Nonlinear Process Fault Diagnosis Based on Serial Principal Component Analysis

Xiaogang Deng, Xuemin Tian, Sheng Chen, *Fellow, IEEE*, and Chris J. Harris

***Abstract*—Many industrial processes contain both linear and nonlinear parts, and kernel principal component analysis (KPCA), widely used in nonlinear process monitoring, may not offer the most effective means for dealing with these nonlinear processes. This paper proposes a new hybrid linear-nonlinear statistical modeling approach for nonlinear process monitoring by closely integrating linear principal component analysis (PCA) and nonlinear KPCA using a serial model structure, which we refer to as serial principal component analysis (SPCA). Specifically, PCA is first applied to extract principal components (PCs) as linear features, and to decompose the data into the PC subspace and residuals subspace (RS). Then KPCA is performed in the RS to extract the nonlinear PCs as nonlinear features. Two monitoring statistics are constructed for fault detection, based on both the linear and nonlinear features extracted by the proposed SPCA. To effectively perform fault identification after a fault is detected, a SPCA similarity factor method is built for fault recognition, which fuses both the linear and nonlinear features. Unlike PCA and KPCA, the proposed method takes into account both linear and nonlinear PCs simultaneously and, therefore, it can better exploit the underlying process's structure to enhance fault diagnosis performance. Two case studies involving a simulated nonlinear process and the benchmark Tennessee Eastman process demonstrate that the proposed SPCA approach is more effective than the existing state-of-the-art approach based on KPCA alone, in terms of nonlinear process fault detection and identification.**

***Index Terms*—Nonlinear process monitoring, kernel principal component analysis, serial principal component analysis, fault detection, fault identification, similarity factor**

I. Introduction

As modern industrial processes become very complicated, large-scale and highly invested, fault diagnosis technology shows its great value in ensuring process safety and improving product quality. In the past several decades, industrial process fault diagnosis methods have been discussed extensively by researchers [1]–[3], and existing methods are usually divided into model, knowledge, and data based classes. Among these three classes, data based fault diagnosis methods have become an increasingly hot topic in recent years because large amounts of historical and real-time data are collected and stored in computer control system database [4]–[7]. Some representative data based methods include principal component analysis (PCA) [8]–[10], partial least squares [11], [12], independent component analysis [13]–[15] and canonical variate analysis [16], [17]. Also some cognitive fault diagnosis methods have been studied by mining online data information, particularly for sensor networks [18]–[20]

As one of most well-known data based fault diagnosis methods, PCA and its extensions have been studied in depth. Miletic *et al.* [21] reported the application of PCA based control charts for a continuous slab caster at Dofasco Company. For multiphase batch process, Zhao and Gao [22] improved PCA by considering the between-phase relative changes. To handle the correlated process data, dynamic PCA based on decorrelated residuals was developed by Rato and Reis [23]. Considering the multimode operations, several multimode PCA methods have been developed in the works [24]–[26]. As linear PCA cannot effectively deal with the nonlinear process monitoring problem, many modified nonlinear PCA methods have been developed. Krammer [27] firstly applied neural network to construct nonlinear PCA model. Dong and MacAvoy [28] generalized the linear PCA to the nonlinear principal curve method. Guo *et al.* [29] combined the radial basis function neural network and PCA to develop a nonlinear monitoring model.

More recently, kernel PCA (KPCA) has attracted great interest from researchers in fault diagnosis field. KPCA proposed by Schölkopf *et al.* [30] avoids complicated nonlinear optimization procedure by the use of kernel function. Lee *et al.* [31] firstly applied KPCA to fault detection, and Choi *et al.* [32] developed a KPCA contribution plot for fault identification. Zhang *et al.* [33] built an improved KPCA method, referred to as SFM-MSKPCA, by combining wavelet decomposition technique and sliding median filter. Since nonlinear principal components (PCs) extracted by kernel transformation may violate the Gaussian distribution assumption, Ge *et al.* [34] incorporated a statistical local approach with KPCA to construct new score variables that follow Gaussian distribution. Fan *et al.* [35] proposed a KICA-PCA method which considers both the nonlinear and non-Gaussian characteristics. Utilizing local data structure analysis, Deng *et al.* [36] proposed a local data structure assisted KPCA method for nonlinear fault detection. More recent researches for KPCA can be found in [37]–[39].

Although KPCA has achieved great success in process fault detection and identification, there exist some open problems or controversial issues to motivate the further study. One important open problem is: Can one single KPCA model be enough to describe the process data exactly? The characteristics of an industrial process are generally unknown and very complex. Although it is usually true that industrial process is nonlinear, both linear and nonlinear relationships often exist among industrial process variables. In some cases, one single nonlinear model may be not the best choice. Therefore, hybrid

X. Deng and X. Tian (dengxg2002@gmail.com, tianxm@upc.edu.cn) are with College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China

S. Chen and C.J. Harris (sqc@ecs.soton.ac.uk, cjh@ecs.soton.ac.uk) are with Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK. S. Chen is also with King Abdulaziz University, Jeddah 21589, Saudi Arabia.

This work was supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2014FL016), and the National Natural Science Foundation of China (Grant Nos. 61403418 and 61273160).

linear-nonlinear modeling can offer a viable alternative to mine the process data features. In the time series prediction field, there are some successful cases of applying hybrid linear and nonlinear modeling [40]–[44]. More specifically, Zhang [40] proposed to build a hybrid ARIMA and neural network model, and the results of [40] indicate that the combined model can be more effective than either single linear model or single nonlinear model. Chen [41] and Xiong *et al.* [42] applied hybrid linear-nonlinear model to predict tourism demand and agricultural commodity futures prices, respectively. Alippi *et al.* [43] applied the ensemble of linear time invariant model and nonlinear reservoir network to reconstruct the missing data in sensor networks. Zhang *et al.* [44] performed nonlinear system identification by combing a linear model with a nonlinear compensation term. As shown in [41], none of a linear model or a nonlinear model alone can offer the solution to all situations and a combination strategy is a better choice for data modeling.

Motivated by the above analysis, we propose a new hybrid linear-nonlinear statistical modeling approach for nonlinear process monitoring by integrating PCA and KPCA closely using a serial model structure, which we refer to as serial principal component analysis (SPCA). To our best knowledge, we are the first to propose a combined linear PCA and nonlinear KPCA method to process fault detection and diagnosis. Our contribution is twofold. Firstly, a hybrid linear-nonlinear fault detection framework is presented based on SPCA. More specifically, linear PCA is first applied to extract PCs as linear features, and then in the PCA residual space (RS), KPCA is used to obtain nonlinear feature extraction. SPCA monitoring statistics are more effective in process fault detection than the monitoring statistics constructed using either PCA or KPCA alone. Secondly, a SPCA similarity factor method is developed for fault recognition, which fuses the linear features and nonlinear features, and therefore it outperforms both the PCA similarity factor method and the KPCA similarity factor method. In general, compared to PCA and KPCA, which only mine the linear or nonlinear features, the proposed SPCA takes into account all the relevant statistical features, including both linear PCs and nonlinear kernel PCs (KPCs).

The rest of this paper is organized as follows, In Section II, after a brief overview of PCA and KPCA, the proposed SPCA is detailed. Section III presents the fault detection procedure based on SPCA, while the fault identification framework based on the proposed SPCA similarity factor is provided in Section IV. Two case studies are used to validate the proposed SPCA approach for process fault detection and identification in Section V, and our conclusions are offered in Section VI.

II. SPCA for Hybrid Linear-Nonlinear Modeling

We begin by a brief overview of PCA and KPCA, followed by the detailed description of our proposed SPCA for hybrid linear-nonlinear modeling.

A. Overview of PCA and KPCA

PCA is a classical linear data dimension reduction technique, which transforms the original variables into new uncorrelated variables arranged by their variances. The new variables with large variances are viewed as the PCs, which represent the dominating data changes, while other variables with small variances are the projections of the original data onto the RS, which are usually thought to be the noise information but this is only true if the data only contains linear features. Thus PCA decomposes the original data space into two subspaces: PC subspace (PCS) and RS. Mathematically, given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ with n samples of m variables, the PCA decomposition is represented by

$$\boldsymbol{X} = \widehat{\boldsymbol{X}} + \widetilde{\boldsymbol{X}} = \sum_{i=1}^{k} \boldsymbol{t}_i \boldsymbol{p}_i^{\mathrm{T}} + \widetilde{\boldsymbol{X}}, \tag{1}$$

where $\boldsymbol{t}_i \in \mathbb{R}^n$ is the ith score vector or PC, $\boldsymbol{p}_i \in \mathbb{R}^m$ is the corresponding loading vector and k is the number of retained PCs, while $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times m}$ is the matrix reconstructed by the PCs and $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{n\times m}$ is the residual matrix.

In kernel PCA, an nonlinear function $\Phi : \mathbb{R}^m \rightarrow \mathcal{F}$ is implicitly assumed that maps the data in the original space onto a new high-dimensional feature space where the data become linearly correlated. Then in the feature space $\mathcal{F}$, linear PCA is applied. As the nonlinear mapping $\Phi(.)$ is unknown, kernel function is used to help completing the nonlinear transformation. Specifically, by using kernel function $\ker(\cdot,\cdot)$, the inner product of two feature data $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$ in the feature space can be calculated in the original data space as

$$\ker(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi^{\mathrm{T}}(\boldsymbol{x}_i)\Phi(\boldsymbol{x}_j), \tag{2}$$

for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^m$, without having to perform the nonlinear mapping $\Phi(\cdot)$ explicitly. A commonly used kernel function is the Gaussian kernel $\ker(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / c)$, where $c > 0$ is known as the kernel width.

For the processes operating in linear zones around some operating points, PCA offers an efficient and robust monitoring method, but it cannot deal with nonlinear processes. By contrast, KPCA is capable of extracting nonlinear features. Since industrial processes are highly complex and the performance of KPCA is often restricted by the chosen kernel parameters, it is often difficult to guarantee that KPCA can precisely capture the process characteristics. Moreover, many industrial processes exhibit both linear and nonlinear characteristics. Combining linear PCA and nonlinear KPCA modeling offers a new and viable process monitoring alternative.

B. Hybrid modeling by combining PCA and KPCA

The proposed hybrid linear-nonlinear SPCA method consists of two modeling steps, as depicted in Fig. 1. In the first step, PCA is executed to extract linear features. The second

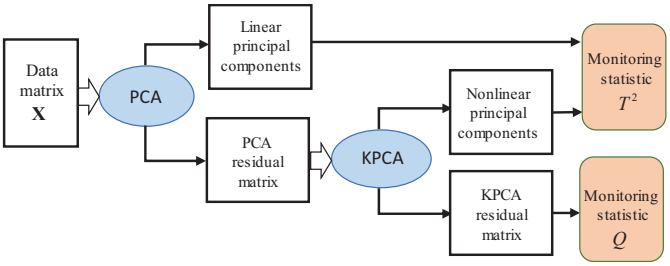


Fig. 1. The schematic of SPCA statistical modeling.

step involves KPCA decomposition, which extracts nonlinear features in the RS of the PCA. We now detail this SPCA.

For the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ which is assumed to be mean centered and variance scaled, linear PCA decomposition of Eq. (1) is rewritten as

$$\boldsymbol{X} = \widehat{\boldsymbol{X}} + \widetilde{\boldsymbol{X}} = \sum_{i=1}^{k_L} \boldsymbol{t}_{L_i}\boldsymbol{p}_{L_i}^{\mathrm{T}} + \widetilde{\boldsymbol{X}}, \tag{3}$$

where $\boldsymbol{t}_{L_i}$ is the ith linear score vector, $\boldsymbol{p}_{L_i}$ is the corresponding loading vector, and k_L is the number of PCs retained in the PCA model. The loading vector $\boldsymbol{p}_{L_i}$ can be obtained by the eigen-decomposition of the covariance matrix as

$$\frac{1}{n-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{p}_{L_i} = \lambda_{L_i}\boldsymbol{p}_{L_i}, \tag{4}$$

where λ_{L_i} denotes the ith eigenvalue of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}/(n-1)$.

Given a testing vector $\boldsymbol{x}_t \in \mathbb{R}^m$, its ith score is given by

$$t_{L_i} = \boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{p}_{L_i}. \tag{5}$$

The first k_L scores of $\boldsymbol{x}_t$, denoted by $\begin{bmatrix} t_{L_1} & t_{L_2} \cdots t_{L_{k_L}} \end{bmatrix}^{\mathrm{T}}$, are available as the linear features of the testing vector. Furthermore, the residual vector of $\boldsymbol{x}_t$ is readily given by

$$\widetilde{\boldsymbol{x}}_t = \boldsymbol{x}_t - \sum_{i=1}^{k_L} t_{L_i}\boldsymbol{p}_{L_i}. \tag{6}$$

In the second step, KPCA is applied to the PCA residual matrix $\widetilde{\boldsymbol{X}}$. The KPCA transformation of $\widetilde{\boldsymbol{X}}$ in $\mathcal{F}$, denoted by $\Phi(\widetilde{\boldsymbol{X}}) \in \mathbb{R}^n \times \mathcal{F}$, can be decomposed as

$$\Phi(\widetilde{\boldsymbol{X}}) = \sum_{i=1}^{k_N} \boldsymbol{t}_{N_i}\left(\boldsymbol{p}_{N_i}\right)^{\mathrm{T}} + \boldsymbol{E}, \tag{7}$$

where $\boldsymbol{t}_{N_i} \in \mathbb{R}^n$ is the ith nonlinear score vector or feature, $\boldsymbol{p}_{N_i} \in \mathcal{F}$ is the corresponding loading vector in KPCA decomposition, and k_N is the number of kernel PCs retained in the model, while $\boldsymbol{E} \in \mathbb{R}^n \times \mathcal{F}$ is the KPCA residual matrix. To obtain the KPCA score and loading vectors, the eigenvalue decomposition of the covariance matrix is formulated as

$$\frac{1}{n-1}\Phi^{\mathrm{T}}(\widetilde{\boldsymbol{X}})\Phi(\widetilde{\boldsymbol{X}})\boldsymbol{p}_{N_i} = \lambda_{N_i}\boldsymbol{p}_{N_i}, \tag{8}$$

with λ_{N_i} denoting the ith eigenvalue of $\frac{1}{n-1}\Phi^{\mathrm{T}}(\widetilde{\boldsymbol{X}})\Phi(\widetilde{\boldsymbol{X}})$. Denote $\widetilde{\boldsymbol{X}} = \begin{bmatrix} \widetilde{\boldsymbol{x}}_1 & \widetilde{\boldsymbol{x}}_2 \cdots \widetilde{\boldsymbol{x}}_n \end{bmatrix}^{\mathrm{T}}$. Then there exists $\boldsymbol{\alpha}_i = \begin{bmatrix} \alpha_{i,1} & \alpha_{i,2} \cdots \alpha_{i,n} \end{bmatrix}^{\mathrm{T}}$ such that [30], [31]

$$\boldsymbol{p}_{N_i} = \sum_{j=1}^{n} \alpha_{i,j}\Phi(\widetilde{\boldsymbol{x}}_j) = \Phi^{\mathrm{T}}(\widetilde{\boldsymbol{X}})\boldsymbol{\alpha}_i. \tag{9}$$

Combining Eqs. (8) and (9) with Eq. (2), we have [30], [31]

$$(n-1)\lambda_{N_i}\boldsymbol{\alpha}_i = \boldsymbol{K}\boldsymbol{\alpha}_i, \tag{10}$$

where $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is the kernel matrix with its ith-row and jth-column element given by $[\boldsymbol{K}]_{i,j} = \ker\left(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j\right) = \Phi^{\mathrm{T}}(\widetilde{\boldsymbol{x}}_i)\Phi(\widetilde{\boldsymbol{x}}_j)$, and $\boldsymbol{K}$ has been been mean centered [31]. It can be seen that λ_{N_i} and $\boldsymbol{\alpha}_i$ are the ith eigenvalue and eigenvector of $\boldsymbol{K}$, and they can be explicitly computed [30].

For the test residual vector $\widetilde{\boldsymbol{x}}_t$, its ith KPCA score is extracted by projecting $\Phi(\widetilde{\boldsymbol{x}}_t)$ onto $\mathbf{p}_{N_i}$ as

$$t_{N_i} = \Phi^{\mathrm{T}}(\widetilde{\boldsymbol{x}}_t)\boldsymbol{p}_{N_i} = \sum_{j=1}^{n} \alpha_{i,j}\Phi^{\mathrm{T}}(\widetilde{\boldsymbol{x}}_j)\Phi(\widetilde{\boldsymbol{x}}_t) = \boldsymbol{k}_t^{\mathrm{T}}\boldsymbol{\alpha}_i, \tag{11}$$

where $\boldsymbol{k}_t \in \mathbb{R}^n$ is the test kernel vector whose jth element is $[\boldsymbol{k}_t]_j = \ker\left(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{x}}_t\right)$. Detailed explanation, discussion and implementation of the KPCA can readily be found in the literature, e.g., [30], [31], [45].

In the literature, there exist some methods for selecting the number of retained PCs in PCA or KPCA model, including the average eigenvalue and cumulative percent variance methods [31], [46]. We adopt the average eigenvalue approach to determine k_L and k_N, owing to its simplicity and robustness. This method retains the PCs whose eigenvalues are larger than the average eigenvalue.

III. Fault Detection Based on SPCA

For the convenience of real-time fault detection, two monitoring statistics are constructed as

$$T^2 = \boldsymbol{t}_{\mathrm{SPCA}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{t}_{\mathrm{SPCA}}, \tag{12}$$

$$Q = \sum_{j=1}^{n}\left(t_{N_j}\right)^2 - \sum_{j=1}^{k_N}\left(t_{N_j}\right)^2, \tag{13}$$

where $\boldsymbol{t}_{\mathrm{SPCA}}^{\mathrm{T}} = \begin{bmatrix} t_{L_1} & t_{L_2} \cdots t_{L_{k_L}} & t_{N_1} & t_{N_2} \cdots t_{N_{k_N}} \end{bmatrix}$ contains the monitored linear and nonlinear PCs, and $\boldsymbol{\Gamma}$ is the covariance matrix of these components computed under the normal operating condition. The T^2 and Q statistics are standard monitoring statistics widely adopted in fault detection application. For example, the two corresponding statistics of the PCA method are constructed using $\boldsymbol{t}_{\mathrm{PCA}}^{\mathrm{T}} = \begin{bmatrix} t_{L_1} & t_{L_2} \cdots t_{L_{k_L}} \end{bmatrix}$, while the two related statistics of the KPCA method are computed with $\boldsymbol{t}_{\mathrm{KPCA}}^{\mathrm{T}} = \begin{bmatrix} t_{N_1} & t_{N_2} \cdots t_{N_{k_N}} \end{bmatrix}$.

To inspect if a fault occurs, the confidence limit is required for each statistic. In order to determine the confidence limit, the distribution for the monitored variables is required. Typically, existing PCA and KPCA methods assume some specified distribution, usually Gaussian, for the monitored variables. For example, the original KPCA method of [31] computes the confidence limits for T^2 and Q statistics based on the F distribution and weighted χ^2 distribution for these two statistics, respectively. Since industrial processes are highly complex, it is difficult to guarantee that process data conform to a specific distribution assumption, such as Gaussian. Kernel density estimation (KDE) [47], [48] is a non-parametric empirical density estimation technique, which does not need any distribution assumption. Hence, data-driven KDE based method has become popular recently for confidence limit determination [14], [16], [36], [39], [49]. Therefore, we apply the KDE to determines the confidence limits for the PCA, KPCA and SPCA based monitoring statistics. Specifically, this KDE based method computes confidence limit based on the values of monitoring statistics under normal testing data. In particular, 95% confidence limit are obtained in our paper.

Similar to the PCA and KPCA based process monitoring procedures, the SPCA based process monitoring procedure

consists of two stages: offline modeling and online monitoring. During the offline modeling stage, normal operating data are acquired and divided into the training and validating datasets. The training dataset is used to construct a SPCA model and the validating dataset is applied to determine the confidence limits. During the online monitoring stage, new measured data is collected and its linear and nonlinear PCs are extracted. Then the two monitoring statistics are computed. If at least one of these two statistics exceeds its confidence limit, a fault alarming signal is given to operator. This SPCA based fault detection procedure is summarized in Fig. 2.

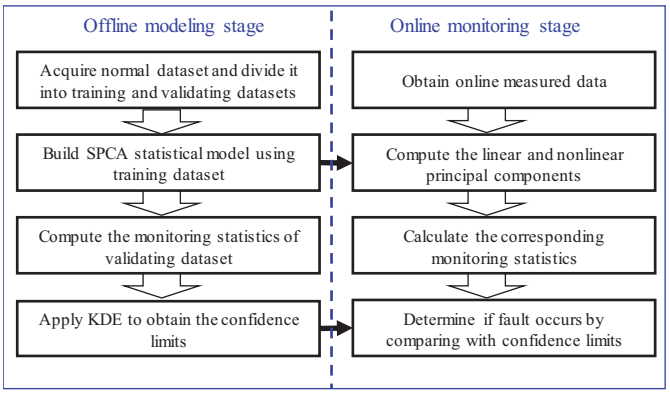


Fig. 2. Flowchart of SPCA based fault detection

IV. Fault Identification Using SPCA Similarity Factor

After a fault is detected, it is vital to diagnose the fault source so that the fault can be repaired quickly. Some methods were discussed for this challenging task in the literature [39], [50], [51]. An early solution is contribution plot [51]–[53], where the largest contribution value indicates the possible fault variable. Contribution plot is easy to implement but performs unsatisfactorily for complicated faults. Often many historical fault datasets are available in computer database and, therefore, pattern recognition strategy can be adopted to identify fault pattern. Hence, we assume that there are many fault pattern datasets available in the historical database.

A. PCA and KPCA similarity factors

PCA similarity factor [54]–[57] computes the similarity factor between the test fault dataset and each fault pattern dataset available in the historical database, and the fault pattern with the highest similarity factor is concluded as the recognition result. Early work on PCA similarity factor was carried out by Johannesmeyer *et al.* [54]. Later, Singhal *et al.* [55] presented a modified PCA similarity factor by considering the weighting of PC directions. This method has been applied in process monitoring and data classification in [56], [57]. To deal with nonlinear processes, Deng *et al.* [58] built a nonlinear similarity factor called KPCA similarity factor. We now briefly review both PCA and KPCA similarity factors.

For datasets $\boldsymbol{S} \in \mathbb{R}^{n\times m}$ and $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ from a process, which have been scaled by the mean and variance of normal operation data, PCA is performed on them to retain the first k_L PCs for each of them. The PCSs of $\boldsymbol{S}$ and $\boldsymbol{H}$ are defined by $\boldsymbol{L} = \left[\boldsymbol{l}_1\ \boldsymbol{l}_2 \cdots \boldsymbol{l}_{k_L}\right] \in \mathbb{R}^{m\times k_L}$ and $\boldsymbol{M} = \left[\boldsymbol{m}_1\ \boldsymbol{m}_2 \cdots \boldsymbol{m}_{k_L}\right] \in \mathbb{R}^{m\times k_L}$, respectively, which contain the first k_L loading vectors of $\boldsymbol{S}$ and $\boldsymbol{H}$. The loading vector is also the PC direction. Let $\theta_{i,j}$ be the angle between the ith PC direction $\boldsymbol{l}_i$ of the dataset $\boldsymbol{S}$ and the j-th PC direction $\boldsymbol{m}_j$ of the dataset $\boldsymbol{H}$. The cosine of $\theta_{i,j}$ is given by

$$\cos\theta_{i,j} = \frac{\boldsymbol{l}_i^{\mathrm{T}}\boldsymbol{m}_j}{\|\boldsymbol{l}_i\|\,\|\boldsymbol{m}_j\|}. \tag{14}$$

Then the weighted PCA similarity factor is defined as [55]

$$\begin{aligned} S_{\mathrm{PCA}}(\boldsymbol{S},\boldsymbol{H}) =& \frac{\sum_{i=1}^{k_L}\sum_{j=1}^{k_L}\lambda_{\boldsymbol{L}_i}\lambda_{\boldsymbol{M}_j}\left(\cos\theta_{i,j}\right)^2}{\sum_{i=1}^{k_L}\lambda_{\boldsymbol{L}_i}\lambda_{\boldsymbol{M}_i}} \\ =& \frac{\mathrm{tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{L}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\Lambda}_{\boldsymbol{M}}^2\boldsymbol{M}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{\Lambda}_{\boldsymbol{L}}\right)}{\sum_{i=1}^{k_L}\lambda_{\boldsymbol{L}_i}\lambda_{\boldsymbol{M}_i}}, \end{aligned} \tag{15}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace operation, $\boldsymbol{\Lambda}_{\boldsymbol{M}}^2 = \boldsymbol{\Lambda}_{\boldsymbol{M}}\boldsymbol{\Lambda}_{\boldsymbol{M}}$, $\boldsymbol{\Lambda}_{\boldsymbol{L}}$ and $\boldsymbol{\Lambda}_{\boldsymbol{M}}$ are the weighting matrices given by

$$\boldsymbol{\Lambda}_{\boldsymbol{L}} = \mathrm{diag}\left\{\sqrt{\lambda_{\boldsymbol{L}_1}}, \sqrt{\lambda_{\boldsymbol{L}_2}}, \cdots, \sqrt{\lambda_{\boldsymbol{L}_{k_L}}}\right\}, \tag{16}$$

$$\boldsymbol{\Lambda}_{\boldsymbol{M}} = \mathrm{diag}\left\{\sqrt{\lambda_{\boldsymbol{M}_1}}, \sqrt{\lambda_{\boldsymbol{M}_2}}, \cdots, \sqrt{\lambda_{\boldsymbol{M}_{k_L}}}\right\}, \tag{17}$$

with $\mathrm{diag}\{a_1, a_2 \cdots, a_p\}$ as the diagonal matrix having the diagonal elements of $a_1, a_2 \cdots, a_p$, while the eigenvalues $\lambda_{\boldsymbol{L}_i}$ and $\lambda_{\boldsymbol{M}_i}$ correspond to the ith PCs of $\boldsymbol{S}$ and $\boldsymbol{H}$, respectively, with the ordered eigenvalues satisfying $\lambda_{\boldsymbol{L}_1} > \lambda_{\boldsymbol{L}_2} > \cdots > \lambda_{\boldsymbol{L}_{k_L}}$ and $\lambda_{\boldsymbol{M}_1} > \lambda_{\boldsymbol{M}_2} > \cdots > \lambda_{\boldsymbol{M}_{k_L}}$. The number of PCs k_L used is a key parameter in PCA similarity factor computation. Let k_S and k_H be the numbers of PCs determined for $\boldsymbol{S}$ and $\boldsymbol{H}$, respectively, based on the average eigenvalue method. According to [54], [55], we can choose $k_L = \max\{k_S, k_H\}$.

The PCA similarity factor $S_{\mathrm{PCA}}(\boldsymbol{S},\boldsymbol{H})$ characterizes the similarity degree of $\boldsymbol{S}$ and $\boldsymbol{H}$ in the linear PC space. Large value indicates high similarity degree. In particular, the two datasets $\boldsymbol{S}$ and $\boldsymbol{H}$ are regarded to come from the same operation pattern if the value of $S_{\mathrm{PCA}}(\boldsymbol{S},\boldsymbol{H})$ is close to 1, while they belong to different operation patterns if $S_{\mathrm{PCA}}(\boldsymbol{S},\boldsymbol{H})$ is close to 0. However, it is clear that PCA similarity factor can only be applied to fault diagnosis of linear processes.

For the datasets $\boldsymbol{S}$ and $\boldsymbol{H}$, we can apply KPCA transformation to obtain their nonlinear PC subspaces respectively as

$$\boldsymbol{L} = \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{S})\boldsymbol{A}_{\boldsymbol{L}} \in \mathcal{F}\times\mathbb{R}^{k_N}, \tag{18}$$

$$\boldsymbol{M} = \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{H})\boldsymbol{A}_{\boldsymbol{M}} \in \mathcal{F}\times\mathbb{R}^{k_N}, \tag{19}$$

where $\boldsymbol{A}_{\boldsymbol{L}}, \boldsymbol{A}_{\boldsymbol{M}} \in \mathbb{R}^{n\times k_N}$ contain the first k_N eigenvectors of the KPCA decomposition on $\boldsymbol{S}$ and $\boldsymbol{H}$, respectively. The KPCA similarity factor [58] can then be written as

$$\begin{aligned} S_{\mathrm{KPCA}}(\boldsymbol{S},\boldsymbol{H}) =& \frac{1}{\sum_{i=1}^{k_N}\lambda_{\boldsymbol{L}_i}\lambda_{\boldsymbol{M}_i}}\mathrm{tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{L}}\boldsymbol{A}_{\boldsymbol{L}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{S})\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{H})\boldsymbol{A}_{\boldsymbol{M}} \\ & \times\boldsymbol{\Lambda}_{\boldsymbol{M}}^2\boldsymbol{A}_{\boldsymbol{M}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{H})\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{S})\boldsymbol{A}_{\boldsymbol{L}}\boldsymbol{\Lambda}_{\boldsymbol{L}}\Big), \end{aligned} \tag{20}$$

where $\boldsymbol{\Lambda}_{\boldsymbol{L}}, \boldsymbol{\Lambda}_{\boldsymbol{M}} \in \mathbb{R}^{k_N\times k_N}$ are the diagonal matrices whose diagonal elements are the square roots of the corresponding eigenvalues $\{\lambda_{\boldsymbol{L}_i}, 1 \le i \le k_N\}$ and $\{\lambda_{\boldsymbol{M}_i}, 1 \le i \le k_N\}$,

respectively, obtained from the KPCA decomposition of $\boldsymbol{S}$ and $\boldsymbol{H}$. Applying kernel trick, we can compute the kernel matrices $\boldsymbol{K_{HS}} = \boldsymbol{\Phi}(\boldsymbol{H})\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{S})$ and $\boldsymbol{K_{SH}} = \boldsymbol{\Phi}(\boldsymbol{S})\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{H})$. Then the KPCA similarity factor can be calculated explicitly as

$$S_{\mathrm{KPCA}}(\boldsymbol{S},\boldsymbol{H}) = \frac{\mathrm{tr}\left(\boldsymbol{\Lambda_L} \boldsymbol{A_L}^{\mathrm{T}} \boldsymbol{K_{SH}} \boldsymbol{A_M} \boldsymbol{\Lambda_M}^2 \boldsymbol{A_M}^{\mathrm{T}} \boldsymbol{K_{HS}} \boldsymbol{A_L} \boldsymbol{\Lambda_L}\right)}{\sum_{i=1}^{k_N} \lambda_{\boldsymbol{L}_i} \lambda_{\boldsymbol{M}_i}}. \tag{21}$$

Since the KPCA similarity factor characterizes the similarity degree of two datasets in the nonlinear PC space, it is capable of applying to fault identification of nonlinear processes.

B. Proposed SPCA similarity factor

The propose hybrid similarity factor, called SPCA similarity factor, integrates both linear and nonlinear features. When SPCA is applied for fault identification, two steps of PCA and KPCA decompositions are involved for $\boldsymbol{S}$ and $\boldsymbol{H}$.

At the PCA step, the datasets are decomposed into

$$\boldsymbol{S} = \widehat{\boldsymbol{S}} + \widetilde{\boldsymbol{S}}, \tag{22}$$

$$\boldsymbol{H} = \widehat{\boldsymbol{H}} + \widetilde{\boldsymbol{H}}, \tag{23}$$

where $\widetilde{\boldsymbol{S}}$ and $\widetilde{\boldsymbol{H}}$ are the resulting PCA residual matrices, and the PCA similarity factor $S_{\mathrm{PCA}}(\boldsymbol{S},\boldsymbol{H})$ is calculated.

The KPCA step then decomposes the PCA residual matrices $\widetilde{\boldsymbol{S}}$ and $\widetilde{\boldsymbol{H}}$ into

$$\boldsymbol{\Phi}\left(\widetilde{\boldsymbol{S}}\right) = \boldsymbol{\Phi}\left(\widehat{\widetilde{\boldsymbol{S}}}\right) + \boldsymbol{E}_{\widetilde{\boldsymbol{S}}}, \tag{24}$$

$$\boldsymbol{\Phi}\left(\widetilde{\boldsymbol{H}}\right) = \boldsymbol{\Phi}\left(\widehat{\widetilde{\boldsymbol{H}}}\right) + \boldsymbol{E}_{\widetilde{\boldsymbol{H}}}, \tag{25}$$

with $\boldsymbol{E}_{\widetilde{\boldsymbol{S}}}$ and $\boldsymbol{E}_{\widetilde{\boldsymbol{H}}}$ being the related KPCA residual matrices, and the KPCA similarity factor $S_{\mathrm{KPCA}}\left(\widetilde{\boldsymbol{S}},\widetilde{\boldsymbol{H}}\right)$ is calculated.

The new SPCA similarity factor is constructed as

$$S_{\mathrm{SPCA}}(\boldsymbol{S},\boldsymbol{H}) = S_{\mathrm{PCA}}(\boldsymbol{S},\boldsymbol{H}) \cdot S_{\mathrm{KPCA}}\left(\widetilde{\boldsymbol{S}},\widetilde{\boldsymbol{H}}\right). \tag{26}$$

In this new SPCA similarity factor, the PCA similarity factor component is capable of describing the similarity degree in linear PC space, while the KPCA similarity factor component is capable of characterizing the similarity degree in nonlinear PC space. Therefore, only when $\boldsymbol{S}$ and $\boldsymbol{H}$ are similar in both the linear and nonlinear spaces, which means that the two datasets are truly similar, this SPCA similarity factor is close to 1. If $\boldsymbol{S}$ and $\boldsymbol{H}$ are not similar, either in the linear feature space or in the nonlinear feature space or in the both spaces, then either the PCA similarity factor or the KPCA similarity factor or the both will be near to 0. This leads the SPCA similarity factor close to 0.

Once a fault is detected, this SPCA similarity factor can be used to identify the pattern of newly occurred fault. The unknown fault data is collected and the similarity factors with all the known fault pattern datasets are computed. The largest SPCA similarity factor indicates the possible fault pattern. If all the similarity factors between the occurring faulty data and all the known historical fault data are all close to zero, then the occurring fault pattern is unknown or unseen to the historical database. In this case, other means must be applied to identify the fault. For example, experience operator with the plan knowledge may have to infer from the operational condition the root cause of the occurring fault. Once this is done, that is, the fault is identified, the fault dataset can be added to the historical database.

V. Case Studies

Two case studies, a simulated nonlinear system [28] and the Tennessee Eastman (TE) process [59], are used to validate our proposed SPCA based approach and to compare its fault detection and identification performance with those of the PCA based and KPCA based methods. The KDE method is used to compute the 95% confidence limits of the two monitoring statistics for the PCA, KPCA and SPCA based fault detection methods. In all monitoring charts, the 95% confidence limit is plotted with dashed line and the monitoring statistic is plotted with solid line. Furthermore, all monitoring charts are normalized by their corresponding 95% confidence limits.

A. A simulated nonlinear system

The simulated nonlinear system, which is a modified version of the example given in [28], is described by

$$\begin{cases} x_1 = u_1 + e_1, \\ x_2 = u_2 + e_2, \\ x_3 = 2u_1 + 3u_2 + e_3, \\ x_4 = 5u_1 - 2u_2 + e_4, \\ x_5 = u_1^2 - 3u_2 + e_5, \\ x_6 = -u_1^3 + 3u_2^2 + e_6, \end{cases} \tag{27}$$

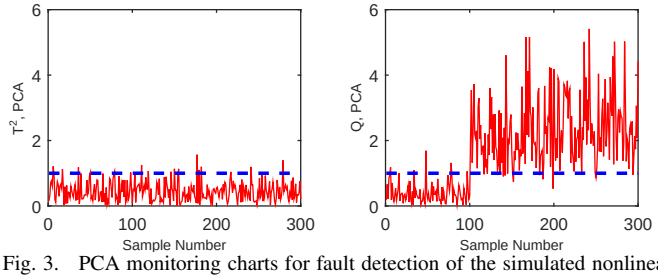


Fig. 3. PCA monitoring charts for fault detection of the simulated nonlinear process.

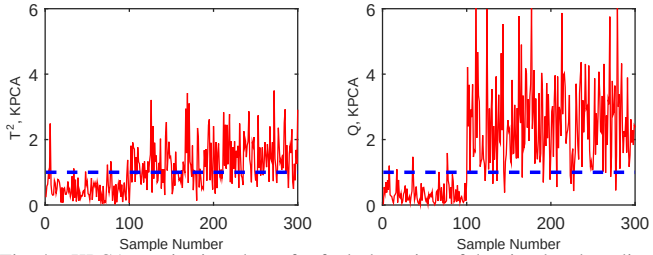


Fig. 4. KPCA monitoring charts for fault detection of the simulated nonlinear process.

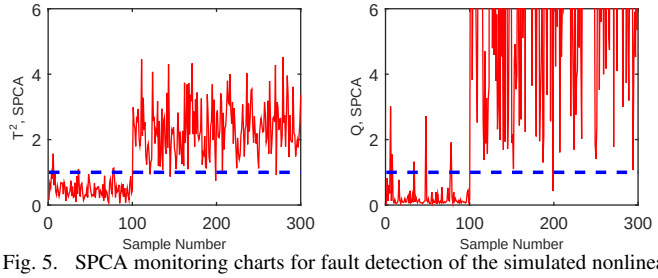


Fig. 5. SPCA monitoring charts for fault detection of the simulated nonlinear process.

where u_1 and u_2 are the independent source variables which follow the uniform distribution in $[0,\ 2]$, while e_1 to e_6 are the independent noise variables obeying the normal distribution with zero mean and variance 0.01. Normal operation dataset consisting of 600 samples is simulated based on the model (27). Among these data, 300 samples are used as the training dataset to build statistical model and the other 300 samples are applied as the validating dataset to determine the confidence limit. Both the KPCA and SPCA adopt the Gaussian kernel with the kernel width $c = 3000$. For fair comparison, the PCA, KPCA and SPCA methods all apply the average eigenvalue method to determine the linear PC and/or nonlinear PC numbers, resulting in $k_L = 2$ PCs for the PCA, $k_N = 4$ nonlinear PCs for the KPCA, and $k_L = 2$ PCs as well as $k_N = 4$ nonlinear PCs for the SPCA, respectively.

A fault with 300 samples is designed, where the variable x_2 has a step change of -0.5 after the 100th sample. The fault detection results obtained by the three methods are shown in Figs. 3 to 5, respectively. It can be seen that the PCA based T^2 statistic cannot detect the fault, and the KPCA based T^2 statistic does better but fluctuates around the confidence limit. By contrast, the SPCA based T^2 statistic clearly goes well above the confidence limit after the occurrence of the fault. The three Q charts all correctly detect the fault, but the SPCA based one yields the best indication of the occurring fault. Detection performance can be quantified by the fault detection rate which is the ratio of the alarming samples over all the fault samples. The fault detection rates of the three methods are summarized in Table I, which clearly confirm that the proposed SPCA method achieves the best fault detection result.

TABLE I
FAULT DETECTION RATES (%) OF THREE METHODS FOR THE SIMULATED NONLINEAR SYSTEM

	T^2	Q
PCA	6.0	94.0
KPCA	66.5	93.0
SPCA	97.5	99.5

.

We further examine the features extracted from the fault data by the three methods. Fig. 6 plots the linear PCs obtained by the PCA method, which are also the linear PCs for the SPCA. Observe from Fig. 6 that there exists no obvious change in these two linear PCs, which leads to the poor fault detection performance of the PCA based T^2 chart as it is constructed based on these two features. Fig. 7 shows the nonlinear PCs extracted by the KPCA, where it can be seen that the 3rd and 4th components exhibit slight changes after the 100th sample. This explains why the KPCA based T^2 chart is able

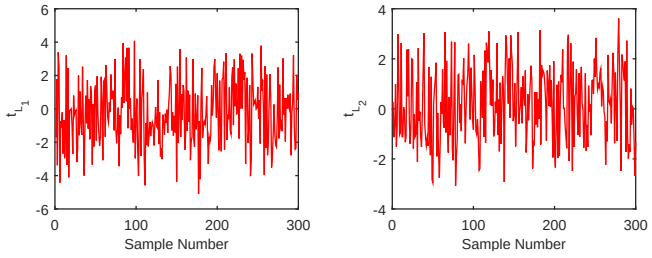


Fig. 6. Linear principal components extracted by the PCA method for the fault data of the simulated nonlinear process.

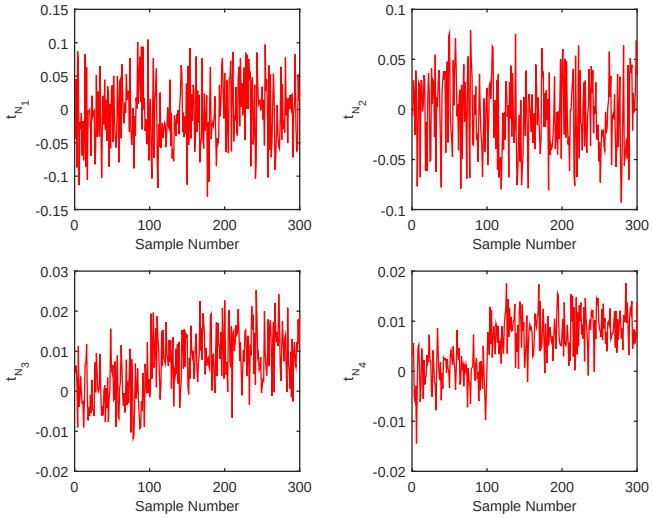


Fig. 7. Nonlinear principal components extracted by the KPCA method for the fault data of the simulated nonlinear process.

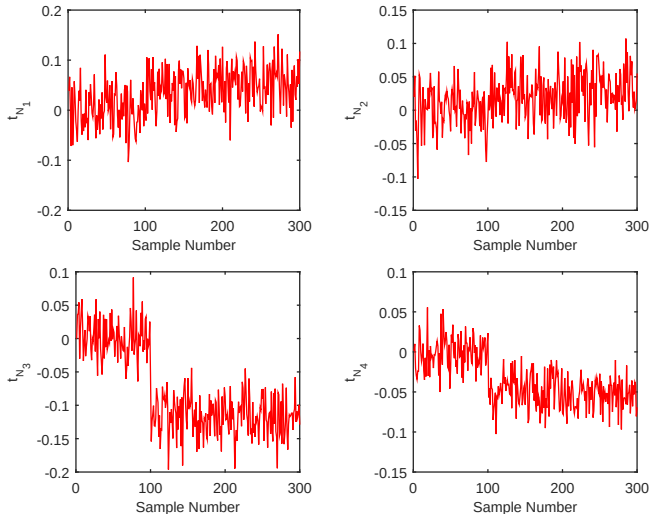


Fig. 8. Nonlinear principal components extracted by the SPCA method for the fault data of the simulated nonlinear process.

to sound alarm but it fluctuates around the confidence limit. Fig. 8 depicts the nonlinear features extracted by the SPCA. It can be seen from Fig. 8 that the 3rd and 4th components exhibit clear changes after the 100th sample. Thus the SPCA based T^2 chart is able to confidently detect the fault.

B. Tennessee Eastman process

The TE process, developed by Downs and Vogel [59], has become a benchmark process for validating process control and fault diagnosis techniques [60]–[62]. This process simulates a realistic chemical process operation and its flowchart is illustrated in Fig. 9. The simulation data with 52 variables can be downloaded from http://web.mit.edu/braatzgroup/links.html, which provides a normal operation case and 21 pre-programmed fault cases, denoted by IDV(1) to IDV(21). These 21 faults involve step changes and random variations in the process variables, slow drift in reaction kinetics, valve sticking and some unknown faults. Detailed fault descriptions can be found in [59], [60], [62]. The normal operation condition includes two datasets,

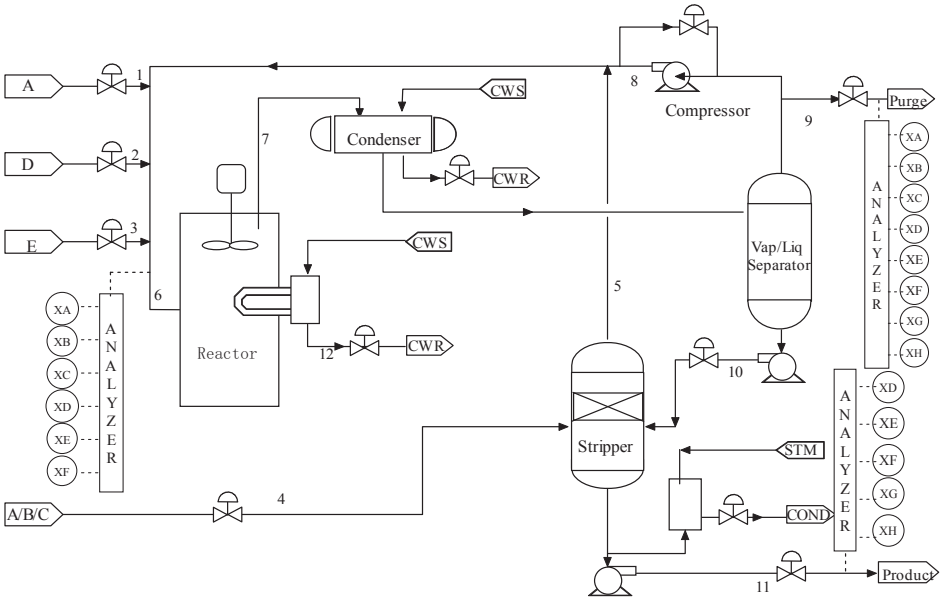


Fig. 9. The flowchart of TE process.

one having 500 samples and the another having 960 samples. Each fault condition also includes two fault datasets. One set contains 960 samples and the fault is introduced after the 160th sample, and the another has 480 samples with the fault occurring after the 20th sample.

We use the normal dataset with 500 samples as the training dataset to build model and apply the another 960 normal samples as the validating dataset to determine the confidence limits. The fault datasets IDV(1) to IDV(21) that each contains 960 samples are used for online fault detection and identification, while the fault datasets IDV(1) to IDV(21) that each contains 480 samples are assumed to form the historical known fault pattern datasets which are relabelled as FP(1) to FP(21). Both the KPCA and SPCA applies the same Gaussian kernel function with the width parameter $c = 500m$, where m is the number of variables. The number of PCs is determined by the average eigenvalue method.

1) Fault detection performance: The fault IDV (4), which is a step change in reactor cooling water inlet temperature, is first used to compare the fault detection performance of the

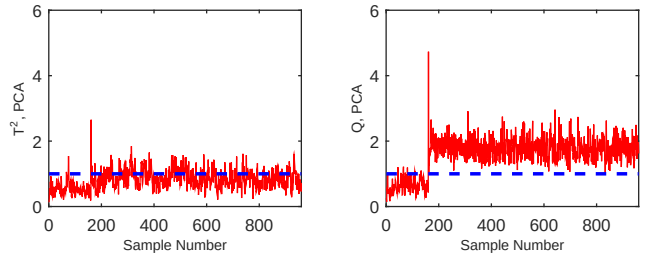


Fig. 10. PCA monitoring charts for the TE process fault IDV(4).

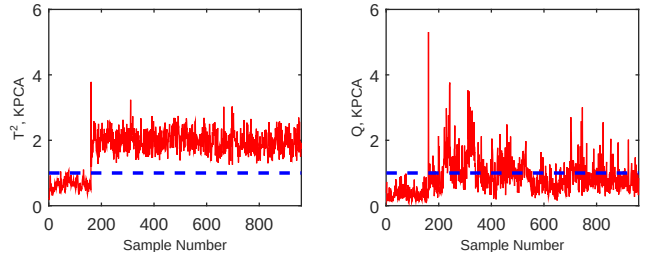


Fig. 11. KPCA monitoring charts for the TE process fault IDV(4).

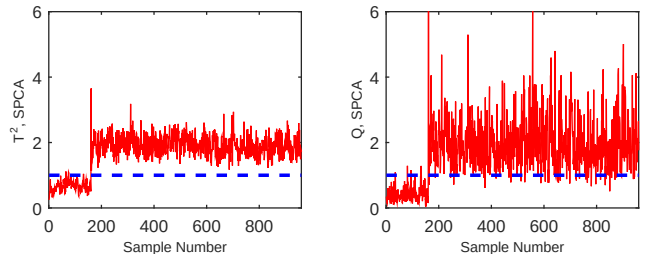


Fig. 12. SPCA monitoring charts for the TE process fault IDV(4).

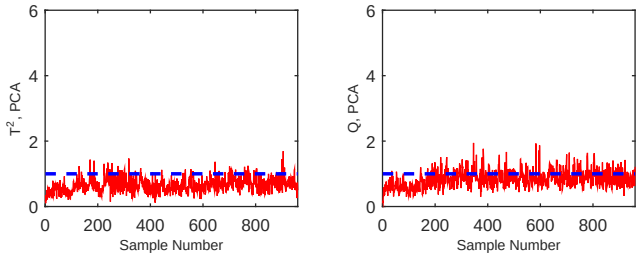


Fig. 13. PCA monitoring charts for the TE process fault IDV(19).

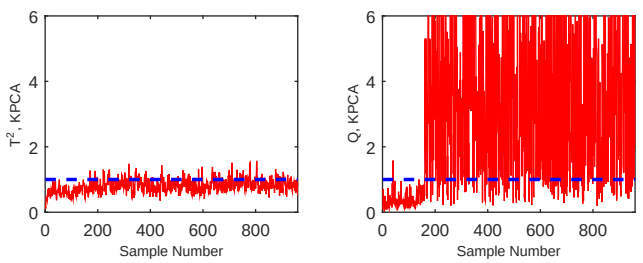


Fig. 14. KPCA monitoring charts for the TE process fault IDV(19).

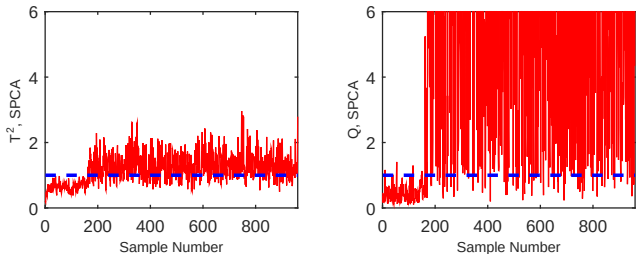


Fig. 15. SPCA monitoring charts for the TE process fault IDV(19).

three methods, and the results obtained are shown in Figs. 10 to 12. It can be seen that the PCA based T^2 statistic fails to detect the fault, while the KPCA based Q statistic fluctuates around its confidence limit. By contrast, both the SPCA based T^2 and Q monitoring charts clearly and confidently detect the occurring fault. By combining both linear and nonlinear features, the SPCA achieves the best detection performance for the fault IDV(4).

The monitoring results obtained by the three methods for the fault IDV(19), which is an unknown fault, are shown in Figs. 13 to 15. It can be seen from Fig. 13 that this fault is difficult to detect by the PCA method because its two monitoring statistics are both below the corresponding confident limits. The KPCA based T^2 monitoring chart also fails to detect this fault, as can be seen from Fig. 14. Again the SPCA method shows its advantages in fault detection. As confirmed by its T^2 and Q monitoring charts given in Fig. 15, the SPCA method confidently detects the fault IDV(19).

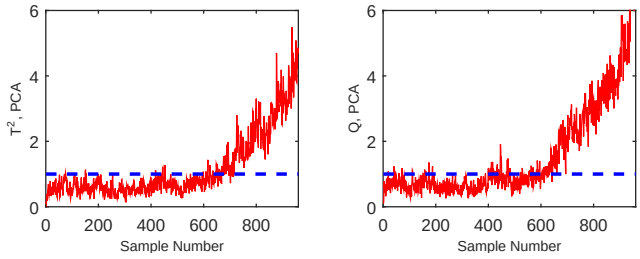


Fig. 16. PCA monitoring charts for the TE process fault IDV(21).

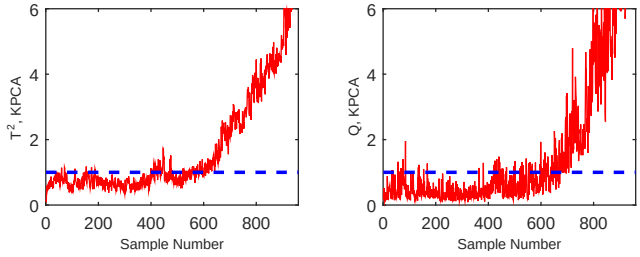


Fig. 17. KPCA monitoring charts for the TE process fault IDV(21).

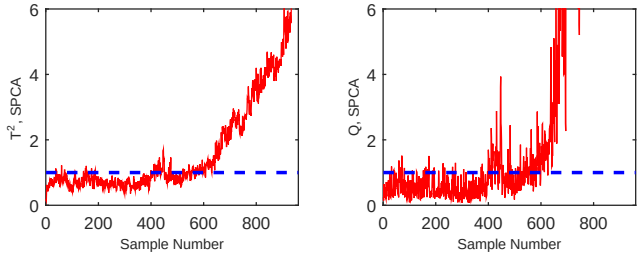


Fig. 18. SPCA monitoring charts for the TE process fault IDV(21).

The fault IDV(21), which involves a valve fault causing slow process degradation, is also used in detection performance comparison, and the results obtained by the three methods are shown in Figs. 16 to 18. It can be seen that all the three methods are able to detect this fault. To further analyze the detection performance, let us define the fault detection time. Specifically, let us consider that a fault is detected only if the six consecutive samples are all above the confidence limit, and the first sample of these six consecutive samples is then defined as the fault detection time. Noting that this fault starts at the 160th sample, the PCA based T^2 statistic detects the fault at the 680th sample, while its Q statistic detects the fault at the 445th sample but then goes below the confidence limit until the 600th sample. The KPCA based T^2 statistic detects this fault at the 427th sample, while its Q statistic indicates the fault until the 672th sample. By contrast, the SPCA based T^2 and Q statistics both detect the fault at the 415th sample, providing the earliest fault alarm.

TABLE II
FAULT DETECTION RATES (%) OF THREE METHODS FOR THE TE PROCESS.

Fault No.	PCA		KPCA		SPCA	
	T^2	Q	T^2	Q	T^2	Q
1	99.5	99.8	99.8	99.8	99.9	99.8
2	98.3	98.8	98.8	98.5	98.9	99.0
3	8.1	7.8	8.0	7.5	7.9	7.4
4	**28.9**	**100**	**100**	**37.3**	**100**	**94.0**
5	**30.6**	**31.3**	**28.6**	**99.5**	**30.5**	**99.9**
6	99.3	100	99.5	100	100	100
7	100	100	100	99.9	100	100
8	97.5	97.9	98.3	98.1	98.1	97.9
9	7.4	6.0	6.5	4.5	7.1	6.0
10	**49.8**	**53.9**	**54.9**	**86.9**	**89.5**	**82.1**
11	**47.4**	**73.3**	**79.3**	**51.8**	**79.8**	**64.8**
12	99.0	97.8	99.1	99.5	99.5	99.8
13	95.0	95.4	95.5	95.9	95.6	95.4
14	99.0	100	100	99.9	100	100
15	12.4	8.8	13.3	14	15.9	10.8
16	**32.5**	**48.3**	**37.0**	**90.0**	**93.0**	**75.8**
17	**81.6**	**93.9**	**96.1**	**90.5**	**96.5**	**92.3**
18	**89.5**	**91.4**	**91.3**	**89.4**	**91.1**	**90.4**
19	**8.4**	**29.1**	**19.1**	**80.8**	**75.0**	**90.4**
20	**47.0**	**57.3**	**68.4**	**72.3**	**73.5**	**82.6**
21	**39.4**	**51.1**	**54.5**	**44.3**	**56.5**	**59.0**

Table II lists the fault detection rates (FDRs) obtained by the three methods for all the 21 faults of the TE process. First thing to note from Table II is that for the faults IDV(3), IDV(9) and IDV(15), all the three methods perform poorly and cannot detect these faults. This is not surprising as it is well-known that these three faults are extremely difficult for data-driven monitoring methods owing to the reason that there exist no observable changes in the mean or variance of these three fault datasets [62]. It can also be seen that for the faults IDV(1), IDV(2), IDV(6), IDV(7), IDV(8), IDV(12), IDV(13) and IDV(14), all the three methods perform similarly well. However, for detecting the faults IDV(4), IDV(5), IDV(10), IDV(11), IDV(16), IDV(17), IDV(18), IDV(19), IDV(20) and IDV(21), the proposed SPCA method clearly outperforms the other two methods. Fig. 19 depicts the average fault detection rates over these 10 faults achieved by the three methods.

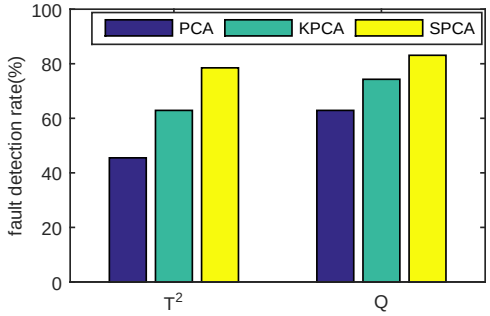


Fig. 19. Average fault detection rates achieved by the three methods averaging over the faults IDV(4), IDV(5), IDV(10), IDV(11), IDV(16), IDV(17), IDV(18), IDV(19), IDV(20) and IDV(21).

TABLE III
FALSE ALARMING RATES (FARS) AND FALSE FAULT DETECTION PROBABILITIES (FFDPS) OF THE THREE METHODS FOR THE TE PROCESS.

Method	PCA		KPCA		SPCA	
Statistic	T^2	Q	T^2	Q	T^2	Q
FAR (%)	3.13	4.58	2.77	2.65	3.04	4.61
FFDP	0		0		0	

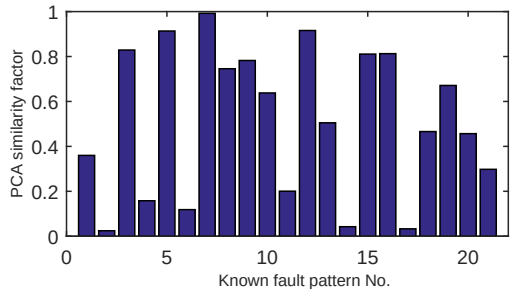
False alarming rate (FAR) is an important metric in classification applications. For our fault detection problem, we define the FAR as the percentage of the normal operating samples exceeding the confidence limit over all the normal operating samples. For the three methods, we compute their FARs on the $160 \times 21 = 3360$ normal samples from the 21 TE process simulation datasets, and the results are listed in Table III. As 95% confidence limit is used as the detection threshold, up to 5% of the samples may exceed the confidence limit statistically. From Table III, it can be seen that although the SPCA does not obtain the lowest FARs, its FAR values for the two monitoring statistics are both lower than 5%, which are consistent to the 95% confidence limit. It is worth emphasizing that the FAR is not the false fault detection probability (FFDP). In a real industrial application, an isolated sample exceeding the confidence limits is never taken to signify that a fault has occurred. Only when several successive samples, e.g., 6, consistently exceed the confidence limits can a fault is detected. By designing the appropriate confidence limits to ensure that the FAR is below 5%, the FFDP is practically zero. This allows us to determine the overall performance of a method by its FDR. In Table III, we also give the FFDPs of the three methods, by assuming that a fault is detected when 3 successive samples exceed the confidence limits.

For the three methods, their computation loads at the online monitoring stage are different. The PCA is the simplest, which costs the least computation time. By contrast, the KPCA, which involves the kernel vector calculation, is more complex and requires more computation time than the PCA. Compared with the KPCA, the SPCA further adds the linear feature extraction step and, therefore, is slightly more complex than the KPCA. However, as the computation of linear features is very simple, the computation time of the SPCA is only marginally more than that of the KPCA. To validate our analysis, we run the TE process online monitoring programmes 10 times in the same computer with the configuration of Intel Core^{TM} i7-5500U processor (2.4GHz) and 8G RAM memory. The average computation times per sample in the online monitoring required by the three methods are is listed Table IV. It can be seen that the PCA only needs 7.25×10^{-5} s for monitoring each sample and the KPCA's computation time increases to 9.35×10^{-3} s, while the SPCA spends 9.54×10^{-3} s for monitoring each sample, which is indeed only marginally higher than that of the KPCA method.

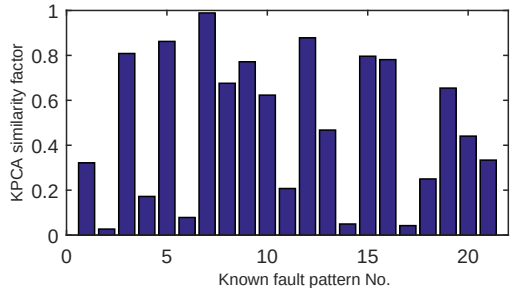
TABLE IV
COMPARISON OF AVERAGE COMPUTATION TIMES PER SAMPLE IN ONLINE MONITORING.

Method	PCA	KPCA	SPCA
Time (s)	7.25×10^{-5}	9.35×10^{-3}	9.54×10^{-3}

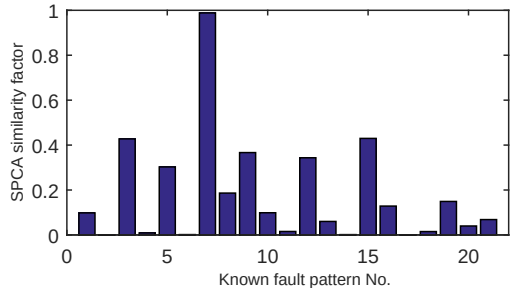
2) Fault identification performance: After a fault is detected, it is necessary to identify what kind of fault is occurring



(a) PCA similarity factor



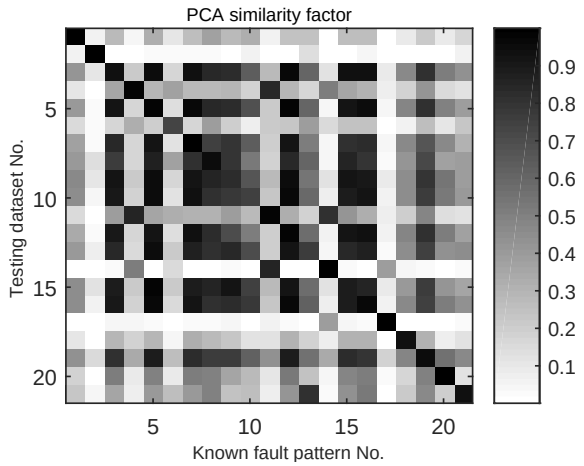
(b) KPCA similarity factor



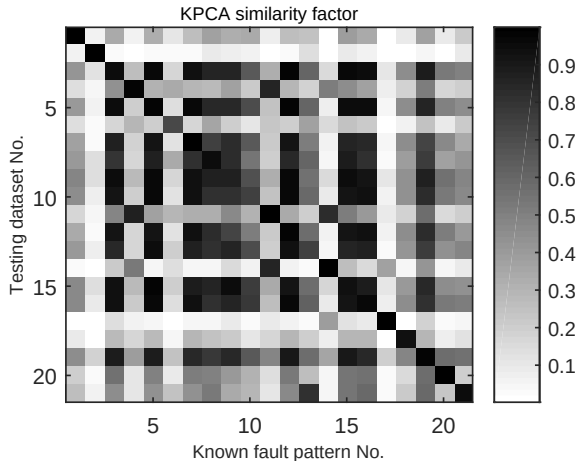
(c) SPCA similarity factor

Fig. 20. Similarity factors between the fault IDV(7) and the historical known fault pattern datasets of FP(1) to FP(21).

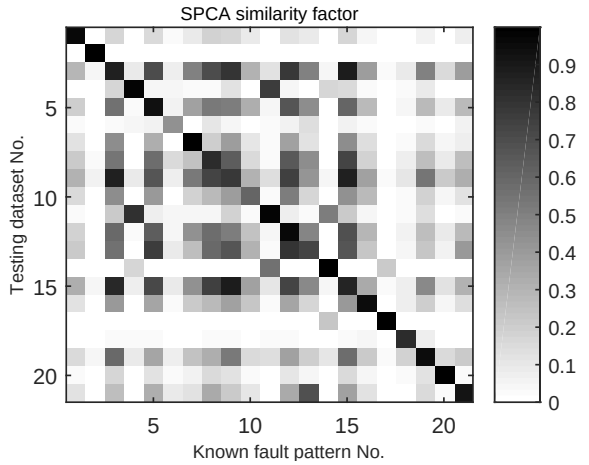
by computing the similarity factors between the fault dataset under investigation with the historical known fault pattern datasets of FP(1) to FP(21). The largest similarity factor then identifies the fault pattern. We first use the fault IDV(7) as an example. The PCA similarity factors, KPCA similarity factors and SPCA similarity factors between the fault IDV(7) and the known fault pattern datasets of FP(1) to FP(21) are calculated and the results are depicted in Fig. 20. It can be seen from Fig. 20 that for all the three methods, the 7th similarity factor, namely, the similarity factor between the fault IDV (7) and the known fault pattern dataset FP(7), attains the largest value of 0.99. Therefore, in theory, all the three methods correctly identify the fault pattern. However, The result of the PCA similarity factor is not very convincing. This is because the 5th and 12th PCA similarity factors are larger than 0.9, while the 3rd, 15th and 16th PCA similarity factors are bigger than 0.8. These large similarity factors are also close to 1 and, therefore, the PCA similarity factor may be easily influenced by the noise, potentially leading to a wrong diagnosis. The KPCA similarity factor is marginally better than the PCA similarity factor, and there are several other large KPCA similarity factor values close to 1, e.g., the 3rd, 5th and 12th KPCA similarity factors. By contrast, all the SPCA similarity factors other than



(a) PCA similarity factor



(b) KPCA similarity factor



(c) SPCA similarity factor

Fig. 21. Gray images of the similarity factors for all the 21 fault datasets of IDV(1) to IDV(21).

the 7th one are smaller than 0.45 and many of them are close to zero. Therefore, by fusing both linear and nonlinear features, the method of SPCA similarity factor is most effective in the recognition of the fault IDV(7).

We then analyze the fault recognition results for all the 21 fault testing datasets as shown in Fig. 21, where the PCA similarity factors, KPCA similarity factors and SPCA similarity factors between all the fault datasets of IDV(1) to IDV(21) and the historical known fault pattern datasets of FP(1) to FP(21) are depicted in gray images. In a gray image, the darkest black colour represents the largest similarity factor that is close to 1, while the lightest black colour, i.e., white colour, is for the smallest similarity factor that is zero or close to 0. From Fig. 21 (a), it can be seen that the gray image or the similarity 'matrix' is not very diagonal, indicating that it is difficult for the PCA based method to identify fault convincingly and correctly. By taking the largest similarity factor as the recognized fault pattern, the PCA based method mis-classifies 7 faults which are indicated in Table V. The clarity of the KPCA similarity factors shown in Fig. 21 (b) is slightly better, and the method makes 5 wrong recognitions which are also listed in Table V. Observe from Fig. 21 (c) that the SPCA based similarity 'matrix' is much close to diagonal and, therefore, the SPCA based method offers the best fault discriminant ability. The number of incorrect recognitions made by the SPCA based method is only 4, as indicated in Table V. As mentioned previously, the faults IDV(3), IDV(9) and IDV(15) are extremely difficult for a data-driven method. Thus, out of the other 18 fault cases, the SPCA based method only makes one error in fault identification. The fault identification rate (FIR), which is the percentage of the correctly identified fault cases over all the fault cases, is also shown in Table V for each method.

TABLE V
FAULT IDENTIFICATION RESULTS FOR THE TE PROCESS BY THE THREE SIMILARITY FACTOR BASED METHODS.

	FIR(%)	No. of wrong identification cases
PCA	66.7	3,5,9,10,13,15,16
KPCA	76.2	3,9,10,13,15
SPCA	81.0	3,9,13,15

VI. CONCLUSIONS

A novel hybrid linear-nonlinear statistical modeling approach, referred to as SPCA, has been proposed for nonlinear process monitoring and fault diagnosis. Our contribution has been twofold. Firstly, we have derived the SPCA based model which fuses both linear and nonlinear features for effective fault detection of nonlinear processes. Specifically, PCA firstly extracts the linear features and KPCA then mines the nonlinear features on the PCA residual subspace. The SPCA based monitoring statistics constructed by fusing both linear and nonlinear features offer more effective fault detection capability than either the PCA or KPCA based monitoring charts. Secondly, a SPCA based similarity factor has been developed for fault identification with the aid of historical fault database, which is more powerful for fault pattern diagnosis than either the PCA or KPCA based similarity factor method. Simulation results involving a simulated nonlinear system and the TE benchmark process have confirmed the superior performance of the proposed SPCA approach over the existing KPCA based approach, in terms of fault detection and identification.

As a note to the related topic, specifically, statistical modeling of time-varying industrial data, we point out that the existing researches of [63]–[65] are worth further investigating.

REFERENCES

[1] V. Agrawal, B. K. Panigrahi, and P. M. V. Subbarao, "Review of control and fault diagnosis methods applied to coal mills," *J. Process Control*, vol. 32, pp. 138–153, Aug. 2015.

[2] K. Serverson, P. Chaiwatanodom, and R. D. Braatz, "Perspective on process monitoring of industrial systems," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 931–939, 2015.

[3] S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems*. Springer-Verlag: London, 2014.

[4] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual Reviews in Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.

[5] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Industrial Electronics*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[6] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, Oct. 2012.

[7] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Industrial & Engineering Chemistry Research*, vol. 52, no. 10, pp. 3543–3562, Feb. 2013.

[8] C. Zhao and F. Gao, "Fault-relevant principal component analysis (FPCA) method for multivariate statistical modeling and process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 133, pp. 1–16, Apr. 2014.

[9] J. J. Hong, J. Zhang, and J. Morris, "Progressive multi-block modelling for enhanced fault isolation in batch processes," *J. Process Control*, vol. 24, no. 1, pp. 13–26, Jan. 2014.

[10] Q. Jiang, B. Huang, and X. Yan, "GMM and optimal principal components-based Bayesian method for multimode fault diagnosis," *Computers & Chemical Engineering*, vol. 84, pp. 338–349, Jan. 2016.

[11] D. Zhou, G. Li, and S. J. Qin, "Total projection to latent structures for process monitoring," *AIChE J.*, vol. 56, no. 1, pp. 168–178, Jan. 2010.

[12] Y. Zhang, W. Du, Y. Fan, and L. Zhang, "Process fault detection using directional kernel partial least squares," *Industrial & Engineering Chemistry Research*, vol. 54, no. 9, pp. 2509–2518, Feb. 2015.

[13] C. Tong, A. Palazoglu, and X. Yan, "Improved ICA for process monitoring based on ensemble learning and Bayesian inference," *Chemometrics and Intelligent Laboratory Systems*, vol. 135, pp. 141–149, Jul. 2014.

[14] X. Tian, X. Zhang, X. Deng, and S. Chen, "Multiway kernel independent component analysis based on feature samples for batch process monitoring," *Neurocomputing*, vol. 72, nos. 7-9, pp. 1584–1596, Mar. 2009.

[15] X. Tian, L. Cai, and S. Chen, "Noise-resistant joint diagonalization independent component analysis based process fault detection," *Neurocomputing*, vol. 149, pp. 652–666, Feb. 2015.

[16] P.-E. P. Odiowei and Y. Cao, "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations," *IEEE Trans. Industrial Informatics*, vol. 6, no. 1, pp. 36–45, Feb. 2010.

[17] B. Jiang, X. Zhu, D. Huang, J. A. Paulson, and R. D. Braatz, "A combined canonical variate analysis and Fisher discriminant analysis (CVA-FDA) approach for fault diagnosis," *Computers & Chemical Engineering*, vol. 77, pp. 1–9, Jun. 2015.

[18] C. Alippi, S. Ntalampiras, and M. Roveri, "A cognitive fault diagnosis system for distributed sensor networks," *IEEE Trans. Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1213–1226, Aug. 2013.

[19] H. Chen, P. Tiňo, A. Rodan, and X. Yao, "Learning in the model space for cognitive fault diagnosis," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 124–136, Jan. 2014.

[20] C. Alippi, M. Roveri, and F. Trovò, "A self-building and cluster-based cognitive fault diagnosis system for sensor networks," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1021–1032, Jun. 2014.

[21] I. Miletic, S. Quinn, M. Dudzic, V. Vaculik, and M. Champagne, "An industrial perspective on implementing on-line applications of multivariate statistics," *J. Process Control*, vol. 14, no. 8, pp. 821–836, Dec. 2004.

[22] C. Zhao and F. Gao, "Statistical modeling and online fault detection for multiphase batch processes with analysis of between-phase relative changes," *Chemometrics and Intelligent Laboratory Systems*, vol. 130, pp. 58–67, Jan. 2014.

[23] T. J. Rato and M. S. Reis, "Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR)," *Chemometrics and Intelligent Laboratory Systems*, vol. 125, pp. 101–108, Jun. 2013.

[24] X. Xu, L. Xie, and S. Wang, "Multimode process monitoring with PCA mixture model," *Computers & Electrical Engineering*, vol. 40, no. 7, pp. 2101–2112, Oct. 2014.

[25] Y. Yang, Y. Ma, B. Song, and H. Shi, "An aligned mixture probabilistic principal component analysis for fault detection of multimode chemical processes," *Chinese J. Chemical Engineering*, vol. 23, no. 8, pp. 1357–1363, Aug. 2015.

[26] C. Tong, A. Palazoglu, and X. Yan, "An adaptive multimode process monitoring strategy based on mode clustering and mode unfolding," *J. Process Control*, vol. 23, no. 10, pp. 1497–1507, Nov. 2013.

[27] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.

[28] D. Dong and T. McAvoy, "Nonlinear principal component analysis – based on principal curves and neural networks," *Computers & Chemical Engineering*, vol. 20, no. 1, pp. 65–78, Jan. 1996.

[29] Y. Guo, K. Li, Z. Yang, J. Deng, and D. M. Laverty, "A novel radial basis function neural network principal component analysis scheme for PMU-based wide-area power system monitoring," *Electric Power Systems Research*, vol. 127, pp. 197–205, Oct. 2015.

[30] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[31] J.-M. Lee, C. K. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, no. 1, pp. 223–234, Jan. 2004.

[32] S. W. Choi, C. K. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 1, pp. 55–67, Jan. 2005.

[33] Y. Zhang, S. Li, and Z. Hu, "Improved multi-scale kernel principal component analysis and its application for fault detection," *Chemical Engineering Research and Design*, vol. 90, no. 9, pp. 1271–1280, Sep. 2012.

[34] Z. Ge, C. Yang, and Z. Song, "Improved kernel PCA-based monitoring approach for nonlinear processes," *Chemical Engineering Science*, vol. 64, no. 9, pp. 2245–2255, May 2009.

[35] J. Fan, S. J. Qin, and Y. Wang, "Online monitoring of nonlinear multivariate industrial processes using filtering KICA-PCA," *Control Engineering Practice*, vol. 22, pp. 205–216, Jan. 2014.

[36] X. Deng, X. Tian, and S. Chen, "Modified kernel principal component analysis based on local structure analysis and its application to nonlinear process fault diagnosis," *Chemometrics and Intelligent Laboratory Systems*, vol. 127, pp. 195–209, Aug. 2013.

[37] M. Yao and H. Wang, "On-line monitoring of batch processes using generalized additive kernel principal component analysis," *J. Process Control*, vol. 28, pp. 56–72, Apr. 2015.

[38] M. Navi, M. R. Davoodi, and N. Meskin, "Sensor fault detection and isolation of an industrial gas turbine using partial kernel PCA," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 1389–1396, 2015.

[39] L. Cai, X. Tian, and S. Chen, "Monitoring nonlinear and non-Gaussian processes using Gaussian mixture model based weighted kernel independent component analysis," *IEEE Trans. Neural Networks and Learning Systems*, to appear, 2016.

[40] G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.

[41] K.-Y. Chen, "Combining linear and nonlinear model in forecasting tourism demand," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10368–10376, Aug. 2011.

[42] T. Xiong, C. Li, Y. Bao, Z. Hu, and L. Zhang, "A combination method for interval forecasting of agricultural commodity futures prices," *Knowledge-Based Systems*, vol. 77, pp. 92–102, Mar. 2015.

[43] C. Alippi, S. Ntalampiras and M. Roveri, "Model ensemble for an effective on-line reconstruction of missing data in sensor networks," in *Proc. IJCNN 2013* (Dallas, TX), Aug. 4-9, 2013, pp. 1–6.

[44] Y. Zhang, T. Chai, and D. Wang, "An alternating identification algorithm for a class of nonlinear dynamical systems," *IEEE Trans. Neural Networks and Learning Systems*, to appear, 2016.

[45] R. T. Samuel and Y. Cao, "Nonlinear process fault detection and identification using kernel PCA and kernel density estimation," *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 165–174, Jun. 2016.

[46] S. Valle, W. Li, and S. J. Qin, "Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods," *Industrial & Engineering Chemistry Research*, vol. 38, no. 11, pp. 4389–4401, Sep. 1999.

[47] X. Hong, S. Chen, and C. J. Harris, "A forward-constrained regression algorithm for sparse kernel density estimation," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 193–198, Jan. 2008.

[48] X. Hong, J. Gao, S. Chen, and T. Zia, "Sparse density estimation on the multinomial manifold," *IEEE Trans. Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2972–2927, Nov. 2015.

[49] J.-M. Lee, C. K. Yoo, and I.-B. Lee, "Statistical process monitoring with independent component analysis," *J. Process Control*, vol. 14, no. 5, pp. 467–485, Aug. 2004.

[50] S. Ntalampiras, "Fault identification in distributed sensor networks based on universal probabilistic modelling," *IEEE Trans. Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 1939–1949, Sep. 2015.

[51] J. Liu and D.-S. Chen, "Fault isolation using modified contribution plots," *Computers & Chemical Engineering*, vol. 61, pp. 9–19, Feb. 2014.

[52] J. A. Westerhuis, S. P. Gurden, and A. K. Smilde, "Generalized contribution plots in multivariate statistical process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 95–114, May 2000.

[53] K. Peng, K. Zhang, G. Li, and D. Zhou, 'Contribution rate plot for nonlinear quality-related fault diagnosis with application to the hot strip mill process," *Control Engineering Practice*, vol. 21, vol. 4, pp. 360–369, Apr. 2013.

[54] M. C. Johannesmeyer, A. Singhal, and D. E. Seborg, "Pattern matching in historical data," *AIChE J.* vol. 48, no. 9, pp. 2022–2038, Sep. 2002.

[55] A. Singhal and D. Seborg, "Pattern matching in historical batch data using PCA," *IEEE Control Systems*, vol. 22, no. 5, pp. 53–63, Oct. 2002.

[56] X. Deng and X. Tian, "Multimode process fault detection using local neighborhood similarity analysis," *Chinese J. Chemical Engineering*, vol. 22, nos. 11-12, pp. 1260–1267, Nov. 2014.

[57] M. Ottavian, P. Facco, L. Fasolato, and M. Barolo, "Multispectral data classification using similarity factors," *Chemometrics and Intelligent Laboratory Systems*, vol. 118, pp. 13–23, Aug. 2012.

[58] X. Deng and X. Tian, "Nonlinear process fault pattern recognition using statistics kernel PCA similarity factor," *Neurocomputing*, vol. 121, pp. 298–308, Dec. 2013.

[59] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, Mar. 1993.

[60] C. K. Lau, K. Ghosh, M. A. Hussain, and C. R. Che Hassan, "Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS," *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 1–14, Jan. 2013.

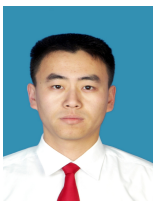
[61] Y. A. W. Shardt, B. Huang, and S. X. Ding, "Minimal required excitation for closed-loop identification: Some implications for data-driven, system identification," *J. Process Control*, vol. 27, pp. 22–35, Mar. 2015.

[62] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag: London, 2001.

[63] C. Alippi, G. Boracchi, and M. Roveri, "Just-in-time classifiers for recurrent concepts," *IEEE Trans. Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 620–634, Apr. 2013.

[64] I. B. Khediri, M. Limam, and C. Weihs, "Variable window adaptive kernel principal component analysis for nonlinear nonstationary process monitoring," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 437–446, Oct. 2011.

[65] W. Shao, X. Tian, P. Wang, X. Deng, and S. Chen, "Online soft sensor design using local partial least squares models with adaptive process state partition," *Chemometrics and Intelligent Laboratory Systems*, vol. 144, pp. 108–121, May 2015.



Xiaogang Deng received the B.Eng. and Ph.D. degrees from China University of Petroleum, Dongying, China, in 2002 and 2008, respectively.
He is currently an associate professor with College of Information and Control Engineering, China University of Petroleum. From October 2015 to October 2016, he was a visiting scholar with the Department of Electronics and Computer Sciences, the University of Southampton, Southampton, U.K. Dr Deng's research interests include industrial process modeling and simulation, data-driven fault detection and diagnosis, and control performance monitoring.

Xuemin Tian received his Bachelor of Engineering from Huadong Petroleum Institute, Dongying, China, in January 1982, and his M.S. degree from Beijing University of Petroleum, Beijing, China, in June 1994. From September 2001 to June 2002, he served as visiting professor at central of process control, University of California in Santa Barbara.
He is a professor of Process Control at China University of Petroleum (Hua Dong). Professor Tian's research interests are in modeling, advanced process control and optimization for petrol-chemical processes as well as fault detection and diagnosis, and process monitoring.

Sheng Chen (M'90-SM'97-F'08) received his BEng degree from Huadong Petroleum Institute, Dongying, China, in January 1982, and his PhD degree from the City University, London, in September 1986, both in control engineering. In 2005, he was awarded the higher doctoral degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK.
From 1986 to 1999, He held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with the Department of Electronics and Computer Science, the University of Southampton, UK, where he holds the post of Professor in Intelligent Systems and Signal Processing. Dr Chen's research interests include adaptive signal processing, wireless communications, modelling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods and optimisation. He has published over 600 research papers.
Dr. Chen is a Fellow of the United Kingdom Royal Academy of Engineering, a Fellow of IET, a Distinguished Adjunct Professor at King Abdulaziz University, Jeddah, Saudi Arabia, and an ISI highly cited researcher in engineering (March 2004).

Chris J. Harris received his BSc and MA degrees from the University of Leicester and the University of Oxford in UK, respectively, and his PhD degree from the University of Southampton, UK, in 1972. He was awarded the higher doctoral degree, the Doctor of Sciences (DSc), by the University of Southampton in 2001.
He is Emeritus Reseach Professor at the University of Southampton, having previously held senior academic appointments at Imperial College, Oxford and Manchester Universities, as well as Deputy Chief Scientist for the UK Government.
Professor Harris was awarded the IEE senior Achievement Medal for Data Fusion research and the IEE Faraday Medal for distinguished international research in Machine Learning. He was elected to a fellow of the UK Royal Academy of Engineering in 1996. He is the co-author of over 450 scientific research papers during a 45 year research career.