Available online at www.sciencedirect.com

SCIENCE DIRECT

Chemometrics and Intelligent Laboratory Systems 75 (2005) 55–67

Chemometrics and intelligent laboratory systems

www.elsevier.com/locate/chemolab

Fault detection and identification of nonlinear processes based on kernel PCA

Sang Wook Choi[a,*], Changkyu Lee[b], Jong-Min Lee[b], Jin Hyun Park[c], In-Beum Lee[b]

[a] *Foresight Centre of Process Analytics and Control Technology, School of Chemical Engineering and Advanced Materials, University of Newcastle upon Tyne, NE1 7RU, UK*

[b] *Department of Chemical Engineering, Pohang University of Science and Technology, San 31 Hyoja Dong, Pohang 790-784, South Korea*

[c] *PNI Consulting Co. Ltd., San 31 Hyoja Dong, Pohang 790-784, South Korea*

Available online 24 July 2004

Abstract

A new fault detection and identification method based on kernel principal component analysis (PCA) is described. In the past, numerous PCA-based statistical process monitoring methods have been developed and applied to various chemical processes. However, these previous methods assume that the monitored process is linear, whereas most of the chemical reactions in chemical processes are nonlinear. For such nonlinear systems, PCA-based monitoring has proved inefficient and problematic, prompting the development of several nonlinear PCA methods. In this paper, we propose a new nonlinear PCA-based method that uses kernel functions, and we compare the proposed method with previous methods. A unified fault detection index is developed based on the energy approximation concept. In particular, a new approach to fault identification, which is a challenging problem in nonlinear PCA, is formulated based on a robust reconstruction error calculation. The proposed monitoring method was applied to two simple nonlinear processes and the simulated continuous stirred tank reactor (CSTR) process. The monitoring results confirm that the proposed methodology affords credible fault detection and identification.

Keywords: Kernel principal component analysis; Data reconstruction; Fault detection and isolation; Monitoring statistics

1. Introduction

Multivariate statistical approaches to process monitoring including fault detection and diagnosis have been rapidly developed in recent decades. The emergence of a range of new sensors and data gathering equipment has enabled data to be collected with greater frequency from almost any process. Many statistical techniques for extracting process information from massive data sets and interpreting this information have been developed in various fields [1,2]. One approach that has proved particularly powerful is the use of principal component analysis (PCA) in combination with T^2 charts, Q charts, and contribution plots [3,4]. The chemometric techniques for multivariate process monitoring have been described in several review papers [4–6].

PCA is one of the most widely used linear techniques for transforming data into a new space. It divides data information into the significant patterns, such as linear tendencies or directions in model subspace, and the uncertainties, such as noises or outliers located in residual subspace. Two different statistics, represented by Mahalanobis and Euclidian distances, are used to elucidate the pattern variations in the model and residual subspaces, respectively.

Application of PCA-based monitoring to inherently nonlinear processes may lead to inefficient and unreliable process monitoring, because a linear PCA is inappropriate to describe the nonlinearity within the process variables [7]. To cope with this problem, extended versions of PCA suitable for handling nonlinear systems have been developed [7–10]. Kramer [8] proposed a nonlinear PCA based on an autoassociative neural network consisting of five layers, specifically the input, mapping, bottleneck, demapping, and output layers. In this formalism, the nonlinear extension of PCA was achieved by inserting nonlinear functions into the nodes of the mapping and demapping layers. Dong and McAvoy [7] formulated an alternative nonlinear version of PCA by combining a principal curve [11] and a neural network. Under their approach, the associated scores and the corrected data points for training samples are obtained by the principal curve method, and

* Corresponding author. Tel.: +82-54-279-2950; fax: +82-54-279-3499.

0169-7439/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2004.05.001

then the neural network model is used to map the original data into the corresponding scores and to map these scores into their corrected ones. In other work in this area, Tan and Mavrovouniotis [9] suggested a nonlinear PCA scheme based on an input-training neural network (ITNN), and Jia et al. [10] proposed an approach that combined linear PCA and ITNN in order to consider linear and nonlinear data correlations separately.

In this study, we develop a new nonlinear monitoring method based on kernel PCA, a type of PCA first put forward by Schölkopf et al. [12]. Conceptually, kernel PCA consists of two steps: (1) the mapping of the data set from its original space into a hyperdimensional feature space, and (2) a PCA calculation in the feature space. It has the following advantages over previous versions of nonlinear PCA. First, unlike nonlinear PCA methods based on neural networks, kernel PCA does not involve a nonlinear optimization procedure. And second, the calculation procedure used in linear PCA can be directly used in kernel PCA. One drawback of kernel PCA is that the computation time may increase with the number of samples, and the data pattern in the feature space is rather hard to interpret in the input space. Nevertheless, this study highlights the power of kernel PCA as a nonlinear process monitoring technique.

Like linear PCA, the kernel PCA approach proposed here utilizes T^2 and Q statistics in the extended feature space for fault detection. In addition, a unified statistic obtained by combining T^2 and Q based on the energy concept is suggested to facilitate the detection work. Fault isolation is a much more difficult problem in nonlinear PCA than in linear PCA, and as a result, there has been little research into fault isolation in nonlinear PCA-based monitoring. Here, we propose a methodology for fault identification in nonlinear systems that is based on the reconstruction-based approach to fault identification proposed by Dunia et al. [13] for linear PCA.

The remaining sections of this paper are structured as follows. Section 2 explains kernel PCA and its properties. In Section 3, the concepts outlined in Section 2 are used to formulate the process fault detection method. In Section 4, the fault identification strategy based on reconstruction is derived. The results of three kinds of experiments are presented and discussed in Section 5. Finally, Section 6 provides concluding remarks.

2. Kernel principal component analysis

PCA is a simple linear transformation technique that compresses high-dimensional data with minimum loss of data information. PCA is performed in the original sample space, whereas kernel PCA is carried out in the extended feature space. Kernel PCA is a type of kernel-based learning machine [12,14].

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n \in R^m$ be the training observations for kernel PCA learning. By the nonlinear mapping $\Phi: \boldsymbol{x} \in \Re^m \rightarrow \boldsymbol{z} \in \Re^h$, the measured inputs are extended into the hyperdimensional feature space as shown in Fig. 1. The dimension of the feature space, h, can be arbitrarily large or even infinite. The sample covariance in the feature space is given by

$$\Sigma_\Phi = \frac{1}{n}\sum_{i=1}^{n}(\Phi(\boldsymbol{x}_i) - \boldsymbol{m}_\Phi)(\Phi(\boldsymbol{x}_i) - \boldsymbol{m}_\Phi)^{\mathrm{T}} \tag{1}$$

where $\boldsymbol{m}_\Phi = \sum_{i=1}^{n}\Phi(\boldsymbol{x}_i)/n$ is the sample mean. Here we denote a mapped point after centering with the corresponding mean as $\bar{\Phi}(\boldsymbol{x}_i) = \Phi(\boldsymbol{x}_i) - \boldsymbol{m}_\Phi$. A principal component $\boldsymbol{v}$ is then computed by solving the eigenvalue problem:

$$\lambda \boldsymbol{v} = \Sigma_\Phi \boldsymbol{v} = \frac{1}{n}\sum_{i=1}^{n}(\bar{\Phi}(\boldsymbol{x}_i)^T \boldsymbol{v})\bar{\Phi}(\boldsymbol{x}_i) \tag{2}$$

where $\lambda > 0$ and $\boldsymbol{v} \neq 0$, which indicates that the principal component is spanned by $\bar{\Phi}(\boldsymbol{x}_1), \bar{\Phi}(\boldsymbol{x}_2), \ldots, \bar{\Phi}(\boldsymbol{x}_n)$, i.e., $\boldsymbol{v} = \sum_{i=1}^{n}\alpha_i\bar{\Phi}(\boldsymbol{x}_i)$. Then, by multiplying with $\bar{\Phi}(\boldsymbol{x}_j)$ from the left of both sides in Eq. (2), we obtain

$$\lambda(\bar{\Phi}(\boldsymbol{x}_j) \cdot \boldsymbol{v}) = \bar{\Phi}(\boldsymbol{x}_j) \cdot (\Sigma_\Phi \boldsymbol{v}) \tag{3}$$

By defining a kernel matrix $K_{ij} = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$ or $\bar{K}_{ij} = \bar{\Phi}(\boldsymbol{x}_i) \cdot \bar{\Phi}(\boldsymbol{x}_j) \in \Re^{n \times n}$, the eigenvalue problem can be represented by the following simple form:

$$\lambda \boldsymbol{\alpha} = (1/n)\bar{\boldsymbol{K}}\boldsymbol{\alpha} \tag{4}$$

for $^{\forall}\lambda > 0$, where $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ldots \alpha_n]^{\mathrm{T}}$. The centered kernel matrix $\bar{\boldsymbol{K}}$ is easily calculated using the noncentered kernel matrix $\boldsymbol{K}$: $\bar{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{E} - \boldsymbol{E}\boldsymbol{K} + \boldsymbol{E}\boldsymbol{K}\boldsymbol{E}$, where $E_{ij} = 1/n$. To ensure the normality of the principal component, i.e., $\|\boldsymbol{v}\|^2 = 1$, the calculated $\boldsymbol{\alpha}$ must be scaled such that they satisfy $\|\boldsymbol{\alpha}\|^2 = 1/n\lambda$.

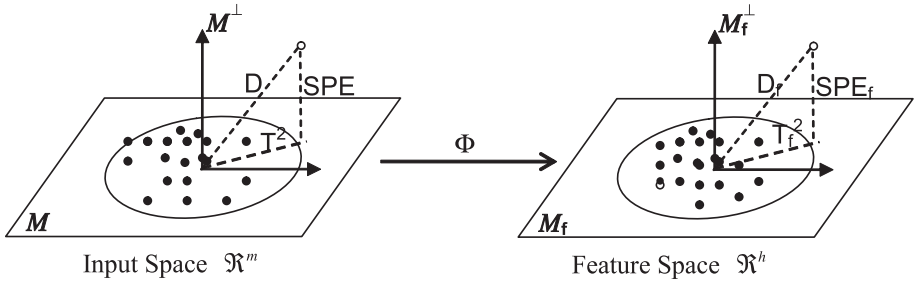

Fig. 1. Graphical depiction of the various metrics in kernel PCA.

After constructing the principal components in the feature space $\mathcal{F}$, the kth projection of the centered value $\bar{\Phi}(\boldsymbol{x}_{\text{new}})$ in $\mathcal{F}$ of the new sample $\boldsymbol{x}_{\text{new}}$ is calculated using

$$t_{\text{new},k} = (\boldsymbol{v}_k \cdot \bar{\Phi}(\boldsymbol{x}_{\text{new}})) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i^k (\bar{\Phi}(\boldsymbol{x}_i) \cdot \bar{\Phi}(\boldsymbol{x}_{\text{new}})) \tag{5}$$

for $k=1,\ldots,p$, where p is the number of principal components in the kernel PCA. Using Eq. (5), we finally obtain a score vector $t_{\text{new}}=[t_{\text{new},1}\ t_{\text{new},2}\ldots t_{\text{new},p}]^{\text{T}}$ for $\boldsymbol{x}_{\text{new}}$.

For the special case in which $\Phi(\boldsymbol{x})=\boldsymbol{x}$, kernel PCA is equivalent to linear PCA. From this viewpoint, kernel PCA can be regarded as a generalized version of linear PCA.

3. Process monitoring: fault detection

3.1. Energy decomposition

A graphical description of the two statistics used for process fault detection is given in Fig. 1. In general, conventional PCA-based fault detection methods use T^2 and the squared prediction error (SPE), which are expressed in terms of Mahalanobis and Euclidian type distances, respectively. T^2 represents the distance in model subspace $\mathcal{M}$, whereas the SPE indicates a distance from model subspace. The confidence limits for these indicators are calculated based on the assumption of the multivariate normality of observations and errors, respectively.

The observations obtained from a significantly nonlinear process are highly non-Gaussian due to the nonlinearity. Hence, the nonlinear mapping to the higher dimensional feature space is formulated such that the training samples conform to a Gaussian distribution after the nonlinear mapping [15]. Then, the distribution of the mapped training data can be estimated by a normal probability density in $\mathcal{F}$. The corresponding energy [15], represented by the negative logarithm of the probability, is given by

$$\Omega_{\Phi}(\boldsymbol{x}) = \bar{\Phi}^{\text{T}}(\boldsymbol{x})(\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\Phi}})^{-1}\bar{\Phi}(\boldsymbol{x}) \tag{6}$$

Here, the regularized covariance $\tilde{\Sigma}_{\Phi}$ is calculated using

$$\tilde{\Sigma}_{\Phi} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\text{T}} + \lambda_c(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^{\text{T}}) \tag{7}$$

where all zero or near-zero eigenvalues of Σ_{Φ} are replaced by a constant λ_c, and $\boldsymbol{\Lambda}$ denotes the diagonal matrix of nonzero or significant eigenvalues. Inserting Eq. (7) into Eq. (6) and using the kernel matrix notation, the energy in Eq. (6) can be divided into two terms:

$$\begin{aligned}\Omega_{\Phi}(\boldsymbol{x}) &= \sum_{k=1}^{p}\frac{1}{\lambda_k}(\boldsymbol{v}_k \cdot \bar{\Phi}(\boldsymbol{x}))^2 + \frac{1}{\lambda_c} \\ &\quad \times \left(||\bar{\Phi}(\boldsymbol{x})||^2 - \sum_{k=1}^{p}(\boldsymbol{v}_k \cdot \bar{\Phi}(\boldsymbol{x}))^2\right) \\ &= \boldsymbol{t}^{\text{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{t} + (\lambda_c)^{-1}(\bar{K}(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{t}^{\text{T}}\boldsymbol{t})\end{aligned} \tag{8}$$

where $\Omega_{\Phi}(\boldsymbol{x})$ represents the estimated statistic for characterizing the likelihood of an input $\boldsymbol{x}$. As shown in Fig. 1, the first and second terms of the right side of Eq. (8) represent a distance in feature space and a distance from feature space, respectively.

3.2. Fault detection using T^2, SPE, and the unified index

Similar to T^2 and SPE in linear PCA, the corresponding metrics in the feature space of kernel PCA, denoted T_{f}^2 ($\boldsymbol{t}^{\text{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{t}$) and SPE_{f} ($\|\boldsymbol{e}\|^2 \equiv \bar{K}(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{t}^{\text{T}}\boldsymbol{t}$), can be monitored in order to detect faults, because we assume that the training data are multivariate normal in the feature space. Given that we do not know the covariance matrix of the data in feature space exactly, T_{f}^2 and SPE_{f} may not conform to a generalized χ^2 distribution. Alternatively, the $100(1-\alpha)\%$ control limit for T_{f}^2 is calculated by means of an F-distribution as

$$T_{\text{f,lim}}^2 = \frac{p(n-1)}{n-p}F(p,n-1;\alpha) \tag{9}$$

where $F(p,n-1;\alpha)$ is an F-distribution with degrees of freedom p and $n-1$ with level of significance α. On the other hand, the $100(1-\alpha)\%$ control limit for SPE_{f} is

$$SPE_{\text{f,lim}} = \theta_1\left[\frac{c_\alpha\sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0(h_0-1)}{\theta_1^2}\right]^{1/h_0} \tag{10}$$

where $\theta_j = \sum_{s+1}^{h}(\Sigma_{ii})^j \text{for} j=1,2,3\ \text{h}_0 = 1-2\theta_1\theta_3/3\theta_2^2$, and c_α c_α is the normal deviate cutting off an area of α upper tail of distribution if h_0 is positive and under the lower tail if h_0 is negative [16]. The T_{f}^2 index reflects a variation in the model subspace, whereas the SPE_{f} index does a variation from the model subspace.

Despite the widespread use of the T^2 and SPE indices in statistical monitoring, and the fact that their detection results derive from different traces of data in mutually complementary spaces, these two indices simultaneously detect abnormal situations very frequently, confusing for process operators carrying out the fault analysis. Thus, to simplify the fault detection task and to avoid confusion, the T^2 and SPE indices can be combined into a single unified index. Yue and Qin [17] proposed such a unified index for use in linear PCA-based monitoring. Here, we use the following weighted sum of T^2 and SPE as a unified index:

$$D_{\text{f}}(\boldsymbol{x}) = T_{\text{f}}^2(\boldsymbol{x}) + (\lambda_c)^{-1}\text{SPE}_{\text{f}}(\boldsymbol{x}). \tag{11}$$

This unified index differs from that proposed by Yue and Qin [17], in that our proposed index is derived not from a combination of two indices but from the estimated

likelihood of sample distribution. Before using the unified index for process monitoring, an alarm limit must be defined. We calculate the confidence limit of the unified index D_f using the kernel density estimation method [18,19].

3.3. Some critical points of kernel PCA

Several points must be considered when applying kernel PCA to chemical process monitoring. First, the dimension of the feature space should be determined. It can be determined by observing the sequence of eigenvalues of data in $\mathcal{F}$. Theoretically, the number of nonzero eigenvalues is equal to the hyperdimension. However, if a very large number of observations are used to train a kernel PCA, the hyperdimension becomes large and, as a consequence, the statistical control limit for SPE or D_f becomes unrealistically large. Hence, we empirically determine the hyperdimension as the smallest number of the ordered eigenvalues whose cumulative sum is above 99% of the sum of all eigenvalues. Second, the number of principal components, which is equal to the dimension of the model feature subspace M_f of kernel PCA, should also be determined. For this purpose, a cross-validation scheme based on the mean squared reconstruction error (MSRE) of data is used. The number of principal components that gives the smallest MSRE when kernel PCA is applied to validation data is determined the dimension of M_f. The reconstruction method is described in the Appendix. Another variable that is critical to the proposed process monitoring scheme is the width of the Gaussian kernel used in kernel PCA. There is no theoretical framework for specifying the optimal value for this parameter; hence, it was empirically set to be $5m$ for monitoring the simulated continuous stirred tank reactor (CSTR) process (Section 5.3), where m is the number of process variables.

4. Process monitoring: fault identification

Newly transformed variables formed by a nonlinear PCA are associated with nonlinear mapping of original process variables. In nonlinear PCA-based monitoring, fault identification is not straightforward because the relationship between the original and transformed variables is nonlinear. In linear PCA-based monitoring, fault identification is often carried out using contribution plots [20], which represent the contribution of each variable to each score, T^2 or SPE. By comparing these quantities for all variables, process operators can identify fault-generating process variable(s). Shao et al. [21] extended this approach to nonlinear cases by expressing the weight of each variable to a score as the first order partial derivative of the score with respect to the variable at the current sampling time, instead of defining the weight of each variable as the loading vector corresponding to the score as is done in linear PCA. However, this approach cannot be used with the kernel PCA approach, because in the kernel PCA, the nonlinear mapping is not represented as a differentiable function, and even it were, the mapping function is not known in most cases. An alternative to the approach of Shao et al. [21] is the reconstruction-based fault identification method for linear PCA-based monitoring proposed by Dunia et al. [13], which has been shown to be useful for discriminating sensor faults. This method uses the ratio of two kinds of squared error, one is the SPE and the other is the SPE, after one variable has been replaced by its value obtained by using the remaining variables to estimate its value based on sample correlation. The core concept of this approach is that, when the reconstructed variable is a faulty process variable, the use of the reconstructed value instead of the true value will significantly reduce the SPE. We exploit this concept to solve the fault identification problem in kernel PCA-based monitoring.

The fault index, expressed as the ratio of two kinds of error, is given by

$$\zeta_i = \frac{\left|\left|\tilde{\boldsymbol{x}}^i - \hat{\tilde{\boldsymbol{x}}}^i\right|\right|^2}{\left|\left|\boldsymbol{x} - \hat{\boldsymbol{x}}\right|\right|^2} \tag{12}$$

where $\tilde{\boldsymbol{x}}^i$ is identical to $\boldsymbol{x}$ except for the ith entry, which is replaced by the corrected value, and $\hat{\tilde{\boldsymbol{x}}}^i$ is its estimate. The denominator of Eq. (12) represents the SPE of $\boldsymbol{x}$ (denoted by SPE_1) in the original variable space, and the numerator represents the SPE of $\tilde{\boldsymbol{x}}^i$ (denoted by $SPE_{2,i}$) in the original variable space. The tracking of an abnormal pattern in the original sample space rather than in the nonlinearly transformed feature space is needed to find a process variable or a variable group responsible for the abnormality. However, the problem to find abnormal trace in the original space does not have an analytical solution in most cases, because it is difficult or even impossible to find an inverse mapping function from the feature space to the original space. As an alternative, the reconstruction of the data in the original space can give clues that can be used to identify process abnormalities. An estimator of faulty values should be very robust to outliers and unseen patterns compared with normal training data. In order to meet this requirement, we slightly modify the denoising method proposed by Takahashi and Kurita [22], which is explained in detail in the Appendix. Then, two kinds of error, SPE_1 and SPE_2, are considered. When a fault occurs, SPE_1 increases substantially. On the other hand, if the value of the ith variable deviates from the normal value, $SPE_{2,j}$ ($j = 1,2,\ldots i-1,i+1,\ m$) also deviates from the normal magnitude because the ith faulty variable is not reconstructed, whereas $SPE_{2,i}$ does not increase considerably, because it is calculated based on the

Table 1
Average detection and isolation error rates of faults

Method		Linear PCA	Kernel PCA
Detection error rate (%)	Min	67.5	22.5
	Max	91.5	49.5
	Average	81.5	34.8
Isolation error rate (%)	Min	–	38.0
	Max		66.3
	Average		51.1

In kernel PCA, the dimension of the extended feature space is 60, and the dimension of model subspace is 37.

reconstructed value of the faulty variable. That is, ζ_i is significantly reduced if the ith process variable is the main source of the abnormal pattern.

5. Results and discussion

5.1. Simulated 2×2 system

First, we carry out kernel PCA-based fault detection and isolation for the following 2×2 system:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} (u_1(t) + u_2(t))^2 \\ (u_1(t) - ku_2(t))^2 \end{bmatrix} \quad (13)$$

where y_1 and y_2 are process outputs, u_1 and u_2 are process inputs, which are taken from a white Gaussian noise distribution with $N(0,0.1)$, and $k=2$ for $t=1-100$, and $k=2.5$ for $t=101-300$. Two inputs and two outputs are used for this process monitoring. We consider first 100 samples as a training set and samples 101–300 as a faulty test set. A total of 100 simulations are performed to obtain an impartial comparison between linear and kernel PCA.

In the linear PCA model, the three principal components, which explain about 81.9% of the total variance in the data, are used for capturing significant patterns. The kernel used in the kernel PCA was a Gaussian kernel of width 0.5, which provides the best performance for this system. Table 1 shows a comparison of the average detection and isolation error rates for the linear and kernel PCA-based monitoring methods, where the error rate of fault detection is defined as the ratio of the samples undetected to all faulty test samples (type II error rate), and the error rate of fault isolation is defined as the ratio of incorrectly identified samples to all faulty test samples. The average detection error rates of linear and kernel PCA are 81.5% and 34.8%, respectively, indicating that use of the kernel PCA model instead of the linear PCA model reduces the detection error rate by 46.7% on average. The kernel PCA method still does not detect 100% of the faulty samples, but the time-series patterns of the T^2 and Q statistics (Fig. 2) are sufficiently different between the faulty and normal condition to detect that the process is in an abnormal condition. As shown in Fig. 2, the Q statistic, represented by the SPE, abruptly increases at the onset of the fault ($t=101$). In contrast, the monitoring charts of linear PCA, shown in Fig. 3, barely capture the fault in this nonlinear system. Moreover, they exhibit several significant false alarms in the normal condition, referred to as type I errors, deteriorating the reliability of online decisions made using these charts. These results show that linear PCA-based monitoring

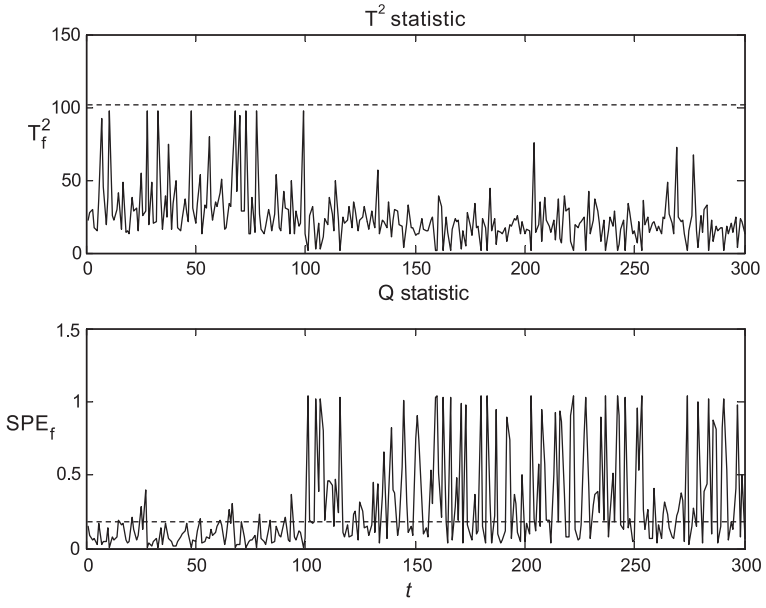

Fig. 2. Monitoring charts of the kernel PCA based monitoring for the simulated 2×2 system: (a) T^2 statistic and (b) Q statistic. The dotted line represents the 99% confidence limit.

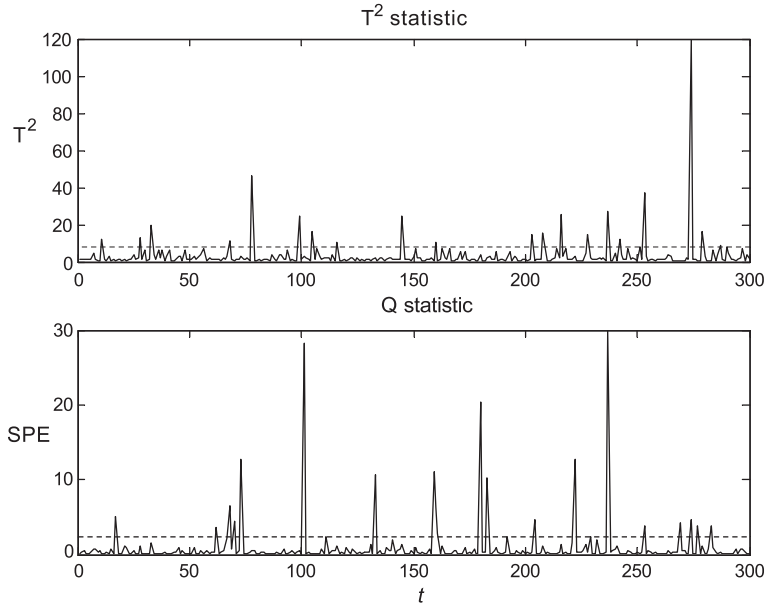

Fig. 3. Monitoring charts of the linear PCA-based monitoring for the simulated 2×2 system: (a) T^2 statistic and (b) Q statistic.

gives spurious results when applied to this inherently nonlinear system, whereas the proposed method behaves correctly.

5.2. Simulated nonlinear system

To compare the proposed method with principal curve-based nonlinear PCA, the data set used in Dong and McAvoy [7] is simulated and analyzed using the two nonlinear PCA-based methods. The system consists of three kinds of variables nonlinearly related with t, which are generated by

$$\boldsymbol{x} = \begin{bmatrix} t \\ t^2 - 3t \\ -t^3 + 3t^2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{14}$$

where $e_i \sim N(0, 0.01)$ is a white Gaussian noise and $t \in [0.01, 1]$. The normal training data comprise 100 samples (samples

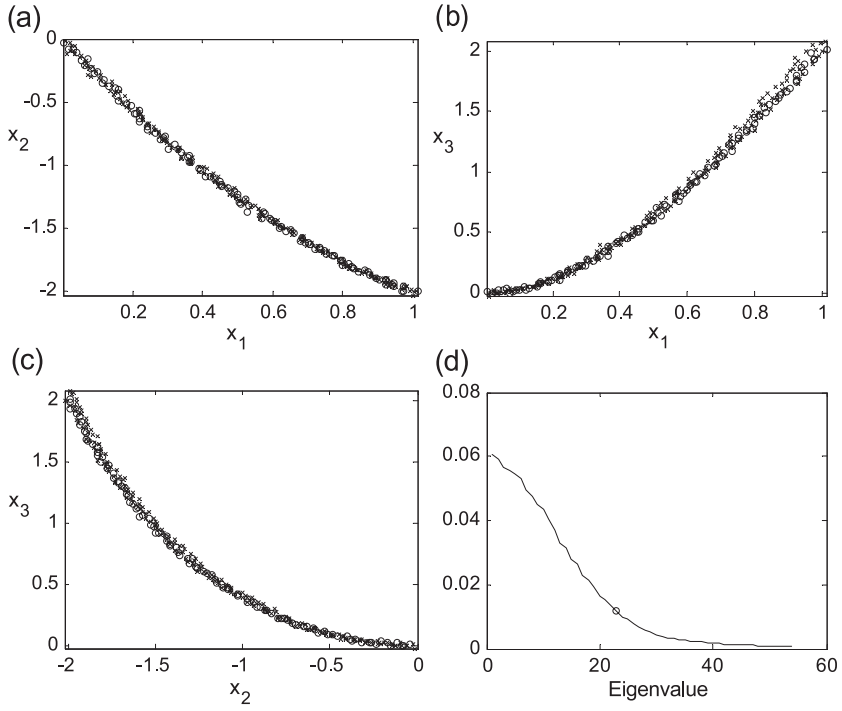

Fig. 4. Data patterns for the normal (circle) and fault (cross) conditions: (a) $\boldsymbol{x}_1$ vs. $\boldsymbol{x}_2$, (b) $\boldsymbol{x}_1$ vs. $\boldsymbol{x}_3$, (c) $\boldsymbol{x}_2$ vs. $\boldsymbol{x}_3$, and (d) eigenvalues obtained from kernel PCA. The circle represents the cutoff number of principal components used for kernel PCA model construction.

Table 2
Average type I and II error rates in kernel PCA

Bias magnitude	Type I error			Type II error		
	Min	Max	Average	Min	Max	Average
0.01	68.0	84.0	76.7	2.0	18.0	9.2
0.02	34.0	62.0	20.6	10.0	33.0	20.6
0.03	18.0	45.0	31.0	23.0	40.0	31.2
0.04	10.0	30.0	19.8	31.0	51.0	40.5
0.05	30.0	23.0	12.5	37.0	59.0	49.9
0.1	0.0	5.0	2.5	70.0	90.0	77.6

1–100) obtained using Eq. (14). The test data set consists of two parts: the first 100 samples (samples 101–200) are collected in the same condition as the normal condition, and the 100 test samples (samples 201–300) are obtained using the same system but with x_3 expressed as $x_3 = -1.1t^3 + 3.2t^2 + e_3$. Thus, the first and second halves of the 200 test samples are considered the normal and fault sets, respectively. As in the first case study, the process simulation and monitoring was performed 100 times to obtain reliable results. Fig. 4(a)–(c) depict the training and test data patterns obtained from one representative simulation, which are represented as 2-D plots spanned by two variables, which shows that the training and test data sets are very similar. Fig. 4(d) shows the eigenvalues obtained using kernel PCA with the number of principal components set to 23.

Table 2 shows the detection results of the kernel PCA-based monitoring. Type I and II errors correspond to false and missing alarms, respectively. As the width of a Gaussian kernel function (σ) increases, the frequency of type II errors tends to increase, whereas type I errors become less frequent. Thus, the choice of σ significantly impacts the process monitoring performance, and its optimal value should be determined taking into consideration the opposing effects of σ on type I and II errors. In this case, a value in the range $\sigma = 0.04-0.05$ may be reasonable. If we focus on the suppression of false alarms, the parameter should optimally be close to 0.05, but if we emphasize minimizing missing alarms, it should be close to 0.04. Figs. 5 and 6 show the process monitoring charts for linear and kernel PCA monitoring using $\sigma = 0.047$, respectively. The two linear PCA control charts hardly detect the abnormality, whereas the kernel PCA charts successfully show a deviation in their patterns over the range $t = 201-300$, with a false alarm rate of 13%. For the same data set, the missing rates of the linear PCA model and the principal curve-based nonlinear PCA model at the 99% control limit were 99% and 74%, respectively. In contrast, the kernel PCA model had a missing rate of 45%. Hence, the kernel PCA approach showed far superior monitoring performance compared to the principal curve-based nonlinear PCA model. Because each score does not satisfy the normality assumption, which is evident from the fact that no T_f^2 value in Fig. 6(a) is close to zero, the control limit for T_f^2 cannot be obtained using Eq. (9). Hence, we determined the upper and lower control limits using the common univariate control scheme. For detecting very small deviations in nonlinear systems, we should set the width of the Gaussian function to be small, which will cause the T_f^2 statistic to have a relatively low value during the faulty condition compared to its average normal value.

5.3. Simulated CSTR process

We now use the proposed method to monitor a non-isothermal CSTR process. The process flow of the single, nonisothermal CSTR considered here is depicted in Fig. 7. We adopt the simulation model parameters and conditions used in Ref. [23]. The reactor model is based on three

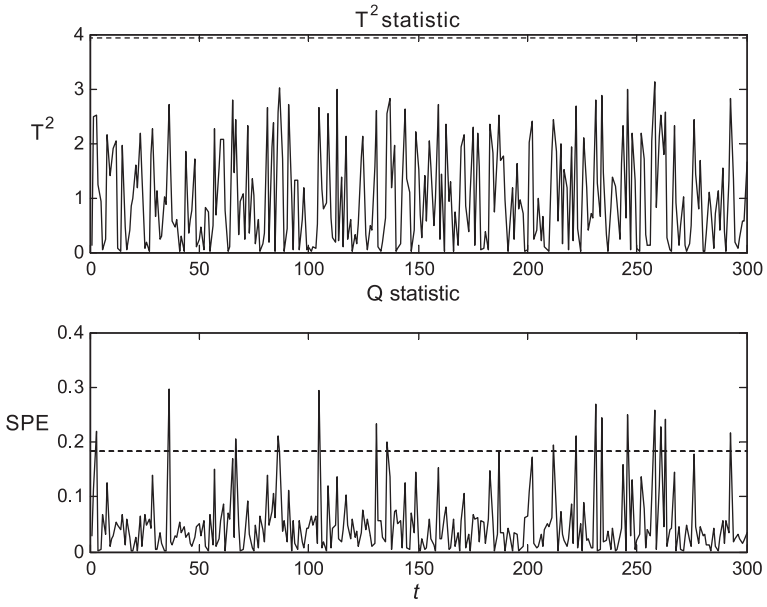

Fig. 5. Monitoring results using linear PCA.

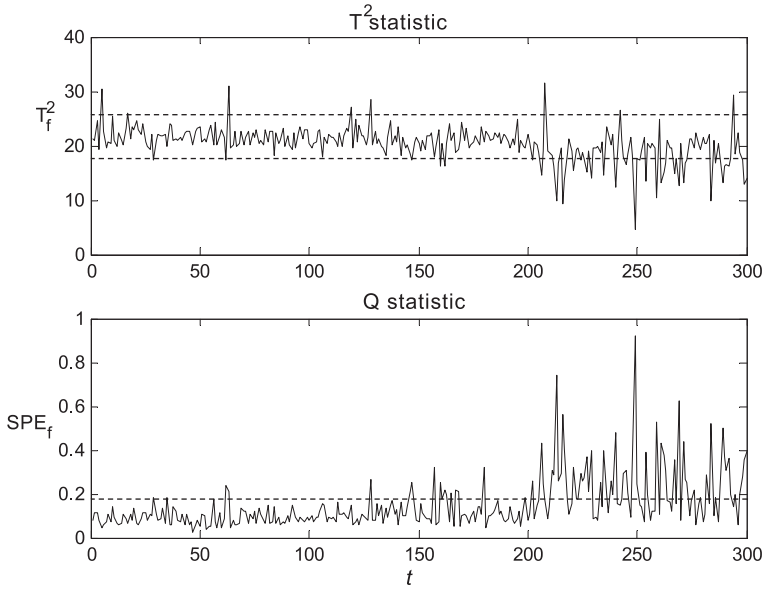

Fig. 6. Monitoring results using kernel PCA. T^2 statistics has 99% upper and lower confidence limits (the upper and lower dotted lines, respectively).

assumptions: perfect mixing, constant physical properties, and negligible shaft work. In the reactor, reactant A premixed with a solvent is converted into product B with the rate $r = k_0 e^{-E/RT} C_a$, where k_0 is the preexponential kinetic constant, E is the activation energy for the reaction, R is the ideal gas constant, and C_a is the reactant concentration. The dynamic behavior of the process is described by the mass balance of reactant A and the total energy balance for the reacting system, which are

$$V\frac{dC}{dt} = F(C - C_i) - Vr \tag{15}$$

$$V\rho C_p \frac{dT}{dt} = \rho C_p F(T_i - T) - \frac{aF_c^{b+1}}{F_c + aF_c^b/2\rho_c C_{pc}}(T - T_{ci}) + (-\Delta H_r)Vr \tag{16}$$

where V is the volume of reaction mixture in the tank, F and F_c are the flow rates of the reaction mixture and the coolant, ρ and ρ_c are the densities of the reaction mixture in the inlet stream and the coolant, respectively, C_p and C_{pc} are the specific heat capacities of the reaction mixture and the coolant, respectively, and ΔH_r is the heat of reaction. The process variables and parameters used in the mathematical modeling and monitoring of CSTR are summarized in Table 3. Only the temperature of reactant A in the outlet stream is controlled, which is implemented by regulating the coolant flow rate. All process disturbances are generated by the first order autoregressive model. In addition, all measured variables are disrupted by Gaussian white noise with different variances. A more detailed description of the process and simulation conditions is given in Ref. [23].

Four fault scenarios are generated to validate the proposed monitoring method. In all cases, the CSTR process is simulated for 350 min, with the fault being introduced at $t = 251$ min. The four types of abnormal condition considered here are summarized in Table 4. These faults can be categorized either as a simple fault,

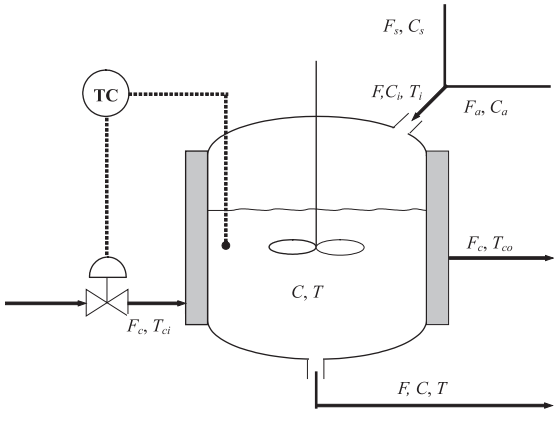

Fig. 7. Simulated nonisothermal CSTR process (F: flow rate; C: concentration; T: temperature/a: reactant; s: solute; c: coolant/i: inlet; o: outlet/TC: temperature controller).

Table 3
Summary of the CSTR process variables and parameters

Process variables
State variables: C, T
Controlled variable: T
Manipulated variable: F_c
Disturbances: F_i (or F_a, F_s), C_i (or C_a, C_s), T_i, T_{ci}, T_{co}
Measured (monitored) variables: $T_{ci}(1)$ $T_i(2)$ $C_a(3)$ $C_s(4)$ $F_s(5)$ $F_c(6)$ $C(7)$ $T(8)$
Parameters
$V = 1$ m^3; $\rho = 10^6$ g/m^3; $\rho_c = 10^6$ g/m^3; $C_p = 1$ cal/g/K; $C_{pc} = 1$ cal/g/K; $k_0 = 10^{10}$ min^{-1};
$a = 1.678 \times 10^6$ cal/min; $b = 0.5$; $\Delta H_r = -1.3 \times 10^7$ cal/kmol

Table 4
Summary of four fault scenarios in the simulated CSTR process

Cases	Fault type	Fault location	Fault pattern
1	Sensor bias	T	bias: $0 \rightarrow 1$
2	Sensor bias	T_i	bias: $0 \rightarrow 1.5$
3	Sensor bias	C_a	bias: $0 \rightarrow 1.0$
4	Sensor drift	C_a	$dC_{ai}/dt = 0.2$

if the effect of the fault is not propagated into other variables, or as a complex fault, if the effect of the fault is propagated into other variables [23]. In this respect, fault case 1 is a complex fault, whereas the others are simple faults. Eight process variables are measured every minute (Table 3). The first 200 samples are used as a training set and the subsequent 150 samples as a test set.

Fig. 8 shows the fault detection and identification result obtained using the proposed method for the bias-type fault at the reactor outlet temperature sensor (fault scenario 1 in Table 4). In the online monitoring result, SPE_f and D_f abruptly increase at the time the fault occurs (i.e., $t = 251$). In particular, the unified index D_f alone is sufficient to detect the sensor bias; hence, we may inspect the unified monitoring chart instead of the conventional T_f^2 and SPE_f charts. As shown in Fig. 8(d), the fault index ζ for the 8th variable T approaches zero with time, but the values of ζ for the other variables do not decrease. The SPE_f of $\tilde{\boldsymbol{x}}^8$, denoted by $SPE_{2,8}$, is relatively small, because the T value from the faulty sensor is fairly well reconstructed. Therefore, we can easily identify this propagated sensor fault using the reconstruction-based fault identification approach.

The second fault scenario involves a bias-type fault at the reactor input temperature sensor. The online monitoring results, depicted in Fig. 9, show that all of the statistics (T_f^2, SPE_f, and D_f) efficiently detect this fault type. Hence, it is sufficient to look at only the unified monitoring chart, which provides an effective alternative to the two conventional charts. The fault is correctly identified as soon as it occurs, as described in Fig. 9(d), and in this fault case, the identification result is more distinct than in the first fault scenario considered above. This result is not surprising given that this second fault case is a type of simple fault, which is easier to isolate than complex faults, such as those in the first fault scenario. The reconstruction of all measurements is illustrated in Fig. 10. When the fault is introduced, the sensor measurement is well recovered from its faulty pattern.

In the third fault scenario, a bias is introduced into the measured reactant concentration values in the inlet. The proposed method provides good detection and identification outcomes, as shown in Fig. 11. The abnormality is evident in the behavior of D_f and SPE_f, but not T^2, indicating that the fault pattern significantly deviates from the model subspace in the feature space. In the fourth fault scenario, a drift fault was introduced into the sensor measuring the inlet reactant concentra-

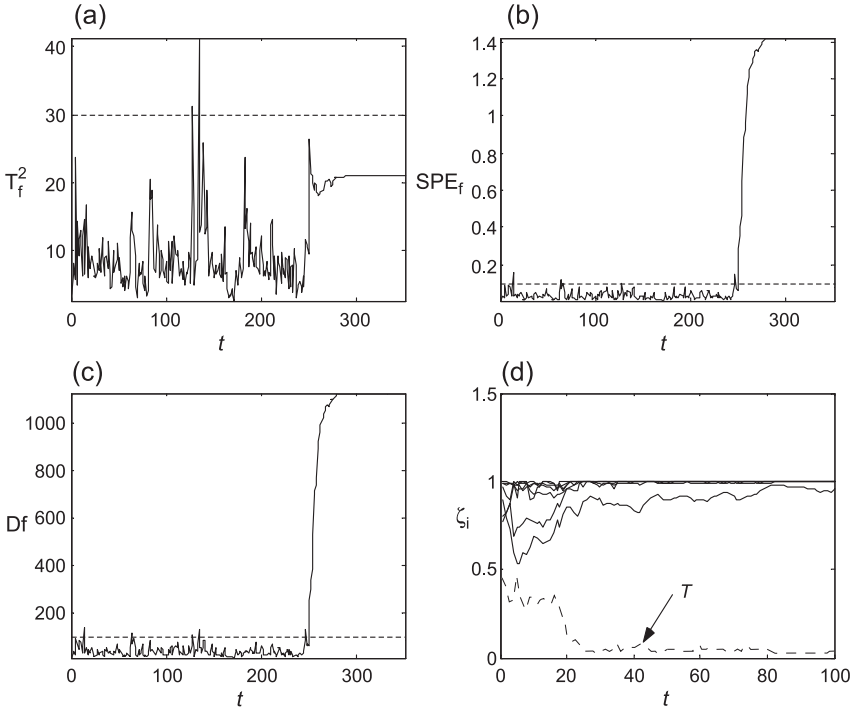

Fig. 8. Fault detection and isolation results for sensor bias fault (T). (a) T_f^2, (b) Q, (c) D_f, and (d) fault identification index. The dotted lines in (a), (b), and (c) are the 95% confidence limits.

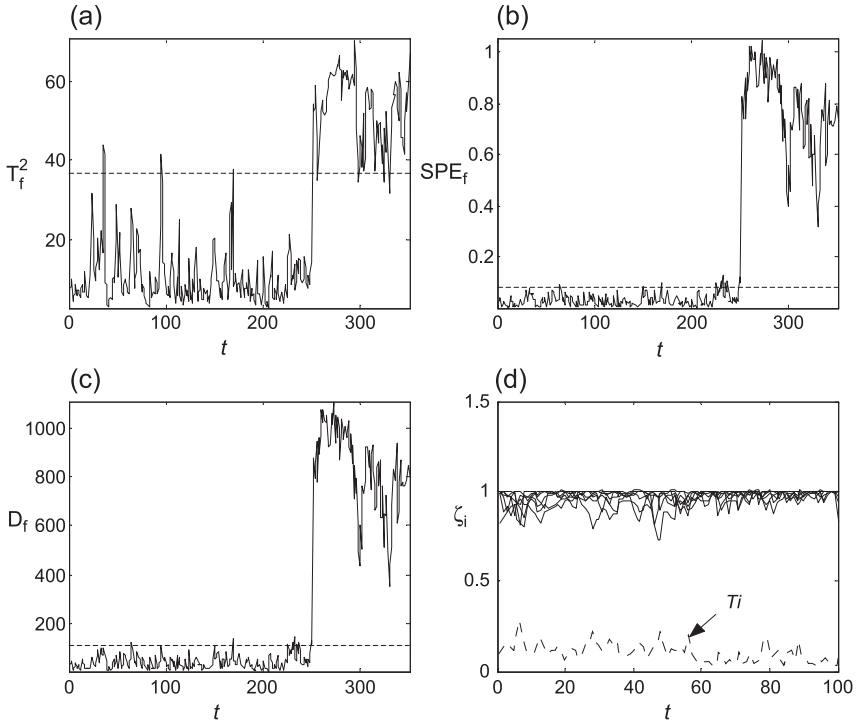

Fig. 9. Fault detection and isolation results for sensor bias fault (T_i). (a) T_f^2, (b) Q, (c) D_f, and (d) fault identification index.

tion. In this case, D_f gradually increases after the onset of the fault, as shown in Fig. 12. The fault is clearly identified after a short delay (20 min), as depicted in Fig. 12(d). For sensor drift faults such as those considered here, a small delay before fault identification may be inevitable because the magnitude of the drift must

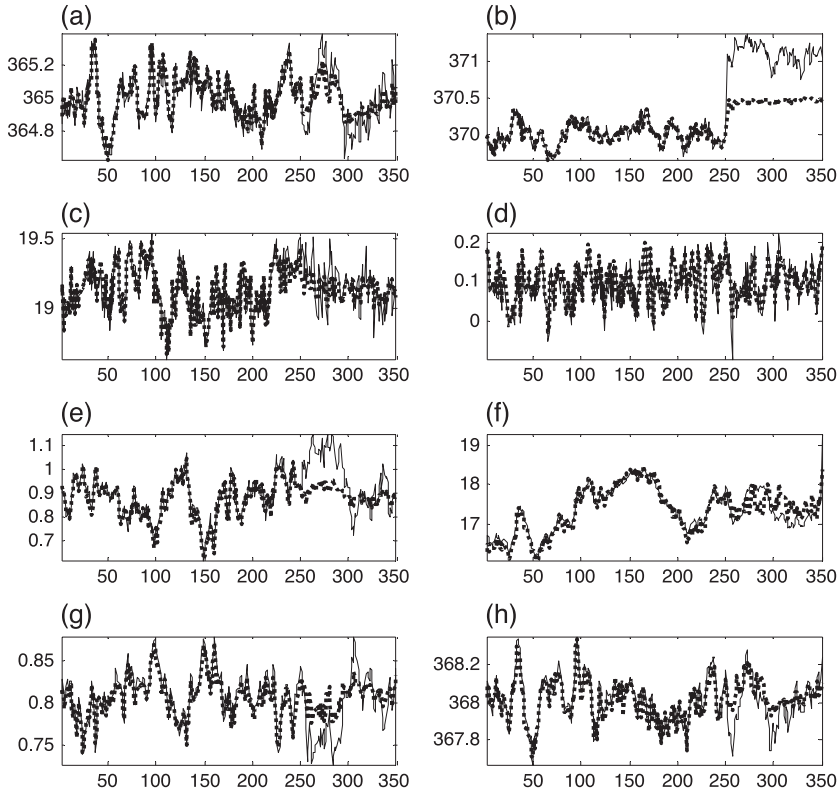

Fig. 10. Patterns of measurements (solid lines) and their reconstructed data (dotted lines) using the robust denoising method. (a) T_{ci}, (b) T_i, (c) C_a, (d) C_s, (e) F_s, (f) F_c, (g) C, and (h) T.

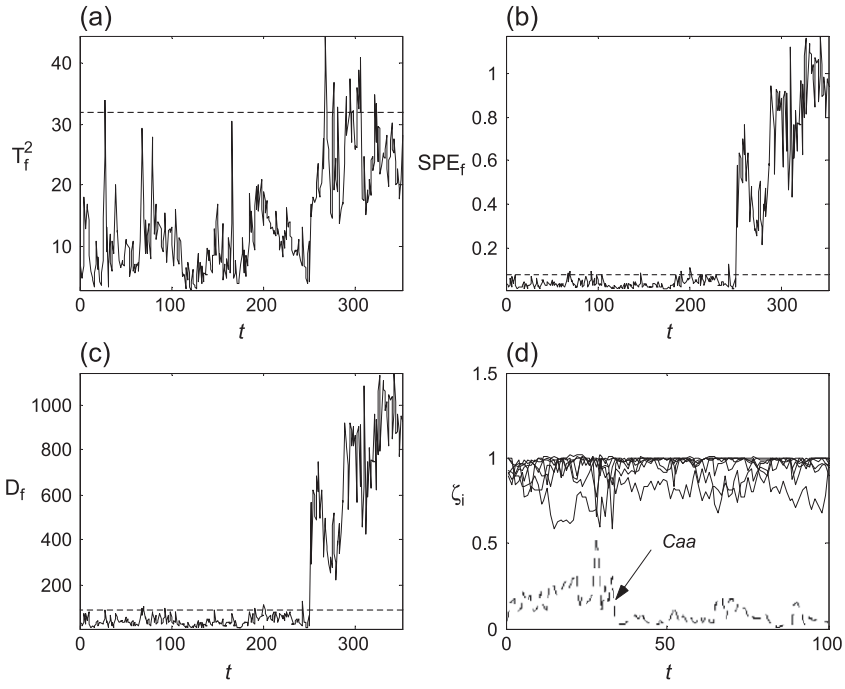

Fig. 11. Fault detection and isolation results for sensor bias fault (C_a). (a) T_f^2, (b)Q, (c)D_f, and (d) fault identification index.

become large enough to discriminate it from ordinary fluctuations.

6. Conclusions

In this paper, a new fault detection and identification method based on kernel PCA was formulated for supervising nonlinear processes. The proposed method gave good monitoring results for faults occurring in apparent nonlinear processes. A unified index, which combines the T^2 statistic with the Q statistic based on the energy concept, is proposed as a single monitoring statistic. The confidence limit for fault detection is developed using the kernel density estimation scheme. Additionally, a reconstruction-based identification index, represented by the

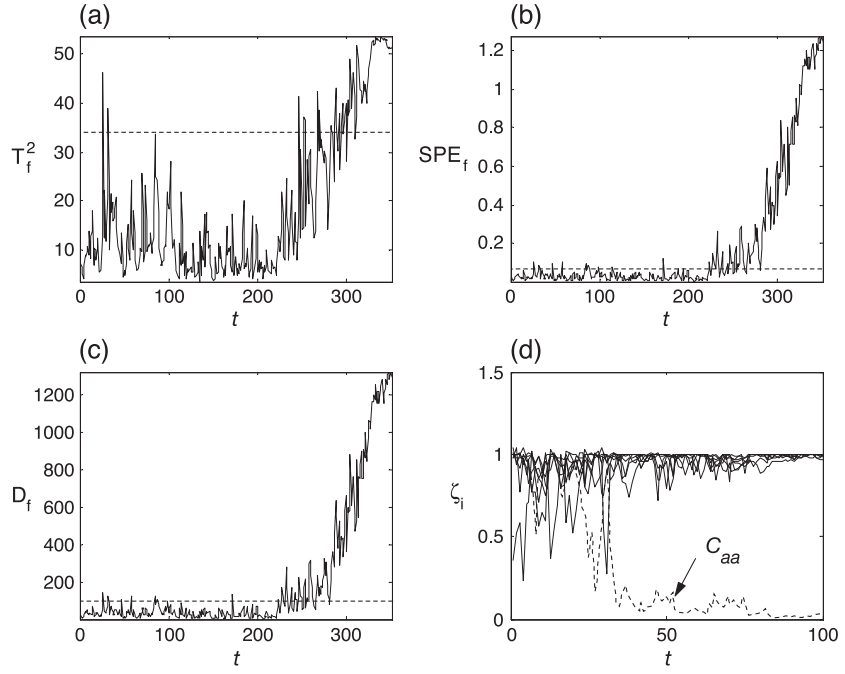

Fig. 12. Fault detection and isolation results for sensor bias fault (T). (a) T_f^2, (b) Q, (c) D_f, and (d) fault identification index.

ratio of two kinds of residuals, is developed for fault identification. Through two simple applications, the superior monitoring performance afforded by the proposed approach compared with linear PCA and principal curve-based nonlinear PCA was demonstrated. In addition, the proposed scheme was successfully applied to a CSTR process under four different fault scenarios. In future work, we plan to consider the optimal tuning of the parameter λ_c in the unified fault detection index for process monitoring, to apply the proposed method to a larger range of fault detection and identification problems, and to attempt to extend the kernel PCA method so as to remove the assumption that the observations in the extended space conform to a Gaussian distribution.

Acknowledgements

This work was supported by the Brain Korea 21 project.

Appendix A

In Section 4, we considered that $\boldsymbol{x}_{\text{new}}$ is reconstructed without the effects of noise and faulty signals. Robust reconstruction is achieved by slightly modifying the algorithm proposed by Takahashi and Kurita [22]. If a kernel function is restricted to a Gaussian form such as $K(\boldsymbol{x},\boldsymbol{y}) = \exp(-||\boldsymbol{x}-\boldsymbol{y}||/\sigma)$, an optimal reconstructed value $\boldsymbol{z}$ can be obtained by minimizing

$$\rho(\boldsymbol{z}) = \left|\left|\Phi(\mathrm{z}) - \boldsymbol{V}\boldsymbol{\Phi}(\boldsymbol{x}_{\text{new}})\right|\right|^2 \tag{A1}$$

To optimize Eq. (A1), the standard gradient method gives the following iterative form for computing $\boldsymbol{z}$:

$$\boldsymbol{z}(t) = \frac{\sum_{i=1}^{n} \beta_i K(\boldsymbol{x}_i, \boldsymbol{z}(t-1))\boldsymbol{x}_i}{\sum_{i=1}^{n} \beta_i K(\boldsymbol{x}_i, \boldsymbol{z}(t-1))} \tag{A2}$$

where $\beta_1 = \sum_{k=1}^{p} t_k \alpha_i^k$. Although an estimate of $\boldsymbol{x}_{\text{new}}$ can be obtained by only reconstructing its projection into the model subspace, this approach is not sufficient for dealing with outliers and unusual patterns. This problem is significantly alleviated by updating not only $\boldsymbol{z}$ but also the principal component [22]. Then, the update formula of Eq. (A2) is changed into the sequence of equations.

$$t_k(t) = \sum_{i=}^{n} \alpha_i^k K(\boldsymbol{x}_i, \dot{\boldsymbol{x}}(t)) \tag{A3}$$

and

$$\beta_i(t) = \sum_{k=1}^{p} t_k(t)\alpha_i^k \tag{A4}$$

where $\dot{\boldsymbol{x}}(t)$ is the weighted sum of two terms such as $\boldsymbol{x}_{\text{new}}$ and $\boldsymbol{z}(t-1)$, given by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{W}(t)\boldsymbol{x}_{\text{new}} + (\boldsymbol{I} - \boldsymbol{W}(t))\boldsymbol{z}(t-1) \tag{A5}$$

Here, $\boldsymbol{W}(\mathrm{t}) = \mathrm{diag}(w_1(\mathrm{t}), w_2(\mathrm{t}),\ldots, w_{\mathrm{m}}(\mathrm{t})) \in \Re^{\mathrm{m}\times\mathrm{m}}$, where $w_j(t)$ denotes the certainty of $\boldsymbol{x}_{\text{new},j}$ $(j = 1,2,\ldots m)$. The certainty $w_j(t)$ is represented by the average difference $\mathrm{D_j} = \mathrm{E}\langle|x_{\mathrm{new,j}} - x_{\mathrm{i,j}}|\rangle_{\mathrm{d}}$ between $\boldsymbol{x}_{\text{new}}$ and the d nearest neighbors of training samples where $E\langle\bullet\rangle_d$ is an expectation of the d smallest $\bullet$ values:

$$w_j(t) = \exp\left[-\frac{(x_j - z_j(t-1))^2}{((1.4826(1 + 5/(n-1)D_j)^2}\right] \tag{A6}$$

Thus, $\boldsymbol{W}(t)$ is relatively large when $\boldsymbol{x}_{\text{new}}$ is similar to the training normal data, and $\boldsymbol{W}(t)$ is small otherwise. In addition, Eq. (A5) indicates that $\boldsymbol{x}_{\text{new},j}$ is used for the next reconstruction step if it has high certainty, otherwise $\boldsymbol{z}_j$ calculated in the previous iteration is mainly used. Then, the estimated $\boldsymbol{z}(t)$ at the tth iteration is

$$\boldsymbol{z}(t) = \frac{\sum_{i=1}^{n} \beta_i K(\boldsymbol{x}_i, \dot{\boldsymbol{x}}(t))\boldsymbol{x}_i}{\sum_{i=1}^{n} \beta_i K(\boldsymbol{x}_i, \dot{\boldsymbol{x}}(t))} \tag{A7}$$

The recursive iteration of (A6) → (A5) → (A3) → (A4) → (A7) until $||\boldsymbol{z}(t) - \boldsymbol{z}(t-1)|| < 10^{-5}$ gives the converged robust estimate $\boldsymbol{z}$ for $\boldsymbol{x}_{\text{new}}$.

References

[1] M. Daszykowski, B. Walczak, D.L. Massart, Projection methods in chemistry, Chemical Intelligence Laboratory Systems 65 (2003) 97–112.
[2] R.A. Johnson, D.W. Wichern, Applied Multivariate Statistical Analysis, Prentice-Hall, New Jersey, 1992.
[3] J. Kresta, J.F. MacGregor, T.E. Marlin, Multivariate statistical monitoring of process operating performance, Canadian Journal of Chemical Engineering 69 (1991) 35–47.
[4] B.M. Wise, N.L. Ricker, Recent advances in multivariate statistical process control: improving robustness and sensitivity, Proceedings of IFAC International Symposium, ADCHEM '91, Paul Sabatier University, Toulouse, 1991, pp. 125–130.
[5] J.F. MacGregor, T. Kourti, Statistical process control of multivariate processes, Control Engineering Practice 3 (1995) 403–414.
[6] B.M. Wise, N.B. Gallagher, The process chemometrics approach to process monitoring and fault detection, Journal of Process Control 6 (1996) 329–348.
[7] D. Dong, T.J. McAvoy, Nonlinear principal component analysis—based on principal curves and neural networks, Computers and Chemical Engineering 20 (1996) 65–78.
[8] M.A. Kramer, Non-linear principal component analysis using autoassociative neural networks, AIChE Journal 37 (1991) 233–243.
[9] S. Tan, M.L. Mavrovouniotis, Reducing data dimensionality through optimising neural networks inputs, AIChE Journal 41 (1995) 1471–1480.

[10] F. Jia, E.B. Martin, A.J. Morris, Non-linear principal component analysis for process fault detection, Computers and Chemical Engineering 22 (1998) S851–S854.
[11] T.J. Hastie, W. Stuetzle, Principal curves, Journal of American Statistical Association 84 (1989) 502–516.
[12] B. Schölkopf, A.J. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.
[13] R. Dunia, S.J. Qin, T.F. Edgar, T.J. McAvoy, Identification of faulty sensors using principal component analysis, AIChE Journal 42 (1996) 2797–2812.
[14] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms, IEEE Transactions Neural Networks 12 (2001) 181–201.
[15] D. Cremers, T. Kohlberger, C. Schnörr, Shape statistics in kernel space for variational image segmentation, Pattern Recognition 36 (2003) 1929–1943.
[16] J.E. Jackson, A User's Guide to Principal Components, Wiley, NY, 1991.
[17] H.H. Yue, S.J. Qin, Reconstruction-based fault identification using a combined index, Industrial and Engineering Chemistry Research 40 (2001) 4403–4414.
[18] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon press, New York, 1995.
[19] C.C. Beardahy, M.J. Baxter, MATLAB routines for kernel density estimation and the graphical representation of archaeological data, Technical Report, 1995.
[20] P. Miller, R.E. Swanson, C.F. Heckler, Contribution plots: the missing link in multivariate quality control, Applied Mathematics and Computer Science 8 (1998) 775–792.
[21] R. Shao, F. Jia, E.B. Martin, A.J. Morris, Wavelets and non-linear principal components analysis for process monitoring, Control Engineering Practice 7 (1999) 865–879.
[22] T. Takahashi, T. Kurita, Robust de-noising by kernel PCA, Lecture Notes in Computer Science 2415 (2002) 739–744.
[23] S. Yoon, J.F. MacGregor, Fault diagnosis with multivariate statistical models: Part I. using steady state fault signatures, Journal of Process Control 11 (2001) 387–400.