data before conducting SVM. Widodo, Yang, Gu, and Choi (2009) developed an intelligent fault diagnosis system of induction motor by using ICA and PCA to extract the optimal features for SVM based classification process. Zhang (2009) applied KPCA, KICA and SVM for non-linear processes fault detection and diagnosis. Several literatures regarding component analysis and SVM can refer to Song and Wyrwicz (2009), Kwak (2008), Lu and Zhang (2007) and Widodo and Yang (2007).

The aforementioned literatures have showed advantages of integrating components analysis and SVM in many application areas. However, their works directly utilize extracted components as the inputs of SVM. Thus, this study further proposes to combine components into a statistic as the input feature of SVM. Generally speaking, the basic idea of this study first uses ICA to reduce the dimensions and extract independent components. Then, the extracted components are utilized to calculate the systematic statistic. To take the autocorrelation into account, the inputs of SVM are: (1) present time systematic statistic, (2) one time delay of systematic statistic and (3) the difference between two successive systematic statistics. The proposed ICA–SVM fault detector will be implemented via three examples: a simulated five-variable dynamic process, a simulated non-linear process and a case study of the Tennessee Eastman (TE) benchmark process. Results indicate that the proposed method possesses superior fault detection when compared to several monitoring schemes, including PCA, ICA, modified ICA, ICA–PCA and PCA–SVM.

The organization of this article is as follows. Section 2 first reviews the ICA algorithm and related process monitoring statistics; the modified ICA and ICA–PCA monitoring methods are also described. The SVM algorithm for a classification problem is introduced in Section 3. The proposed ICA–SVM fault detector is specified in Section 4. Section 5 first reports the simulation results of the proposed algorithm. Then, a case study of the Tennessee Eastman benchmark process is used to confirm the efficiency of the proposed methodology by comparing it to several traditional methods. The conclusion is finally given in Section 6.

2. ICA-based process monitoring methods

2.1. Theory of ICA algorithm

Fig. 1 shows the flowchart of the ICA algorithm. Consider a centered data matrix $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)] \in R^{d\times n}$, where $\mathbf{x} \in R^d$ is a column vector with d measured variables and n is the number of measurements which can be expressed as a linear combination of m unknown independent components (ICs) $\mathbf{s} = [s_1, s_2, \ldots, s_m] \in R^m$ (i.e. assume $E(\mathbf{s}\mathbf{s}^T) = \mathbf{I}$) that is,

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

where $\mathbf{A} \in R^{d\times m}$ is the mixing matrix and $\mathbf{E} \in R^{d\times n}$ is the residual matrix. ICA tries to estimate $\mathbf{A}$ and $\mathbf{s}$ from the only known $\mathbf{x}$. It is necessary to find a de-mixing matrix $\mathbf{W}$ which is given as

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

such that the reconstructed vector $\hat{\mathbf{s}}$ becomes as independent as possible.

The eigen-decomposition of covariance matrix $\mathbf{R_x} = E(\mathbf{x}\mathbf{x}^T)$ is given as

$$\mathbf{R_x} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{3}$$

where T denotes transpose operator, $\mathbf{U} \in R^{d\times d}$ is an orthogonal matrix of eigenvectors, and $\mathbf{\Lambda} \in R^{d\times d}$ is the diagonal matrix of eigenvalues. All score principal components (PCs) can be expressed as:

$$\mathbf{t} = \mathbf{U}^T\mathbf{x} \in R^d \tag{4}$$

By using only the first few several eigenvectors in descending order of the eigenvalues, the number of principal components in $\mathbf{t}$ can be reduced, and the reduced $\mathbf{t}$ is denoted as $\mathbf{t}' \in R^a$ where $a \leqslant d$. Denote $\mathbf{P} \in R^{d\times a}$ and $\mathbf{D} \in R^{a\times a}$ as the matrices of eigenvectors and eigenvalues, respectively. They are associated with the retained principal components such that $\mathbf{t}' = \mathbf{P}^T\mathbf{x}$.

Hotelling's T^2 can be used to measure the variation of the systematic part of the PCA model. T^2 is the sum of the normalized squared scores, that is

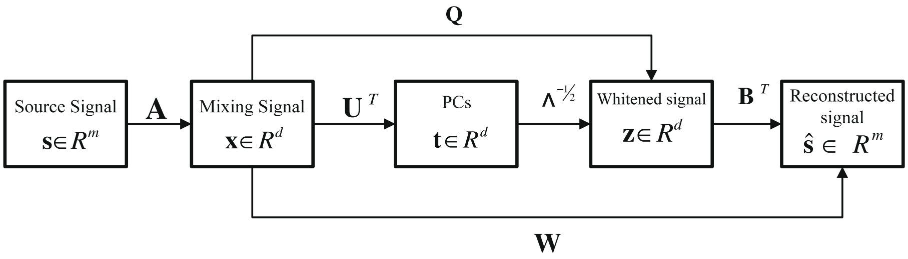


Fig. 1. Flowchart of ICA algorithm.

$$T^2 = \mathbf{t}'^T\mathbf{D}^{-1}\mathbf{t}' = \mathbf{x}^T\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^T\mathbf{x} \tag{5}$$

The upper confidence limit for T^2 is obtained by using F distribution

$$T^2_{a,n,\alpha} = \frac{a(n-1)}{n-a}F_{a,n-a,\alpha} \tag{6}$$

A measure of variation not captured by the PCA model can be monitored by squared prediction error (*SPE*).

$$SPE = \mathbf{e}^T\mathbf{e} = \mathbf{x}^T(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x} \tag{7}$$

where residual is $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \mathbf{P}\mathbf{t}' = (\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x}$ and $\mathbf{e} = 0$ when $a = d$. The upper control limit for *SPE* is

$$SPE_\alpha = \theta_1\left[\frac{c_\alpha\sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0(h_0-1)}{\theta_1^2}\right]^{1/h_0} \tag{8}$$

$\theta_g = \sum_{j=a+1}^{d}\lambda_j^g$ for $g = 1, 2, 3$, $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$ and c_α is the normal deviate corresponding to the upper $1-\alpha$ percentile.

The initial step for ICA is whitening, also known as sphering. The whitening transformation of $\mathbf{x}$ can be expressed as

$$\mathbf{z} = \mathbf{Q}\mathbf{x} = \mathbf{Q}\mathbf{A}\mathbf{s} = \mathbf{B}\mathbf{s} \tag{9}$$

where the whitening matrix $\mathbf{Q} = \Lambda^{-1/2}\mathbf{U}^T$ and $\mathbf{B}$ is an orthogonal demixing matrix (i.e.$E(\mathbf{z}\mathbf{z}^T) = \mathbf{B}E(\mathbf{s}\mathbf{s}^T)\mathbf{B}^T = 1$). The relationship between $\mathbf{W}$ and $\mathbf{B}$ is

$$\mathbf{W} = \mathbf{B}^T\mathbf{Q} \tag{10}$$

Therefore, Eq. (2) becomes

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} = \mathbf{B}^T\mathbf{z} = \mathbf{B}^T\mathbf{Q}\mathbf{x} = \mathbf{B}^T\Lambda^{-1/2}\mathbf{U}^T\mathbf{x} \tag{11}$$

According to Eq. (11), the ICA problem has therefore reduced the problem of finding an arbitrary $\mathbf{W}$ to the simpler problem of finding the matrix $\mathbf{B}$.

The objective of found $\mathbf{B}$ is to make $\hat{\mathbf{s}}$ becomes as independent as possible. Two non-Gaussian measurements methods can be applied: kurtosis and negentropy. The kurtosis method is simple but it is sensitive to outliers. The negentropy method is based on the information theoretic algorithm. Hyvärinen (1997) proposed a fix-point algorithm for ICA (fastICA). The algorithm calculates the column vector $\mathbf{b}_i$ $(i = 1, 2, \ldots, m)$ of $\mathbf{B}$ through iterative steps. For the detailed procedure, refer to Hyvärinen (1999), Hyvärinen and Oja (2000) and Hyvärinen, Karhunen, and Oja (2001). After $\mathbf{B}$ was obtained, we can calculate $\hat{\mathbf{s}}$ by using Eq. (11).

2.2. ICA-based process monitoring

To divide de-mixing matrix $\mathbf{W}$ into two parts: dominant part ($\mathbf{W}_d$) and excluded part ($\mathbf{W}_e$), three monitoring statistics were developed by Lee et al. (2004a) and they are shown as follows:

$$I^2 = \hat{\mathbf{s}}_e^T\hat{\mathbf{s}}_d \tag{12a}$$

$$I_e^2 = \hat{\mathbf{s}}_e^T\hat{\mathbf{s}}_e \tag{12b}$$

$$SPE = \mathbf{e}^T\mathbf{e} = (\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}) \tag{12c}$$

where $\hat{\mathbf{s}}_d = \mathbf{W}_d\mathbf{x}$, $\hat{\mathbf{s}}_e = \mathbf{W}_e\mathbf{x}$ and $\hat{\mathbf{x}} = \mathbf{Q}^{-1}\mathbf{B}_d\hat{\mathbf{s}} = \mathbf{Q}^{-1}\mathbf{B}_d\mathbf{W}_d\mathbf{x}$.

In PCA monitoring, the latent variables are assumed to be Gaussian distributed; hence the upper control limit for T^2 can be directly determined from the fitted distribution. However, the ICA components do not follow a specific distribution. Lee et al. (2004a) proposed to use the non-parametric technique, kernel density estimation (KDE), to determine the control limits for ICA. But using KDE has some shortcomings, such as the performance depending on the choice of window width, and KDE does not operate well when the dataset is autocorrelated. Thus, this study will apply an intelligent method, SVM, for detecting the faults. The next section will introduce the related algorithm.

3. Support vector machine

Support vector machine (SVM) is an effective machine learning method for classification problems, and eventually results in better generalization performance than most traditional methods (Cao, Chua, Chong, Lee, & Gu, 2003). SVM first maps input vectors into a higher feature space, either linearly or non-linearly, where a maximum separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separates the data. The separating hyperplane maximizes the distance between the two parallel hyperplanes. An excellent description of the SVM theory can be seen in Vapnik's book (1995). We give a brief overview of SVM for binary classification problem herein.

3.1. The linearly separable case

The input vectors $\mathbf{x}_i \in R^d$ $(i = 1, 2, \ldots, n)$ correspond to labels $y_i \in \{-1, +1\}$. There exists a separating hyperplane, the function of which is

$$\boldsymbol{\delta}\mathbf{x} + b = 0 \tag{13}$$

where $\boldsymbol{\delta} \in R^n$ is a normal vector, the bias b is a scale. Two parallel hyperplanes can be represented as

$$Y_i(\boldsymbol{\delta}\mathbf{x}_i + b) \geqslant 1 \tag{14}$$

SVM tries to maximize the margin between two classes, where the margin width between the two parallel hyperplanes equals $\frac{2}{\|\boldsymbol{\delta}\|}$. Therefore, for a linearly separable case, one can find the optimal hyperplane by solving the following quadratic optimization problem:

$$\begin{aligned} &\text{Min} \quad \frac{1}{2}\|\boldsymbol{\delta}\|^2 \\ &\text{s.t.} \quad y_i(\boldsymbol{\delta}\mathbf{x}_i + b) \geqslant 1 \end{aligned} \tag{15}$$

By introducing Lagrange multipliers α_i ($i = 1, 2, \ldots, n$) for the constraint, the primal problem becomes a task of finding the saddle point of Lagrange. Thus, the dual problem becomes

$$\begin{aligned} \text{Max} \quad & L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \geqslant 0 \end{aligned} \tag{16}$$

By applying Karush–Kukn–Tucker (KKT) conditions, the following relationship holds.

$$\alpha_i[y_i(\boldsymbol{\delta}\mathbf{x}_i + b) - 1] = 0 \tag{17}$$

If $\alpha_i > 0$, the corresponding data points are called support vectors (SVs). Hence, the optimal solution for the normal vector is given by

$$\boldsymbol{\delta}^* = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{18}$$

where N is the number of SVs. By choosing any SVs ($\mathbf{x}_k$, y_k), we can obtain $b^* = y_k - \boldsymbol{\delta}^*\mathbf{x}_k$. After ($\boldsymbol{\delta}^*$, b^*) is determined, the discrimination function can be given by

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i) + b^*\right) \tag{19}$$

where sgn(.) is the sign function.

3.2. The linearly non-separable case

SVM tries to map input vector $\mathbf{x}_i \in R^d$ into a higher feature space, and can thus solve the linearly non-separable case. The mapping process is based on the chosen kernel function. Some popular kernel functions are listed as follows:

$$\text{Linear kernel} \quad K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i\mathbf{x}_j \tag{20a}$$

$$\text{Polynomial kernel of degree } g \quad K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma\mathbf{x}_i\mathbf{x}_j + r)^g, \quad \gamma > 0 \tag{20b}$$

$$\text{Radial basis function} \quad K(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\}, \quad \gamma > 0 \tag{20c}$$

$$\text{Sigmoid kernel} \quad K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma\mathbf{x}_i\mathbf{x}_j + r), \quad \gamma > 0 \tag{20d}$$

where r, γ and g are kernel parameters. Hence, the discrimination function takes the form

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \cdot K(\mathbf{x} \cdot \mathbf{x}_i) + b^*\right) \tag{21}$$

Unlike most of the traditional methods which implement the empirical risk minimization principle, SVM implements the structural risk minimization principle, which can eventually result in better generalization performance. Besides, SVM makes no assumptions regarding the dataset, and only requires 'normal' and 'abnormal' data. Therefore, this study applies SVM as the classifier for fault detection. The next section interprets the proposed methodology.

4. Develop ICA–SVM fault detection method

The ICA-based statistic, I^2, is usually time dependent, especially for chemical or biological processes, for example, a dataset from the Tennessee Eastman (TE) benchmark process, which was generated by Chiang, Russell, and Braatz (2001) and can be downloaded from http://brahms.scs.uiuc.edu. To execute FastICA for the TE normal operation dataset (i.e. IDV(0)) and then plot the autocorrelation function (ACF) to I^2, Fig. 2 demonstrates high autocorrelation of I^2. Even though the dynamic ICA (Lee et al., 2004c) can eliminate the autocorrelation of data, the original data matrix should be extended to include lagged variables which will increase the computation complexity, and more ICs should be extracted for analysis.

Therefore, this study aims to develop an ICA–SVM fault detection method for a non-Gaussian multivariate process. Fig. 3 shows the architecture of the proposed methodology. At first, the feature extraction based on ICA is used to project the high dimension dataset into a lower one. The extracted ICs are then used to calculate the systematic part statistic. To take autocorrelation into account, the time delay and time difference of systematic statistics are also considered as input vectors for ICA–SVM. Development of ICA–SVM fault detector contains two phases, off-line training and on-line testing. The detailed procedure is described as follows.

4.1. Phase I: off-line ICA–SVM training

This phase attempts to build a referenced knowledge for ICA–SVM which considers the development of normal operation condition (NOC) and fault operation condition (FOC) datasets.

4.2. NOC training dataset development

Step 1: Scale NOC dataset

Obtain a NOC dataset (without shifts in the process), denoted as $\mathbf{x}_{normal}$. The first step focuses on centering and whitening $\mathbf{x}_{normal}$, and then denoted as $\mathbf{z}_{normal}$. This step eliminates most cross-correlation between the observed variables.

Step 2: Execute FastICA algorithm

Initially let $d = m$. By using the FastICA algorithm over $\mathbf{z}_{normal}$, we can obtain the de-mixing matrix $\mathbf{W}_{normal}$ and the corresponding orthogonal de-mixing matrix $\mathbf{B}_{normal}$. Therefore, the reconstructed signal is given by $\hat{\mathbf{s}}_{normal} = \mathbf{B}_{normal}^T\mathbf{z}_{normal}$.

Step 3: Perform dimension reduction

The order of $\hat{\mathbf{s}}_{normal}$ is determined by the largest sum of squares coefficient, that is

$$\text{Arg}_i\text{Max}\|\mathbf{w}_i\|_2 \tag{22}$$

where $\|\cdot\|$ denotes Euclidean norm (L_2) and $\mathbf{w}_i$ is the row vector in $\mathbf{W}_{normal}$. There are several methods for selecting the number of ICs such as cross-validation (Wold, 1978), majority of non-Gaussianity, and variance of reconstruction error (Valle, Li, & Qin, 1999). However, there is no standard criterion to determine the number of ICs.

Step 4: Calculate the systematic part statistics

After performing dimension reduction, the dominant de-mixing matrix $\mathbf{W}_d$ can be obtained and the relationship to $\mathbf{B}_d$ is

$$\mathbf{B}_d = (\mathbf{W}_d\mathbf{Q}^{-1})^T \tag{23}$$

Hence, the dominant ICs are calculated by

$$\hat{\mathbf{s}}_{normal\_d} = \mathbf{B}_d^T\mathbf{z}_{normal} \tag{24}$$

The systematic part statistic at sample t is

$$I_{normal}^2(t) = \hat{\mathbf{s}}_{normal\_d}^T(t)\hat{\mathbf{s}}_{normal\_d}(t) \tag{25}$$

Eq. (25) is one of the input features for ICA–SVM. Additionally, the time delay and time difference are also considered as input features for ICA–SVM, that is

$$I_{normal}^2 = [I_{normal}^2(t), I_{normal}^2(t-1), I_{normal}^2(t) - I_{normal}^2(t-1)] \tag{26}$$

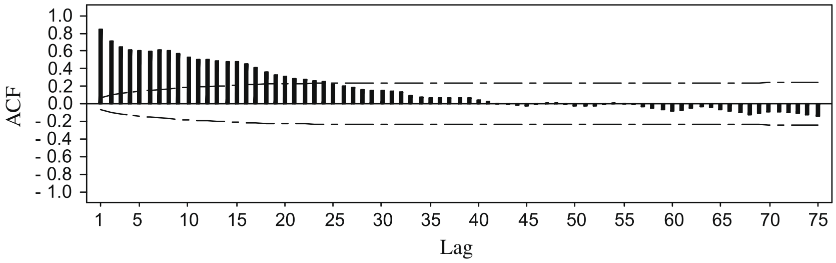


Fig. 2. ACF plot of I^2 for the TE process with normal operation.

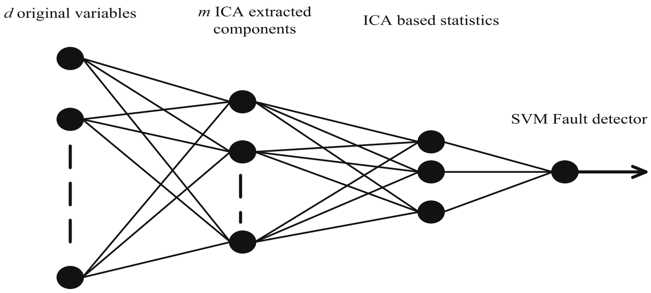


Fig. 3. Architecture of ICA–SVM fault detector.

4.3. FOC training dataset development

FOC dataset is also scaled at first, denoted as $\mathbf{z}_{fault}$. The dominant ICs under FOC can be calculated by

$$\hat{\mathbf{s}}_{fault\_d} = \mathbf{B}_d^T\mathbf{z}_{fault} \tag{27}$$

The systematic part statistic at sample t under FOC is

$$I^2_{fault}(t) = \hat{\mathbf{s}}^T_{fault\_d}(t)\hat{\mathbf{s}}_{fault\_d}(t) \tag{28}$$

The time delay and time difference are considered as ICA–SVM input vectors,

$$I^2_{fault} = [I^2_{fault}(t), I^2_{fault}(t-1), I^2_{fault}(t) - I^2_{fault}(t-1)] \tag{29}$$

After separately developing the NOC and FOC datasets, the training dataset merges Eqs. (26) and (29) for ICA–SVM training, and denoted as $I^2_{train} = [I^2_{normal}, I^2_{fault}]$.

4.4. Phase II: on-line ICA–SVM testing

The objective of this phase is to test the trained ICA–SVM model. Once the new data are obtained, the same scaling is then applied, and the scaled dataset is then denoted as $\mathbf{z}_{new}$. The dominant ICs of $\mathbf{z}_{new}$ can be obtained from

$$\hat{\mathbf{s}}_{new\_d} = \mathbf{B}_d^T\mathbf{z}_{new} \tag{30}$$

The statistic of systematic part at time t is

$$I^2_{new}(t) = \hat{\mathbf{s}}^T_{new\_d}(t)\hat{\mathbf{s}}_{new\_d}(t) \tag{31}$$

The testing input vector for ICA–SVM is

$$I^2_{new} = [I^2_{new}(t), I^2_{new}(t-1), I^2_{new}(t) - I^2_{newt}(t-1)] \tag{32}$$

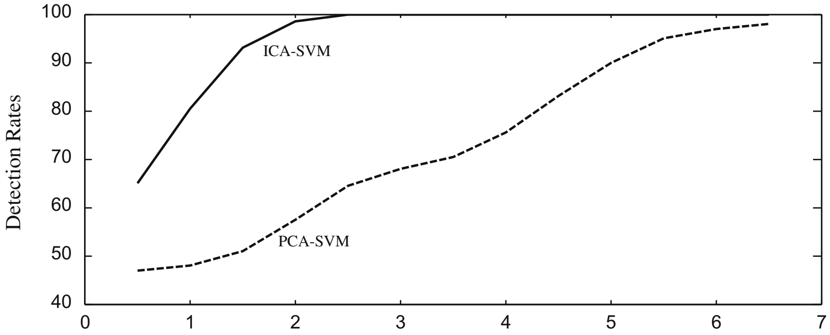


Fig. 4. Compare detection rates against step changed sizes.

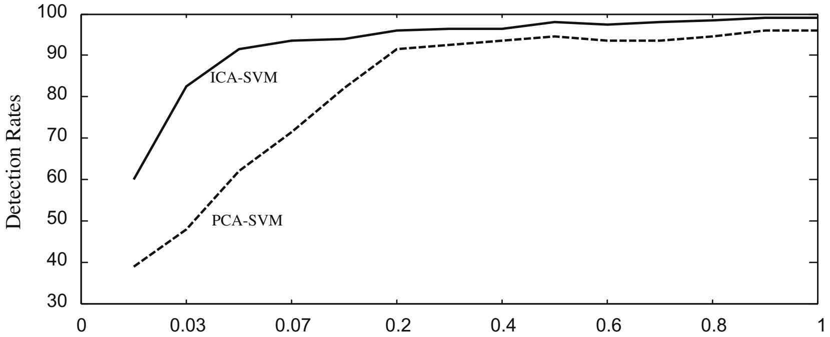


Fig. 5. Compare detection rates against slope sizes.

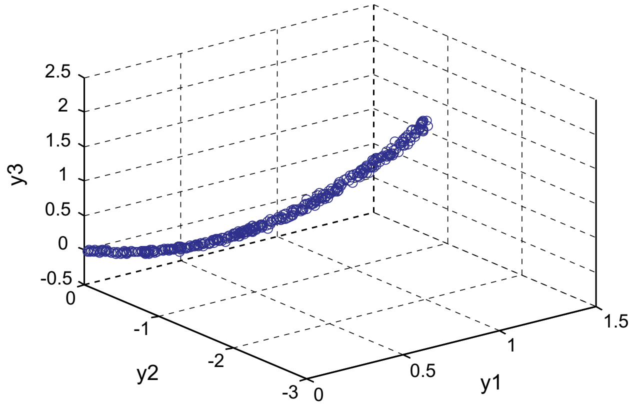


Fig. 6. Scatter plot of y_1, y_2 and y_3.

Table 1
Comparison between PCA–SVM and ICA–SVM.

	PCA–SVM	ICA–SVM
C	0.5	2048
γ	2.0	0.0078125
Detection rate (%)	50.75	92.5

and the discrimination function is given by

$$f(I^2_{new}) = \operatorname{sgn}\left(\sum_{i=1} \alpha_i y_i \cdot K(I^2_{train,i} \cdot I^2_{new}) + b^*\right) \tag{33a}$$

$$I^2_{new} \in \begin{cases} +1 & \text{if} \quad f(I^2_{new}) > 0 \\ -1 & \text{if} \quad f(I^2_{new}) < 0 \end{cases} \tag{33b}$$

When $I^2_{new} \in 1$, it denotes that the process is under NOC condition, whereas $I^2_{new} \in -1$ indicates the process is in an FOC condition.

The proposed method is verified in the next section by implementing two simulated multivariate processes and a case study of the TE process, which characterize non-Gaussian property. Besides, several traditional methods are also performed in order to describe the efficiency of the proposed methodology.

5. Implementation

At first, the ICA–SVM fault detector was implemented via a simulated five-variable dynamic process in which the step and linear disturbance models were introduced in the process. Further, the proposed ICA–SVM is also applied to monitor the non-linear process. After that, a case study of Tennessee Eastman (TE) benchmark process was implemented to illustrate the efficiency of ICA–SVM, in which several traditional methods including PCA, ICA, modified ICA and ICA–PCA were also implemented in order to make a comparison.

5.1. A five-variable simulation example

Consider a multivariate process with five variables similar to Lee et al. (2004a)

$$\mathbf{R}(k) = \begin{bmatrix} 0.118 & -0.191 & 0.287 \\ 0.847 & 0.264 & 0.943 \\ -0.333 & 0.514 & -0.217 \end{bmatrix} \mathbf{R}(k-1) + \begin{bmatrix} 1 & 2 \\ 3 & -4 \\ -2 & 1 \end{bmatrix} \mathbf{H}(k-1)$$
$$\mathbf{C}(k) = \mathbf{R}(k) + \mathbf{E}(k) \tag{34}$$

where **E** is assumed to be normally distributed, with zero mean and variance of 0.1. **H** is the input that can be obtained as follows:

$$\mathbf{H}(k) = \begin{bmatrix} 0.811 & -0.226 \\ 0.477 & 0.415 \end{bmatrix} \mathbf{H}(k-1) + \begin{bmatrix} 0.193 & 0.689 \\ -0.320 & -0.749 \end{bmatrix} \mathbf{G}(k-1) \tag{35}$$

G = [g_1, g_2] is the input that follows a uniform distribution between −2 and 2. Both input **H** = [h_1, h_2] and output **C** = [c_1, c_2, c_3] are used for analysis.

Two faults of step and linear change for g_1 were introduced respectively after sampling 50 through 200 observations. Three

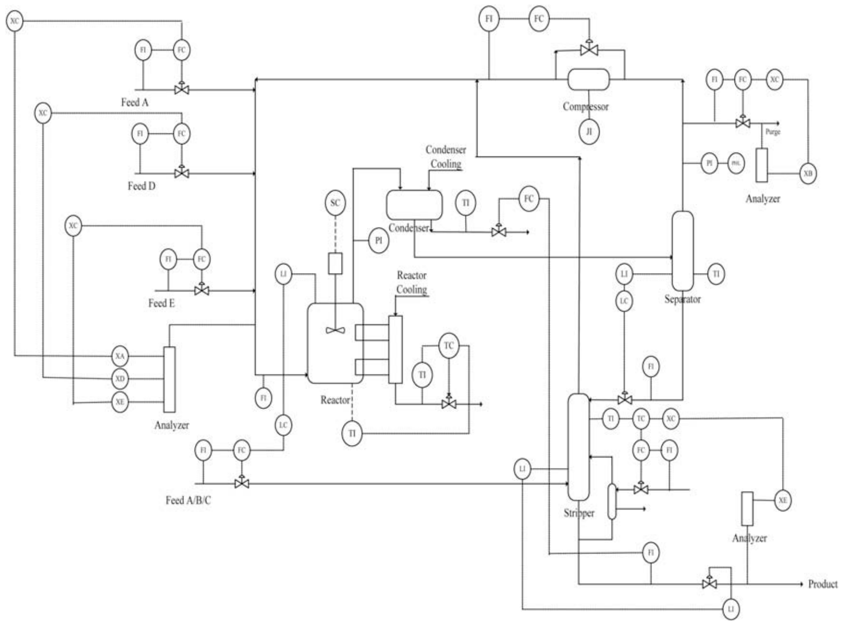


Fig. 7. Process layout of the TE process (Lee et al., 2006).

PCs were selected that capture approximately 88.1% of the variance. Also, the three ICs were chosen for analysis. The PCA-based SVM (PCA–SVM) is used as the benchmark for comparison, and the input vector for PCA–SVM is $[T^2(t), T^2(t-1), T^2(t) - T^2(t-1)]$. To perform SVM, the RBF is utilized which is one of the most widely used kernel functions in SVM applications (Keerthi & Lin, 2003; Lessmann & Voß, 2009). The RBF kernel function only needs to tune of its two parameters penalty cost C and γ, which facilitate adapting the classifier to a particular task. However, there is no theoretical framework to specify the optimal values of kernel parameters. In LIBSVM (Chang & Lin, 2001), the optimal settings of these two parameters can be found by using grid search. Readers can find more detailed information regarding gird search in (Hsu, Chang, & Lin, 2006).

Fig. 4 shows the detection rates against step changed sizes. It indicates ICA–SVM can detect the fault more efficiently than PCA–SVM, irrespective of shift sizes. It is noted that even with small shifts of process (say, 1.5), the ICA–SVM can achieve a high detection rate.

Fig. 5 shows the detection rates against slope sizes. It indicates that ICA–SVM and PCA–SVM can get good results for the large slope sizes (>0.2). It is noted that ICA–SVM has the capability to achieve above a 90% detection rate when the slope size is near 0.05. In the next section, the efficiency of proposed ICA–SVM will be further verified via implementing a non-linear process.

5.2. A simulated non-linear process

In this section, the proposed method will be implemented to monitor a non-linear process. Consider a non-linear system given by (Dong & McAvoy, 1996; Ge & Song, 2008):

$$\begin{aligned} y_1 &= t + e_1 \\ y_2 &= t^2 - 3t + e_2 \\ y_3 &= -t^3 + 3t^2 + e_3 \end{aligned} \tag{36}$$

where e_1, e_2, e_3 are independent random noises following the distribution of $N(0, 0.01)$ and the system input is $t \in [0.7, 1.2]$. In the first 200 samples, data are calculated according to Eq. (36) and these data are taken as the normal operating condition. After the first 200 samples, the system is then changed to

$$\begin{aligned} y_1 &= t + e_1 \\ y_2 &= t^2 - 3t + e_2 \\ y_3 &= -1.1t^3 + 3.2t^2 + e_3 \end{aligned} \tag{37}$$

From Eq. (37), it is known that there is a small change in y_3. This condition can be viewed as a fault condition for the system. Fig. 6 shows the data sets for the normal and condition and fault condition. Clearly, it is difficult to directly discriminate the two conditions.

To perform SVM, 400 samples are obtained to build the model. Besides, 400 samples are simulated as the testing dataset in which the first 200 samples are simulated according to Eq. (36) and the latter 200 samples are simulated from Eq. (37). The RBF kernel function is used to conduct SVM and the optimal settings of parameters can be found by using grid search.

To compare the performance between PCA–SVM and ICA–SVM, the percentage of correctly identify abnormal samples after the fault occurrence (i.e. after sample 200) will be calculated. Table 1 shows the comparison result along with the optimal settings of SVM parameters. It can be seen from the results that ICA–SVM can be used to monitor the non-linear system more effectively than PCA–SVM. This is because the ICA considers higher-order statistics than PCA, so can provide more information for SVM to detect faults. The next section is a case study of the TE benchmark process, and several multivariate monitoring methods will be compared to interpret the efficiency of ICA–SVM.

Table 2
Monitored variables in the TE process.

No.	Process measurements	No.	Manipulated variables
XMEAS(1)	*A* feed (stream 1)	XMV(1)	*D* feed flow (stream 2)
XMEAS(2)	*D* feed (stream 2)	XMV(2)	*E* feed flow (stream 3)
XMEAS(3)	*E* feed (stream 3)	XMV(3)	*A* feed flow (stream 1)
XMEAS(4)	*A* and *C* feed (stream 4)	XMV(4)	*A* and *C* feed flow (stream 4)
XMEAS(5)	Recycle flow (stream 8)	XMV(5)	Compressor recycle valve
XMEAS(6)	Reactor feed rate (stream 6)	XMV(6)	Purge valve (stream 9)
XMEAS(7)	Reactor pressure	XMV(7)	Separator pot liquid flow (stream 10)
XMEAS(8)	Reactor level	XMV(8)	Stripper liquid product flow (stream 11)
XMEAS(9)	Reactor temperature	XMV(9)	Stripper steam valve
XMEAS(10)	Purge rate (stream 9)	XMV(10)	Reactor cooling water valve
XMEAS(11)	Product separator temperature	XMV(11)	Condenser cooling water flow
XMEAS(12)	Product separator level		
XMEAS(13)	Product separator pressure		
XMEAS(14)	Product separator underflow (stream 10)		
XMEAS(15)	Stripper level		
XMEAS(16)	Stripper pressure		
XMEAS(17)	Stripper underflow (stream 11)		
XMEAS(18)	Stripper temperature		
XMEAS(19)	Stripper steam flow		
XMEAS(20)	Compressor work		
XMEAS(21)	Reactor cooling water outlet temp		
XMEAS(22)	Separator cooling water outlet temp		

Table 3
Process disturbance.

No.	State	Disturbance
IDV(1)	*A*/*C* feed ratio, *B* composition constant (stream 4)	Step
IDV(2)	*B* composition, *A*/*C* ratio constant (stream 4)	Step
IDV(3)	*D* feed temperature (stream 2)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	*A* feed loss (stream 1)	Step
IDV(7)	*C* header pressure loss – reduced availability (stream 4)	Step
IDV(8)	*A*, *B*, *C* feed composition (stream 4)	Random variation
IDV(9)	*D* feed temperature (stream 2)	Random variation
IDV(10)	*C* feed temperature (stream 4)	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)	Unknown	Unknown
IDV(17)	Unknown	Unknown
IDV(18)	Unknown	Unknown
IDV(19)	Unknown	Unknown
IDV(20)	Unknown	Unknown

5.3. A case study: Tennessee Eastman benchmark process

5.3.1. Description of Tennessee Eastman process

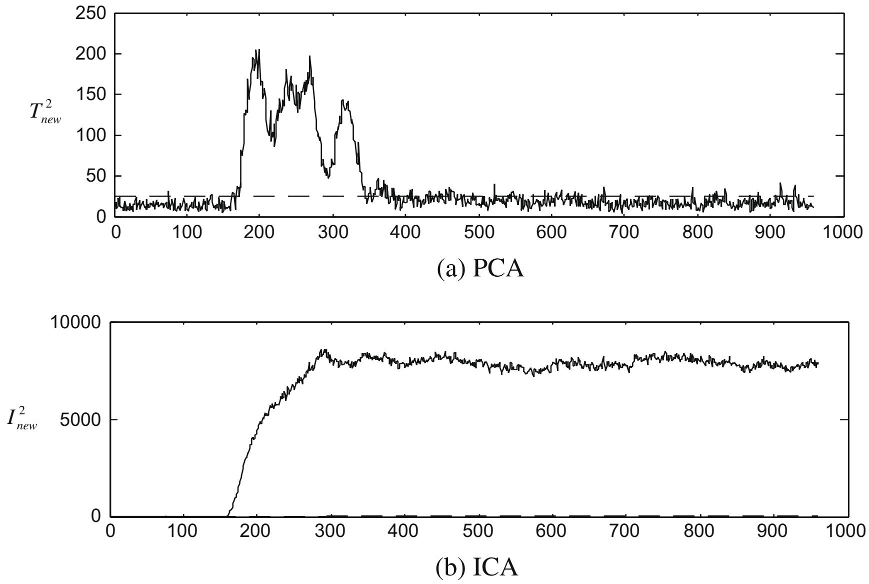
The Tennessee Eastman (TE) process proposed by Downs and Vogels (1993) has been a benchmark problem for testing the performance of various multivariate monitoring approaches (Chen & Liao, 2002; Ge & Song, 2007; Lee et al., 2004c; Lee et al., 2006). Fig. 7 shows the layout of the TE process. The process contains five major parts: reactor, condenser, compressor, separator and stripper. The gaseous A, C, D and E and the inert B are fed to the reactor, where the liquid products G and H are formed and a byproduct F is also produced. The reactions in the reactor are

$$A(\mathrm{g}) + C(\mathrm{g}) + D(\mathrm{g}) \rightarrow G(\mathrm{l}), \text{ product 1}$$

$$A(\mathrm{g}) + C(\mathrm{g}) + E(\mathrm{g}) \rightarrow H(\mathrm{l}), \text{ product 2}$$

$$A(\mathrm{g}) + E(\mathrm{g}) \rightarrow F(\mathrm{l}), \text{ byproduct}$$

$$3D(\mathrm{g}) \rightarrow 2F(\mathrm{l}), \text{ byproduct}$$



(a) PCA

(b) ICA

Fig. 8. Monitoring result of IDV(5).

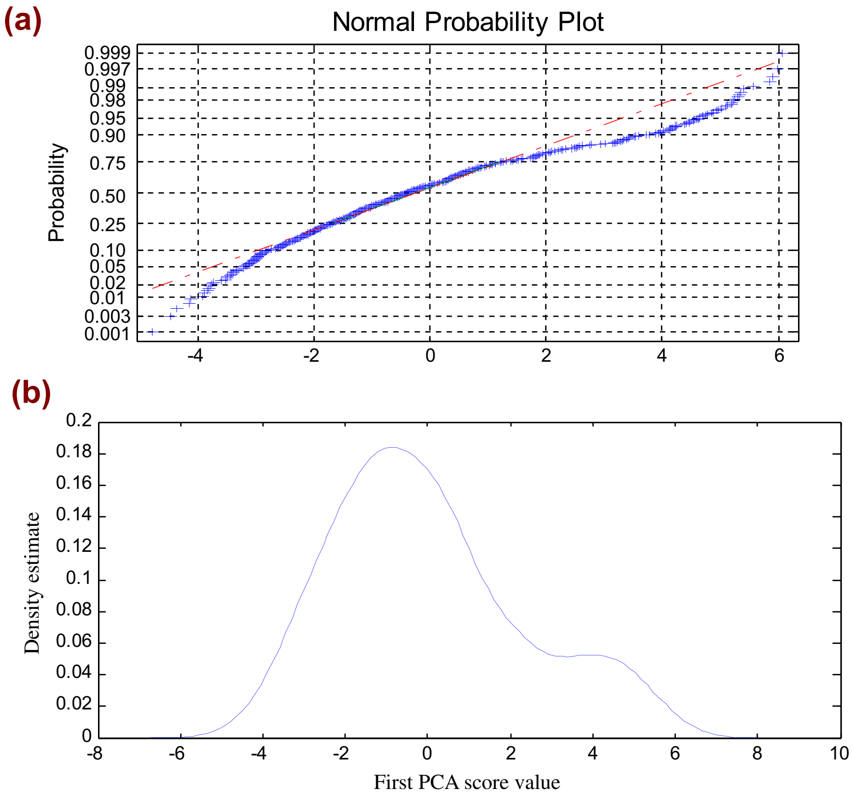


Fig. 9. (a) Normal probability plot (b) density estimate for the first PCA score.

Table 4
Related information of compared methods.

Methods	Statistics	Discrimination method
PCA	T^2	$T^2_{a,n,\alpha} = \frac{a(n-1)}{n-a} F_{a,n-a,\alpha}$
	SPE	$SPE_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0}$
ICA	I^2	99% upper control limit (KDE)
	SPE	99% upper control limit (KDE)
Modified ICA	T^2	99% upper control limit (KDE)
	SPE	$SPE_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0}$
ICA–PCA	I^2	99% upper control limit (KDE)
	T^2	$T^2_{a,n,\alpha} = \frac{a(n-1)}{n-a} F_{a,n-a,\alpha}$
	SPE	$SPE_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0}$
PCA–SVM	None	$f(T^2_{new}) = \mathrm{sgn}\left(\sum_{i=1} \alpha_i y_i \cdot K(T^2_{train,i} \cdot T^2_{new}) + b^* \right)$ $T^2_{new} \in \begin{cases} +1 & \text{if } f(T^2_{new}) > 0 \\ -1 & \text{if } f(T^2_{new}) < 0 \end{cases}$

The reactor product stream is cooled via a condenser and then fed into a vapor–liquid separator. The vapor exiting the separator is recycled to the reactor feed through the compressor. The condensed components from the separator stream 10 are pumped to the stripper. Stream 4 is used to strip the remaining reactants in stream 10 and is combined with the recycle stream. The products *G* and *H* exiting the base of the stripper are sent to a downstream process which is not included in this process. The TE process contains 22 continuous process measurements, nineteen composition measurements and 12 manipulated variables. Measurements are taken every 3 min. A detailed description of the TE process can been found in Chiang et al. (2001).

In this study, we use the data set which is generated by Chiang et al. (2001) and can be downloaded from http://brahms.scs.uiuc.edu. There are 33 monitored variables which are listed in Table 2: 22 continuous process measurements (XMEAS) and 11 manipulated variables (XMV). The agitation speed is excluded, because it is not manipulated. 19 composition measurements are also excluded, because they are hard to measure on-line in practice. There are 21 programmed process disturbance (IDV) models which are listed in Table 3. Faults IDV(1)–IDV(7) are a step change in a process variable. Faults IDV(8)–IDV(12) are an increase in the variability of some process variables. Fault IDV(13) is a slow drift in the reaction kinetics. Faults IDV(14), IDV(15) and IDV(21) are associated with sticking valves. IDV(16)–IDV(20) are unknown fault types.

5.3.2. Implementation

Fig. 8 shows the PCA-based T^2 and ICA-based I^2 monitoring result for IDV(5) in which the condenser cooling water inlet temperature (XMV(21) in Table 2) is step changed at sample 160. Fig. 8a indicates PCA can detect the fault approximately at sample 160. However, the T^2 sequence returns within the control limit approximately at sample 355. This result may confuse operators when judging the process condition after sample 355. By plotting the normal probability plot (Fig. 9a) and probability density estimation plot (Fig. 9b) to the first PCA score and results clearly indicate the PCA score does not follow the Gaussian assumption. Therefore, ICA is more suitable to deal with TE process. Fig. 8b shows that ICA not only can detect fault at sample 160, but also keeps the I^2 sequence all above the control limit after sample 160, which indicates that a fault remains in the process. Besides, the I^2 sequence can clearly exhibit a step-changed fault type, but T^2 cannot indicate the fault type. Therefore, an ICA-based monitoring method can improve monitoring performance and provide more information for operators to rectify the process. This is because PCA uses only the information of the covariance matrix, whereas ICA uses information on distribution which is not contained in the covariance matrix.

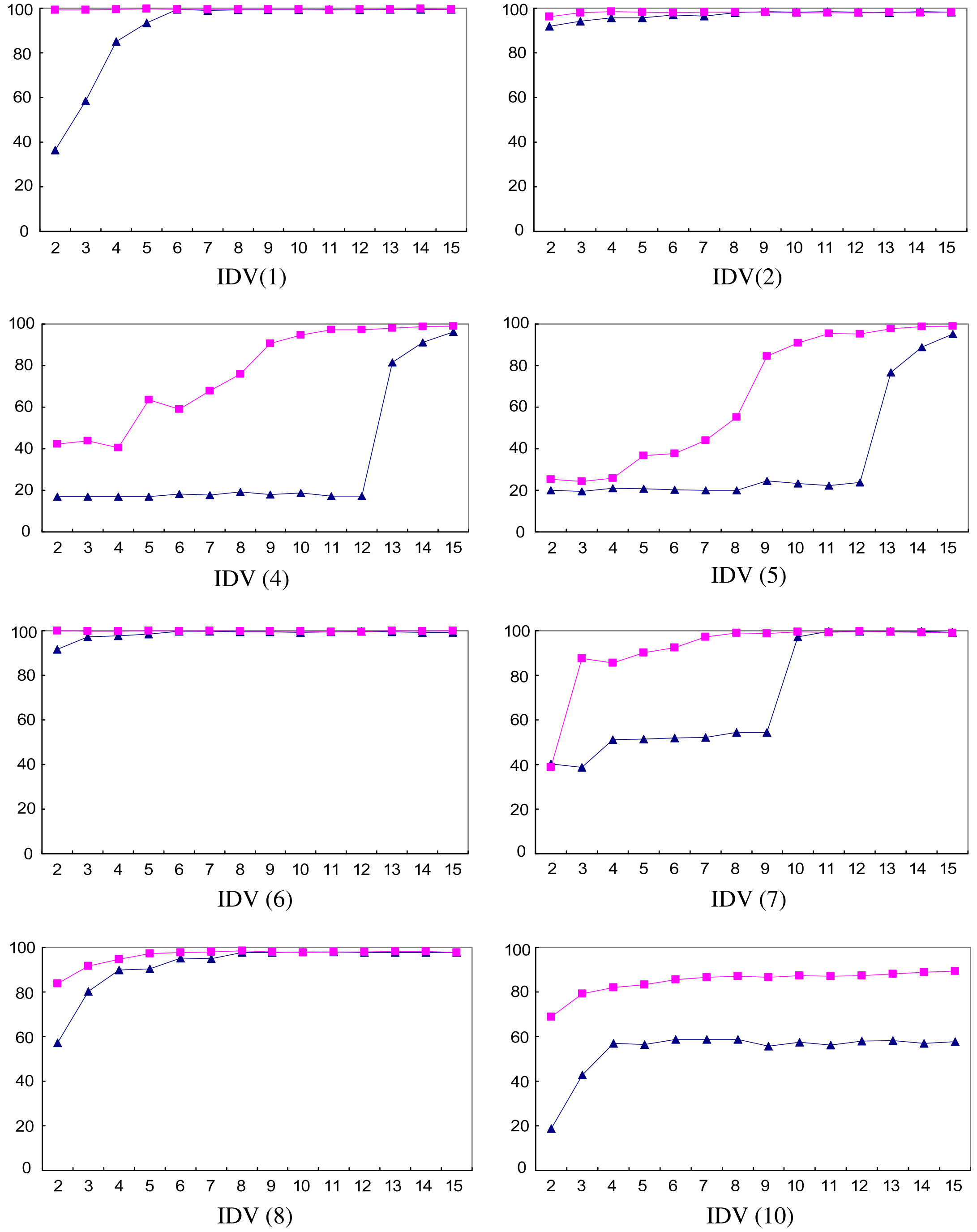
The TE process training dataset contains 500 observations which were generated without faults (IDV(0)), so it is used to develop the NOC dataset. The TE process training dataset for each fault contains 480 observations and it is used to develop the FOC dataset. Each TE process testing data set contains 960 observations, and all faults were introduced at sample 160; we use it to test the off-line built ICA–SVM fault detector. Because faults IDV(3), IDV(9) and IDV(15) are quite small and have almost no effect on the overall process, these faults will not be analyzed in our study.

The efficiency of the ICA–SVM fault detector will be verified by comparing it to PCA, original ICA, modified ICA and ICA–PCA monitoring methods. The PCA–SVM is also compared. In this study, 15 PCs were selected for PCA, ICA–PCA and PCA–SVM methods. Nine ICs were selected for ICA, modified ICA, ICA–PCA and ICA–SVM methods. Table 4 summarizes the relevant information of these compared methods.

For the data obtained after the fault occurrence, the percentage of the samples detected by the applied method was calculated and it was termed as the detection rate. Table 5 shows the fault detec-

Table 5
Fault detection rates for testing dataset in TE process.

Fault	PCA(15 PCs)		ICA (9 ICs)		Modified ICA(9 ICs)		ICA–PCA(9 ICs and 15 PCs)			PCA–SVM(15 PCs)	ICA–SVM(9 ICs)
	T^2	SPE	I^2	SPE	T^2	SPE	I^2	T^2	SPE		
IDV(1)	99.3	99.6	99.4	99.8	100	100	99.5	99.7	100	99.4	99.6
IDV(2)	97.8	98.5	98.0	99.0	98.0	98.0	98.2	98.5	99.2	98.3	98.1
IDV(4)	2.5	95.5	96.5	100	65.0	96.0	96.8	98.0	100	96.1	99.1
IDV(5)	25.0	25.5	100	100	24.0	24.0	100	100	100	95.1	99.0
IDV(6)	99.0	100	100	100	100	100	100	100	100	99.3	100
IDV(7)	91.0	100	93.5	100	100	100	95.0	96.5	100	99.3	99.1
IDV(8)	96.5	97.5	97.0	97.9	97.0	98.0	97.2	97.8	98.9	97.8	97.8
IDV(10)	29.0	34.9	78.5	80.0	70.0	64.0	78.8	84.4	83.8	57.7	89.4
IDV(11)	18.9	63.3	24.4	81.8	43.0	66.0	28.8	47.4	81.8	68.6	82.7
IDV(12)	96.9	97.3	97.5	100	98.0	97.0	97.6	99.5	100	97.6	99.5
IDV(13)	93.5	95	97.5	100	95.0	94.0	98.4	98.9	100	95.5	95.5
IDV(14)	83.8	100	95.5	100	100	100	95.8	98.7	100	90.7	99.0
IDV(16)	18.8	22.5	78.7	87.5	76.0	73.0	81.5	85.4	87.5	63.8	93.0
IDV(17)	75.9	88.8	78.9	80.4	87.0	94.0	79.4	89.3	80.5	62.9	95.7
IDV(18)	88.8	90	89.8	91.5	90.0	90.0	90.0	91.5	90.2	90.4	91.5
IDV(19)	0.2	23.8	64.9	67.8	25.0	29.0	65.4	76.2	79.2	56.0	96.6
IDV(20)	28.8	47.6	57.4	64.2	70.	66.0	58.8	75.3	69.4	75.4	92.7
Average	61.5	75.3	85.1	91.2	78.7	81.7	86	90.4	92.4	84.9	95.8



IDV(1)

IDV(2)

IDV (4)

IDV (5)

IDV (6)

IDV (7)

IDV (8)

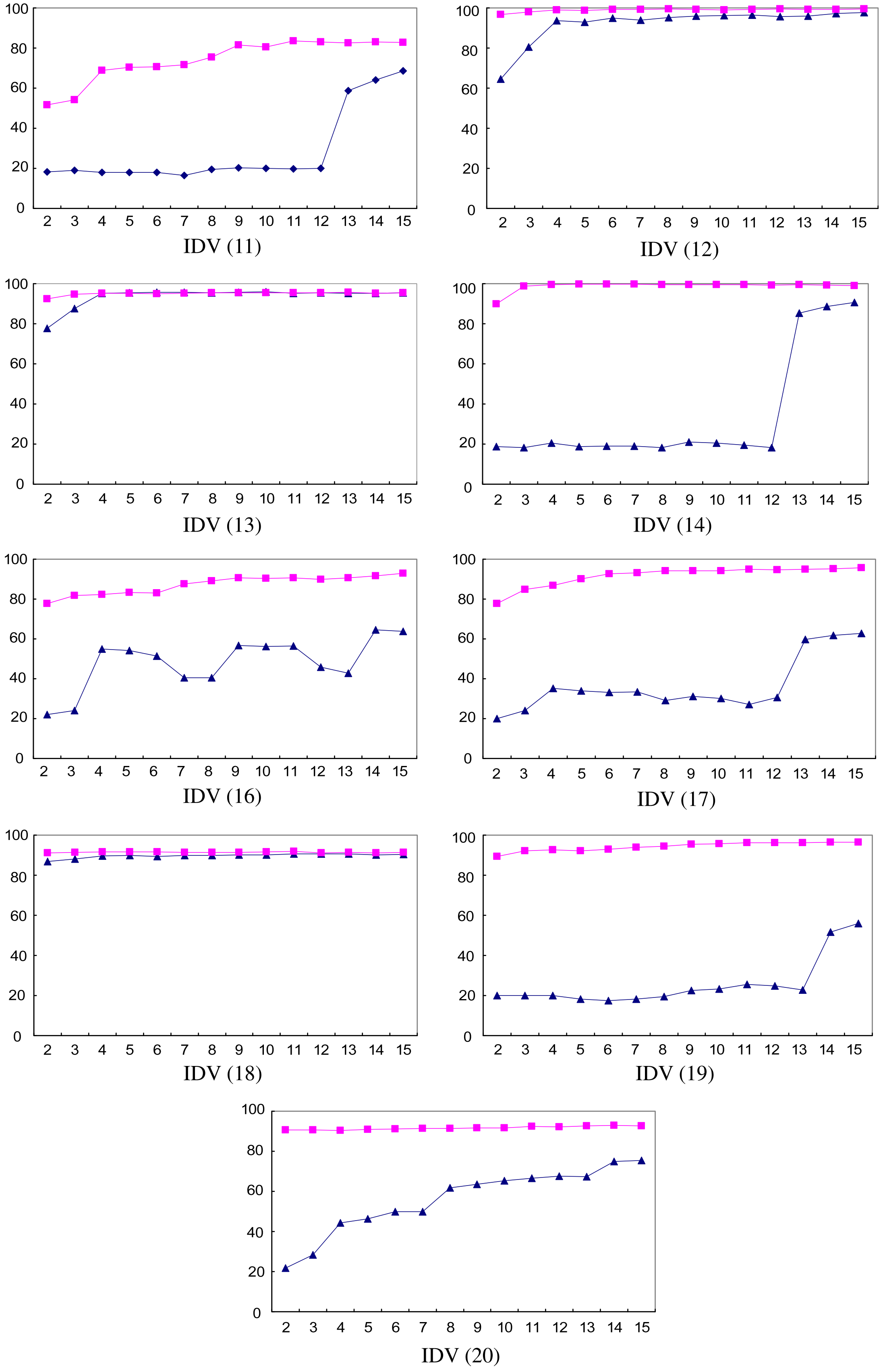
IDV (10)

Fig. 10. TE process fault detection rates against the number of latent components (■: ICA–SVM fault detector, ▲: PCA–SVM fault detector).

tion rates under several monitoring schemes. The detection rate of ICA outperforms PCA, especially for faults IDV(5), IDV(10), IDV(11), IDV(16), IDV(19) and IDV(20). The modified ICA also produces higher detection rates than PCA for most faults. However, ICA performs better than modified ICA, in particular for faults IDV(5), IDV(11) and IDV(19). The detection rates of ICA–PCA are even higher than those of PCA, ICA and modified ICA, because ICA–PCA considers both Gaussian and non-Gaussian information simultaneously. The PCA–SVM outperforms PCA and modified ICA in most cases, but does not perform well when compared to ICA and ICA–PCA. Overall, the excellent fault detection performance is observed in the case of ICA–SVM, in particular, when compared to PCA, modified ICA, ICA/ I^2, ICA–PCA/I^2 and PCA–SVM.

Fig. 10 shows ICA–SVM fault detection rates against the number of chosen ICs. In addition, the plots also exhibit results of PCA–SVM in order to visualize the efficiency of ICA–SVM. It indicates that the detection rate of PCA–SVM increases with the number of chosen PCs, whereas the ICA–SVM changes with smaller variation, except fault IDV(4), IDV(5) and IDV(7), which implies ICA–SVM can have good results with a small number of ICs in most faults. It should be noted that the overall detection rate of ICA–SVM is higher than that of PCA–SVM, irrespective of the number of components.

Through this case study, results show that the proposed ICA–SVM has superior capability in detecting faults when compared to traditional methods. The use of the ICA–SVM fault detector can help operators immediately rectify the process when a fault



IDV (11)

IDV (12)

IDV (13)

IDV (14)

IDV (16)

IDV (17)

IDV (18)

IDV (19)

IDV (20)

Fig. 10 (*continued*)

occurs. This is particularly important for the process industries such as chemical or biological processes which may involve plant safety problems.

6. Conclusion

In order to maintain yield and quality in a process, providing correct information of process status for operators is an important issue. A novel method based on integrating independent component analysis and support vector machine (ICA–SVM) for multivariate process fault detection has been developed in this study. Through implementing two simulated multivariate processes, the results show ICA–SVM is superior to PCA–SVM in terms of fault detection rates. This is because the ICA considers higher ordered statistics, so can provide more useful information for SVM to detect faults. A case study of the Tennessee Eastman (TE) benchmark process is also applied to evaluate ICA–SVM. The results clearly show the superiority of proposed ICA–SVM over traditional monitoring methods, including PCA, ICA, modified ICA and ICA–PCA for detecting most TE fault models. Further research can extend the proposed method to a batch process which is more complex in contrast to a continuous process.

Acknowledgement

This work was supported in part by National Science Council of Taiwan (NSC 98-2410-H-324-006-MY2).

References

Cao, L. J., Chua, K. S., Chong, W. K., Lee, H. P., & Gu, Q. M. (2003). A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing, 55*, 321–336.

Chang, C. C., & Lin, C. J. (2001). LIBSVM: A library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Chen, J., & Liao, C. M. (2002). Dynamic process fault monitoring based on neural network and PCA. *Journal of Process Control, 12*, 277–289.

Chiang, L. H., Russell, E. L., & Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. London: Springer-Verlag.

Dong, D., & McAvoy, T. J. (1996). Nonlinear principal component analysis-based on principal curves and neural networks. *Computers and Chemical Engineering, 20*, 65–78.

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering, 17*(3), 245–255.

Ge, Z., & Song, Z. (2007). Process monitoring based on independent component analysis- principal component analysis (ICA–PCA) and similarity factors. *Industrial and Engineering Chemistry Research, 46*, 2054–2063.

Ge, Z., & Song, Z. (2008). Online monitoring of nonlinear multiple mode processes based on adaptive local model approach. *Control Engineering Practice, 16*, 1427–1437.

Hsu, C. C., Chen, L. S., & Liu, C. H. (2010). A process monitoring scheme based on independent component analysis and adjusted outliers. *International Journal of Production Research, 48*(6), 1727–1743.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2006). A practical guide to support vector classification. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>.

Hyvärinen, A. (1997). A fast-point algorithm for independent component anlaysis. *Neurocomputing, 9*, 1483–1492.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions Neural Networks, 10*, 626–634.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis*. New York, USA: John Wiley & Sons, Inc..

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks, 13*(4-5), 411–430.

Jackson, J. E. (1959). Quality control methods for several related variables. *Technometrics, 1*(4), 359–377.

Jackson, J. E., & Mudholkar, G. S. (1979). Control procedures for residual associated with principal component analysis. *Technometrics, 21*(3), 341–349.

Jia, F., Martin, E. B., & Morris, A. J. (1998). Non-linear principal components analysis for fault detection. *Computers and Chemical Engineering, 22*, 851–854.

Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., & Ohno, H. (2003). Monitoring independent components for fault detection. *AIChE Journal, 49*(4), 969–976.

Keerthi, S. S., & Lin, C. J. (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation, 15*(7), 1667–1689.

Kourti, T., & MacGregor, J. F. (1996). Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology, 28*, 409–428.

Ku, W., Storer, R. H., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 30*, 179–196.

Kwak, N. (2008). Feature extraction for classification problems and its application to face recognition. *Pattern Recognition, 41*(5), 1701–1717.

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

Lee, J. M., Yoo, C. K., & Lee, I. B. (2004a). Statistical process monitoring with independent component analysis. *Journal of Process Control, 14*, 467–485.

Lee, J. M., Yoo, C. K., & Lee, I. B. (2004b). Fault detection of batch processes using multiway kernel principal component analysis. *Computers and Chemical Engineering, 28*(9), 1837–1847.

Lee, J. M., Yoo, C. K., & Lee, I. B. (2004c). Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical Engineering Science, 59*, 2995–3006.

Lessmann, S., & Voß, S. (2009). A reference model for customer-centric data mining with support vector machines. *European Journal of Operational Research, 199*(2), 520–530.

Li, T. S. (2009). Applying wavelets transform and support vector machine for copper clad laminate defects classification. *Computers & Industrial Engineering, 56*(3), 1154–1168.

Liu, Y. (2009). Wavelet feature extraction for high-dimensional microarray data. *Neurocomputing, 72*, 985–990.

Lu, C. J., Wu, C. M., Keng, C. J., & Chiu, C. C. (2006). Integrated application of SPC/EPC/ICA and neural networks. *International Journal of Production Research, 46*(4), 873–893.

Lu, J., & Zhang, E. (2007). Gait recognition for human identification based on ICA and fuzzy SVM through multiple views fusion. *Pattern Recognition Letters, 28*(16), 2401–2411.

Martin, E. B., & Morris, A. J. (1996). Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control, 6*(6), 349–358.

Montgomery, D. C. (2005). *Introduction to statistical quality control*. New York: John Wiley & Sons.

Nomikos, P., & MacGregor, J. F. (1994). Monitoring batch processes using multiway principal component analysis. *American Institute of Chemical Engineers Journal, 40*, 1361–1375.

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

Shao, R., Jia, F., & Morris, A. J. (1999). Wavelets and non-linear principal components analysis for process monitoring. *Control Engineering Practice, 7*(7), 865–879.

Shieh, M. D., & Yang, C. C. (2008). Classification model for product form design using fuzzy support vector machines. *Computers & Industrial Engineering, 55*(1), 150–164.

Shin, H. J., Eom, D. H., & Kim, S. S. (2005). One-class support vector machines—an application in machine fault detection and classification. *Computers & Industrial Engineering, 48*(2), 359–408.

Song, X., & Wyrwicz, A. M. (2009). Unsupervised spatiotemporal fMRI data analysis using support vector machines. *NeuroImage, 47*, 204–212.

Valle, S., Li, W., & Qin, S. J. (1999). Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial and Engineering Chemistry Research, 38*, 4389–4401.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Wasterhuis, J. A., Gurden, S. P., & Smilde, A. K. (2000). Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems, 51*, 95–114.

Widodo, A., & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing, 21*(6), 2560–2574.

Widodo, A., Yang, B. S., Gu, D. S., & Choi, B. K. (2009). Intelligent fault diagnosis system of induction motor based on transient current signal. *Mechatronics, 19*, 680–689.

Wold, S. (1978). Cross-validatory estimation of components in factor and principal components models. *Technometrics, 20*, 397–405.

Yoo, C. K., Lee, J. M., Vanrolleghem, P. A., & Lee, I. B. (2004). On-line monitoring of batch processes using multiway independent component analysis. *Chemometrics and Intelligent Laboratory Systems, 71*(2), 151–163.

Zhang, Y. (2009). Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. *Chemical Engineering Science, 64*, 801–811.