A multi-feature extraction technique based on principal component analysis for nonlinear dynamic process monitoring

Guo, Lingling; Wu, Ping; Lou, Siwei; Gao, Jinfeng; Liu, Yichao

DOI
10.1016/j.jprocont.2019.11.010

Publication date
2020

Document Version
Final published version

Published in
Journal of Process Control

Contents lists available at ScienceDirect

Journal of Process Control

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

A multi-feature extraction technique based on principal component analysis for nonlinear dynamic process monitoring

Lingling Guo[a], Ping Wu[a,*], Siwei Lou[a], Jinfeng Gao[a], Yichao Liu[b]

[a] Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, PR China
[b] Delft Center for Systems and Control, Delft University of Technology, Delft 2628CD, the Netherlands

ARTICLE INFO

Article history:
Received 23 July 2019
Revised 6 November 2019
Accepted 24 November 2019

Keywords:
Principal component analysis
Multi-feature extraction
Nonlinear dynamic process
Process monitoring

ABSTRACT

Principal component analysis (PCA) and its modified methods have been widely applied in industrial process monitoring. In practice, industrial processes are with disparate characteristics, the process monitoring system should consider as many process characteristics as possible, such as dynamic and nonlinear characteristics. In this paper, a multi-feature extraction technique based on PCA is proposed for nonlinear dynamic process monitoring. The proposed method integrates dynamic inner PCA (DiPCA), PCA and kernel PCA (KPCA) methods through a serial structure to extract the dynamic, linear and nonlinear features among the process data. Along with the proposed method, the original data space is decomposed into several orthogonal subspaces, in which abnormal variations of different features can be monitored. For real-time process monitoring, a combined Hotelling's T^2 statistic based on the extracted multi-feature and a squared prediction error (SPE or Q) statistic are established. Case studies on a numerical example and the Tennessee Eastman process are carried out to demonstrate the superior process monitoring performance of the proposed method compared with other relevant methods.

1. Introduction

Process monitoring has become a hot topic research area in both academic and industry fields in the past decades [1,2]. Among various monitoring methods, data-driven methods have attracted considerable attention, especially multivariate statistical process monitoring (MSPM) methods [3–5]. Some representative MSPM methods including principal component analysis (PCA) [6,7], canonical variate analysis (CVA) [8,9] and partial least squares (PLS) [10,11], etc. have been widely developed for the process monitoring purpose. As the most ubiquitous MSPM method, PCA has been thoroughly studied. However, for nonlinear dynamic processes, the application of traditional PCA tends to have an inefficient and unreliable process monitoring performance.

To cope with the dynamic characteristic of industrial processes, Ku et al. [12] directly extended the ordinary PCA to dynamic PCA by performing PCA procedure on the augmented data matrix. However, the derived dynamic relationships are implicit and have poor interpretability. To improve the interpretability, Li et al. [13] suggested a dynamic latent variable (DLV) method, in which an inner vector auto-regressive (VAR) model is built to explore the dynamic relationships and a compact outer model is built to represent the data structure. However, the inner VAR model is not consistent with the outer model. Inspired by the idea of DLV, Dong and Qin [14] developed a dynamic-inner PCA (DiPCA) method to extract dynamically correlated latent components, in which the current latent components are predictable from the past correlated latent components.

In addition, several nonlinear PCA methods have been developed to address the nonlinear problem. Compared with the neural network-based [15,16], lwpr-based [17] and randomized nonlinear PCA methods [18], the kernel-based PCA (KPCA) methods are simpler and more elegant. In KPCA, the original input space is mapped into a high, possibly infinite-dimensional feature space through nonlinear mapping, in which the data can be treated with traditional PCA method. Since the nonlinear mapping is implicit, the nonlinear features are usually extracted with the help of the kernel function computed in original space [19]. KPCA has been widely adopted in nonlinear processes [20–24].

For complex industrial processes, the monitoring system should extract as many features as possible. Several studies have pointed out that the combination strategy may be a better choice for process monitoring [25,26]. Chen et al. [27] developed a multi-layer data processing method based on PCA to extract several signal features for real-time incipient fault detection and diagnosis. Deng et al. [28] devised a layerwise feature extraction strategy to

[*] Corresponding author.
E-mail address: pingwu@zstu.edu.cn (P. Wu).

https://doi.org/10.1016/j.jprocont.2019.11.010
0959-1524/© 2019 Elsevier Ltd. All rights reserved.

capture multiple linear and nonlinear features hierarchically for nonlinear process monitoring. In addition, a new hybrid approach referred to as serial PCA (SPCA) was developed through integrating PCA and KPCA using a serial structure for nonlinear process monitoring [29]. However, the above-mentioned methods do not take dynamic characteristic of the process data into account.

Motivated by the above discussions, a novel multi-feature extraction technique is proposed for nonlinear dynamic process monitoring. Through a serial structure, the proposed method integrates DiPCA, PCA and KPCA methods to extract the multi-feature including dynamic, linear and nonlinear features. The proposed method is referred to as MFPCA in this paper. Specifically, DiPCA is first conducted to capture the dynamic features. After the dynamic components are extracted, PCA is applied in the residuals for additional research and analysis. Furthermore, PCA model residuals still contain nonlinear information that can be retrieved by KPCA. As a result, the original data space is decomposed into several subspaces, in which the variations of different features can be detected. A combined Hotelling's T^2 statistic is constructed based on all the extracted features from DiPCA, PCA and KPCA. And a squared prediction error (SPE or Q) statistic is established.

The main contributions of the proposed MFPCA method are as follows.

- It takes the unique advantage of DiPCA, PCA and KPCA methods in modeling different features and then develops a monitoring index through combining these features for real-time process monitoring.
- The dynamic, linear and nonlinear features contained in the process data are considered simultaneously and the abnormal variations of these features are reflected in the corresponding subspaces.

The remaining part of this paper is organized as follows. In Section 2, a brief review of PCA, KPCA and DiPCA methods is presented. Section 3 describes the proposed MFPCA method. Then, the MFPCA-based process monitoring scheme is presented in Section 4. In Section 5, case studies on a numerical example and the Tennessee Eastman (TE) process are presented to illustrate the applicability and efficiency of the proposed MFPCA-based process monitoring method. Finally, conclusions are drawn in Section 6.

2. Review of PCA, KPCA and DiPCA

2.1. PCA

For PCA, the objective is to find an orthogonal transformation so that the new variables after transformation, i.e., principal components (PCs), retain as much variance information of original data as possible. Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ with n samples and m variables, the PCA model is represented as,

$$\mathbf{X} = \sum_{i=1}^{l} \mathbf{t}_i \mathbf{p}_i^T + \mathbf{E} \tag{1}$$

where $\mathbf{t}_i$ is the ith PC, $\mathbf{p}_i \in \mathbb{R}^m$ is the loading vector corresponding to ith PC, $\mathbf{E}$ is the residual matrix and l ($l < m$) is the number of retained PCs.

2.2. KPCA

The nonlinear structure among the original data is more likely to be linear after high-dimensional nonlinear mapping. In KPCA, the original data $\mathbf{x}$ is mapped onto the feature space $\mathcal{H}$ through a nonlinear function Φ,

$$\mathbf{x} \in \mathbb{R}^m \longrightarrow \Phi(\mathbf{x}) \in \mathcal{H}$$

Then PCA is performed on the data $\Phi(\mathbf{x})$ to extract kernel PCs (KPCs) as nonlinear feature. Since $\Phi(\cdot)$ is implicit and unknown, kernel function $\kappa(\cdot)$ is used to help completing the nonlinear transformation. Specifically, the dot products in $\mathcal{H}$ can be calculated by the kernel function in original space.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \tag{2}$$

For KPCA, there are several types of kernel functions. In this study, the commonly used radial basis function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = exp(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2/2\sigma^2)$ is adopted. And σ is a scaling constant. More detailed explanation and implementation of KPCA can readily be found in the literature [6,19,21].

2.3. DiPCA

Assumed that $\mathbf{x}_k \in \mathbb{R}^m$ is the sample vector at time k and $\mathbf{w} \in \mathbb{R}^m$ is the weight vector, then its latent score can be expressed as $t_k = \mathbf{x}_k^T\mathbf{w}$. The future latent score can be predicted from the past scores.

$$\begin{aligned}\hat{t}_k &= \beta_1 t_{k-1} + \beta_2 t_{k-2} + \cdots + \beta_s t_{k-s} \\ &= [\mathbf{x}_{k-1}^T, \mathbf{x}_{k-2}^T, \ldots, \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w})\end{aligned} \tag{3}$$

where $\hat{t}_k$ is the predicted latent score, s is the dynamic order, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_s]^T$ is the coefficient vector, $\boldsymbol{\beta}\otimes\mathbf{w}$ is the Kronecker product. Based on the dynamic model (3), the objective of DiPCA is to maximize the covariance between the t_k and $\hat{t}_k$,

$$\begin{aligned}&\max_{\mathbf{w},\boldsymbol{\beta}} \sum_{k=s+1}^{n} \mathbf{w}^T \mathbf{x}_k [\mathbf{x}_{k-1}^T, \mathbf{x}_{k-2}^T, \ldots, \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w}) \\ &s.t. \quad \|\mathbf{w}\| = 1, \left\|\boldsymbol{\beta}\right\| = 1\end{aligned} \tag{4}$$

3. The proposed MFPCA method

To extract the dynamic, linear and nonlinear features, DiCPA, PCA and KPCA are performed sequentially through a serial structure. The schematic of the proposed MFPCA method is shown in Fig. 1.

3.1. Dynamic feature extraction

Firstly, the DiPCA method is used to capture the dynamic characteristic of the process. For the normalized process data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$, the optimization problem (4) can be rewritten as,

$$\begin{aligned}&\max_{\mathbf{w},\boldsymbol{\beta}} \mathbf{w}^T \mathbf{X}_{s+1}^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{w}) \\ &s.t. \quad \|\mathbf{w}\| = 1, \left\|\boldsymbol{\beta}\right\| = 1\end{aligned} \tag{5}$$

where $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+n-s-1}]^T$ and $\mathbf{Z}_s = [\mathbf{X}_s, \mathbf{X}_{s-1}, \ldots, \mathbf{X}_1]$. After weight vector $\mathbf{w}$ and latent scores $\mathbf{t} = \mathbf{X}\mathbf{w}$ are obtained, the loading vector is calculated as $\mathbf{p} = \mathbf{X}^T\mathbf{t}/\mathbf{t}^T\mathbf{t}$. The objective (5) can be solved through an iterative algorithm (see [14]). Following the iterative algorithm, we can obtain several latent scores $\mathbf{t}_j$ and loading vectors $\mathbf{p}_j$ for $j = 1, 2, \ldots, L_d$, where L_d is the number of retained dynamic PCs (DPCs). Then, we have the following relation,

$$\mathbf{E} = \mathbf{X} - \sum_{j=1}^{L_d} \mathbf{t}_j \mathbf{p}_j^T = \mathbf{X} - \mathbf{T}_{L_d}\mathbf{P}_{L_d}^T \tag{6}$$

where loading matrix $\mathbf{P}_{L_d} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{L_d}]$, latent scores matrix $\mathbf{T}_{L_d} = \mathbf{X}\mathbf{R}$, $\mathbf{R} = \mathbf{W}_{L_d}(\mathbf{P}_{L_d}^T\mathbf{W}_{L_d})^{-1}$ and weight matrix $\mathbf{W}_{L_d} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{L_d}]$.

By constructing $\mathbf{T}_{L_d}^i$ in the same way as $\mathbf{X}_i$, an inner VAR model is built to represent the dynamic relationships between $\mathbf{T}_{L_d}^{s+1}$

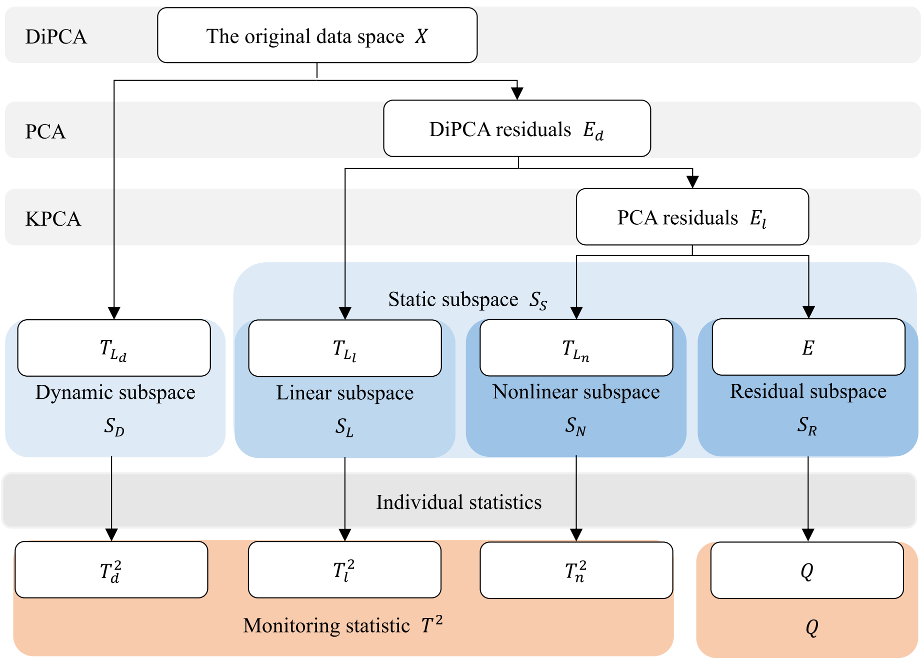


Fig. 1. The schematic of MFPCA method.

and $\mathbf{T}_{L_d}^1, \mathbf{T}_{L_d}^2, \ldots, \mathbf{T}_{L_d}^s$,

$$\begin{aligned}\mathbf{T}_{L_d}^{s+1} &= \mathbf{T}_{L_d}^1\boldsymbol{\Theta}_s + \mathbf{T}_{L_d}^2\boldsymbol{\Theta}_{s-1} + \cdots + \mathbf{T}_{L_d}^s\boldsymbol{\Theta}_1 + \mathbf{V}\\ &= \bar{\mathbf{T}}_{L_d}^s\boldsymbol{\Theta} + \mathbf{V}\end{aligned} \tag{7}$$

where $\bar{\mathbf{T}}_{L_d}^s = [\mathbf{T}_{L_d}^1, \mathbf{T}_{L_d}^2, \cdots, \mathbf{T}_{L_d}^s]$, $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_s; \boldsymbol{\Theta}_{s-1}; \ldots; \boldsymbol{\Theta}_1]$ is the coefficient matrix and $\mathbf{V}$ is the inner VAR model residuals. The model parameter $\boldsymbol{\Theta}$ can be estimated by least squares,

$$\hat{\boldsymbol{\Theta}} = (\bar{\mathbf{T}}_{L_d}^{sT}\bar{\mathbf{T}}_{L_d}^s)^{-1}\bar{\mathbf{T}}_{L_d}^{sT}\mathbf{T}_{L_d}^{s+1} \tag{8}$$

Then the prediction of $\mathbf{T}_{L_d}^{s+1}$ can be derived,

$$\hat{\mathbf{T}}_{L_d}^{s+1} = \bar{\mathbf{T}}_{L_d}^s\hat{\boldsymbol{\Theta}} \tag{9}$$

Finally, $\mathbf{X}$ is decomposed as,

$$\mathbf{X} = \hat{\mathbf{T}}_{L_d}^{s+1}\mathbf{P}_{L_d}^T + \mathbf{E}_d \tag{10}$$

where $\mathbf{E}_d$ is the DiPCA model residuals which contain little or no auto-covariance information.

Given a test vector $\mathbf{x}_t \in \mathbb{R}^m$, its dynamic latent scores $\mathbf{t}_t^d$, predicted dynamic latent scores $\hat{\mathbf{t}}_t^d$ and DiPCA model residuals $\mathbf{e}_t^d$ are calculated,

$$\begin{aligned}\hat{\mathbf{t}}_t^d &= \sum_{i=1}^{s}\boldsymbol{\Theta}_i^T\mathbf{t}_{t-i}^d, \mathbf{t}_t^d = \mathbf{R}^T\mathbf{x}_t\\ \mathbf{e}_t^d &= \mathbf{x}_t - \mathbf{P}_{L_d}\hat{\mathbf{t}}_t^d\end{aligned} \tag{11}$$

3.2. Linear feature extraction

After the dynamic relationships are extracted through DiPCA, the residuals still contain some useful information which can be treated by PCA for additional exploration and analysis [14]. Thus, we perform singular value decomposition on $\mathbf{E}_d$,

$$\mathbf{E}_d = \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{Z}\mathbf{V}^T, \mathbf{Z} = \mathbf{U}\mathbf{D} \tag{12}$$

where $\mathbf{V}$ and $\mathbf{Z}$ represent the loading matrix and PCs, respectively. And they are sorted in columns based on the order of the singular values in $\mathbf{D}$.

In the PCA model (12), the PCs with the first L_l largest singular values are selected. Then the linear PCA model of $\mathbf{E}_d$ is expressed as,

$$\mathbf{E}_d = \mathbf{T}_{L_l}\mathbf{P}_{L_l}^T + \mathbf{E}_l \tag{13}$$

where $\mathbf{T}_{L_l}$ and $\mathbf{P}_{L_l}$ are the first L_l columns of matrices $\mathbf{Z}$ and $\mathbf{V}$, $\mathbf{E}_l$ represents the PCA model residuals.

For the test residual vector $\mathbf{e}_t^d$, the linear latent scores $\mathbf{t}_t^l$ and PCA model residuals $\mathbf{e}_t^l$ can be calculated as follows,

$$\begin{aligned}\mathbf{t}_t^l &= \mathbf{P}_{L_l}^T\mathbf{e}_t^d\\ \mathbf{e}_t^l &= (\mathbf{I} - \mathbf{P}_{L_l}\mathbf{P}_{L_l}^T)\mathbf{e}_t^d\end{aligned} \tag{14}$$

3.3. Nonlinear feature extraction

Most industrial processes are nonlinear in nature. After performing PCA as stated in (13), the residuals maybe contain nonlinear information [29]. Thus, KPCA is used to further explore the nonlinear features in the PCA model residuals $\mathbf{E}_l$.

Denote the nonlinear mapping as $\mathbf{E}_l \in \mathbb{R}^{n\times m} \rightarrow \Phi(\mathbf{E}_l) \in \mathbb{R}^n \times \mathcal{H}$. To perform KPCA on the residuals $\mathbf{E}_l$, the eigenvalue decomposition method is adopted,

$$(n-1)\lambda_i\boldsymbol{\alpha}_i = \mathbf{K}\boldsymbol{\alpha}_i \tag{15}$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is kernel matrix with its element given by $[\mathbf{K}]_{ij} = \kappa(\mathbf{e}_i^l, \mathbf{e}_j^l)$. λ_i and $\boldsymbol{\alpha}_i$ represent the ith eigenvalue and eigenvector of $\mathbf{K}$. Then the loading vector and KPCs can be expressed as $\mathbf{p}_i = \Phi(\mathbf{E}_l)^T\boldsymbol{\alpha}_i$ and $\mathbf{t}_i = \mathbf{K}\boldsymbol{\alpha}_i$, respectively. Assumed L_n as the number of retained KPCs, $\Phi(\mathbf{E}_l)$ can be decomposed as follows,

$$\Phi(\mathbf{E}_l) = \mathbf{T}_{L_n}\mathbf{P}_{L_n}^T + \mathbf{E} \tag{16}$$

where $\mathbf{T}_{L_n} = \mathbf{K}\mathbf{A}_{L_n}$. The coefficient matrix $\mathbf{A}_{L_n} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{L_n}]$ and loading matrix $\mathbf{P}_{L_n} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{L_n}]$.

For the test residual vector $\mathbf{e}_t^l$, the nonlinear latent scores are computed by projecting $\Phi(\mathbf{e}_t^l)$ onto $\mathbf{P}_{L_n}$ as

$$\mathbf{t}_t^n = \mathbf{P}_{L_n}^T\Phi(\mathbf{e}_t^l) = \mathbf{A}_{L_n}^T\mathbf{k}_t \tag{17}$$

where $\mathbf{k}_t = [\kappa(\mathbf{e}_t^l, \mathbf{e}_1^l), \kappa(\mathbf{e}_t^l, \mathbf{e}_2^l), \ldots, \kappa(\mathbf{e}_t^l, \mathbf{e}_n^l)]^T$ is the test kernel vector.

Table 1
The comparisons of MFPCA with other relevant methods.

Method	Focusing on different characteristics			Computation time
	Dynamic	Nonlinear	Interpretability	
SPCA [29]		√	Easy	Longer
DiPCA [14]	√		Easy	Shortest
DKPCA [30]	√	√	Difficult, the extracted nonlinear dynamics of the process are implicit	Longest, performing kernel function on augmented data
MFPCA	√	√	Easy	Longer compared to SPCA, but shorter than DKPCA

By synthesizing the above multi-feature extraction (11), (14) and (17), the original data $\mathbf{x}$ can be fully decomposed as follows,

$$\begin{cases} \mathbf{x} = \hat{\mathbf{x}}^d + \hat{\mathbf{x}}^l + \hat{\mathbf{x}}^n + \mathbf{e} \\ \hat{\mathbf{x}}^d = \mathbf{P}_{L_d}\hat{\mathbf{t}}^d \in S_D \\ \hat{\mathbf{x}}^l = \mathbf{P}_{L_l}\mathbf{P}_{L_l}^T(\mathbf{x} - \mathbf{P}_{L_d}\hat{\mathbf{t}}^d) \in S_L \\ \hat{\mathbf{x}}^n = \mathbf{P}_{L_n}\mathbf{P}_{L_n}^T\Phi((\mathbf{I} - \mathbf{P}_{L_l}\mathbf{P}_{L_l}^T)(\mathbf{x} - \mathbf{P}_{L_d}\hat{\mathbf{t}}^d)) \in S_N \\ \mathbf{e} = (\mathbf{I} - \mathbf{P}_{L_n}\mathbf{P}_{L_n}^T)\Phi((\mathbf{I} - \mathbf{P}_{L_l}\mathbf{P}_{L_l}^T)(\mathbf{x} - \mathbf{P}_{L_d}\hat{\mathbf{t}}^d)) \in S_R \end{cases} \tag{18}$$

Along with the MFPCA model (18), the original data space is decomposed into four orthogonal subspaces: dynamic subspace S_D, linear subspace S_L, nonlinear subspace S_N and residual subspace S_R. In [14], the orthogonality between S_D and static subspace S_S has been proved. Both PCA and KPCA methods can provide orthogonal decomposition. Therefore, the derived subspaces are orthogonal to each other. The comparisons of MFPCA with other relevant methods are listed in Table 1. Compared with SPCA, DiPCA and dynamic kernel PCA (DKPCA), it can be seen that the proposed MFPCA method can capture the dynamic and nonlinear characteristics of process data in more interpretable way at lower computational cost.

3.4. Determination of model parameters

In the MFPCA method, there are three types of parameters to be determined: the dynamic order s in DiPCA model, the retained number L_d, L_l and L_n for DPCs, PCs, and KPCs, and the kernel parameter σ in KPCA.

- Obviously, the choice of the dynamic order s will have significant influence on the decomposition of dynamic and static parts. To determine s, a cross validation (CV) method is adopted as in [14]. The criterion is to observe the sample crosscorrelation of any two residual variables. An optimal s should make the sample crosscorrelation close to 0.
- The average eigenvalue (AE) method is usually adopted to determine L_d, L_l and L_n for DPCs, PCs and KPCs owing to its simplicity and robustness. It determines the number of retained components by observing whether the corresponding eigenvalues are larger than the average eigenvalue [29].
- The kernel parameter σ is critical to process monitoring. As σ becomes large, the false alarm rate (FAR) decreases meanwhile the fault detection rate (FDR) also decreases. Here, we use the cross validation method to find an optimal σ that minimizes the FAR [6,21,30].

4. MFPCA-based process monitoring scheme

Based on the features extracted from MFPCA model (18), we can obtain the following monitoring statistics,

$$T_d^2 = \hat{\mathbf{t}}_t^{dT}\Lambda_d^{-1}\hat{\mathbf{t}}_t^d$$
$$T_l^2 = \mathbf{t}_t^{lT}\Lambda_l^{-1}\mathbf{t}_t^l$$
$$T_n^2 = \mathbf{t}_t^{nT}\Lambda_n^{-1}\mathbf{t}_t^n$$
$$Q = \mathbf{e}^T\mathbf{e}$$

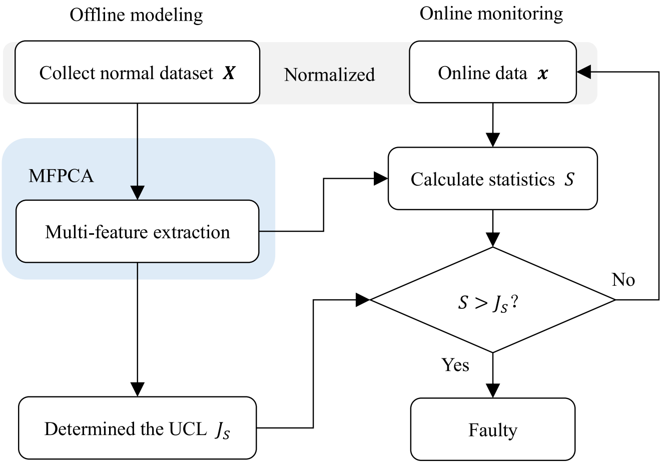


Fig. 2. The MFPCA-based process monitoring scheme.

where Λ_d, Λ_l and Λ_n are covariance matrices for scores $\hat{\mathbf{t}}_t^d$, $\mathbf{t}_t^l$ and $\mathbf{t}_t^n$ under the normal operation condition, respectively. These statistics T_d^2, T_l^2, T_n^2 and Q monitor the abnormal variations in subspace S_D, S_L, S_N and S_R, respectively. Except the individual statistics, the derived three features are also incorporated into a multi-feature $\mathbf{t}_t^m = [\hat{\mathbf{t}}_t^d, \mathbf{t}_t^l, \mathbf{t}_t^n]$ to reflect the process status,

$$T^2 = \mathbf{t}_t^{mT}\Lambda^{-1}\mathbf{t}_t^m \tag{19}$$

where $\Lambda = \{\Lambda_d, \Lambda_l, \Lambda_n\}$ is diagonal matrix.

To determine whether a fault occurs, the upper control limits (UCLs) are required for the developed statistics. For notational convenience, we denote S as one of the above statistics. In this study, the kernel density estimation (KDE) method is used to estimate the unknown distribution which statistic S obeys [8,31,32]. For statistic S, its probability density function $p(S)$ can be estimated through the kernel function $\mathcal{K}$,

$$p(S) = \frac{1}{Mh}\sum_{j=1}^{M}\mathcal{K}\left(\frac{S - S_j}{h}\right) \tag{20}$$

where M is the number of samples, h is the bandwidth. In this paper, kernel function $\mathcal{K}(g) = e^{-g^2/2}/\sqrt{2\pi}$. The selections of kernel function and parameter h are referred to [32]. Given a significance level α, the upper control limit J_S is calculated by solving the following equation,

$$\int_{-\infty}^{J_S} p(S)dS = \alpha \tag{21}$$

In case studies, α is set to 95% for all statistics. To sum up, the proposed MFPCA-based process monitoring scheme is presented in Fig. 2.

5. Case studies

In this section, a numerical example and the TE process are presented to illustrate the proposed MFPCA-based process monitoring performance. The experiment environment is Windows 10 operating system, Intel(R) Xeon(R) Bronze 3104 CPU @ 1.70GHz, 64.0GB RAM, and Matlab(R2018b).

5.1. Numerical example

Consider a nonlinear dynamic system, which is an integrated version of the two examples given in [14,29].

$$\begin{cases} \mathbf{z}_k = \mathbf{c} + \mathbf{A}\mathbf{z}_{k-1} + \mathbf{w}_k \\ \mathbf{x}_k = \mathbf{P}\mathbf{z}_k + f(\mathbf{u}_k) + \mathbf{e}_k \end{cases} \tag{22}$$

$$\mathbf{A} = \begin{bmatrix} 0.5205 & 0.1022 & 0.0599 \\ 0.5367 & -0.0139 & 0.4159 \\ 0.0412 & 0.6054 & 0.3874 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 0.5205 \\ 0.5367 \\ 0.0412 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 0.4316 & 0.1723 & -0.0574 \\ 0.1202 & -0.1463 & 0.5348 \\ 0.2483 & 0.1982 & 0.4797 \\ 0.1151 & 0.1557 & 0.3739 \\ 0.2258 & 0.5461 & -0.0424 \end{bmatrix}$$

where state variables $\mathbf{z}_k \in \mathbb{R}^3$ is generated from a VAR(1) process, input variables $\mathbf{u} = [u_1, u_2]^T \in \mathbb{R}^2$ obeys a uniform distribution on the closed interval [0,2]. $\mathbf{w}_k \in \mathbb{R}^3 \sim \mathbf{N}(0, 1^2)$ and $\mathbf{e}_k \in \mathbb{R}^5 \sim \mathbf{N}(0, 0.1^2)$ denote the independent noises. The nonlinear mapping function $f(\mathbf{u})$ is expressed as follows,

$$f(\mathbf{u}) = 0.05[u_1, u_2, 5u_1 - 2u_2, u_1^2 - 3u_2, -u_1^3 + 3u_2^2]^T$$

Based on system (22), 1000 samples are generated to build the MFPCA model. Here, SPCA, DiPCA and DKPCA methods are also employed for comparison. By using the method described in Section 3.4, $s = 1$, $L_d = 3$ and $L_l = 2$ are selected for DiPCA and MFPCA methods. In MFPCA, kernel parameter σ and L_n are determined as 40 and 2. SPCA and DKPCA methods also adopt the same kernel parameter. Also, $L_l = 1$ and $L_n = 3$ for SPCA, $L_n = 8$ for DKPCA.

To validate the process monitoring performance, the following artificial faults are designed,

- Fault 1: $\mathbf{z}_k := \mathbf{z}_k + [1.8452, 1.5656, 0.0640]^T$ for $k > 500$. The fault is injected to the dynamic components. Fault 1 should be detected by SPCA T_l^2, DiPCA T_d^2 and T_l^2, DKPCA T^2, MFPCA T_d^2 and T_l^2 indices. Since this fault does not affect the residual part of the static relationships, it is not detected by Q index [14]. The monitoring charts are depicted in Fig. 3. The four methods all correctly detect the fault with satisfactory results. However, DKPCA Q exceeds the corresponding UCL after the occurrence of the fault.
- Fault 2: $\mathbf{u}_k := \mathbf{u_k} + [-1.7070, 1.6926]^T$ for $k > 500$. In this case, the fault causes the change of nonlinear part and should be monitored by SPCA T_n^2, DKPCA T^2, MFPCA T_n^2 statistics. Fig. 4 plots the monitoring charts. As shown in Fig. 4, for DiPCA, T_d^2 and T_l^2 statistics can not detect Fault 2 due to the nonlinearity. On the other hand, since the nonlinear features are left in the residual part, the fault is detected partly by Q statistic.
- Fault 3: $\mathbf{x}_k := \mathbf{x_k} + [0.2980, -0.3039, -0.4443, 0.4151, -0.1047]^T$ for $k > 500$. It is an additional disturbance of $\mathbf{x}_k$, which occurs in the residual part of the static relationships. Therefore, it should be only detected by Q statistic. The results are shown in Fig. 5. It can be seen that SPCA T_n^2 and DKPCA T^2 statistics fluctuate around the UCLs.

Based on the above discussions and results, it can be observed that the proposed MFPCA method has a better representation for the data structure. To further quantify the monitoring performance, two indices are introduced. The first one is FDR, which is defined as the ratio of detected faulty samples to all faulty samples. Another index is FAR, which is defined as the ratio of false detected normal samples to all fault-free samples. Table 2 compares the averaged monitoring results. It can be seen that the FARs of MFPCA are at the same level as that of the SPCA method. In addition, the proposed MFPCA method can offer comparable even better FDRs to SPCA and DiPCA methods, which maybe because it takes advantage of DiPCA, PCA and KPCA to extract different features and improves the monitoring performance. For FDRs and FARs, the statistics T_d^2, T_l^2 and T_n^2 also provide similar conclusions as described.

Table 3 gives the averaged computation time for online process monitoring. MFPCA method (0.4756ms) requires shorter computation time than DKPCA method (0.5327ms) for real-time process monitoring. Since DKPCA performs kernel function after augmenting the data, it requires more time than other methods.

Table 3
Numerical example: the comparisons of the averaged computation time for online process monitoring, which is based on 100 simulations.

Method	SPCA	DiPCA	DKPCA	MFPCA
Time (ms)	0.5173	0.0189	0.5327	0.4756

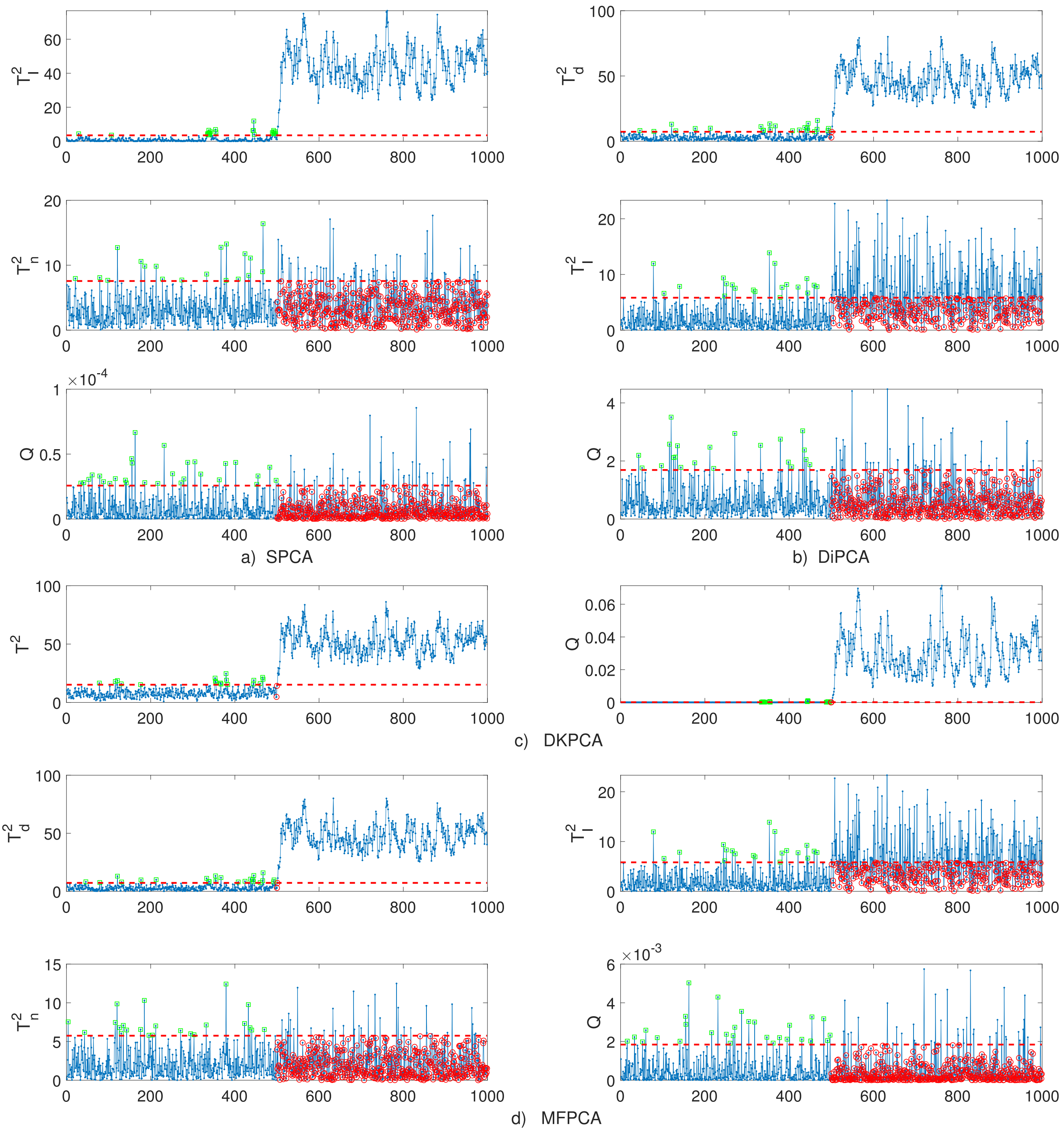
5.2. TE process

The Tennessee Eastman (TE) process, which is based on a real chemical process, has become a widely acceptable benchmark process for developing, studying and evaluating process monitoring method [33]. The flowchart of TE process is depicted in Fig. 6. It consists of five major unit operations: a condenser, a reactor, a stripper, a compressor, and a vapor-liquid separator. Four gaseous reactants A, C, D, E and the inert B are fed to the reactor in which

Table 2
Numerical example: averaged monitoring performance of SPCA, DiPCA, DKPCA and MFPCA methods, which is based on 100 simulations.

Fault	SPCA		DiPCA		DKPCA		MFPCA	
	T^2	Q	T^2	Q	T^2	Q	T^2	Q
1	99.62[a]	8.76	99.65	10.57	99.38	99.41	99.60	7.30
	6.28[b]	6.29	5.95	5.77	5.64	5.63	6.27	6.39
2	80.07	42.79	40.95	74.73	95.09	24.20	88.44	29.58
	6.06	6.42	5.71	5.69	5.32	5.11	5.99	6.47
3	29.15	99.98	7.41	74.90	39.14	99.76	8.28	99.99
	5.88	6.35	5.51	5.70	5.27	5.29	5.81	6.66

[a] First row: Fault detection rates (FDRs).
[b] Second row: False alarm rates (FARs).



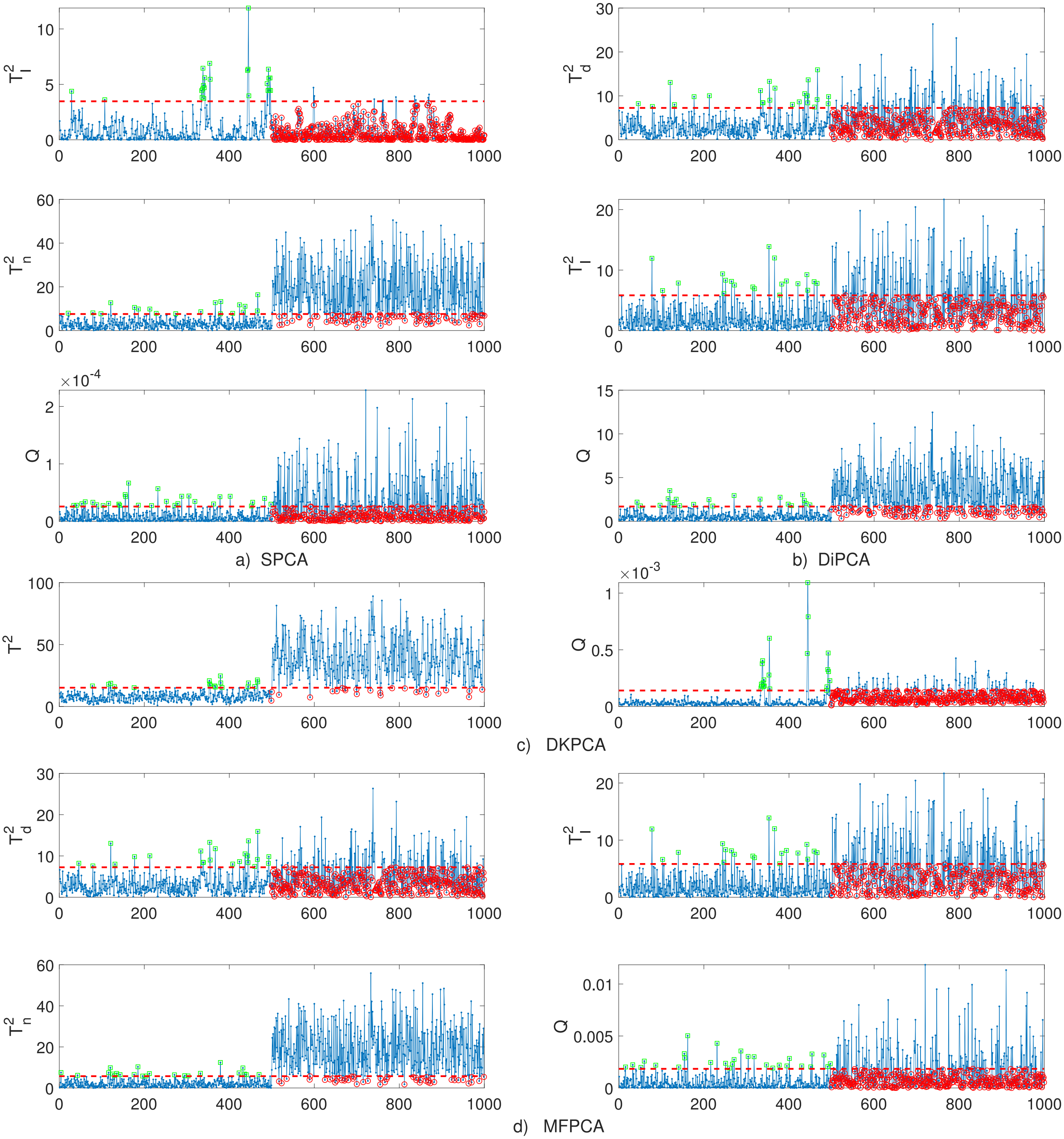
a) SPCA

b) DiPCA

c) DKPCA

d) MFPCA

Fig. 3. The monitoring charts of SPCA, DiPCA, DKPCA and MFPCA for Fault 1. Red dashed lines indicate the 95% control limit J_S, blue dash-dot lines represent the corresponding statistic, green square and red circle represent false-alarm points and missed-detection points, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



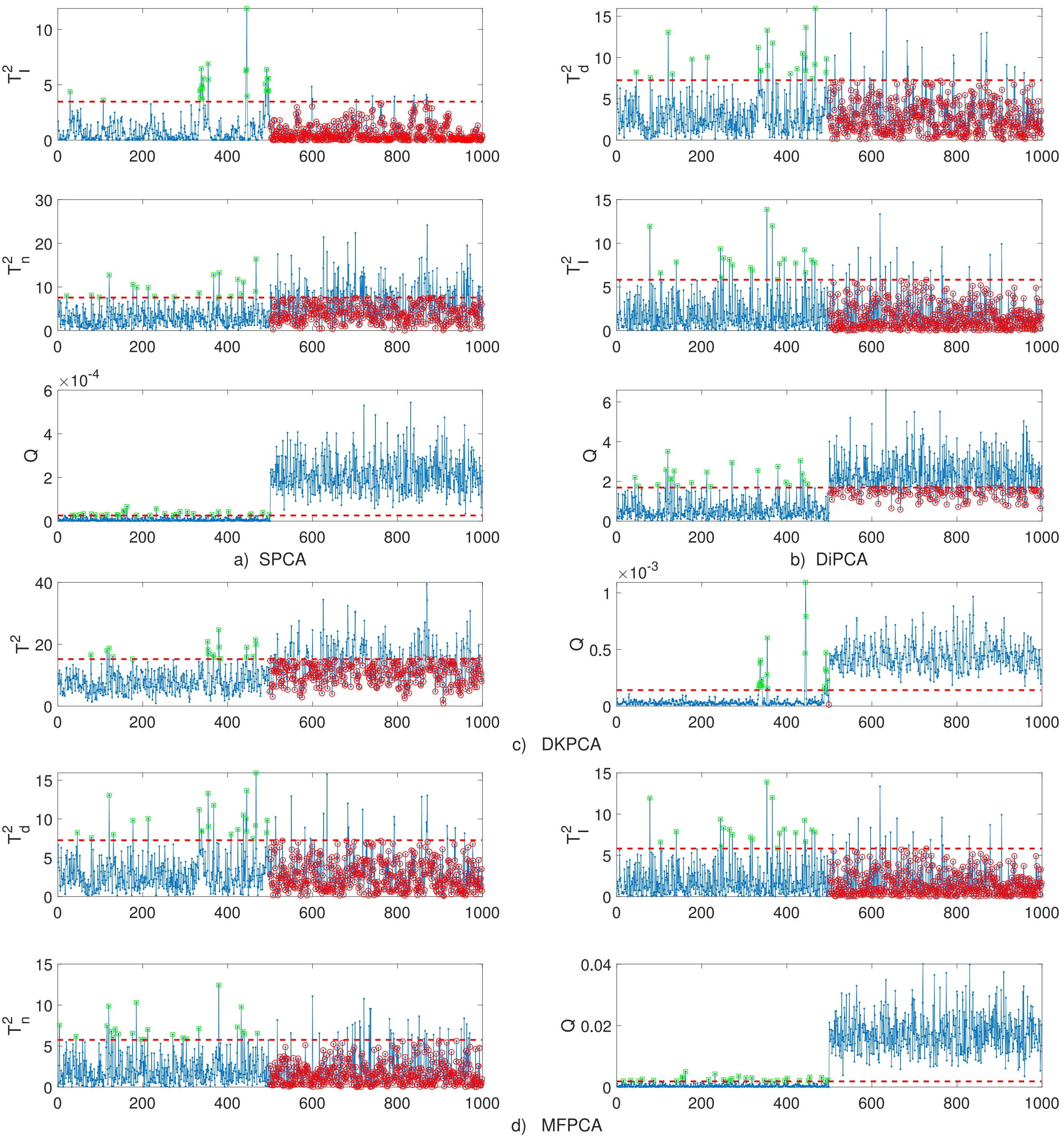
a) SPCA

b) DiPCA

c) DKPCA

d) MFPCA

Fig. 4. The monitoring charts of SPCA, DiPCA, DKPCA and MFPCA for Fault 2. Red dashed lines indicate the 95% control limit J_S, blue dash-dot lines represent the corresponding statistic, green square and red circle represent false-alarm points and missed-detection points, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



a) SPCA

b) DiPCA

c) DKPCA

d) MFPCA

Fig. 5. The monitoring charts of SPCA, DiPCA, DKPCA and MFPCA for Fault 3. Red dashed lines indicate the 95% control limit J_S, blue dash-dot lines represent the corresponding statistic, green square and red circle represent false-alarm points and missed-detection points, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

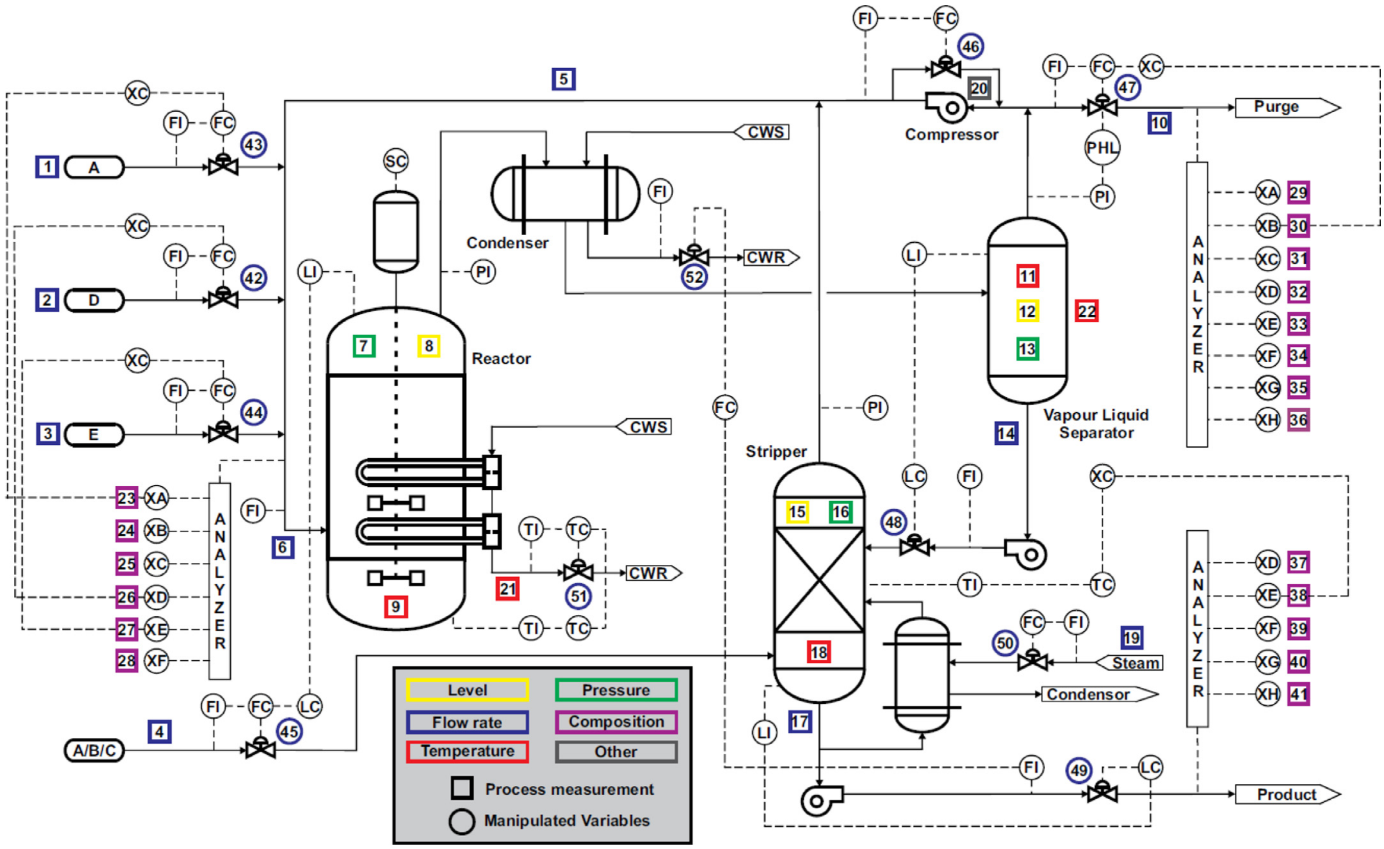


Fig. 6. The TE process flowsheet [34].

the liquid products G, H and a by-product F are produced through the following reactions,

$$
\begin{array}{ll}
A(g)+C(g)+D(g) & \rightarrow G(liq) \\
A(g)+C(g)+E(g) & \rightarrow H(liq) \\
A(g)+E(g) & \rightarrow F(liq) \\
3D(g) & \rightarrow 2F(liq)
\end{array}
$$

There are 41 process measurements XMEAS(1–41) and 12 manipulated variables XMV(1–12) in TE process [33]. A widely used TE dataset for process monitoring is available from http://web.mit.edu/braatzgroup/links.html. In this dataset, 52 variables are included, except for XMV(12) that is the agitator speed. 22 different simulation conditions including a normal situation IDV(0) and 21 fault modes IDV(1–21) are simulated. The descriptions are listed in Table 4.

In this study, 33 variables (XMEAS(1–22) and XMV(1–11)) are selected for process monitoring. And the normal dataset with 960 samples is used to build model. According to Section 3.4, $s=3$, $L_d=13$ and $L_l=14$ are set for DiPCA and MFPCA methods, $\sigma=200$ and $L_n=16$ is chosen for MFPCA method, $\sigma=165$, $L_l=13$ and $L_n=14$ are selected for SPCA method, $s=3$, $\sigma=30$ and $L_n=80$ are determined for DKPCA method.

To illustrate the performance of the proposed MFPCA-based process monitoring method, the detection delay time (DD) between fault occurrence time and fault detection time is also calculated for comparison, except FAR and FDR. In general, a better monitoring performance means that the higher FDR, the lower FAR and shorter DD [8,21]. Tables 5 and 6 summarize the monitoring results including FAR, FDRs and DDs based on SPCA, DiPCA, DKPCA and MFPCA methods, respectively.

Here, we compute the FAR on the normal dataset consisting of 500 samples. Although the FAR of MFPCA T^2 statistic (7.44%) is slightly higher than SPCA T^2 statistic (5.80%), it is lower than DKPCA T^2 statistic (8.47%). As can be seen in Table 5, MFPCA method can derive better monitoring performance than SPCA, DiPCA and DKPCA methods, particulary for IDV(10), IDV(20). For MFPCA, most of the faults can be successfully detected and their FDRs reach more than 90%, except IDV(3), IDV(9), IDV(15) and IDV(21). For IDV(3), IDV(9) and IDV(15), since they have little impact on the overall process behavior due to feedback control, they are extremely difficult to be detected. However, MFPCA method still gives better performance than SPCA and DiPCA methods for the three faults. For IDV(21), the FDR derived by MFPCA is lower than 90%, but it is higher than other methods.

Table 4
TE process: Description of fault modes.

Fault	Description	Type
IDV(0)	Normal situation	–
IDV(1)	A/C feed ratio, B composition constant	Step
IDV(2)	B composition, A/C ration constant	Step
IDV(3)	D feed temperature	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss	Step
IDV(7)	C header pressure loss-reduced availability	Step
IDV(8)	A, B, and C feed composition	Random
IDV(9)	D feed temperature	Random
IDV(10)	C feed temperature	Random
IDV(11)	Reactor cooling water inlet temperature	Random
IDV(12)	Condenser cooling water inlet temperature	Random
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16–20)	Unknown	Unknown
IDV(21)	The valve fixed at steady state position	Constant position

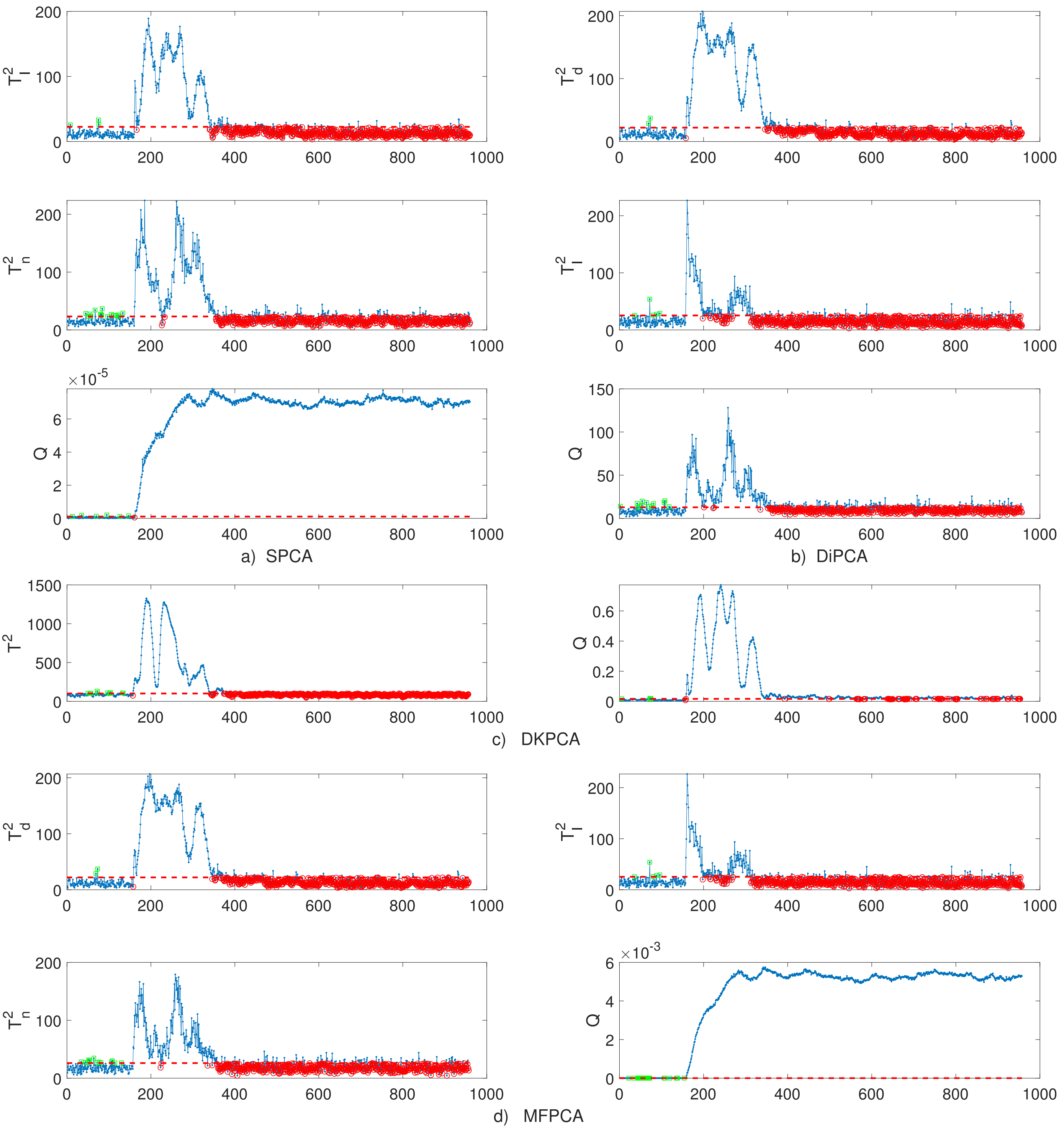
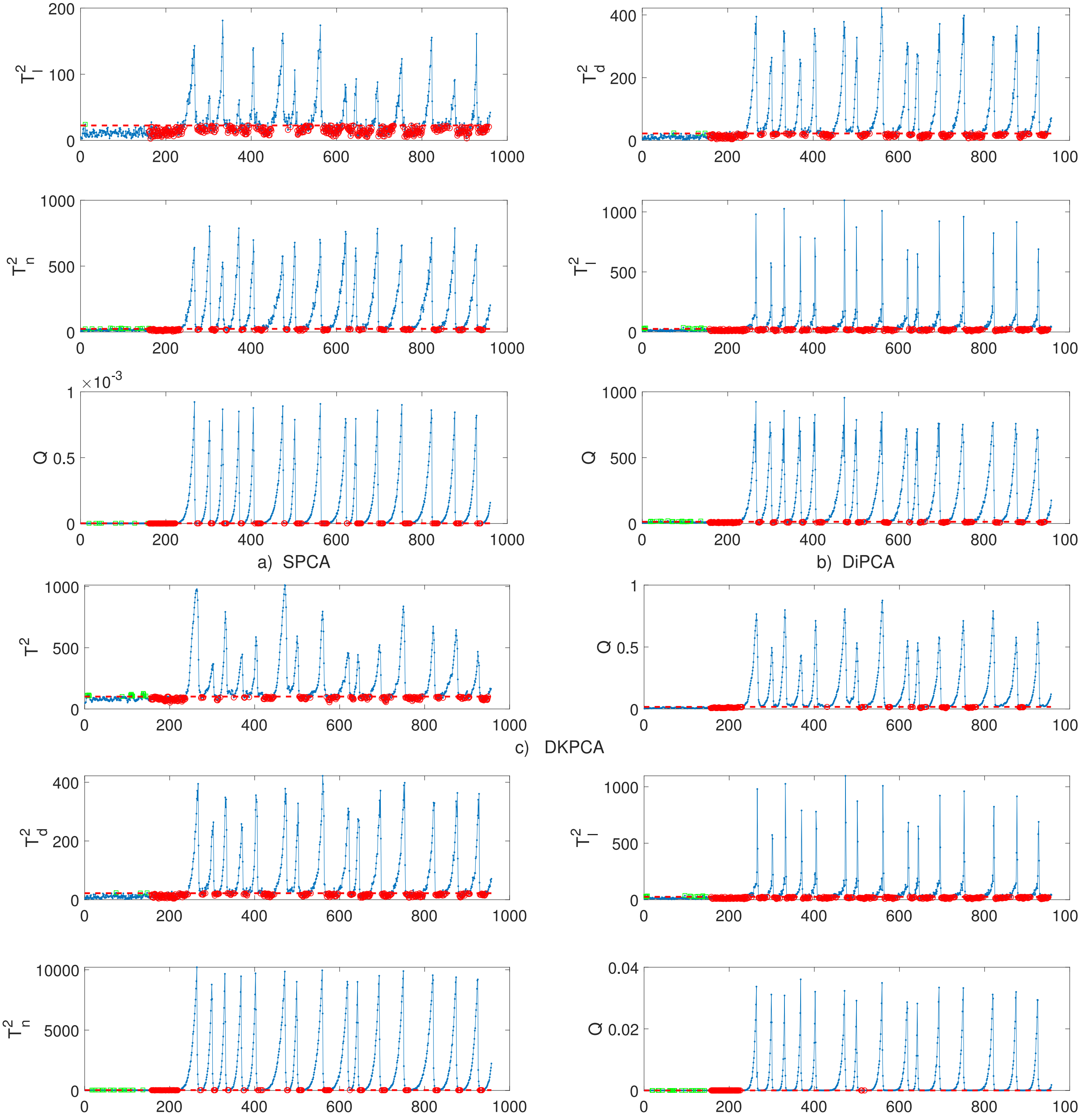


Fig. 7. The monitoring charts of SPCA, DiPCA, DKPCA and MFPCA methods for IDV(5). Red dashed lines indicate the 95% control limit J_S, blue dash-dot lines represent the corresponding statistic, green square and red circle represent false-alarm points and missed-detection points, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



a) SPCA

b) DiPCA

c) DKPCA

d) MFPCA

Fig. 8. The monitoring charts of SPCA, DiPCA, DKPCA and MFCPA methods for IDV(20). Red dashed lines indicate the 95% control limit J_S, blue dash-dot lines represent the corresponding statistic, green square and red circle represent false-alarm points and missed-detection points, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

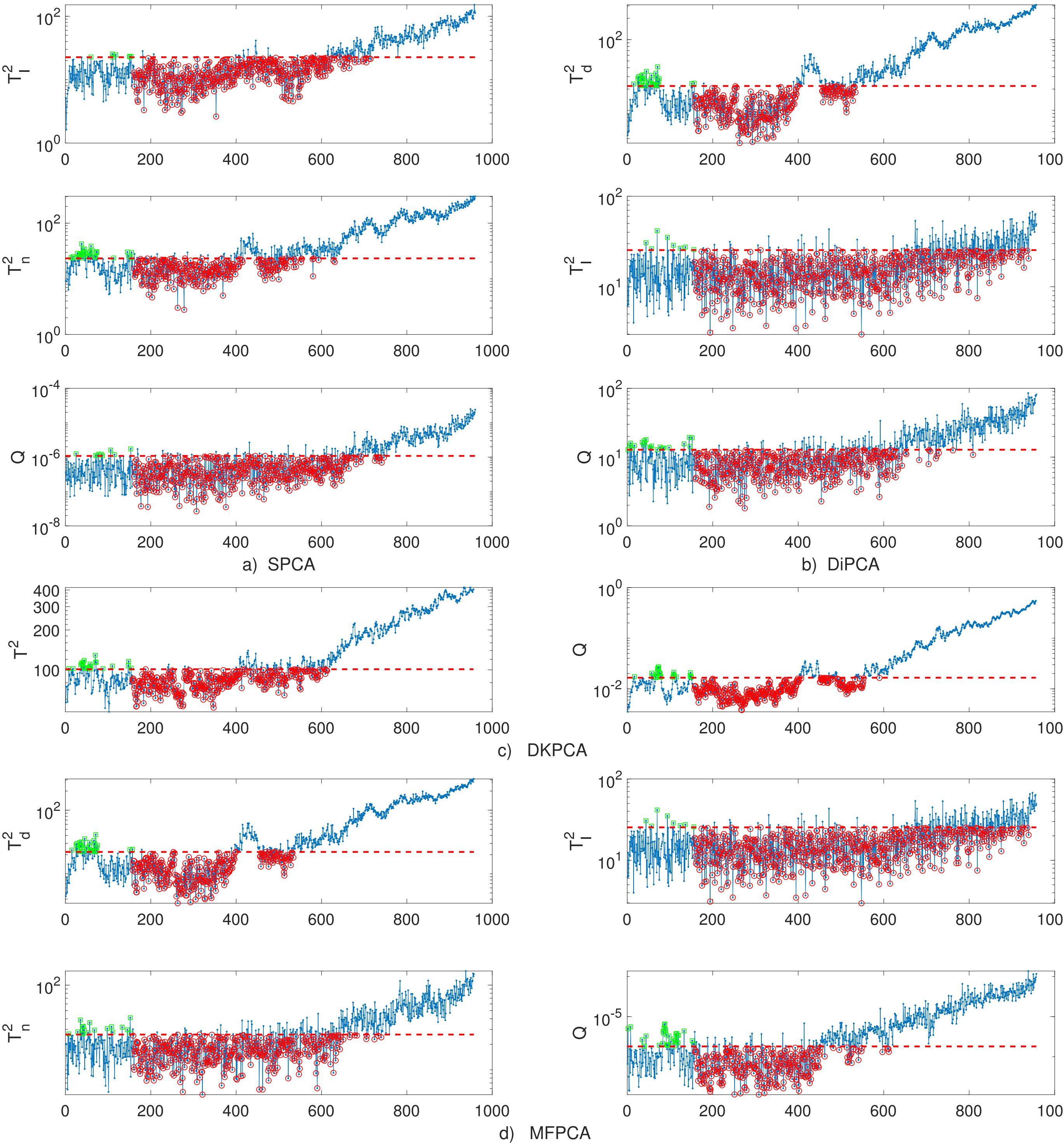


Fig. 9. The monitoring charts of SPCA, DiPCA, DKPCA and MFPCA methods for IDV(21). Red dashed lines indicate the 95% control limit J_S, blue dash-dot lines represent the corresponding statistic, green square and red circle represent false-alarm points and missed-detection points, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
TE process: FAR for IDV(0) and FDRs for IDV(1–21) with different methods.

Fault	SPCA		DiPCA		DKPCA		MFPCA	
	T^2	Q	T^2	Q	T^2	Q	T^2	Q
IDV(0)	5.80	4.80	4.63	10.66	8.47	2.62	7.44	9.05
IDV(1)	100.00	99.50	99.88	100.00	99.63	99.75	100.00	99.88
IDV(2)	98.88	94.13	98.88	96.63	98.63	98.38	99.00	96.25
IDV(3)	12.63	7.88	10.88	13.63	15.50	28.38	18.00	16.50
IDV(4)	100.00	84.50	89.00	100.00	99.88	99.88	100.00	100.00
IDV(5)	39.00	99.88	33.63	43.25	36.88	88.75	39.38	100.00
IDV(6)	100.00	100.00	99.75	100.00	99.38	99.88	100.00	100.00
IDV(7)	100.00	99.88	100.00	100.00	99.88	99.88	100.00	100.00
IDV(8)	98.50	84.13	98.25	96.00	97.63	99.63	99.00	97.50
IDV(9)	10.38	7.13	11.25	14.25	12.63	23.38	18.25	16.75
IDV(10)	89.88	71.63	84.25	84.63	56.50	82.50	91.00	93.50
IDV(11)	87.00	55.50	81.75	92.00	97.63	92.25	93.13	90.75
IDV(12)	99.50	99.13	99.63	99.25	99.63	100.00	99.63	100.00
IDV(13)	96.00	94.75	95.13	96.38	95.50	96.13	96.13	96.38
IDV(14)	100.00	99.88	100.00	100.00	99.88	99.88	100.00	100.00
IDV(15)	15.75	6.50	14.13	13.38	19.00	27.00	19.50	27.25
IDV(16)	93.50	68.50	75.75	93.00	37.50	84.38	94.00	92.50
IDV(17)	97.88	87.25	94.00	97.75	97.38	97.63	98.00	97.25
IDV(18)	91.50	90.13	91.00	91.00	90.63	93.25	92.00	91.25
IDV(19)	90.50	89.25	49.25	90.38	78.38	93.25	97.63	92.00
IDV(20)	76.75	80.25	67.63	75.75	68.13	82.50	88.25	91.38
IDV(21)	61.63	40.50	59.88	47.88	55.25	59.50	60.88	64.50
Average	79.01	74.30	73.99	78.34	74.07	83.15	81.13	83.98

Table 6
TE process: DDs (minutes) for IDV(1–21) with different methods.

Fault	SPCA		DiPCA		DKPCA		MFPCA	
	T^2	Q	T^2	Q	T^2	Q	T^2	Q
IDV(1)	3	15	6	3	12	9	3	6
IDV(2)	15	12	15	12	24	42	15	3
IDV(3)	45	9	54	45	90	39	45	3
IDV(4)	3	3	3	3	6	6	3	3
IDV(5)	3	6	3	3	6	9	3	3
IDV(6)	3	3	9	3	18	6	3	3
IDV(7)	3	3	3	3	6	6	3	3
IDV(8)	9	33	45	9	51	3	9	9
IDV(9)	3	30	3	3	6	3	3	3
IDV(10)	18	51	24	3	27	27	18	21
IDV(11)	18	6	9	3	3	3	3	3
IDV(12)	9	3	9	9	6	3	9	3
IDV(13)	6	24	81	6	3	3	81	33
IDV(14)	3	6	3	3	6	6	3	3
IDV(15)	117	78	189	15	84	264	198	15
IDV(16)	3	33	15	6	60	6	3	21
IDV(17)	3	45	3	54	9	9	3	57
IDV(18)	39	54	12	39	51	48	12	27
IDV(19)	3	6	33	6	36	9	6	3
IDV(20)	15	45	18	42	201	27	30	45
IDV(21)	120	27	63	6	252	741	3	3
Average	21	23	29	13	46	60	22	13

To further demonstrate the superiority of the MFPCA method, Figs. 7 and 8 plot the monitoring charts obtained by SPCA, DiPCA, DKPCA and MFPCA methods for IDV(5) and IDV(20), respectively. From Fig. 7, the proposed MFPCA-based process monitoring method combines the advantage of SPCA and DiPCA methods. For IDV(20), as can be seen from Fig. 8, there are quite a few missed-detection points in SPCA Q, DiPCA Q and DKPCA Q charts. However, in MFPCA Q chart, the missed-detection points are greatly reduced so that the FDR is as high as 91.38%. As shown in Table 5, the average FDRs of the MFPCA Q statistic (83.98%) is higher than SPCA Q statistic (74.30%), DiPCA Q statistic (78.34%) and DKPCA Q statistic (83.15%).

In Table 6, it can be seen that the average DDs of MFPCA method is either equal to or lower than the other three methods. Taking IDV(21) as an example, the monitoring charts are shown in Fig. 9. The MFPCA Q statistic can detect the disturbance earlier than other Q statistics. Although the SPCA T_n^2, DiPCA T_d^2 and DKPCA Q statistics are over 95% control limits after the 400th sample, many faulty samples are not detected from 450th sample to 600th sample. On the contrary, the MFPCA Q statistic can observe almost faulty samples after the 450th sample.

Table 7
TE process: the comparisons of the averaged computation time for online process monitoring.

Method	SPCA	DiPCA	DKPCA	MFPCA
Time(ms)	0.4808	0.0220	0.5226	0.4551

Table 7 compares the averaged computation time of the four methods. It can be observed that the computation cost of MFPCA (0.4551ms) is much lower than DKPCA (0.5226ms) for real-time process monitoring. It can be concluded that the proposed MFPCA-based process monitoring method can detect faults more effectively and timely, compared to other relevant methods.

6. Conclusions

In this paper, a multi-feature extraction technique based on principal component analysis is proposed for nonlinear dynamic process monitoring. The proposed MFPCA method takes the dynamic, linear and nonlinear characteristics of process data into consideration. Then multi-feature is extracted separately from process data through using several PCA-like methods, i.e., DiPCA, PCA and KPCA. Case studies show the extracted rich information improves the process monitoring performance by comparison with DiPCA and SPCA methods. And compared with DKPCA methods, the interpretability of the multi-feature extraction is improved. Based on existing results, the MFPCA-based fault identification and fault diagnosis method is worth further investigating in future work.

Declaration of Competing Interest

We declare that we have no financial interests or personal relationships that could have appeared to influence the work reported in this paper. This manuscript has not been published and is not under consideration for publication elsewhere. There is no conflict of interest when submitting manuscripts.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61703371, in part by the Social Development Project of Zhejiang Provincial Public Technology Research under Grant LGF19F030004, and in part by the Foundation of Key Laboratory of Advanced Process Control for Light Industry (Jiangnan University), Ministry of Education, P. R. China under Grant APCLI1805.

References

[1] K. Severson, P. Chaiwatanodom, R.D. Braatz, Perspectives on process monitoring of industrial systems, Annu. Rev. Control 42 (2016) 190–200, doi:10.1016/j.arcontrol.2016.09.001.

[2] S. Yin, H. Gao, O. Kaynak, Data-driven control and process monitoring for industrial applications-part i, IEEE Trans. Ind. Electron. 61 (11) (2014) 6356–6359, doi:10.1109/TIE.2014.2312885.

[3] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, Chemom. Intell. Lab. Syst. 171 (2017) 16–25, doi:10.1016/j.chemolab.2017.09.021.

[4] Q.P. He, J. Wang, Statistical process monitoring as a big data analytics tool for smart manufacturing, J. Process Control 67 (2018) 35–43, doi:10.1016/j.jprocont.2017.06.012.

[5] Y. Wang, Y. Si, B. Huang, Z. Lou, Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008-2017, Can. J. Chem. Eng. 96 (10) (2018) 2073–2085, doi:10.1002/cjce.23249.

[6] R. Fezai, M. Mansouri, O. Taouali, M.F. Harkat, N. Bouguila, Online reduced kernel principal component analysis for process monitoring, J. Process Control 61 (2018) 1–11, doi:10.1016/j.jprocont.2017.10.010.

[7] N. Li, S. Guo, Y. Wang, Weighted preliminary-summation-based principal component analysis for non-gaussian processes, Control Eng. Pract. 87 (2019) 122–132, doi:10.1016/j.conengprac.2019.03.015.

[8] K.E.S. Pilario, Y. Cao, M. Shafiee, Mixed kernel canonical variate dissimilarity analysis for incipient fault monitoring in nonlinear dynamic processes, Comput. Chem. Eng. 123 (2019) 143–154, doi:10.1016/j.compchemeng.2018.12.027.

[9] Q. Zhu, Q. Liu, S.J. Qin, Concurrent quality and process monitoring with canonical correlation analysis, J. Process Control 60 (2017) 95–103, doi:10.1016/j.jprocont.2017.06.017. DYCOPS-CAB 2016

[10] G. Wang, S. Yin, Quality-related fault detection approach based on orthogonal signal correction and modified PLS, IEEE Trans. Ind. Inf. 11 (2) (2015) 398–405, doi:10.1109/TII.2015.2396853.

[11] Z. Zhao, Q. Li, M. Huang, F. Liu, Concurrent PLS-based process monitoring with incomplete input and quality measurements, Comput. Chem. Eng. 67 (2014) 69–82, doi:10.1016/j.compchemeng.2014.03.022.

[12] W. Ku, R.H. Storer, C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis, Chemom. Intell. Lab. Syst. 30 (1) (1995) 179–196, doi:10.1016/0169-7439(95)00076-3.

[13] G. Li, S.J. Qin, D. Zhou, A new method of dynamic latent-variable modeling for process monitoring, IEEE Trans. Ind. Electron. 61 (11) (2014) 6438–6445, doi:10.1109/TIE.2014.2301761.

[14] Y. Dong, S.J. Qin, A novel dynamic pca algorithm for dynamic data modeling and process monitoring, J. Process Control 67 (2018) 1–11, doi:10.1016/j.jprocont.2017.05.002.

[15] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE J. 37 (2) (1991) 233–243, doi:10.1002/aic.690370209.

[16] D. Dong, T. McAvoy, Nonlinear principal component analysis-based on principal curves and neural networks, Comput. Chem. Eng. 20 (1) (1996) 65–78, doi:10.1016/0098-1354(95)00003-K.

[17] G. Wang, S. Yin, O. Kaynak, An LWPR-based data-driven fault detection approach for nonlinear process monitoring, IEEE Trans. Ind. Inf. 10 (4) (2014) 2016–2023, doi:10.1109/TII.2014.2341934.

[18] P. Wu, L. Guo, S. Lou, J. Gao, Local and global randomized principal component analysis for nonlinear process monitoring, IEEE Access 7 (2019) 25547–25562, doi:10.1109/ACCESS.2019.2901128.

[19] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319, doi:10.1162/089976698300017467.

[20] J.-M. Lee, C. Yoo, S.W. Choi, P.A. Vanrolleghem, I.-B. Lee, Nonlinear process monitoring using kernel principal component analysis, Chem. Eng. Sci. 59 (1) (2004) 223–234, doi:10.1016/j.ces.2003.09.012.

[21] L. Guo, P. Wu, J. Gao, S. Lou, Sparse kernel principal component analysis via sequential approach for nonlinear process monitoring, IEEE Access 7 (2019) 47550–47563, doi:10.1109/ACCESS.2019.2909986.

[22] M. Yao, H. Wang, On-line monitoring of batch processes using generalized additive kernel principal component analysis, J. Process Control 28 (2015) 56–72, doi:10.1016/j.jprocont.2015.02.007.

[23] X. Deng, X. Tian, S. Chen, C.J. Harris, Fault discriminant enhanced kernel principal component analysis incorporating prior fault information for monitoring nonlinear processes, Chemom. Intell. Lab. Syst. 162 (2017) 21–34, doi:10.1016/j.chemolab.2017.01.001.

[24] Q. Jiang, X. Yan, Nonlinear plant-wide process monitoring using mi-spectral clustering and Bayesian inference-based multiblock KPCA, J. Process Control 32 (2015) 38–50, doi:10.1016/j.jprocont.2015.04.014.

[25] J. Chen, C.-M. Liao, Dynamic process fault monitoring based on neural network and PCA, J. Process Control 12 (2) (2002) 277–289, doi:10.1016/S0959-1524(01)00027-0.

[26] K.-Y. Chen, Combining linear and nonlinear model in forecasting tourism demand, Expert Syst. Appl. 38 (8) (2011) 10368–10376, doi:10.1016/j.eswa.2011.02.049.

[27] H. Chen, B. Jiang, N. Lu, Z. Mao, Deep PCA based real-time incipient fault detection and diagnosis methodology for electrical drive in high-speed trains, IEEE Trans. Veh. Technol. 67 (6) (2018) 4819–4830, doi:10.1109/TVT.2018.2818538.

[28] X. Deng, X. Tian, S. Chen, C.J. Harris, Deep principal component analysis based on layerwise feature extraction and its application to nonlinear process monitoring, IEEE Trans. Control Syst. Technol. (2018) 1–15, doi:10.1109/TCST.2018.2865413.

[29] X. Deng, X. Tian, S. Chen, C.J. Harris, Nonlinear process fault diagnosis based on serial principal component analysis, IEEE Trans. Neural Netw. Learn. Syst. 29 (3) (2018) 560–572, doi:10.1109/TNNLS.2016.2635111.

[30] S.W. Choi, I.-B. Lee, Nonlinear dynamic process monitoring based on dynamic kernel PCA, Chem. Eng. Sci. 59 (24) (2004) 5897–5908, doi:10.1016/j.ces.2004.07.019.

[31] R.T. Samuel, Y. Cao, Improved kernel canonical variate analysis for process monitoring, in: Proceedings of the Twenty-First International Conference on Automation and Computing (ICAC), 2015, pp. 1–6, doi:10.1109/IConAC.2015.7313990.

[32] P.P. Odiowei, Y. Cao, Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations, IEEE Trans. Ind. Inf. 6 (1) (2010) 36–45, doi:10.1109/TII.2009.2032654.

[33] J. Downs, E. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (3) (1993) 245–255, doi:10.1016/0098-1354(93)80018-I.

[34] S. Yin, H. Luo, S.X. Ding, Real-time implementation of fault-tolerant control systems with performance optimization, IEEE Trans. Ind. Electron. 61 (5) (2014) 2402–2411, doi:10.1109/TIE.2013.2273477.