Expert Systems with Applications 37 (2010) 3264–3273

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Intelligent ICA–SVM fault detector for non-Gaussian multivariate process monitoring

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

Keywords:
ICA
SVM
PCA
Fault detector
Autocorrelated

ABSTRACT

Recently, the independent component analysis (ICA) has been widely used for multivariate non-Gaussian process monitoring. For principal component analysis (PCA) based monitoring method, the control limit can be determined by a specific distribution (*F* distribution) due to the PCA extracted components are assumed to follow multivariate Gaussian distribution. However, the control limit for ICA based monitoring statistic is determined by using kernel density estimation (KDE). It is well known that the KDE is sensitive to the smoothing parameter, and it does not perform well with autocorrelated data. In most cases, the calculated ICA based monitoring statistic is usually autocorrelated. Thus, this study aims to integrate ICA and support vector machine (SVM) in order to develop an intelligent fault detector for non-Gaussian multivariate process monitoring. Simulation study indicates that the proposed method can effectively detect faults when compare to methods of original SVM and PCA based SVM in terms of detection rate.

1. Introduction

Statistical process control (SPC) has been successfully applied to analyzing systems or processes in which only one variable is measured and tested. Generally, there exist many variables need to be monitored and even controlled in a process. Under such a condition, it may produce false alarms if the univariate SPC is applied to monitor the process. Multivariate statistical process control (MSPC) provides a way for engineers to test products, and it provides several advantages over univariate models. The traditional MSPC models such as Shewhart chart, EWMA (exponentially weighted moving average) chart and CUMSUM (cumulative sum) chart are usually used for monitoring final product variables. The details of related models can be found in Montgomery (2005).

A problem is emerged with utilizing the traditional MSPC models, since they may be impractical for high-dimensional systems with collinearities. Hence, it needs to reduce the dimensionality of variable space. Principal component analysis (PCA) is one of the well known projection methods. PCA technique is primarily used in the area of chemometrics (e.g., Kourti & MacGregor, 1996; Wasterhuis, Gurden, & Smilde, 2000) and it also very promising in any kind of multivariate processes. PCA benefits from the capability of handling both process variables and product quality variables. PCA considers up to the second-order statistics which means that the latent variables capture the most variance of original variables. PCA tries to decorrelate variables, but not to make them independent. PCA works well in many cases, but it is seldom applied to a higher order characteristic process. T^2 and Squared Prediction Error (*SPE)* statistics are used for the PCA process monitoring, however the control limits of T^2 and *SPE* are determined based on the assumption that the latent variables are multivariate Gaussian distributed.

Independent component analysis (ICA) is originally developed for signal processing applications including speech signal processing, communications, medical image processing, financial engineering and so forth. Later, it has been generalized for feature extraction. ICA can be taken as an extension of PCA. However, the objectives for both algorithms are quite different. PCA extracts components by only considering variance–covariance matrix, and it aims at making the latent variables to be orthogonally uncorrelated. ICA has no orthogonality constraint which not only allows to decorrelates variables, but also to consider the higher order statistics for making latent variables to be independent. Therefore, ICA can be used to deal with a non-Gaussian process which is more practical in a real-world manufacturing environment, especially for the process industry.

Recently, ICA has been used for monitoring the multivariate processes. Lee, Yoo, and Lee (2004a) developed three control charts based on ICA statistics including I^2, I_e^2 and *SPE*. I^2 is used to monitor the systematic part of process variation. I_e^2 can compensate for the error which is stemmed from an incorrect number of independent

* Corresponding author. Tel.: +886 4 2332300/7732.
E-mail address: cchsu@cyut.edu.tw (C.-C. Hsu).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.09.053

components selected for the dominant part. *SPE* is used to monitor the non-systematic part of common cause variation.

The control limits of PCA systematic part statistic T^2 can be determined by the *F* distribution. Unlike PCA, the extracted independent components do not follow Gaussian distribution, Lee et al. (2004a) therefore used the non-parametric technique based on Kernel Density Estimation (KDE) to determine the control limit for the ICA systematic part statistic I^2. However, KDE has two main limitations, requirement of large dataset and high sensitivity to the smoothing parameter (Yoo, Lee, Vanrolleghem, & Lee, 2004). Although the least-squared cross-validation (LSCV) works well in most types of data, it does not perform well if data is autocorrelated (Blundell, Maier, & Debevec, 2001).

Support vector machine (SVM) is an effective machine learning method for classification problem and eventually result in better generalization performance than most of traditional methods. In this study, we propose an ICA–SVM fault detector for multivariate process monitoring. The basic idea of proposed methodology is to use ICA for feature extraction and then calculate the systematic part statistic. SVM is used as a classifier to detect the process conditions. Due to the calculated I^2 tends to be highly autocorrelated. Therefore, we not only take I^2 to be the input vector of SVM, but also consider the time delay and time difference of I^2 as inputs in SVM. The effectiveness of proposed methodology is investigated by comparing to that of PCA–SVM and original SVM.

The organization of this article is as follows. Section 2 reviews PCA and ICA based projection methods for multivariate process monitoring. SVM algorithm for classification problem is introduced in Section 3. The proposed ICA–SVM fault detector is described in Section 4. Section 5 reports the simulation results of the proposed algorithm when step and linear disturbances are introduced in the process. The comparisons are presented in this section as well. The conclusion is finally given in Section 6.

2. Projection methods for multivariate process monitoring

In the case of monitoring all available process variables, the traditional multivariate control charts are limited since the process variables may be highly correlated with one another and their covariance matrix is nearly singular. Therefore, it is required to project a high dimensional space into a lower subspace to simplify the requirements to accurately describe a large dataset. PCA and ICA both are popular statistical projection tools and have been widely used in multivariate process monitoring. We introduce the PCA based monitoring method and the ICA based monitoring method in Sections 2.1 and 2.2, respectively.

2.1. PCA based fault detection method

Jackson (1959) initially used PCA as a feature extraction tool for multivariate monitoring. Later, Jackson and Mudholkar (1979) developed a residual analysis for the PCA based control charts. After these two initial works, the alternative PCA based approaches have thereafter been developed in the literature. These approaches include the PCA based method for monitoring multiway batch processes (Nomikos & MacGregor, 1994, 1995), kernel PCA for monitoring batch processes (Lee, Yoo, & Lee, 2004b), dynamic PCA for monitoring time-dependent measurements (Ku, Storer, & Georgakis, 1995), integrated PCA-wavelet method for process monitoring (Shao, Jia, & Morris, 1999), integrated neural network and PCA for fault detection (Jia, Martin, & Morris, 1998) and so forth.

PCA intends to linearly transform high-dimensional input vector into a lower dimensional one whose components are uncorrelated. $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)] \in R^{d\times n}$ denotes a centered data matrix, where $\mathbf{x} \in R^d$ is a column vector with *d* measured variables and *n* is the number of measurements. The covariance matrix is $\mathbf{R_x} = E(\mathbf{x}\mathbf{x}^T)$, where *E* represents expectation and *T* denotes the transpose operator. The eigen-decomposition of $\mathbf{R_x}$ can be given by

$$\mathbf{R_x} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T \tag{1}$$

where $\mathbf{U} \in R^{d\times d}$ is an orthogonal matrix of eigenvectors and $\boldsymbol{\Lambda} \in R^{d\times d}$ is the diagonal matrix of eigenvalues. All score principal components can be expressed as

$$\mathbf{t} = \mathbf{U}^T\mathbf{x} \tag{2}$$

$\mathbf{t} \in R^d$. 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 $\mathbf{t}' = \mathbf{P}^T\mathbf{x}$. The principal components have the following properties (Cao, Chua, Chong, Lee, & Gu, 2003):

1. $\mathbf{t}$ are uncorrelated.
2. $E\{\mathbf{t}'\mathbf{t}'^T = \mathbf{D} = diag\{\lambda_1, \lambda_2, \ldots, \lambda_a\},\ \lambda_i$ have been sorted with sequentially maximum variances.
3. The mean-squared approximation error in the representation of original data by $\mathbf{t}'$ is minimal.

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

$$T^2 = \mathbf{t}'^T\mathbf{D}^{-1}\mathbf{t}' = \mathbf{x}^T\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^T\mathbf{x} \tag{3}$$

The upper confidence limit for T^2 can be obtained by using *F* distribution, and it takes the form as

$$T^2_{a,n,\alpha} = \frac{a(n-1)}{n-a}F_{a,n-a,\alpha} \tag{4}$$

A measure of variation not captured by the PCA model can be monitored by the Squared Prediction Error (*SPE*).

$$SPE = \mathbf{e}^T\mathbf{e} = \mathbf{x}^T(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x} \tag{5}$$

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 $\mathbf{e} = 0$ if $a = d$. The upper control limit for *SPE* can be expressed as

$$SPE_\alpha = \theta_1\left[\frac{c_\alpha\sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0(h_0-1)}{\theta_1^2}\right]^{1/h_0} \tag{6}$$

where $\theta_g = \sum_{j=a+1}^{d}\lambda_j^g$ 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 upper $1-\alpha$ percentile.

2.2. ICA based fault detection method

Kano, Tanaka, Hasebe, Hashimoto, and Ohno (2003) developed the ICA-based SPC and showed the superiority over the PCA based monitoring method. Yoo et al. (2004) utilized the multiway ICA for batch processing monitoring. Lee et al. (2004a) developed I^2, I_e^2 and *SPE* control charts as well as variable contribution plot for fault diagnosis. Lee, Yoo, and Lee (2004c) presented a dynamic ICA (DICA) monitoring method. In which, ICA was applied to the augment matrix with time lagged variables. Lee, Qin, and Lee (2006) proposed a modified ICA to relax the drawbacks of original ICA algorithm such as the pre-determination of number of extracted independent components and the pre-determination of the proper order of independent components. Lu, Wu, Keng, and Chiu (2006) applied ICA for integrating SPC and Engineering Process Control (EPC). Ge and Song (2007) proposed PCA–ICA to extract Gaussian and non-Gaussian information for fault detection and diagnosis.

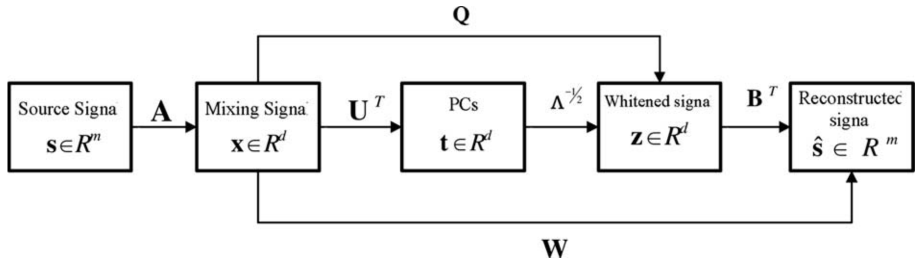

Fig. 1. Flowchart of ICA algorithm.

Fig. 1 illustrates the flowchart of ICA. In the ICA algorithm, the measured variables $\mathbf{x} \in R^d$ can be expressed as linear combination of m unknown independent components $\mathbf{s} = [s_1, s_2, \ldots, s_m] \in R^m$, that is,

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{7}$$

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

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} \tag{8}$$

such that the reconstructed vector $\hat{\mathbf{s}}$ becomes as independent as possible. For convenience, we assume $d = m$, and $E(\mathbf{s}\mathbf{s}^T) = \mathbf{I}$. The whitening transformation is expressed as

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

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

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

Hence, Eq. (8) can be rewritten as

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

According to Eq. (11), the ICA problem can be reduced to find an orthogonal matrix $\mathbf{B}$.

To calculate $\mathbf{B}$, Hyvärinen (1999) introduced a fast fixed-point algorithm for ICA (FastICA). This algorithm calculates the column vector $\mathbf{b}_i$ $(i = 1, 2, \ldots, m)$ of $\mathbf{B}$ through iterative steps. The detailed procedure can refer to Hyvärinen and Oja (2000) and Hyvärinen et al. (2001). After obtaining $\mathbf{B}$, we can calculate $\hat{\mathbf{s}}$ by using Eq. (11) and $\mathbf{W}$ from Eq. (10).

To divide $\mathbf{W}$ into two parts, dominant part ($\mathbf{W}_d$) and excluded part ($\mathbf{W}_e$), Lee et al. (2004a) proposed three statistics for process monitoring as follows:

$$I^2 = \hat{\mathbf{s}}_d^T\hat{\mathbf{s}}_d$$
$$I_e^2 = \hat{\mathbf{s}}_e^T\hat{\mathbf{s}}_e$$
$$SPE = \mathbf{e}^T\mathbf{e} = (\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}) \tag{12}$$

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 determined by using Eq. (4). In ICA, I^2, I_e^2 and SPE depart from the normality assumption. Lee et al. (2004a) therefore proposed to use KDE to determine the control limits for I^2, I_e^2 and SPE statistics for ICA monitoring. Additionally, the variable contributions of $\mathbf{x}(t)$ for $I^2(t)$ and $I_e^2(t)$ can be defined as

$$\mathbf{x}_{cd}(t) = \frac{\mathbf{Q}^{-1}\mathbf{B}_d\hat{\mathbf{s}}_d(t)}{\left\|\mathbf{Q}^{-1}\mathbf{B}_d\hat{\mathbf{s}}_d(t)\right\|}\|\hat{\mathbf{s}}_d(t)\| \tag{13}$$

$$\mathbf{x}_{ce}(t) = \frac{\mathbf{Q}^{-1}\mathbf{B}_e\hat{\mathbf{s}}_e(t)}{\left\|\mathbf{Q}^{-1}\mathbf{B}_e\hat{\mathbf{s}}_e(t)\right\|}\|\hat{\mathbf{s}}_e(t)\| \tag{14}$$

ICA is interested in non-Gaussian information, whereas PCA seeks to treat Gaussian information. Due to many manufacturing processes exhibit non-Gaussian characteristic, hence this study adopts ICA as the feature extraction method. After extracting essential features, it requires introducing a method for the classification problem. Next section reviews the theory of support vector machine (SVM).

3. Support vector machines

Support vector machines (SVMs) are supervised learning methods used for classification and regression. Recently, SVM has been applied to process monitoring and fault diagnosis. Chinnam (2002) used SVM to recognize the process shifts, and showed that SVM outperforms radial basis function (RBF) neural networks. Sun and Tsung (2003) developed a kernel-distance based multivariate control chart (K chart) by using one-class SVM and demonstrated that K chart outperforms the conventional charts when the data distribution departs from normality. In advance, Kumar, Choudhary, Kumar, Shankar, and Tiwari (2006) presented a robust K chart which aims at solving the over-fitting problems when outliers exist in the SVM training dataset. Widodo, Yang, and Han (2007) integrated ICA and SVM for faults diagnosis of induction motors.

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 to the SVM theory can be seen in Vanpinik'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 with labels $y_i \in \{-1, +1\}$. There exists a separating hyperplane and its function is as

$$\boldsymbol{\delta} \cdot \mathbf{x} + b = 0 \tag{15}$$

where $\boldsymbol{\delta} \in R^n$ is a normal vector, the bias b is a scale, and $\frac{|b|}{\|\boldsymbol{\delta}\|}$ represents the perpendicular distance from the separating hyperplane to the origin. Two parallel hyperplanes can be represented as

$$y_i(\boldsymbol{\delta} \cdot \mathbf{x}_i + b) \geqslant 1 \tag{16}$$

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

$$\begin{aligned} &\text{Min} \quad \frac{1}{2}\|\boldsymbol{\delta}\|^2 \\ &\text{s.t.} \quad y_i(\boldsymbol{\delta} \cdot \mathbf{x}_i + b) \geqslant 1 \end{aligned} \tag{17}$$

By introducing Lagrange multipliers $\alpha_i (i = 1, 2, \ldots, n)$ for the constraint, the primal problem (Model (17)) becomes a task of finding the saddle point of Lagrange. Thus, the dual problem becomes

$$\text{Max} \quad L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$
$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$\alpha_i \geqslant 0 \tag{18}$$

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

$$\alpha_i [y_i (\boldsymbol{\delta} \cdot \mathbf{x}_i + b) - 1] = 0 \tag{19}$$

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

$$\boldsymbol{\delta}^* = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{20}$$

where N is the number of SVs. From Eq. (19), by choosing any SVs $(\mathbf{x}_k, y_k)$, we can obtain $b^* = y_k - \boldsymbol{\delta}^* \cdot \mathbf{x}_k$. After $(\boldsymbol{\delta}^*, b^*)$ is determined, the discrimination function can be given by

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

where sgn(.) is the sign function, and $\mathbf{x} \in \begin{cases} +1 & \text{if} \quad f(\mathbf{x}) > 0 \\ -1 & \text{if} \quad f(\mathbf{x}) < 0 \end{cases}$.

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:

Linear kernel	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \mathbf{x}_j$
Polynomial kernel of degree g	$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \mathbf{x}_j + r)^g, \gamma > 0$
Radial basis function	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\}, \gamma > 0$
Sigmoid kernel	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i \mathbf{x}_j + r), \gamma > 0$

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

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \cdot K(\mathbf{x} \cdot \mathbf{x}_i) + b^* \right) \tag{22}$$

Unlike most of the traditional methods which implement the empirical risk minimization principal, SVM implements the structure risk minimization principal that 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 applied SVM as classifier for process monitoring. Next section interprets the proposed methodology.

4. ICA–SVM fault detector

For training SVM, the irrelevant input variables can have a negative impact on the performance. Cao et al. (2003) had shown that the SVM with feature extraction performs better than that without feature extraction. Similarly, to develop the SVM fault detector, the first step is feature extraction in this study. The architecture of proposed ICA–SVM fault detector is shown in Fig. 2. At first, the feature extraction based on ICA is used to project the high dimension dataset into a lower one ($m \leqslant d$). The extracted independent components are then used to calculate the systematic part statistic. Because the calculated statistic is autocorrelated, we hence additionally consider time delay and time difference of systematic statistics 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.

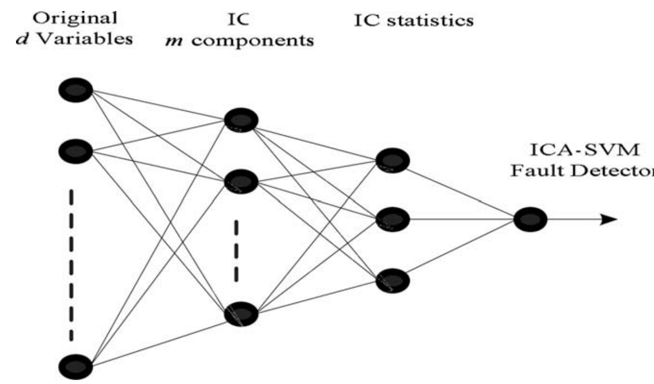

Fig. 2. Architecture of ICA–SVM fault detector.

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.

4.1.1. NOC training dataset development

Step 1: Scale NOC dataset. Obtain an 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 denote 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 FastICA algorithm over $\mathbf{z}_{normal}$, we can obtain an orthogonal matrix $\mathbf{B}_{normal}$. Therefore, the reconstructed dataset is given by $\hat{\mathbf{s}}_{normal} = \mathbf{B}_{normal}^T \mathbf{z}_{normal}$.

Step 3: Determine the order of $\hat{\mathbf{s}}_{normal}$. In this step, the order of $\hat{\mathbf{s}}_{normal}$ is determine by using Euclidean norm (L_2) of each row ($\mathbf{w}_i$) in $\mathbf{W}_{normal}$ (Lee et al., 2004a), $\text{Arg}_i\text{Max}\|\mathbf{w}_i\|_2$. Hence, we can obtain a sorted de-mixing matrix.

Step 4: Perform dimension reduction. There are several methods for selecting the number of independent components 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 independent components. In this study, the number of independent components is set to be the same as the number of principal components.

Step 5: Calculate the systematic part statistics. The dominant independent components represent the systematic part of data structure. From the above steps, we can obtain a dominant de-mixing matrix, $\mathbf{W}_d$. From Eq. (10), $\mathbf{B}_d = (\mathbf{W}_d \mathbf{Q}^{-1})^T$. Hence, the dominant independent components can be calculated by $\hat{\mathbf{s}}_{normal\_d} = \mathbf{B}_d^T \mathbf{z}_{normal}$, and the systematic part statistic at sample t can be obtained, that is $I_{normal}^2(t) = \hat{\mathbf{s}}_{normal\_d}^T(t) \hat{\mathbf{s}}_{normal\_d}(t)$. The obtained I_{normal}^2 is usually autocorrelated. Hence, the time delay $I_{normal}^2(t-1)$ and time difference $I_{normal}^2(t) - I_{normal}^2(t-1)$ are additionally taken as input vectors for ICA–SVM.

4.1.2. FOC training dataset development

FOC dataset is also scaled at first, denoted as $\mathbf{z}_{fault}$. $\hat{\mathbf{s}}_{fault\_d}$ represents the dominant independent components under FOC, and it can be calculated by $\hat{\mathbf{s}}_{fault\_d} = \mathbf{B}_d^T \mathbf{z}_{fault}$. The statistic of FOC systematic part at sample t is calculated from $I^2_{fault}(t) = \hat{\mathbf{s}}^T_{fault\_d}(t)\hat{\mathbf{s}}_{fault\_d}(t)$. Also, $I^2_{fault}(t-1)$ and $I^2_{fault}(t) - I^2_{fault}(t-1)$are taken as the input vectors of ICA–SVM.

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

The objective of this phase is to test the trained ICA–SVM model. Once the new data is obtained, the same scaling is then applied, and the scaled dataset is then denoted as $\mathbf{z}_{new}$. The dominant independent components of $\mathbf{z}_{new}$ can be obtained from $\hat{\mathbf{s}}_{new\_d} = \mathbf{B}_d^T \mathbf{z}_{new}$, and the statistic of systematic part at time t can be calculated by $I^2_{new}(t) = \hat{\mathbf{s}}^T_{new\_d}(t)\hat{\mathbf{s}}_{new\_d}(t)$. The statistics, $I^2_{new}(t), I^2_{new}(t-1)$ and $I^2_{new} - I^2_{new}(t-1)$ are fed into trained ICA–SVM for on-line process monitoring.

5. Implementation

In this section, a dynamic multivariate system is used to verify the effectiveness of ICA–SVM. Section 5.1 introduces the multivariate simulation model which is constructed by Matlab/Simulink. Two common faults, step and linear disturbances are introduced into the process for on-line testing of the built ICA–SVM detector.

5.1. Multivariate process model

Ku et al. (1995) suggested a first-order autoregressive multivariate simulation process, and thereafter Chen and Liao (2002) and Lee et al. (2004a) simulated a modified version of Ku et al. Without lose of generality, this study adopts the version of Lee et al. (2004a) to simulate the first-order autoregressive multivariate process.

A discrete state-space is used to represent the one-order autoregressive multivariate process, and it takes the form as

$$\mathbf{F}(k) = \begin{bmatrix} 0.118 & -0.191 & 0.287 \\ 0.847 & 0.264 & 0.943 \\ -0.333 & 0.514 & -0.217 \end{bmatrix} \mathbf{F}(k-1) + \begin{bmatrix} 1 & 2 \\ 3 & -4 \\ -2 & 1 \end{bmatrix} \mathbf{H}(k-1)$$
$$\mathbf{C}(k) = \mathbf{F}(k) + \mathbf{E}(k) \tag{23}$$

Eq. (23) express the state equation and the output equation, respectively. $\mathbf{F}$ represents the state, and $\mathbf{C}$ represents the output. $\mathbf{E}$ is assumed to be normally distributed with zero mean and variance of 0.1. $\mathbf{H}$ is the input that can be obtained as follows:

$$\mathbf{H}(k) = \begin{bmatrix} 0.811 & -0.226 \\ 0.477 & 0.415 \end{bmatrix} \mathbf{H}(k-1) + \begin{bmatrix} 0.193 & 0.689 \\ -0.320 & -0.749 \end{bmatrix} \mathbf{G}(k-1) \tag{24}$$

$\mathbf{G} = [g_1, g_2]$ is the input that follows a uniform distribution between $[-2, 2]$. Both input $\mathbf{H} = [h_1, h_2]$ and output $\mathbf{C} = [c_1, c_2, c_3]$ are used for analysis, whereas $\mathbf{F}$ and $\mathbf{G}$ are not, Totally, five variables therefore are monitored in this process. Matlab/Simulink is adopted to construct the one-order autoregressive multivariate process. Fig. 3 shows the blocks of programmed Matlab/Simulink. Eq. (24) is designed in 'discrete state-space' block and Eq. (23) is designed in 'discrete state-space 1' block.

5.2. Step disturbance

A step change of g_1 by 2 is introduced at the sample of $t = 50$. Fig. 4 shows the simulated 200 observations, and c_2 exhibits an obvious step sequence in response to the change of g_1. In PCA model, three principal components are selected, and they capture approximately 88.1% of the variance. Fig. 5 shows the T^2_{new} values for PCA. Observing this figure, T^2_{new} can not indicate the disturbance type, it seems to only be "randomness". As the same of PCA, three independent components are retained in the ICA model. Fig. 6 shows the I^2_{new} statistic for ICA, and it indicates an obvious step disturbance pattern. From Fig. 6, I^2_{new} can provide more disturbance information than T^2_{new}. By using KDE, the control limit for ICA systematic statistic can be determined. The dash line in Fig. 6 represents the 99% control limit. Observing Fig. 6, the detection rate is relatively low, that is many points fall inside the control limit after the 50th sample.

By plotting the autocorrelation function (ACF) to I^2_{new} (see Fig. 7), it indicates that I^2_{new} is highly autocorrelated. Therefore, we consider to take $I^2(t), I^2(t-1)$ and $I^2(t) - I^2(t-1)$ as the input vectors for ICA–SVM fault detector. In this study, LIBSVM 2.82 developed by Chang and Lin (2001) is used to perform SVM. The users' guide is referred to Hsu, Chang, and Lin (2007). In LIBSVM, the default kernel function is radial basis function (RBF), and the optimal algorithmic parameters can be automatically generated.

Table 1 summarizes the training and testing accuracies of SVM, PCA–SVM and ICA–SVM under a set of step sizes. Inputs of original SVM are normalized dataset $\mathbf{z}(t), \mathbf{z}(t-1)$ and $\mathbf{z}(t) - \mathbf{z}(t-1)$. Inputs of PCA–SVM are $T^2(t),\ T^2(t-1)$ and $T^2(t) - T^2(t-1)$. The training and testing accuracies are close in ICA–SVM whatever the shift size is small or large. However, the accuracies of SVM and PCA–SVM are not favorable compared to that of ICA–SVM. For a clearer view of comparisons, Fig. 8 shows the fault detection rates against step sizes. From this figure, SVM performs better than PCA–SVM in

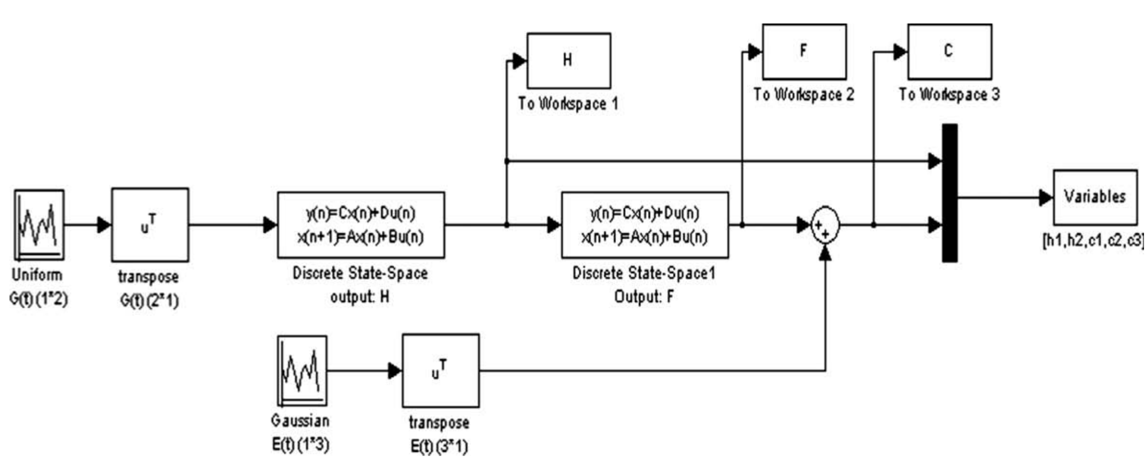

Fig. 3. Matlab/Simulink blocks for multivariate process simulation.

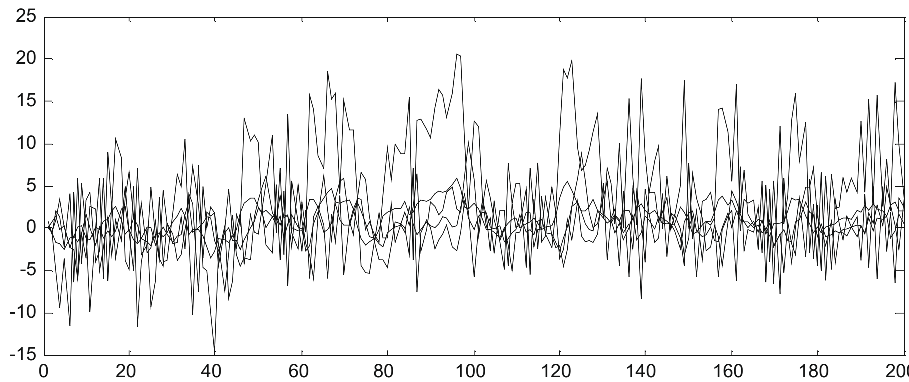

Fig. 4. Response for a step change of w_1 by 2 at sample 50.

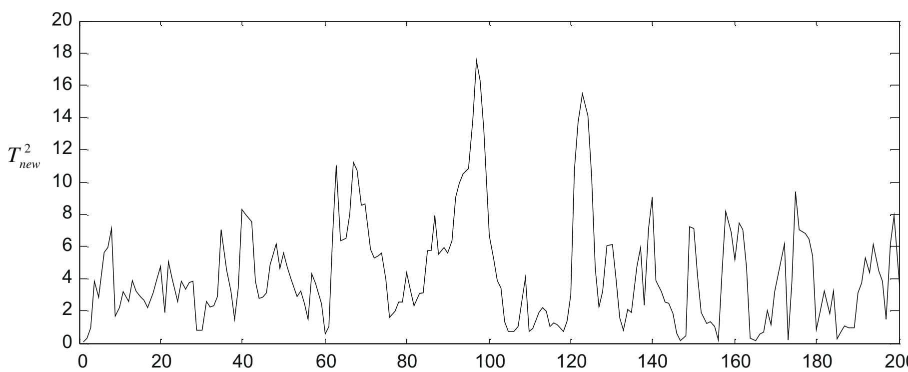

Fig. 5. T^2_{new} for step change with size 2 introduced at sample 50.

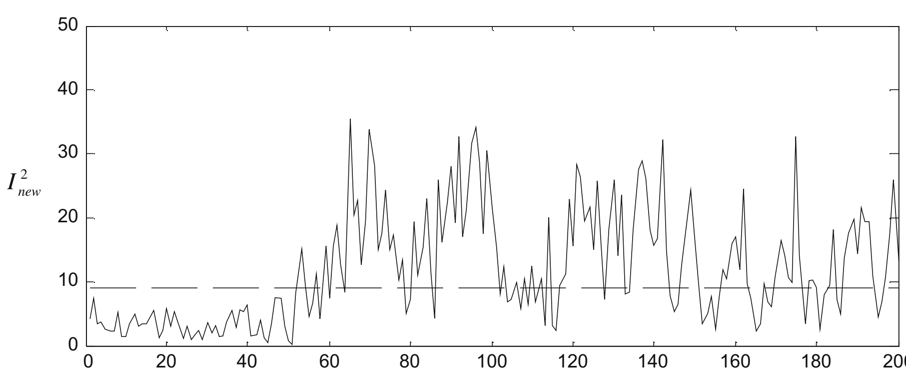

Fig. 6. I^2_{new} for step change with size 2 introduced at sample 50.

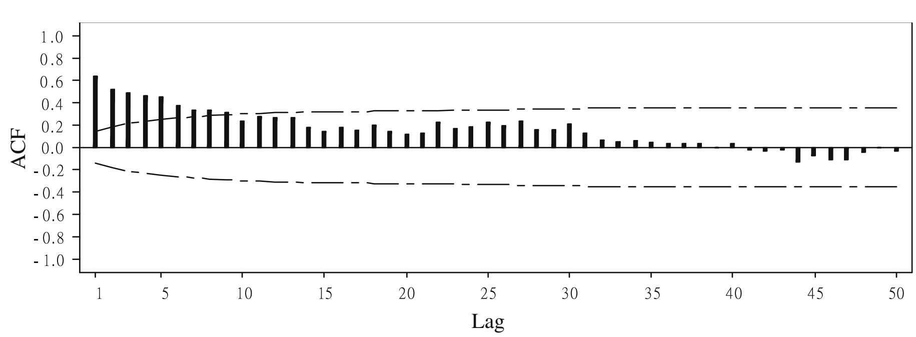

Fig. 7. ACF plot for I^2_{new} with step disturbance.

Table 1
Training and testing accuracies (step disturbance introduced at sample 50).

Step size	SVM without feature extraction		PCA–SVM fault detector		ICA–SVM fault detector	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
1	67.3	55.0	66.3	47.0	75.3	65.0
1.5	77.0	62.5	65.5	48.0	89.0	80.5
2	84.8	65.5	70.0	51.0	97.0	93.0
2.5	89.3	71.0	76.8	57.5	99.0	98.5
3	92.8	74.0	81.8	64.5	99.0	100
3.5	95.5	78.0	86.5	68.0	99.25	100
4	96.5	81.5	90.5	70.5	99.25	100
4.5	96.5	84.0	93.5	75.5	99.25	100
5	97.3	84.5	96.0	83.0	99.5	100
5.5	98.5	87.0	97.5	90.0	99.5	100
6	98.8	88.5	99.0	95.0	99.5	100
6.5	99.0	90.0	99.25	97.0	99.5	100
7	99.3	91.5	99.25	98.0	99.5	100

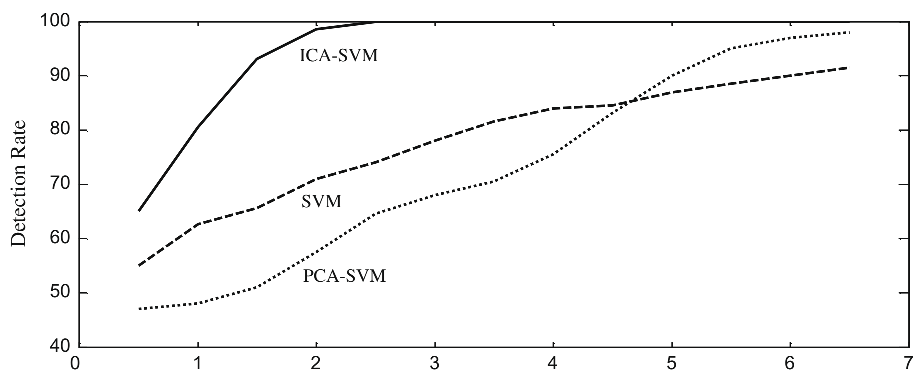

Fig. 8. Compare detection rates against shift sizes (solid line: ICA–SVM, dash line: SVM only, dot line: PCA–SVM).

small shifts, whereas PCA–SVM detects better than SVM in large shifts. Overall, the results clearly show that ICA–SVM can more effectively detect faults than SVM and PCA–SVM. For example, in the case of step size being 2.5, the proposed ICA–SVM can achieve nearly 100% detection rate; however detection rates of SVM and PCA–SVM are only 74% and 64.5%, respectively. Besides, ICA–SVM can produce above 90% detection rate when the step size is only 2, but 6.5 and 5.5 in SVM and PCA–SVM, respectively. These results indicate that ICA–SVM has the better ability to detect small shifts.

As a fault is detected, Eq. (13) can be used to diagnose the fault in process. Fig. 9 plots the variable contribution for I^2_{new} when a fault of a step change of g_1 by 2 is detected by ICA–SVM. It indicates that variable c_2 is primarily responsible for the step change of g_1.

5.3. Linear disturbance

A linear change of g_1 with a slope size of 0.05 introduced at the 50th sample is used for the simulation study of linear disturbance. Fig. 10 shows the obtained I^2_{new}, and it can illustrate the simulated disturbance type. The dash line is the 99% control limit obtained by KDE. From Fig. 10, the detection rate is not satisfactory either. Although the disturbance is introduced at the 50th sample, it is detected at around the 80th sample. The ACF plot for I^2_{new} shown in Fig. 11 demonstrates the high autocorrelation of I^2_{new}. The optimal SVM parameters as well are automatically generated by LIBSVM.

Similar to Tables 1, 2 summarizes the training and testing accuracies of the three approaches investigated herein under various slope sizes. From Table 2, there is no significant difference between

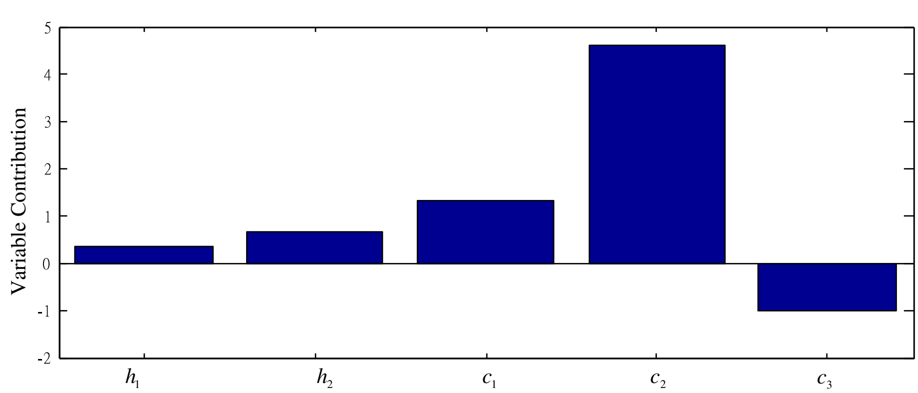

Fig. 9. Variables contribution plot for I^2_{new} (Step change of g_1 with 2 at sample 50).

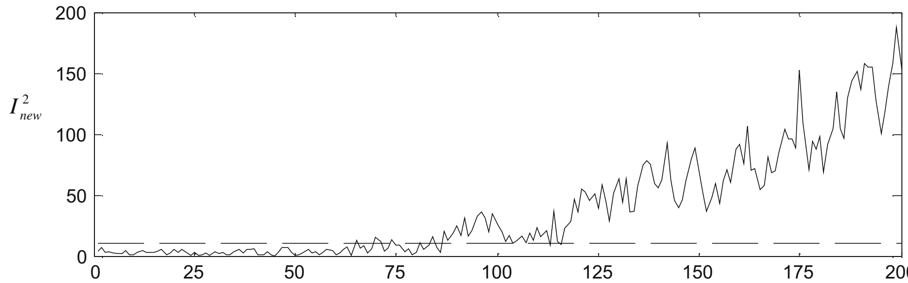

Fig. 10. I^2_{new} for linear change with slope 0.05 at sample 50.

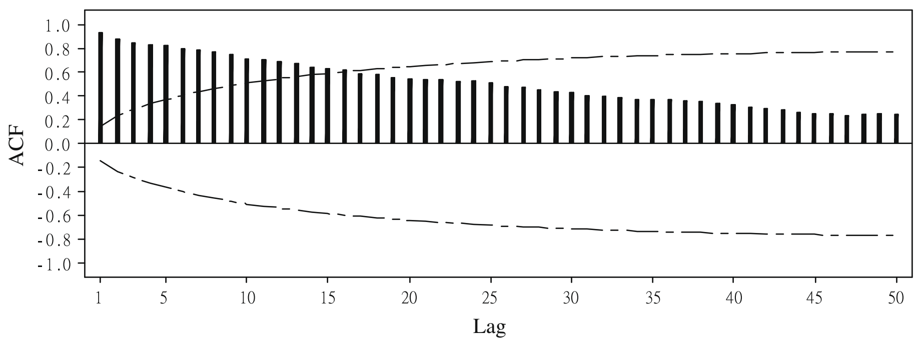

Fig. 11. ACF plot for I^2_{new} with linear disturbance.

Table 2
Training and testing accuracies (linear disturbance introduced at sample 50).

Slope size	SVM without feature extraction		PCA–SVM fault detector		ICA–SVM fault detector	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
0.01	66.0	53.0	63.0	39.0	73.8	60.0
0.03	78.5	63.0	75.8	48.0	88.5	82.5
0.05	86.5	75.0	82.8	62.0	92.8	91.5
0.07	89.0	77.5	88.5	71.5	94.3	93.5
0.1	90.5	79.0	91.8	82.0	96.3	94.0
0.2	93.0	82.5	94.5	91.5	97.3	96.0
0.3	94.5	83.5	96.3	92.5	97.8	96.5
0.4	94.5	84.0	96.5	93.5	98.5	96.5
0.5	94.5	84.0	97.5	94.5	98.8	98.0
0.6	94.8	84.0	97.8	93.5	99.0	97.5
0.7	94.8	84.5	98.0	93.5	99.0	98.0
0.8	94.8	85.0	98.0	94.5	99.0	98.5
0.9	94.8	85.0	98.5	96.0	99.0	99.0
1	95.0	85.0	98.5	96.0	99.0	99.0

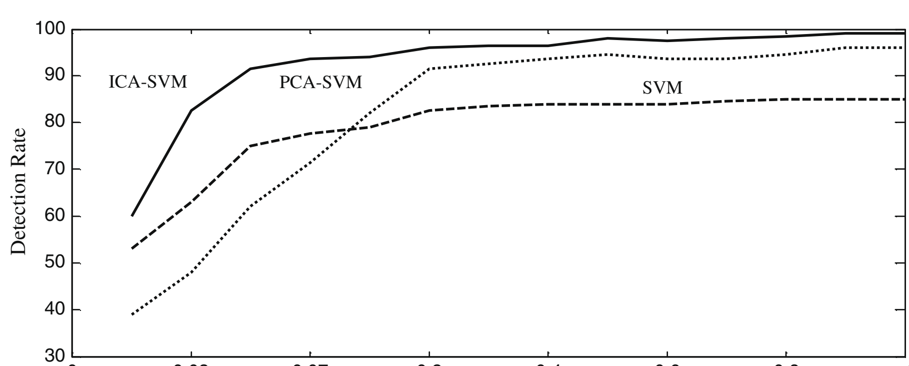

Fig. 12. Compare detection rates against slope sizes (solid line: ICA–SVM, dash line: SVM only, dot line: PCA–SVM).

training and testing accuracies in ICA–SVM, whereas SVM and PCA–SVM produce an obvious difference, especially in small slope sizes. Fig. 12 exhibits a clearer view of comparison results. SVM with feature extraction (PCA–SVM and ICA–SVM) can produce

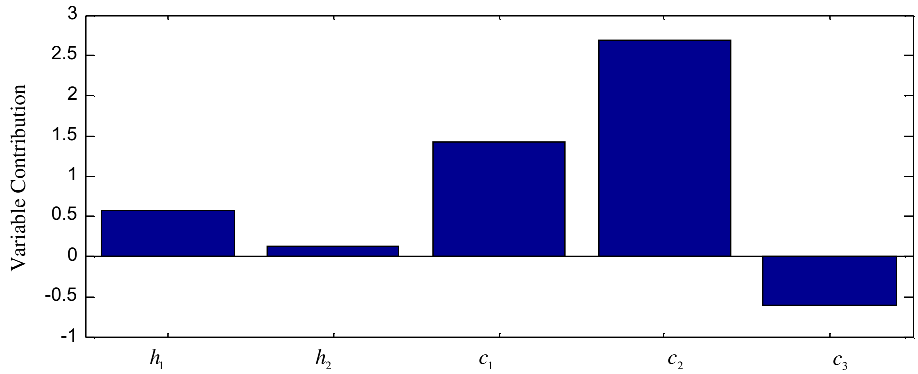

Fig. 13. Variable contribution plot for I^2_{new} at sample 70. (Linear change of g_1 with slope 0.05 at sample 50).

higher detection rates if slope sizes are greater than 0.1. Furthermore, the original SVM can only achieve a maximum of 85% detection rate, but PCA–SVM and ICA–SVM can achieve above 90% detection rate. Overall, ICA–SVM is clearly shown to be more effective for detecting the linear disturbance. For example, ICA–SVM can produce above 90% detection rate when the slope size is only 0.04, but 0.2 for PCA–SVM.

By using Eq. (13), Fig. 13 plots the variable contribution for I^2_{new} at sample 70, if a linear change of g_1 by a slope of 0.05 is introduced at sample 50. From Fig. 13, it indicates that variables c_1 and c_2 are primarily responsible for the linear change of g_1.

6. Conclusions

The ICA based statistic I^2 can reveal more disturbance type information than PCA based statistic T^2 when deal with a non-Gaussian multivariate process. However, by using KDE to I^2 may produce unsatisfactory detection rate due to the autocorrelation. This study proposes a simple but effective fault detector, ICA–SVM, for multivariate process monitoring. In ICA–SVM, ICA is used to project the high dimension variable space into a lower one, and then calculate the systematic part statistics such as $I^2(t)$, $I^2(t-1)$ and $I^2(t) - I^2(t-1)$ as input vectors for SVM. Two common encountered process disturbance types of step and linear are introduced to verify the efficiency of proposed methodology. From the simulation results, ICA–SVM can well detect small step change in the process. In linear disturbance, SVM with feature extraction (i.e., ICA–SVM and PCA–SVM) can achieve a higher detection rate than that of SVM without feature extraction. General speaking, the results clearly show that ICA–SVM can effectively detects faults in a multivariate dynamic process.

Even though the effectiveness of ICA–SVM is verified via simulation, nevertheless it is anticipated to improve the performance of multivariate process monitoring in an actual process such as semiconductor etch process, Tennessee Eastman process and so forth. This study considers only the original ICA model, further researches can extend the proposed method by using the dynamic ICA model and modified ICA model.

Acknowledgement

This work was supported in part by National Science Council of Taiwan (Grant No. NSC 97-2410-H-324-015).

References

Blundell, G., Maier, J., & Debevec, E. (2001). Linear home ranges: Effects of smoothing, sample size, and autocorrelation on kernel estimates. *Ecological Monographs, 71*, 469–489.

Cao, L. J., Chua, K. S., Chong, W. K., Lee, H. P., & Gu, Q. M. A. (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. *Software*. Available from: <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.

Chinnam, R. B. (2002). Support vector machines for recognizing shifts in correlated and other manufacturing processes. *International Journal of Production Research, 40*(17), 4449–4466.

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.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2007). *A practical guide to support vector classification*. Available from: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide>.

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.

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

Kumar, S., Choudhary, A. K., Kumar, M., Shankar, R., & Tiwari, M. K. (2006). Kernel distanced-based robust support vector methods and its application in developing a robust K-chart. *International Journal of Production Research, 44*(1), 77–96.

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.

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.

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, Preview Article*, 1–21.

Montgomery, D. C. (2005). *Introduction to statistical quality control* (5th ed.). New York, USA: John Wiley & Sons, Inc..

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.

Sun, R., & Tsung, F. (2003). A kernel-distance-based multivariate control chart using support vector methods. *International Journal of Production Research, 41*(13), 2975–2989.

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., & Han, T. (2007). Combination of independent component analysis and support vector machine for intelligent faults diagnosis of induction motors. *Expert Systems with Applications, 32*, 299–312.

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.