

Fault Discriminant Enhanced Kernel Principal Component Analysis Incorporating Prior Fault Information for Monitoring Nonlinear Processes

Xiaogang Deng^a, Xuemin Tian^a, Sheng Chen^{b,c,*}, Chris J. Harris^b.

^aCollege of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China

^bElectronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

^cKing Abdulaziz University, Jeddah 21589, Saudi Arabia

Abstract

Kernel principal component analysis (KPCA) based fault detection method, whose statistical model only utilizes normal operating data and ignores available prior fault information, may not provide the best fault detection performance for nonlinear process monitoring. In order to exploit available prior fault data to enhance fault detection performance, a fault discriminant enhanced KPCA (FDKPCA) method is proposed, which simultaneously monitors two types of data features, nonlinear kernel principal components (KPCs) and fault discriminant components (FDCs). More specifically, based on the normal operating data, KPCs are extracted by usual KPCA modeling, while with the normal operating data and prior fault data, FDCs are obtained by the kernel local-nonlocal preserving discriminant analysis (KLNPD). Monitoring statistics are constructed for both the KPCA and KLNPD sub-models. Moreover, Bayesian inference is employed to transform the corresponding monitoring statistics into fault probabilities, and the overall probability-based monitoring statistics are constructed by weighting the results of the two sub-models, which provides more effective on-line fault detection capability. Two case studies involving a simulated nonlinear system and a continuous stirred tank reactor demonstrate the superior fault detection performance of the proposed KLNPD scheme over the traditional KPCA method.

Key words: Fault detection, nonlinear process, prior fault information, kernel principal component analysis, fault discriminant enhanced kernel principal component analysis

1. Introduction

With the increasing attention on process safety and product quality, fault diagnosis technologies have become ever-increasingly important for industrial plants. As computer control systems are widely used in modern industry, abundant process data are collected and stored in historical database. Therefore, data-driven fault diagnosis methods have received significant interests from academics and engineers [1, 2, 3, 4, 5]. Many data-driven methods have been developed, including principal component analysis (PCA), independent component analysis (ICA) and Fisher discriminant analysis (FDA) [6, 7, 8]. Among all these methods, PCA is a popular one and it has been extensively studied.

To effectively deal with diverse and complicated characteristics of real industrial data, many extended PCA methods have been reported. For monitoring the process data with serial correlation, Ku *et al.* [9] proposed a dynamic PCA (DPCA) by considering lagged measurements, while Huang and Yan [10]

*Corresponding author. Email address: sqc@ecs.soton.ac.uk

further combined DPCA with dynamic ICA to monitor the data exhibiting both Gaussian and non-Gaussian dynamic features. Aiming at the industrial systems with multiple operating modes, improved PCA methods were studied, which include multiple PCA [11], local neighborhood standardization based PCA [12] and local neighborhood similarity analysis [13]. To reduce the modeling complexity of plant-wide process, Ge and Song [14] proposed a distributed PCA method to divide variables into different sub-blocks for sub-PCA modeling, while Jiang and Yan [15] developed a mutual information based multi-block PCA method. To cope with multiscale characteristics of process data, multiscale PCA methods were derived by combining PCA with wavelet analysis [16, 17] or ensemble empirical mode decomposition [18].

All the aforementioned PCA methods assume the linearity of the monitored system. As nonlinearity is common in industrial processes, nonlinear PCA has become an attractive research topic over the past two decades. Classic nonlinear PCA methods include neural network PCA [19] and kernel PCA (KPCA) [20]. Because of its simplicity and effectiveness, KPCA gains most attention in the research field of nonlinear PCA. KPCA was firstly proposed by Schölkopf *et al.* [20] and then was introduced for nonlinear process monitoring by Lee *et al.* [21]. Later, Lee *et al.* [22] built multiway KPCA for batch process fault detection. To distinguish the importance of different kernel principal components (KPCs), Jiang and Yan [23] developed a weighted KPCA. Aiming at nonlinear nonstationary processes, Khediri *et al.* [24] built a variable window adaptive KPCA. Considering the local data structure, Deng *et al.* [25] proposed a modified KPCA assisted by local structure analysis for nonlinear fault detection. There are also many other important studies on KPCA which can be found in the literature, e.g., [26, 27, 28, 29].

Although KPCA has achieved successful applications in many industrial cases, there are some critical issues deserving further research. One important problem is related to the modeling assumption of KPCA. As an unsupervised approach, all the existing KPCA based fault detection methods only utilize normal operation data for statistical modeling. For a real industrial process, even though it may be impossible to collect all types of fault data, there are always some classes of known fault data available in the historical database. The current KPCA approach completely ignore these prior fault data which can in fact provides some very valuable and powerful fault discriminant information for aiding on-line fault detection. Therefore, how to integrate prior fault discriminant information to enhance fault detection performance of KPCA is a very valuable research topic.

To extract fault discriminant information, a usual strategy is to construct fault classifiers using discriminant analysis techniques. FDA is a most widely used discriminant analysis method for fault classification [30, 31, 32]. To improve classification performance, many improved FDA methods have been derived, including semi-supervised FDA [33], and kernel FDA [34, 35]. Recently, based on local data structure analysis, localized FDA methods have been developed [36, 37, 38]. Furthermore, by utilizing nonlocal structure information, local and nonlocal preserving discriminant analysis (LNPD) methods were developed to improve FDA based fault discrimination [39, 40, 41]. All the above-mentioned discriminant analysis methods focus

on fault classification tasks. However, in this paper, we apply discriminant analysis technique in a very different way, specifically, to assist KPCA based fault detection method by incorporating a small amount of prior fault data available.

Against the above background, in this paper, a fault discriminant enhanced KPCA (FDKPCA) is proposed to monitor nonlinear processes by incorporating prior fault information. Fundamentally different from the traditional KPCA based fault detection method which performs unsupervised modeling on normal operating data, our FDKPCA applies both unsupervised and supervised modeling techniques to build the two sub-models. For normal operating data, a KPCA sub-model is developed to extract nonlinear principal components (PCs) or KPCs, which is a usual unsupervised modeling procedure. However, a kernel LNPDA (KLNPDA) sub-model is also built based on both normal operating data and available prior fault data to obtain the fault discriminant components (FDCs), which is a supervised modeling strategy. The monitoring statistics provided by the two sub-models are transformed into fault probabilities by Bayesian inference and the overall probability-based monitoring statistics are constructed by weighting the results of the two sub-models, which provides more effective on-line fault detection capability. To our best knowledge, our work is the first to effectively integrate KPCA and KLNPDA together for better fault detection, by exploiting available prior fault information.

The remainder of this paper is structured as follows. In Section 2, the KPCA method is briefly reviewed. In Section 3, our proposed FDKPCA method is detailed. The monitoring scheme based on the FDKPCA is also presented in this section. The proposed method is then evaluated in Section 4, in comparison with the standard KPCA method, using two case studies involving a simulated nonlinear system and a benchmark continuous stirred tank reactor (CSTR) system. Finally, the conclusions are drawn in Section 5.

2. KPCA Based Fault Detection Method

KPCA is popular in the field of process monitoring and fault diagnosis because of its excellent nonlinear feature extraction ability. Its idea is to firstly map the original nonlinear data onto a linear high-dimensional feature space and then to perform linear PCA in the feature space. Specifically, given a training data matrix $\mathbf{X}_n \in \mathbb{R}^{N \times M}$, where M is the number of measured variables and N is the number of samples while the subscript n represents normal operation status, a nonlinear mapping projects the original data space onto a new high-dimensional feature space as: $\Phi: \mathbb{R}^M \rightarrow \mathcal{F}$. In the feature space, the data matrix \mathbf{X}_n is mapped onto $\Phi(\mathbf{X}_n) \in \mathbb{R}^N \times \mathcal{F}$, and its linear PCA decomposition can be expressed as

$$\Phi(\mathbf{X}_n) = \sum_{i=1}^K \mathbf{t}_i \mathbf{p}_i^T + \mathbf{E}, \quad (1)$$

where $\Phi(\mathbf{X}_n)$ has been mean-centered, $\mathbf{t}_i \in \mathbb{R}^N$ is the i -th score vector of $\Phi(\mathbf{X}_n)$, also called the i -th PC, and $\mathbf{p}_i \in \mathcal{F}$ is the corresponding loading vector, while K denotes the number of KPCs retained in the KPCA model and $\mathbf{E} \in \mathbb{R}^N \times \mathcal{F}$ is the residual matrix.

The loading vector in (1) can be solved by an eigenvalue decomposition of the covariance matrix of $\Phi(\mathbf{X}_n)$ as

$$\frac{1}{N-1} \Phi^T(\mathbf{X}_n) \Phi(\mathbf{X}_n) \mathbf{p}_i = \lambda_i \mathbf{p}_i, \quad (2)$$

where λ_i is the i -th eigenvalue of the covariance matrix $\frac{1}{N-1} \Phi^T(\mathbf{X}_n) \Phi(\mathbf{X}_n)$. There exists a coefficient vector $\boldsymbol{\alpha}_i \in \mathbb{R}^N$ satisfying [20, 21]

$$\mathbf{p}_i = \Phi^T(\mathbf{X}_n) \boldsymbol{\alpha}_i. \quad (3)$$

Combining (2) and (3) leads to the expression

$$\frac{1}{N-1} \Phi^T(\mathbf{X}_n) \Phi(\mathbf{X}_n) \Phi^T(\mathbf{X}_n) \boldsymbol{\alpha}_i = \lambda_i \Phi^T(\mathbf{X}_n) \boldsymbol{\alpha}_i, \quad (4)$$

which can further be formulated as [20]

$$\frac{1}{N-1} \Phi(\mathbf{X}_n) \Phi^T(\mathbf{X}_n) \Phi(\mathbf{X}_n) \Phi^T(\mathbf{X}_n) \boldsymbol{\alpha}_i = \lambda_i \Phi(\mathbf{X}_n) \Phi^T(\mathbf{X}_n) \boldsymbol{\alpha}_i. \quad (5)$$

Let us express the training data matrix as $\mathbf{X}_n = [\mathbf{x}_n(1) \ \mathbf{x}_n(2) \ \cdots \ \mathbf{x}_n(N)]^T$, where $\mathbf{x}_n(i) \in \mathbb{R}^M$ for $1 \leq i \leq N$ denotes the i -th sample data vector. To avoid explicitly defining a high-dimensional nonlinear mapping, a kernel matrix is introduced as $\mathbf{K}_n = \Phi(\mathbf{X}_n) \Phi^T(\mathbf{X}_n) \in \mathbb{R}^{N \times N}$, whose (i, j) -th element is $\mathbf{K}_n(i, j) = (\Phi(\mathbf{x}_n(i)))^T \Phi(\mathbf{x}_n(j))$. By this kernel trick [20], the inner product of two feature vectors in the feature space can be computed by a kernel function as

$$(\Phi(\mathbf{x}_n(i)))^T \Phi(\mathbf{x}_n(j)) = \ker(\mathbf{x}_n(i), \mathbf{x}_n(j)). \quad (6)$$

In this paper, we use the well-known Gaussian kernel function $\ker(\mathbf{x}_n(i), \mathbf{x}_n(j)) = \exp(-\|\mathbf{x}_n(i) - \mathbf{x}_n(j)\|^2 / \sigma)$, where $\sigma > 0$ is the kernel width parameter. With the introduction of a kernel function, explicit use of a high-dimensional nonlinear mapping $\Phi(\cdot)$ is avoided and (5) is expressed as

$$\mathbf{K}_n \mathbf{K}_n \boldsymbol{\alpha}_i = (N-1) \lambda_i \mathbf{K}_n \boldsymbol{\alpha}_i. \quad (7)$$

The solutions of (7) are obtained by solving the following problem

$$\mathbf{K}_n \boldsymbol{\alpha}_i = (N-1) \lambda_i \boldsymbol{\alpha}_i \quad (8)$$

for the \bar{N} non-zero eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{\bar{N}} > 0$ with the corresponding eigenvectors α_i for $1 \leq i \leq \bar{N}$, where $\bar{N} \leq N$. Noted that to ensure the coefficient vector \mathbf{p}_i to be a unity-norm vector, α_i meets the following constraint

$$\alpha_i^T \mathbf{K}_n \alpha_i = 1. \quad (9)$$

The schematic of this KPCA modeling for process monitoring application is illustrated in Fig. 1.

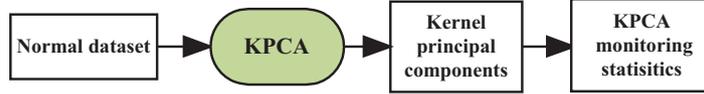


Figure 1: The schematic of KPCA modeling for fault detection application.

For a testing data vector $\mathbf{x}_t \in \mathbb{R}^M$ at the time instant t , its i -th KPC can be calculated as

$$t_i = (\Phi(\mathbf{x}_t))^T \mathbf{p}_i = (\Phi(\mathbf{x}_t))^T \Phi^T(\mathbf{X}_n) \alpha_i = \mathbf{k}_n^T \alpha_i, \quad (10)$$

where $\mathbf{k}_n = [\ker(\mathbf{x}_n(1), \mathbf{x}_t) \dots \ker(\mathbf{x}_n(N), \mathbf{x}_t)]^T = \Phi(\mathbf{X}_n) \Phi(\mathbf{x}_t)$. For process monitoring, two monitoring statistics T^2 and Q are usually constructed as

$$T^2 = [t_1 \ t_2 \ \dots \ t_K] \mathbf{A}^{-1} [t_1 \ t_2 \ \dots \ t_K]^T, \quad (11)$$

$$Q = \sum_{i=1}^{\bar{N}} t_i^2 - \sum_{i=1}^K t_i^2, \quad (12)$$

where t_i for $1 \leq i \leq \bar{N}$ are computed by (10) and \mathbf{A} is the $K \times K$ diagonal matrix with the eigenvalues λ_i for $1 \leq i \leq K$ as its diagonal elements. How to select the number of retained KPCs K can be found in [6, 21]. In this paper, we opt for the average eigenvalue method [21], which retains the KPCs whose eigenvalues are larger than the average eigenvalue.

3. The proposed method

As can be seen from Fig. 2, our proposed FDKPCA method integrates the KPCA and KLNPPDA sub-models. Since the KPCA modeling was discussed in the previous section, we only need to detail the proposed KLNPPDA modeling as well as the Bayesian inference strategy to construct the final probability-based monitoring statistics for fault detection.

3.1. Fault Discriminant Modeling Using KLNPPDA

We begin by discussing the principle of linear LNPDA. LNPDA originates from local and nonlocal preserving projection (LNPP), which combines the ideas of local preserving projection and nonlocal preserving projection [42, 43]. Basically, LNPP finds the low dimensional projection of a data matrix such that the distances of local neighborhood points are as small as possible while the distances of nonlocal neighborhood

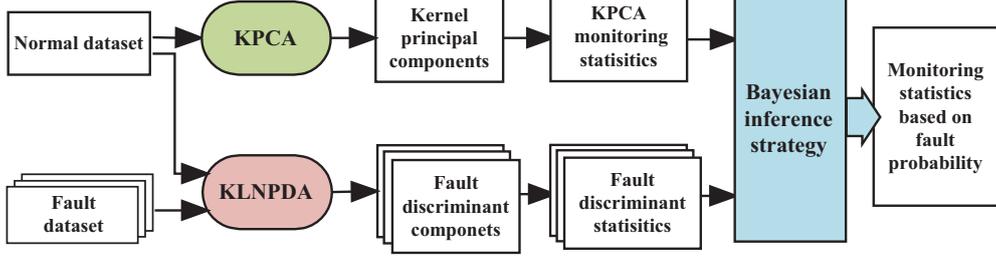


Figure 2: The schematic of FDKPCA modeling for fault detection application.

points are as large as possible. If we define local neighborhood points as the points from a same class and nonlocal neighborhood points as the points from different classes, then LNPP becomes a supervised method known as LNPDA [39, 40, 41]. Therefore, LNPDA is performed to find the data projection direction so that the points of different classes are far apart while the points of a same class are close.

To cope with the nonlinearity of industrial processes, we propose to use the kernel version of LNPDA, referred to as KLNPDPA, for obtaining FDCs. For training a fault discriminant model, we assume that there are the normal operation dataset $\mathbf{X}_n \in \mathbb{R}^{N \times M}$ and one class of prior fault dataset denoted as $\mathbf{X}_f \in \mathbb{R}^{F \times M}$, where the known fault dataset has F sample points each having the same M variables. Then an augmented data matrix is constructed as $\mathbf{X}_{nf} = [\mathbf{X}_n^T \ \mathbf{X}_f^T]^T = [\mathbf{x}_{nf}(1) \ \mathbf{x}_{nf}(2) \ \cdots \ \mathbf{x}_{nf}(N_f)]^T \in \mathbb{R}^{N_f \times M}$, where $N_f = N + F$ is the total number of samples in the augmented training dataset.

Each data vector $\mathbf{x}_{nf}(i) \in \mathbb{R}^M$ is firstly mapped onto a high dimensional feature space $\Phi(\mathbf{x}_{nf}(i)) \in \mathcal{F}$. Then in the feature space, a linear transformation vector $\mathbf{q} \in \mathcal{F}$ is sought to obtain the projection $y(i) = (\Phi(\mathbf{x}_{nf}(i)))^T \mathbf{q}$ for $1 \leq i \leq N_f$ so that the points of a same class stay as close as possible while the points of different classes are as far apart as possible. The former is a local projection optimization problem while the latter is a nonlocal projection optimization problem.

The local projection optimization can be defined as [42]

$$\begin{aligned} \min_{\mathbf{q}} J_{Local}(\mathbf{q}) &= \min_{\mathbf{q}} \frac{1}{2} \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} (y(i) - y(j))^2 w_l(i, j) \\ &= \min_{\mathbf{q}} \frac{1}{2} \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} \left((\Phi(\mathbf{x}_{nf}(i)))^T \mathbf{q} - (\Phi(\mathbf{x}_{nf}(j)))^T \mathbf{q} \right)^2 \times w_l(i, j), \end{aligned} \quad (13)$$

where \mathbf{q} is subject to the unity-norm constraint of $\mathbf{q}^T \mathbf{q} = 1$, while $w_l(i, j)$ is a weighting parameter for the distance between $y(i)$ and $y(j)$, which puts a high value for the data pair from a same class and a low value for the data pair from different classes. One simple way to set this weighting parameter is according to

$$w_l(i, j) = \begin{cases} 1, & \text{if } \mathbf{x}_{nf}(i), \mathbf{x}_{nf}(j) \in \mathbf{X}_n, \\ 1, & \text{if } \mathbf{x}_{nf}(i), \mathbf{x}_{nf}(j) \in \mathbf{X}_f, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Subject to the constraint $\mathbf{q}^T \mathbf{q} = 1$, the optimization (13) can be reformulated as

$$\begin{aligned} \min_{\mathbf{q}} J_{Local}(\mathbf{q}) = & \min_{\mathbf{q}} \sum_{i=1}^{N_f} \mathbf{q}^T \Phi(\mathbf{x}_{nf}(i)) d_l(i, i) (\Phi(\mathbf{x}_{nf}(i)))^T \mathbf{q} \\ & - \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} \mathbf{q}^T \Phi(\mathbf{x}_{nf}(i)) w_l(i, j) (\Phi(\mathbf{x}_{nf}(j)))^T \mathbf{q}, \end{aligned} \quad (15)$$

where $d_l(i, i) = \sum_{j=1}^{N_f} w_l(i, j)$. In matrix form, the optimization (15) can be rewritten as

$$\min_{\mathbf{q}} J_{Local}(\mathbf{q}) = \min_{\mathbf{q}} \mathbf{q}^T \Phi^T(\mathbf{X}_{nf}) (\mathbf{D}_l - \mathbf{W}_l) \Phi(\mathbf{X}_{nf}) \mathbf{q} = \min_{\mathbf{q}} \mathbf{q}^T \Phi^T(\mathbf{X}_{nf}) \mathbf{L}_l \Phi(\mathbf{X}_{nf}) \mathbf{q}, \quad (16)$$

where \mathbf{D}_l is the $N_f \times N_f$ diagonal matrix with $d_l(i, i)$ as its i -th diagonal element, \mathbf{W}_l is the $N_f \times N_f$ matrix whose (i, j) -th element is $w_l(i, j)$, and the weighting matrix $\mathbf{L}_l = \mathbf{D}_l - \mathbf{W}_l$.

Similarly, the nonlocal projection optimization can be expressed as [43]

$$\begin{aligned} \max_{\mathbf{q}} J_{NonLocal}(\mathbf{q}) = & \max_{\mathbf{q}} \frac{1}{2} \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} (y(i) - y(j))^2 w_{nl}(i, j) \\ = & \max_{\mathbf{q}} \frac{1}{2} \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} \left((\Phi(\mathbf{x}_{nf}(i)))^T \mathbf{q} - (\Phi(\mathbf{x}_{nf}(j)))^T \mathbf{q} \right)^2 w_{nl}(i, j), \end{aligned} \quad (17)$$

subject to the constraint $\mathbf{q}^T \mathbf{q} = 1$, where $w_{nl}(i, j)$ is a weighting parameter, which imposes a high value on the data pair from different classes and a low value for the data pair from a same class, according to

$$w_{nl}(i, j) = \begin{cases} 0, & \text{if } \mathbf{x}_{nf}(i), \mathbf{x}_{nf}(j) \in \mathbf{X}_n, \\ 0, & \text{if } \mathbf{x}_{nf}(i), \mathbf{x}_{nf}(j) \in \mathbf{X}_f, \\ 1, & \text{otherwise.} \end{cases} \quad (18)$$

The optimization (17) can be formulated in matrix form as

$$\max_{\mathbf{q}} J_{NonLocal}(\mathbf{q}) = \max_{\mathbf{q}} \mathbf{q}^T \Phi^T(\mathbf{X}_{nf}) (\mathbf{D}_{nl} - \mathbf{W}_{nl}) \Phi(\mathbf{X}_{nf}) \mathbf{q} = \max_{\mathbf{q}} \mathbf{q}^T \Phi^T(\mathbf{X}_{nf}) \mathbf{L}_{nl} \Phi(\mathbf{X}_{nf}) \mathbf{q}, \quad (19)$$

where \mathbf{W}_{nl} is the $N_f \times N_f$ matrix whose (i, j) -th element is $w_{nl}(i, j)$, \mathbf{D}_{nl} is the $N_f \times N_f$ diagonal matrix whose i -th diagonal element $d_{nl}(i, i) = \sum_{j=1}^{N_f} w_{nl}(i, j)$, and the weighting matrix $\mathbf{L}_{nl} = \mathbf{D}_{nl} - \mathbf{W}_{nl}$.

By combing the optimizations (16) and (19), the KLNPDPA optimization problem can be expressed as

$$\max_{\mathbf{q}} \frac{J_{NonLocal}(\mathbf{q})}{J_{Local}(\mathbf{q})} = \max_{\mathbf{q}} \frac{\mathbf{q}^T \Phi^T(\mathbf{X}_{nf}) \mathbf{L}_{nl} \Phi(\mathbf{X}_{nf}) \mathbf{q}}{\mathbf{q}^T \Phi^T(\mathbf{X}_{nf}) \mathbf{L}_l \Phi(\mathbf{X}_{nf}) \mathbf{q}}, \quad (20)$$

subject to the constraint $\mathbf{q}^T \mathbf{q} = 1$. Similar to KPCA, the projection vector \mathbf{q} can be spanned using the training samples as

$$\mathbf{q} = \sum_{j=1}^{N_f} \beta_j \Phi(\mathbf{x}_{nf}(j)) = \Phi^T(\mathbf{X}_{nf}) \boldsymbol{\beta} \quad (21)$$

Where $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ \cdots \ \beta_{N_f}]^T$. Substituting (21) into (20) results in the following optimization

$$\max_{\mathbf{q}} \frac{J_{NonLocal}(\mathbf{q})}{J_{Local}(\mathbf{q})} = \max_{\mathbf{q}} \frac{\boldsymbol{\beta}^T \Phi(\mathbf{X}_{nf}) \Phi^T(\mathbf{X}_{nf}) \mathbf{L}_{nl} \Phi(\mathbf{X}_{nf}) \Phi^T(\mathbf{X}_{nf}) \boldsymbol{\beta}}{\boldsymbol{\beta}^T \Phi(\mathbf{X}_{nf}) \Phi^T(\mathbf{X}_{nf}) \mathbf{L}_l \Phi(\mathbf{X}_{nf}) \Phi^T(\mathbf{X}_{nf}) \boldsymbol{\beta}}. \quad (22)$$

Kernel trick is again applied to avoid the difficulty of explicitly defining the nonlinear high-dimensional mapping. Denote $\mathbf{K}_{nf} = \Phi(\mathbf{X}_{nf}) \Phi^T(\mathbf{X}_{nf})$ with its (i, j) -th element $\mathbf{K}_{nf}(i, j) = (\Phi(\mathbf{x}_{nf}(i)))^T \Phi(\mathbf{x}_{nf}(j)) = \ker(\mathbf{x}_{nf}(i), \mathbf{x}_{nf}(j))$. Then we have the optimization

$$\max_{\boldsymbol{\beta}} \frac{J_{NonLocal}(\boldsymbol{\beta})}{J_{Local}(\boldsymbol{\beta})} = \max_{\boldsymbol{\beta}} \frac{\boldsymbol{\beta}^T \mathbf{K}_{nf} \mathbf{L}_{nl} \mathbf{K}_{nf} \boldsymbol{\beta}}{\boldsymbol{\beta}^T \mathbf{K}_{nf} \mathbf{L}_l \mathbf{K}_{nf} \boldsymbol{\beta}}. \quad (23)$$

To ensure a unique non-zero solution, the constraint

$$\boldsymbol{\beta}^T \mathbf{K}_{nf} \boldsymbol{\beta} = 1 \quad (24)$$

should be imposed. To solve the optimization (23), we resort to the following generalized eigen-value decomposition

$$\mathbf{K}_{nf} \mathbf{L}_{nl} \mathbf{K}_{nf} \boldsymbol{\beta} = \gamma \mathbf{K}_{nf} \mathbf{L}_l \mathbf{K}_{nf} \boldsymbol{\beta}. \quad (25)$$

Solving (25) subject to the constraint (24) yields the \bar{N}_f non-zero eigenvalues $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_{\bar{N}_f}$ with the corresponding eigenvectors $\boldsymbol{\beta}_i$ for $1 \leq i \leq \bar{N}_f$, where $\bar{N}_f \leq N_f$.

For a test vector \mathbf{x}_t at the time instant t , its discriminant component is given by

$$y_i = (\Phi(\mathbf{x}_t))^T \Phi^T(\mathbf{X}_{nf}) \boldsymbol{\beta}_i = \mathbf{k}_{nf}^T \boldsymbol{\beta}_i, \quad (26)$$

where $\mathbf{k}_{nf} = [k_{nf}(1) \ k_{nf}(2) \ \cdots \ k_{nf}(N_f)]^T \in \mathbb{R}^{N_f}$ and $k_{nf}(j) = \ker(\Phi(\mathbf{x}_t), \Phi(\mathbf{x}_{nf}(j)))$ for $1 \leq j \leq N_f$. For fault detection, two fault discriminant statistics are constructed as

$$T_f^2 = [y_1 \ y_2 \ \cdots \ y_{K_f}] \boldsymbol{\Gamma}^{-1} [y_1 \ y_2 \ \cdots \ y_{K_f}]^T, \quad (27)$$

$$Q_f = [y_{K_f+1} \ \cdots \ y_{\bar{N}_f}] \boldsymbol{\Omega}^{-1} [y_{K_f+1} \ \cdots \ y_{\bar{N}_f}]^T, \quad (28)$$

where $[y_1 \ y_2 \ \cdots \ y_{K_f}]^T$ contains the principal FDCs corresponding to the first K_f largest eigenvectors and $[y_{K_f+1} \ \cdots \ y_{\bar{N}_f}]^T$ contains the remaining $\bar{N}_f - K_f$ minor FDCs, while $\boldsymbol{\Gamma}$ is the covariance matrix of the

principal FDCs and $\mathbf{\Omega}$ is the covariance matrix of the remaining minor FDCs, both calculated based on the training data \mathbf{X}_{nf} . Similarly to KPCA modeling, the value of K_f is selected by the average eigenvalue method [21].

The above analysis is based on the pair of the normal operating dataset and a single prior fault dataset. In fact, if there are C classes of prior fault data $\{\mathbf{X}_f^{(c)}, c = 1, 2, \dots, C\}$ available, the procedure can be executed for each pair of the normal dataset and a class of fault dataset, leading to the group of C fault discriminant statistics denoted as $\{T_f^{2(c)}, Q_f^{(c)}, c = 1, 2, \dots, C\}$.

3.2. Monitoring Statistics Based on Bayesian Inference

With the proposed FDKPCA method, we obtain two kinds of monitoring statistics from the KPCA and KLNPDA sub-models. For notational convenience, the KPCA monitoring statistics T^2 and Q are also denoted as $T_f^{2(0)}$ and $Q_f^{(0)}$ and, therefore, all the monitoring statistics can be denoted by $T_f^{2(c)}$ and $Q_f^{(c)}$ for $c = 0, 1, 2, \dots, C$.

Under the normal operation, all the monitoring statistics should be smaller than their respective confidence limits, which are denoted by $T_{f,\text{lim}}^{2(c)}$ and $Q_{f,\text{lim}}^{(c)}$. In most of the existing KPCA studies, certain statistical distributions of observed data are assumed for confidence limit computation [21, 22]. However, industrial data have highly complex characteristics and they may not obey these assumed distributions. Therefore, we use a data-driven confidence limit computation method based on the well-known kernel density estimation (KDE) method [25, 44, 45, 46]. Specifically, the normal operation data are first projected onto the statistical models and the monitoring statistics $T_f^{2(c)}$ and $Q_f^{(c)}$ are computed. Then the density function is estimated for each statistic by the KDE method. Finally, with the given significance level δ , the confidence limit is determined for every statistic by finding the point occupying the $1 - \delta$ area of the estimated density function. In this paper, the significance level is set to $\delta = 5\%$, and thus the 95% confidence limit is applied for each statistic.

Up to now, the KPCA sub-model and fault discriminant sub-models are built and the monitoring statistics $T_f^{2(c)}$ and $Q_f^{(c)}$ for $c = 0, 1, 2, \dots, C$ are designed for process monitoring. As different monitoring statistics are provided, how to combine them to generate an overall decision is critically important. We propose to use the Bayesian inference to turn each monitoring statistic into a fault probability. Then all the probability values from different sub-models are combined to construct the probability-based monitoring statistics with a weighting strategy for enhancing fault detection performance.

According to Bayesian inference [47, 48], the monitoring statistics $T_f^{2(c)}$ and $Q_f^{(c)}$ can be transformed into the posterior fault probabilities $P_{T^2}^{(c)}(f|\mathbf{x}_t)$ and $P_Q^{(c)}(f|\mathbf{x}_t)$, computed as

$$P_{T^2}^{(c)}(f|\mathbf{x}_t) = \frac{P_{T^2}^{(c)}(\mathbf{x}_t|f)P_{T^2}^{(c)}(f)}{P_{T^2}^{(c)}(\mathbf{x}_t)}, c = 0, 1, \dots, C, \quad (29)$$

$$P_Q^{(c)}(f|\mathbf{x}_t) = \frac{P_Q^{(c)}(\mathbf{x}_t|f)P_Q^{(c)}(f)}{P_Q^{(c)}(\mathbf{x}_t)}, \quad c = 0, 1, \dots, C, \quad (30)$$

where $P_{T^2}^{(c)}(f)$ and $P_Q^{(c)}(f)$ are the prior fault probabilities equivalent to the significance level δ , $P_{T^2}^{(c)}(\mathbf{x}_t|f)$ and $P_Q^{(c)}(\mathbf{x}_t|f)$ are the occurrence probabilities of \mathbf{x}_t under fault condition, which are defined as

$$P_{T^2}^{(c)}(\mathbf{x}_t|f) = \exp\left(-T_{f,\text{lim}}^{2(c)}/T_f^{2(c)}\right), \quad (31)$$

$$P_Q^{(c)}(\mathbf{x}_t|f) = \exp\left(-Q_{f,\text{lim}}^{(c)}/Q_f^{(c)}\right), \quad (32)$$

while $P_{T^2}^{(c)}(\mathbf{x}_t)$ and $P_Q^{(c)}(\mathbf{x}_t)$ are the occurrence probabilities of \mathbf{x}_t , which are computed according to

$$P_{T^2}^{(c)}(\mathbf{x}_t) = P_{T^2}^{(c)}(\mathbf{x}_t|f)P_{T^2}^{(c)}(f) + P_{T^2}^{(c)}(\mathbf{x}_t|n)P_{T^2}^{(c)}(n), \quad (33)$$

$$P_Q^{(c)}(\mathbf{x}_t) = P_Q^{(c)}(\mathbf{x}_t|f)P_Q^{(c)}(f) + P_Q^{(c)}(\mathbf{x}_t|n)P_Q^{(c)}(n). \quad (34)$$

In (33) and (34), $P_{T^2}^{(c)}(n)$ and $P_Q^{(c)}(n)$ are the prior normal probabilities equivalent to the confidence level $1-\delta$, while $P_{T^2}^{(c)}(\mathbf{x}_t|n)$ and $P_Q^{(c)}(\mathbf{x}_t|n)$ are the occurrence probabilities of \mathbf{x}_t under normal operating condition, defined as

$$P_{T^2}^{(c)}(\mathbf{x}_t|n) = \exp\left(-T_f^{2(c)}/T_{f,\text{lim}}^{2(c)}\right), \quad (35)$$

$$P_Q^{(c)}(\mathbf{x}_t|n) = \exp\left(-Q_f^{(c)}/Q_{f,\text{lim}}^{(c)}\right). \quad (36)$$

Two probability-based monitoring statistics, PT^2 and PQ , are then constructed by weighting all the corresponding posterior fault probabilities according to

$$PT^2 = \frac{1}{\sum_{c=0}^C w_{T^2}^{(c)}} \sum_{c=0}^C w_{T^2}^{(c)} P_{T^2}^{(c)}(f|\mathbf{x}_t) = \sum_{c=0}^C \bar{w}_{T^2}^{(c)} P_{T^2}^{(c)}(f|\mathbf{x}_t), \quad (37)$$

$$PQ = \frac{1}{\sum_{c=0}^C w_Q^{(c)}} \sum_{c=0}^C w_Q^{(c)} P_Q^{(c)}(f|\mathbf{x}_t) = \sum_{c=0}^C \bar{w}_Q^{(c)} P_Q^{(c)}(f|\mathbf{x}_t), \quad (38)$$

where $w_{T^2}^{(c)}$ and $w_Q^{(c)}$ are the weighting factors, while $\bar{w}_{T^2}^{(c)} = w_{T^2}^{(c)}/\sum_{c=0}^C w_{T^2}^{(c)}$ and $\bar{w}_Q^{(c)} = w_Q^{(c)}/\sum_{c=0}^C w_Q^{(c)}$ are the normalized weighting factors.

We now discuss how to design these weighting factors appropriately. Note that the weighting factors should ensure that the KPCA sub-model can always function well while the fault discriminant sub-models can provide useful fault information to improve the monitoring performance. For the KPCA sub-model, we

design its weighting factors as

$$w_{T^2}^{(0)} = \text{sat} \left(\exp \left((T_f^{2(0)} - T_{f,\text{lim}}^{2(0)}) / T_{f,\text{lim}}^{2(0)} \right) \right), \quad (39)$$

$$w_Q^{(0)} = \text{sat} \left(\exp \left((Q_f^{(0)} - Q_{f,\text{lim}}^{(0)}) / Q_{f,\text{lim}}^{(0)} \right) \right), \quad (40)$$

where $\text{sat}(\cdot)$ represents the saturation function defined by

$$\text{sat}(z) = \begin{cases} 0, & z < 0, \\ z, & 0 \leq z \leq z_{\max}, \\ z_{\max}, & z > z_{\max}. \end{cases} \quad (41)$$

In our case studies, we empirically set $z_{\max} = 200$.

Since the fault discriminant statistics should provide more information only in the case of fault occurrence, we set the weighting factors $w_{T^2}^{(c)}$ and $w_Q^{(c)}$ for $1 \leq c \leq C$ according to

$$w_{T^2}^{(c)} = \begin{cases} \text{sat} \left(\exp \left(\frac{T_f^{2(c)} - T_{f,\text{lim}}^{2(c)}}{T_{f,\text{lim}}^{2(c)}} \right) \right), & \text{if } P_{T^2}^{(c)}(f|\mathbf{x}_t) > \delta \ \& \ \bar{P}_{T^2}^{(c)}(f|\mathbf{x}_t) > \delta, \\ 0, & \text{otherwise,} \end{cases} \quad (42)$$

$$w_Q^{(c)} = \begin{cases} \text{sat} \left(\exp \left(\frac{Q_f^{(c)} - Q_{f,\text{lim}}^{(c)}}{Q_{f,\text{lim}}^{(c)}} \right) \right), & \text{if } P_Q^{(c)}(f|\mathbf{x}_t) > \delta \ \& \ \bar{P}_Q^{(c)}(f|\mathbf{x}_t) > \delta, \\ 0, & \text{otherwise,} \end{cases} \quad (43)$$

where

$$\bar{P}_{T^2}^{(c)}(f|\mathbf{x}_t) = \frac{1}{L} \sum_{i=t-L+1}^t P_{T^2}^{(c)}(f|\mathbf{x}_i), \quad (44)$$

$$\bar{P}_Q^{(c)}(f|\mathbf{x}_t) = \frac{1}{L} \sum_{i=t-L+1}^t P_Q^{(c)}(f|\mathbf{x}_i), \quad (45)$$

are the corresponding mean posterior fault probabilities over the L test samples $\{\mathbf{x}_{t-L+1}, \mathbf{x}_{t-L+2}, \dots, \mathbf{x}_t\}$. In (42) and (43), $P_{T^2}^{(c)}(f|\mathbf{x}_t) > \delta$ and $P_Q^{(c)}(f|\mathbf{x}_t) > \delta$ mean that the current sample indicates a fault, while $\bar{P}_{T^2}^{(c)}(f|\mathbf{x}_t) > \delta$ and $\bar{P}_Q^{(c)}(f|\mathbf{x}_t) > \delta$ mean that the average probability of the past L samples indicates a fault. When both these two conditions are satisfied, the weighting factors will become non-zero values and the fault discriminant model is therefore activated to contribute to process monitoring. Otherwise, the fault discriminant model is deactivated. In our study, the value of L is empirically selected as $L = 6$.

With (37) and (38), two probability-based monitoring statistics are constructed for fault detection. Different from traditional monitoring statistics, they indicates fault occurrence probabilities. Hence, if $PT^2 \leq \delta$ and $PQ \leq \delta$, the process is under normal operation status. Otherwise, a fault is detected.

3.3. FDKPCA Based Process Monitoring Procedure

We are now ready to summarize our proposed FDKPCA based process monitoring procedure. Similar to other data-driven methods, the FDKPCA based process monitoring involves two stages: offline model training and online fault detection. In the offline model training stage, the training dataset including the normal operating data and some priori fault data are used to build the FDKPCA model. During the online fault detection stage, new data are projected onto the FDKPCA model and the probability-based monitoring statistics PT^2 and PQ are computed to judge the process operation status. More specifically, the two stages of the FDKPCA based process monitoring are as follows.

Offline model training stage

- 1) Collect normal operating data, divide them into the normal training dataset $\mathbf{X}_n = \mathbf{X}_f^{(0)}$ and validation dataset \mathbf{X}_v , and normalize the both datasets with the mean and variance of the normal training dataset.
- 2) Gather existing fault datasets $\{\mathbf{X}_f^{(c)}, c = 1, 2, \dots, C\}$ and normalize them with the mean and variance of the normal training dataset.
- 3) Build the FDKPCA model, which includes the KPCA sub-model constructed based on the normal training dataset $\mathbf{X}_f^{(0)}$, and the fault discriminant sub-models each constructed based on a dataset pair $\{\mathbf{X}_f^{(0)}, \mathbf{X}_f^{(c)}\}$.
- 4) Project the validation data \mathbf{X}_v onto the FDKPCA model, and calculate the monitoring statistics $T_f^{2(c)}$ and $Q_f^{(c)}$ for $c = 0, 1, 2, \dots, C$.
- 5) Apply the KDE to the monitoring statistics obtained in 4) to estimate the confidence limits $T_{f,\text{lim}}^{(c)}$ and $Q_{f,\text{lim}}^{(c)}$ for $c = 0, 1, 2, \dots, C$.

Online monitoring stage

- 1) Acquire a new data vector \mathbf{x}_t from the plant at time instant t , and normalize it with the mean and variance of the normal training dataset.
- 2) Project \mathbf{x}_t onto the FDKPCA model, and calculate the corresponding monitoring statistics.
- 3) Apply the Bayesian inference to calculate the posterior fault probabilities using (29) and (30).
- 4) Calculate the probability-based monitoring statistics PT^2 and PQ using (37) and (38).
- 5) Determine if a fault occurs by comparing the monitoring statistics with the significance level.

Finally, we note that when no prior fault data are available, the FDKPCA naturally degenerates to the traditional KPCA.

4. Simulation Studies

Two case studies, a simulated nonlinear system and a benchmark CSTR system, are used to evaluate our proposed FDKPCA based monitoring scheme, in comparison with the traditional KPCA method. For the both KPCA and FDKPCA based monitoring statistics, their thresholds are set to the corresponding 95% confidence limits obtained by the KDE method. In all monitoring charts, the monitoring statistics are plotted with solid curves and their thresholds are plotted with dashed lines. Two performance metrics are used to evaluate the fault detection results, namely, the fault detection rate, which is defined as the percentage of the samples exceeding the limits over all the faulty samples, and the fault detection time, which is defined as the number of the alarming sample from which successive 6 samples exceed the threshold.

4.1. A Simulated Nonlinear System

The following nonlinear system with $M = 6$ measured variables, which is the modified version of the example given in [21], is simulated

$$\begin{cases} x_1 = s_1 + e_1, \\ x_2 = s_2 + e_2, \\ x_3 = s_3 + e_3, \\ x_4 = s_1^2 - 3s_2 + e_4, \\ x_5 = -s_1^3 + 3s_3^2 + e_5, \\ x_6 = s_1^2 + 2s_3^3 + e_6, \end{cases} \quad (46)$$

where the source signals s_i for $1 \leq i \leq 3$ are independently and uniformly distributed in $[0, 4]$, the noises e_i for $1 \leq i \leq 6$ are independent and each follows the normal distribution with zero mean and variance 0.01, while x_i for $1 \leq i \leq 6$ are the monitored variables. Normal operation dataset with 600 samples are generated based on the model (46). Among these data, 300 samples are used as the normal training data to build statistical model and the other 300 samples are used as the validation dataset to compute confidence limits. Two faults are designed for this system, which are listed in Table 1. For both faults, the training datasets, i.e., prior fault datasets, and the testing datasets are generated, respectively, each includes 300 samples with the fault introduced after 100-th sample.

Table 1: Two faults of the simulated nonlinear system

Fault class	Fault description	Fault magnitude	
		Training data	Testing data
D1	A step change in variable x_2	-0.4	-0.35
D2	A step change in variable x_5	+3.0	+2.8

Gaussian kernel function is used. According to the empirical rules of [21], the kernel width for the KPCA model is related to the number of monitored variables M . In our simulation study, the kernel width is set

to $\sigma = 100M$ empirically for the KPCA model. The KPCA sub-model of the FDKPCA has the same kernel width. For the fault discriminant sub-models of the FDKPCA, the kernel width parameters are chosen by the cross-validation method [49, 50].

4.1.1. Case 1. Prior D1 fault data available

In this case, the FDKPCA model includes one fault discriminant sub-model related to the prior D1 fault training dataset. Figs. 3 and 4 show the monitoring charts of the KPCA and FDKPCA, respectively, for the D1 fault testing data. It can be seen from Fig. 3 that the KPCA cannot detect this fault because its monitoring statistics T^2 and Q stay well below the respective confidence limits with no obvious changes after the occurrence of fault D1. The fault detection rates based on the T^2 and Q statistics of the KPCA are 7.5% and 8%, respectively. By contrast, the PT^2 chart of the FDKPCA shown in Fig. 4 can detect this fault at the 116-th sample with a high fault detection rate of 82.0%. However, the PQ chart of the FDKPCA offers no clear improvement in detecting this fault and its fault detection rate is also 8%. The reason for good PT^2 performance and poor PQ one is that most significant discriminant information is extracted by

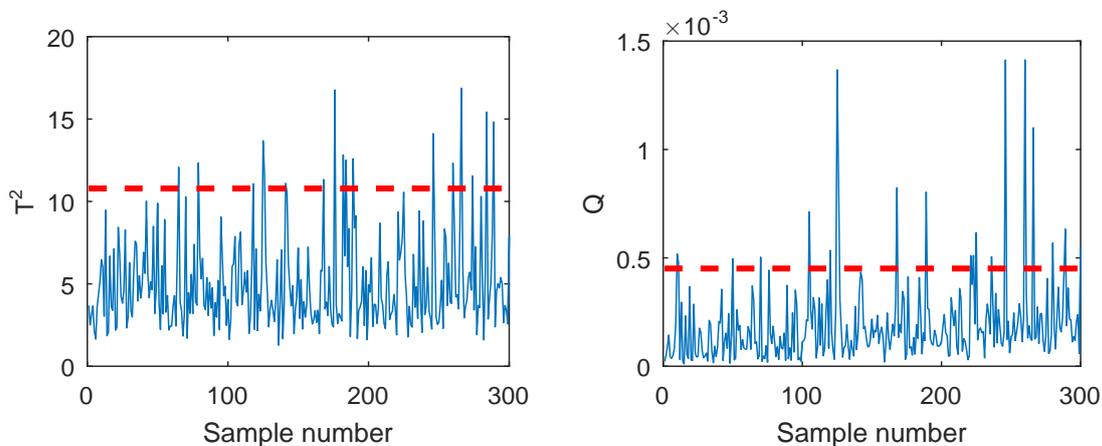


Figure 3: KPCA monitoring charts for fault D1 of the simulated nonlinear system.

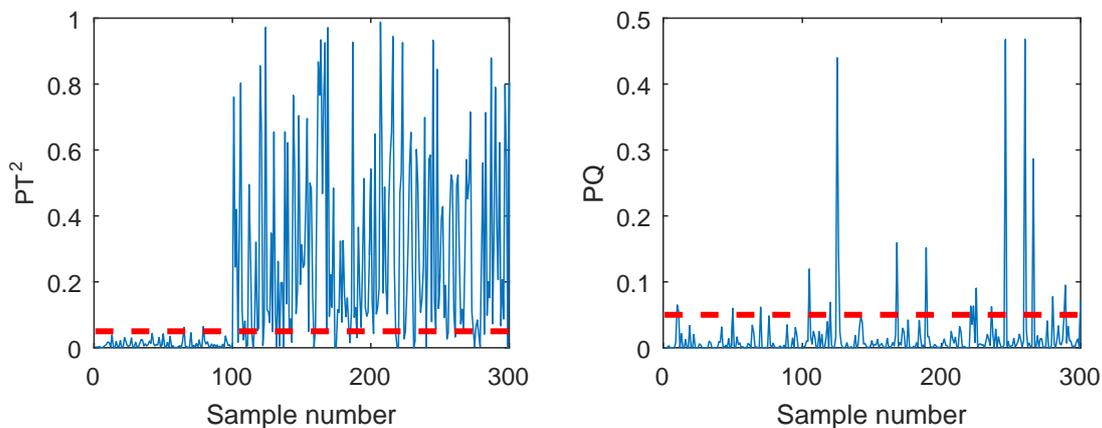


Figure 4: FDKPCA monitoring charts for fault D1 of the simulated nonlinear system under the case 1 training.

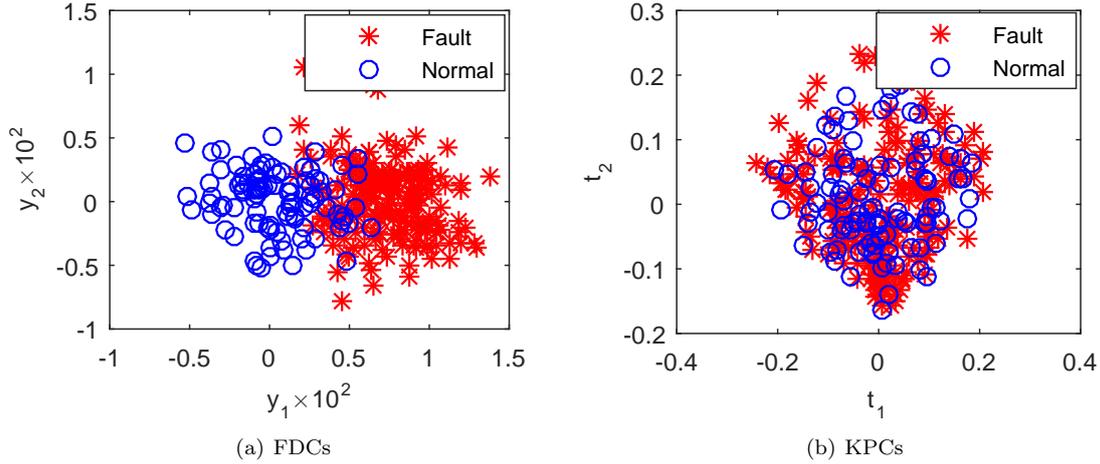


Figure 5: FDCs and KPCs for the fault D1 test dataset of the simulated nonlinear system under the case 1 training.

the first several principal FDCs which are included in the PT^2 statistic.

To investigate the FDKPCA method further, we provide the KPCs and FDCs for the D1 fault testing data in Fig. 5. Note that there are 100 normal operation samples and 200 faulty samples in the D1 fault testing dataset. It is clear from Fig. 5 that the normal and faulty KPCs overlap heavily but the fault FDCs can mostly be separated from the normal FDCs. This explains the much better fault detection performance of the FDKPCA in this case. We also plot the monitoring statistics generated by the fault discriminant sub-model only in Fig. 6, where it can be seen that the $T_f^{2(1)}$ chart clearly detects the occurrence of fault D1 with 82.5% fault detection rate.

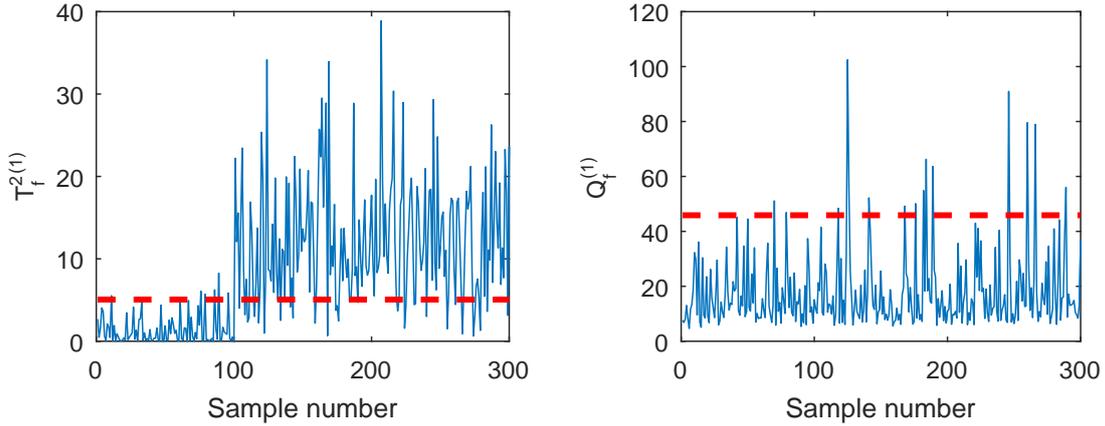


Figure 6: Monitoring statistics using the fault discriminant sub-model only for fault D1 of the simulated nonlinear system under the case 1 training.

The weighting factors to combine the KPCA sub-model and KLNPDFA fault discriminant sub-model are illustrated in Fig. 7. It can be seen that under the normal operation mode (the first 100 samples), $\bar{w}_{T_2}^{(1)}$ for the fault discriminant sub-model is 0 and $\bar{w}_{T_2}^{(0)}$ for the KPCA sub-model is 1. Thus, the fault discriminant sub-model is deactivated as it should be. After the fault occurs, $\bar{w}_{T_2}^{(1)}$ becomes non-zero and mostly takes

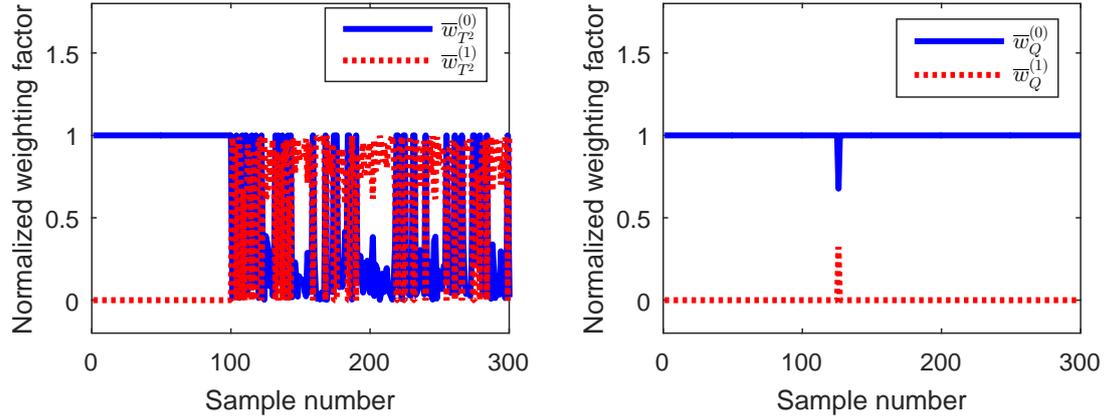


Figure 7: Normalized weighting factors of fault probabilities for fault D1 of the simulated nonlinear system under the case 1 training.

large values close to the maximum 1, indicating that the fault discriminant sub-model is correctly activated, while $\bar{w}_{T^2}^{(0)}$ although becoming smaller still maintains non-zero values most of the time. In other words, the KPCA sub-model is always active. This confirms the effectiveness of our weighting design strategy which aims to ensure that the KPCA sub-model always functions well while the fault discriminant sub-models are only activated under a fault condition. As explained previously, for this specific system, little fault information is involved in $Q_f^{2(1)}$ statistic of the fault discriminant sub-model and, therefore, $\bar{w}_Q^{(1)}$ is 0 for most of the faulty samples.

When detecting an unknown fault which is related to the prior fault data used in training the FDKPCA model, the fault detection performance of the FDKPCA will obviously be better than that of the KPCA. Intuitively, when detecting an unknown fault which is not related to any prior fault data for training the FDKPCA model, the fault detection performance of the FDKPCA should be at least no worse than that of the KPCA. Figs. 8 and 9 depict the monitoring performance of the KPCA and FDKPCA for the D2 fault test dataset, respectively. From Fig. 8, it can be seen that the KPCA exhibits a poor performance

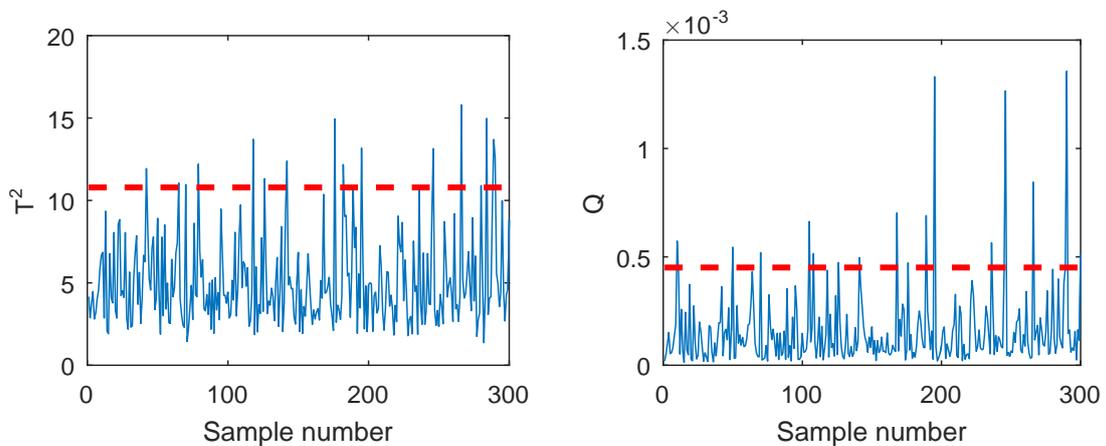


Figure 8: KPCA monitoring charts for fault D2 of the simulated nonlinear system.

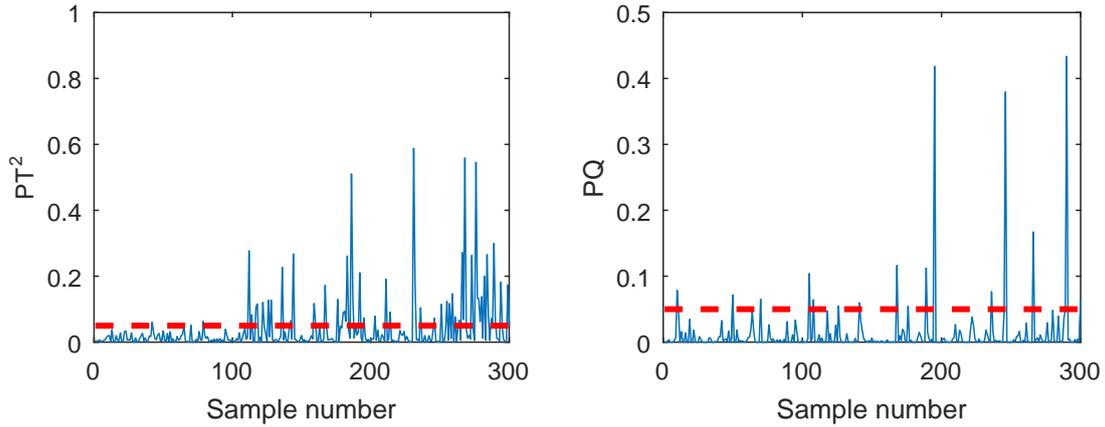


Figure 9: FDKPCA monitoring charts for fault D2 of the simulated nonlinear system under the case 1 training.

with the T^2 statistic based fault detection rate of 7% and the Q statistic based fault detection rate of 6.5%. According to Fig. 9, the performance of the FDKPCA is certainly no worse than that of the KPCA. In fact, some improvement is achieved by the FDKPCA with its PT^2 statistic attaining the fault detection rate of 27.5% and its PQ statistic achieving the same fault detection rate of 6.5% as the Q statistic of the KPCA.

4.1.2. Case 2. Prior D2 fault data available

After the FDKPCA model is built using the normal operation dataset and prior D2 fault dataset, the monitoring charts of the constructed FDKPCA model for the D1 fault testing dataset are shown in Fig. 10. It can be seen from Fig. 10 that a fault detection rate of 44% is obtained based on the FDKPCA's PT^2 statistic, which is a large improvement over the fault detection rate of 7.5% obtained by the KPCA's T^2 statistic. The monitoring charts of the constructed FDKPCA model for the D2 fault testing dataset are depicted in Fig.11, where it can be seen that the FDKPCA's PT^2 statistic is capable of detecting this fault with a high fault detection rate of 74.5%.

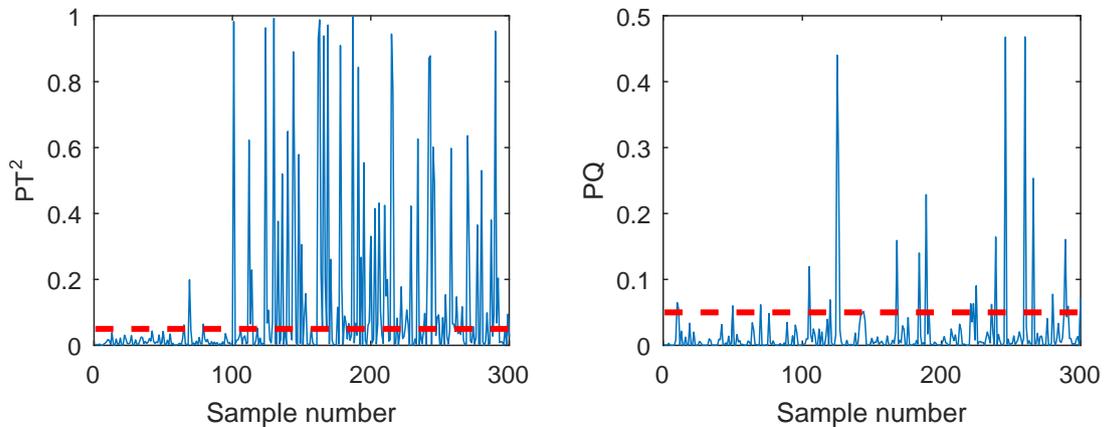


Figure 10: FDKPCA monitoring charts for fault D1 of the simulated nonlinear system under the case 2 training.

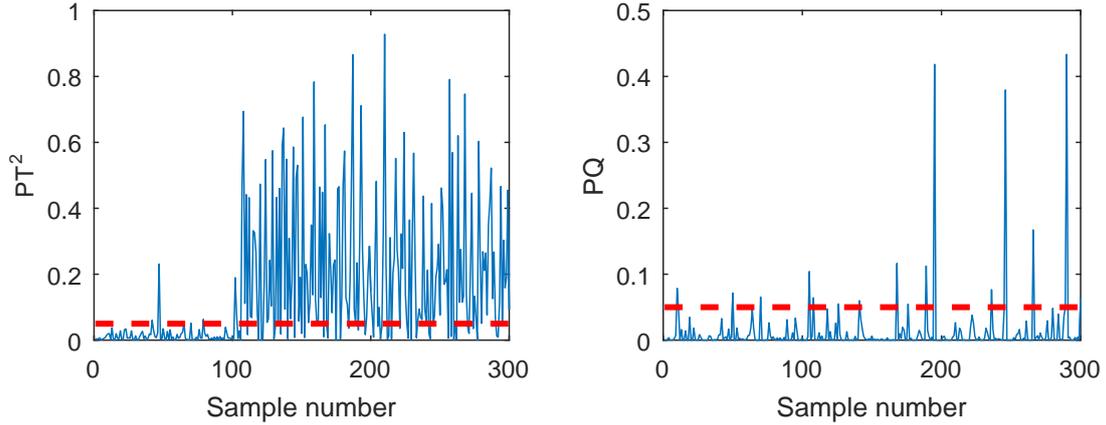


Figure 11: FDKPCA monitoring charts for fault D2 of the simulated nonlinear system under the case 2 training.

4.1.3. Case 3: Both prior D1 and D2 fault data available

The FDKPCA modeling involves building the KPCA sub-model and the two fault discriminant sub-models. The monitoring results depicted in Figs. 12 and 13 clearly demonstrate that the constructed FDKPCA model is capable of detecting both the D1 and D2 faults. More specifically, for the D1 fault testing dataset, the FDKPCA's PT^2 and PQ statistics achieve the fault detection rates of 84.5% and 10.5%, respectively, while for the D2 fault testing dataset, the PT^2 and PQ statistics of the FDKPCA model are 75.5% and 6.5%, respectively.

The results of all the three cases are listed in Table 2. The KPCA model is unable to detect the faults D1 and D2 of this simulated nonlinear system. By contrast, the FDKPCA model incorporating both the prior D1 and D2 fault information is capable of detecting the both faults. Also the FDKPCA model integrating one prior fault information is able to detect the fault for which it has some prior information. Moreover, the FDKPCA model trained with one prior fault data is capable of improving the fault detection performance for the other unseen fault, in comparison with the KPCA model.

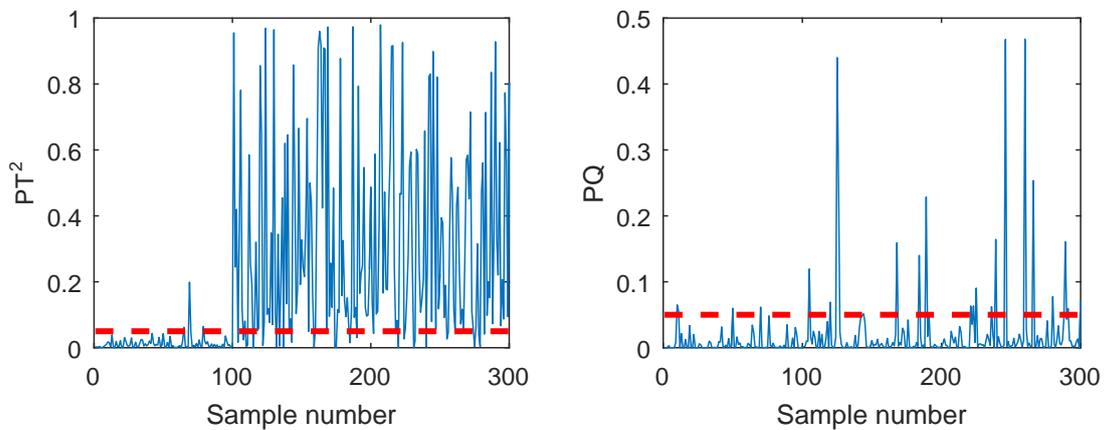


Figure 12: FDKPCA monitoring charts for fault D1 of the simulated nonlinear system under the case 3 training.

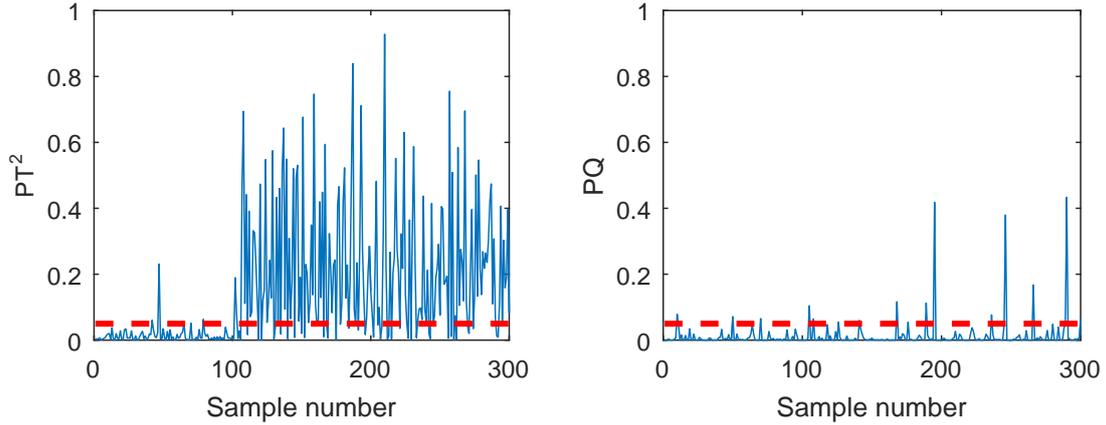


Figure 13: FDKPCA monitoring charts for fault D2 of the simulated nonlinear system under the case 3 training.

Table 2: Fault detection rates (%) of faults D1 and D2 obtained by KPCA and FDKPCA.

Testing Fault No.	KPCA		FDKPCA					
			Case 1		Case 2		Case 3	
	T^2	Q	PT^2	PQ	PT^2	PQ	PT^2	PQ
D1	7.5	8.0	82.0	8.0	44.0	10.5	84.5	10.5
D2	7.0	6.5	27.5	6.5	74.5	6.5	75.5	6.5

4.2. The continuous stirred tank reactor System

The CSTR is a benchmark chemical process widely used in the evaluation of fault detection and process control methods [7, 51, 52]. In this CSTR system shown in Fig. 14, the input stream including reactant A flows into reactor where first-order irreversible reaction $A \rightarrow B$ occurs to produce the component B as an output stream. Cooling jacket is used to remove the reaction heat. To ensure a stable operation, two cascade control systems are designed to control reactor level and temperature, respectively.

Based on the mechanical model presented in [51, 52], this CSTR system is simulated. In the simulation procedure, ten variables are collected and Gaussian noises are added to them as the measurement noises.

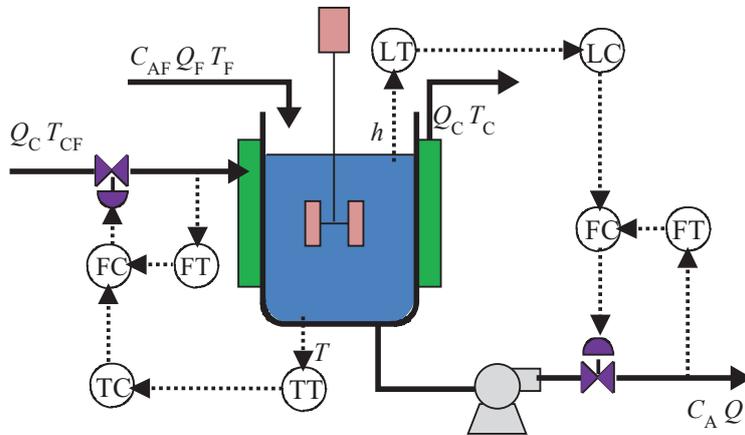


Figure 14: Schematic of the CSTR system.

Table 3: Monitored CSTR system variables and their noise standard deviations (Std.).

Variable name	Variable description	Normal value	Noise Std.	Unit
C_A	Concentration of reactant A in reactor	0.0372	0.0024	mol/L
T	Reactor temperature	402.35	0.71	K
T_C	Coolant temperature in jacket	345.44	0.63	K
h	Reactor level	6.0	0.04	dm
C_{AF}	Concentration of reactant A in reactor feed stream	1.0	0.0024	mol/L
T_F	Reactor feed temperature	320.0	0.71	K
T_{CF}	Coolant inlet temperature	300	0.63	K
Q_F	Reactor inlet flow rate	100	0.71	L/min
Q	Reactor outlet flow rate	100	0.71	L/min
Q_C	Coolant flow rate	15	0.1	L/min

Detailed description of these variables and the corresponding noise strengths are given in Table 3. Normal operation and nine types of fault situations are designed. The designed faults include process condition changes, sensor malfunctions and valve sticking, as can be seen from Table 4.

Table 4: Fault classes of the CSTR system.

Fault class	Fault description	Fault magnitude	
		Training data	Testing data
F1	Coolant inlet temperature ramps down.	-0.8 K/min	-0.9 K/min
F2	Concentration of reactant A in reactor feed stream ramps up.	+0.003 (mol/L)/min	+0.002 (mol/L)/min
F3	Reactor feed temperature ramps up.	0.6 K/min	0.8 K/min
F4	Coolant valve is stuck at some position.	29.5%	30.5%
F5	Reactor temperature sensor has a bias.	+1K	+0.7K
F6	Catalyst activation energy ramps up.	+6 K/min	+4 K/min
F7	Heat-transfer coefficient ramps down.	-200 (J/(min K))/min	-220 (J/(min K))/min
F8	Coolant temperature sensor has a bias.	+2K	+1.5K
F9	Coolant flow rate sensor has a bias.	+2L/min	+0.5L/min

For the normal operation mode, 500 samples are generated as the normal training dataset and another 800 samples are generated as the validating data. For each of the nine fault classes, the prior fault training dataset and the fault testing dataset are simulated, which include 500 and 800 samples, respectively. In the prior fault training data, the fault is introduced after the 50-th sample while in the fault testing data, the fault is introduced after the 180-th sample. The kernel width parameters for the KPCA and FDKPCA are determined using the same strategy as given in Section 4.1.

4.2.1. Three prior fault data available

We start by investigating the case that the prior F2, F4 and F8 fault data are available. In this case, three fault discriminant sub-models are utilized by the FDKPCA.

Firstly, the KPCA and FDKPCA monitoring charts for the F2 Fault testing dataset are depicted in

Figs. 15 and 16, respectively. The results of Fig. 15 show that the T^2 and Q statistics of the KPCA both detect this fault at the 367-th sample with the fault detection rates of 75% and 74.8%, respectively. The FDKPCA results given in Fig. 16 indicate that its PT^2 statistic alarms this fault at the 302-th sample while its PQ statistic detects this fault at the 347-th sample. Therefore, the PT^2 statistic reduce the fault detection delay by 65 samples and the PQ statistic alarms the fault 20 samples earlier, compared with the KPCA. The fault detection rates of the FDKPCA are 82.4% and 79.7% based on the PT^2 and PQ statistics, respectively, which are both higher than the detection rates of the KPCA method.

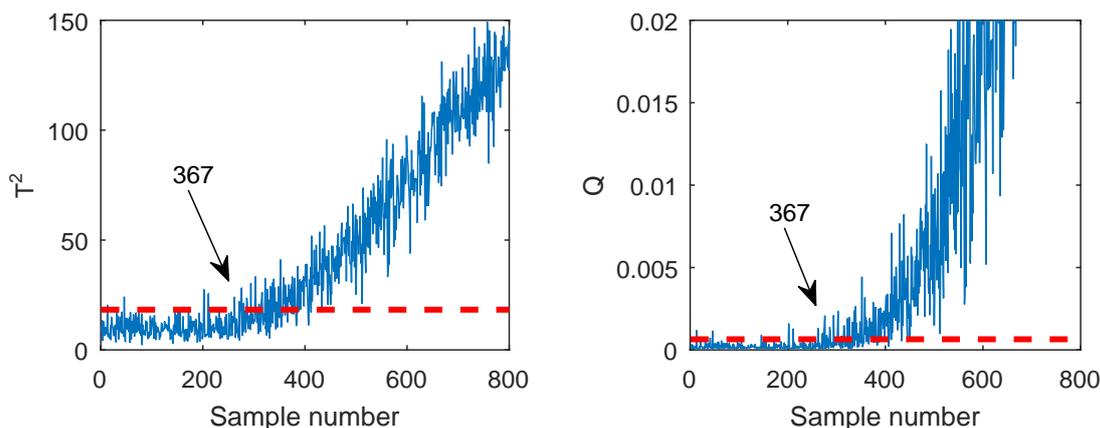


Figure 15: KPCA monitoring charts for fault F2 of the CSTR system.

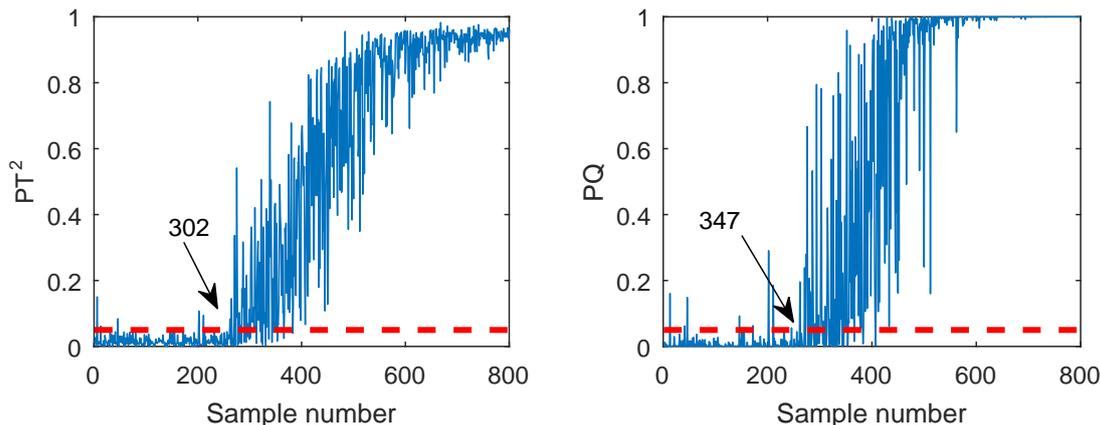


Figure 16: FDKPCA monitoring charts for fault F2 of the CSTR system. The model is trained with prior F2, F4 and F8 fault data.

Furthermore, the KPCA and FDKPCA monitoring results for the F8 fault testing dataset are shown in Figs. 17 and 18, respectively. As this fault only involves a small sensor bias of coolant temperature sensor and it does not affect other measured variables, it is difficult for the KPCA method to detect this fault. Observe from Fig. 17 that the monitoring statistics of the KPCA fluctuate around the confidence limits and cannot clearly signal an alarm. The fault detection rates of the KPCA's T^2 and Q statistics are 28.6% and 29.8%, respectively. By contrast, the FDKPCA exhibits significantly better fault detection performance.

As can be seen from Fig. 18, the fault detection rates are 74.8% and 32.9% based on the PT^2 and PQ statistics, respectively. Moreover, the FDKPCA's PT^2 statistic detects this fault at the 231-th sample and its PQ detects this fault at the 313-th sample.

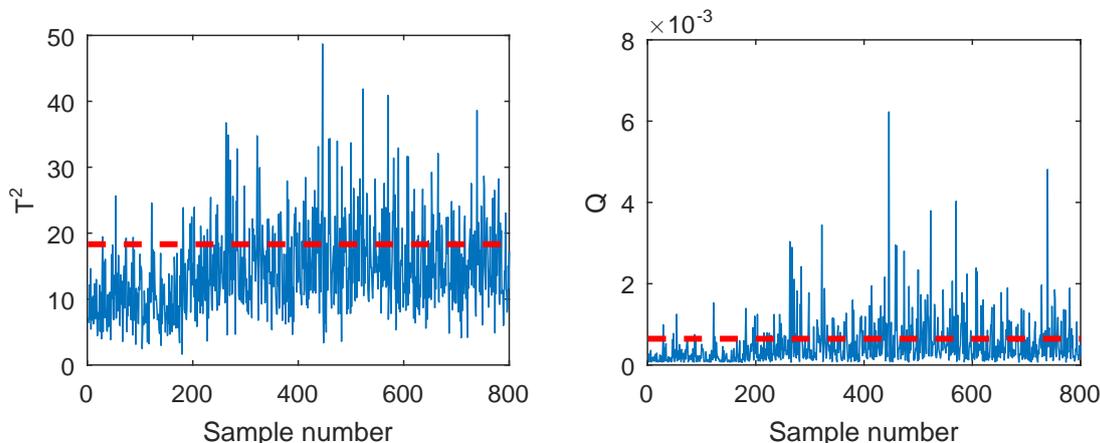


Figure 17: KPCA monitoring charts for fault F8 of the CSTR system.

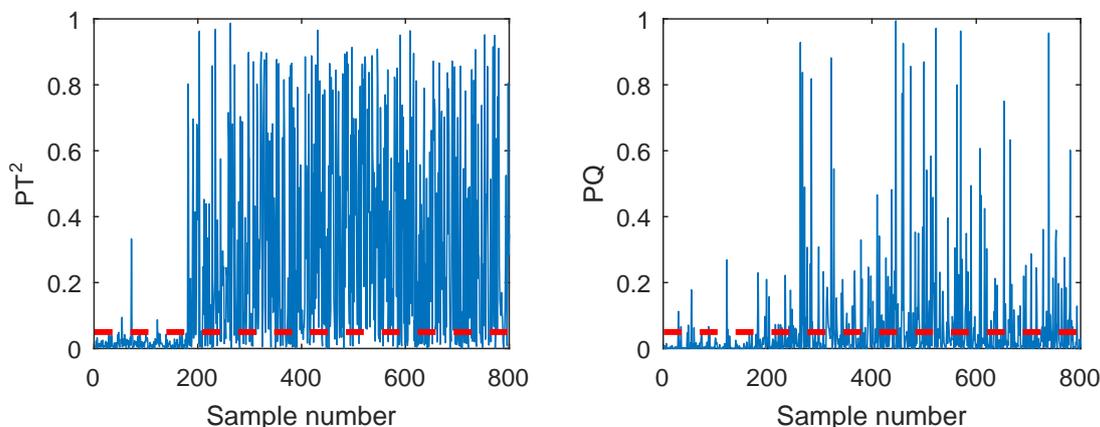


Figure 18: FDKPCA monitoring charts for fault F8 of the CSTR system. The model is trained with prior F2, F4 and F8 fault data.

4.2.2. Three cases of different prior fault data available

Next we compare the performance of the KPCA method with that of the FDKPCA method given the three cases with different available prior fault data:

Case 1. One prior F2 fault dataset is available.

Case 2. Three prior F2, F4 and F8 fault datasets are available.

Case 3. All the nine prior F1 to F9 fault datasets are available.

Under these three cases, the FDKPCA statistical models are built and tested on all the 9 fault testing datasets, in comparison with the KPCA. The fault detection rates and fault detection times obtained are tabulated in Tables 5 and 6, respectively, where symbol / represents failure to detect the fault. For the FDKPCA under the case 1 training, its monitoring results for fault F2 are significantly better than that of

Table 5: Fault detection rates (%) for the CSTR system obtained by the KPCA and FDKPCA.

Testing Fault No.	KPCA		FDKPCA					
			Case 1		Case 2		Case 3	
	T^2	Q	PT^2	PQ	PT^2	PQ	PT^2	PQ
F1	85.5	85.5	85.5	85.7	85.5	85.7	89.4	86.3
F2	75.0	74.8	82.1	79.5	82.4	79.7	83.7	79.8
F3	84.0	83.6	84.0	84.5	84.0	84.7	87.1	85.5
F4	34.4	39.0	34.8	49.8	61.5	56.3	69.7	58.6
F5	69.0	74.2	69.2	81.3	84.8	85.3	93.6	87.1
F6	48.9	48.2	50.3	49.7	51.1	50.5	63.1	54.4
F7	61.6	61.8	62.4	64.4	63.9	65.2	71.0	66.1
F8	28.6	29.8	28.7	30.2	74.8	32.9	75.7	35.0
F9	86.3	86.5	87.3	91.9	89.8	92.6	97.7	94.7

the KPCA, while its monitoring performance for the other 8 faults is at least no worse than those of the KPCA. In fact, according to Table 6, the FDKPCA obtained under the case 1 training also reduces the fault detection time for fault F5 clearly. As explained previously, fault F8 is the most difficult fault to detect by data-driven methods. Observe from Table 6 that the FDKPCA obtained under the case 1 training also fails to detect this fault. With the FDKPCA obtained under the case 2 training, its monitoring performance for faults F2, F4 and F8 are significantly better than those of the KPCA. Moreover, its monitoring performance for the other 5 unseen faults are also improved, compared with the KPCA model. With all the 9 classes of prior fault information available, the FDKPCA obtained under the case 3 training significantly improves the monitoring performance for all the 9 faults. The average fault detection rates over all the 9 faults obtained by the KPCA and FDKPCA models are compared Fig. 19.

Table 6: Fault detection times (sample numbers) for the CSTR system obtained by the KPCA and FDKPCA.

Testing Fault No.	KPCA		FDKPCA					
			Case 1		Case 2		Case 3	
	T^2	Q	PT^2	PQ	PT^2	PQ	PT^2	PQ
F1	308	295	308	295	308	295	273	295
F2	367	367	302	347	302	347	300	347
F3	312	312	312	308	312	308	256	308
F4	/	/	/	318	238	318	203	318
F5	278	277	241	241	241	241	233	241
F6	506	580	506	506	506	478	425	478
F7	472	472	472	472	419	419	405	405
F8	/	/	/	/	231	313	231	313
F9	226	226	226	222	186	186	186	186

4.3. Summary of the Simulation Results

Based on the results of the above two case studies, we can draw the following observations regarding the features of the proposed FDKPCA monitoring procedure, in comparison with the traditional KPCA method.

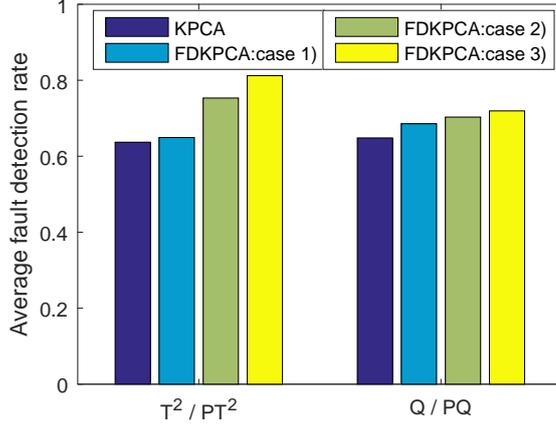


Figure 19: Average fault detection rates of the KPCA and FDKPCA for the CSTR system.

1. When detecting an unknown fault which is related to the prior fault data used in training the FDKPCA model, the FDKPCA significantly enhances the fault detection performance, in comparison with the model incorporating no prior fault information.
2. When detecting an unknown fault which is not related to any prior fault data used in training the FDKPCA model, the fault detection performance of the FDKPCA is at least no worse than that of the KPCA. Often, however, the FDKPCA can improve the fault detection performance for an unseen fault, over the KPCA method. We believe this is because different faults may influence some same monitored variables.
3. Generally, utilization of prior fault information is beneficial and helps the FDKPCA method to monitor the process changes better. More specifically, incorporating more prior fault information increases the fault detection capability of the FDKPCA model.

In addition, the simulation results also verify that the weighting strategy designed in Section 3.2 for constructing the probability-based monitoring statistics is appropriate and achieves its design goal, which is to ensure that the KPCA sub-model is always active while the fault discriminant sub-models are only activated when a fault is detected.

5. Conclusions

A basic principle in data learning or modeling is to incorporate available *a priori* information regarding the underlying data generating mechanism into the learning process. In this paper, we have adopted this principle in data-driven nonlinear process monitoring application. In particular, a novel FDKPCA procedure has been proposed by integrating KPCA modeling with fault discriminant analysis for nonlinear process monitoring, which incorporates the available prior fault data effectively to enhance fault detection performance. More specifically, two different types of features, KPCs and FDCs, have been considered in our

FDKPCA based fault detection. We have proposed to utilize a Bayesian inference to transform the monitoring statistics built by the KPCA sub-model and the fault discriminant sub-models into fault probabilities and have further designed an appropriate weighting strategy to combine the resulting fault probabilities into the overall probability-based monitoring statistics. Two case studies, involving a simulated nonlinear system and the benchmark CSTR system, have demonstrated that the proposed FDKPCA method achieves superior fault detection performance over the widely used KPCA method.

The proposed approach thus has opened a new research direction for data-driven nonlinear process monitoring, where many issues warrant further study. For example, it is worth considering how to design alternative weighting factors for monitoring statistics in order to further enhance monitoring performance. A more challenging problem is how to utilizing the prior fault information to solve much more difficult fault identification or diagnosis.

Acknowledgement

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

References

- [1] C.F. Alcala, S.J. Qin, Analysis and generalization of fault diagnosis methods for process monitoring, *J. Process Control*, 21(3) (2011) 322–330.
- [2] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, *Industrial & Engineering Chemistry Research*, 52(10) (2013) 3543–3562.
- [3] Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques – part I: fault diagnosis with model-based and signal-based approaches, *IEEE Trans. Industrial Electronics*, 62(6) (2015) 3757–3767.
- [4] V. Agrawal, B.K. Panigrahi, P.M.V. Subbarao, Review of control and fault diagnosis methods applied to coal mills, *J. Process Control*, 32 (2015) 138–153.
- [5] Y. Liu, Y. Pan, Q. Wang, D. Huang, Statistical process monitoring with integration of data projection and one-class classification, *Chemometrics and Intelligent Laboratory Systems*, 149 (2015) 1–11.
- [6] L.H. Chiang, E.L. Russell, R.D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, Springer, London UK, 2001.
- [7] X. Tian, L. Cai, S. Chen, Noise-resistant joint diagonalization independent component analysis based process fault detection, *Neurocomputing*, 149 (2015) 652–666.
- [8] C. Zhao, F. Gao, A nested-loop Fisher discriminant analysis algorithm, *Chemometrics and Intelligent Laboratory Systems*, 146 (2015) 396–406.
- [9] W. Ku, R.H. Storer, C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis, *Chemometrics and Intelligent Laboratory Systems*, 30(1) (1995) 179–196.
- [10] J. Huang, X. Yan, Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and Bayesian inference, *Chemometrics and Intelligent Laboratory Systems*, 148 (2015) 115–127.
- [11] S.J. Zhao, J. Zhang, Y.M. Xu, Monitoring of processes with multiple operating modes through multiple principle component analysis models, *Industrial and Engineering Chemistry Research*, 43(22) (2004) 7025–7035.

- [12] H. Ma, Y. Hu, H. Shi, A novel local neighborhood standardization strategy and its application in fault detection of multimode processes, *Chemometrics and Intelligent Laboratory Systems*, 118 (2012) 287–300.
- [13] X. Deng, X. Tian, Multimode process fault detection using local neighborhood similarity analysis, *Chinese J. Chemical Engineering*, 22(11-12) (2014) 1260–1267.
- [14] Z. Ge, Z. Song, Distributed PCA model for plant-wide process monitoring, *Industrial and Engineering Chemistry Research*, 52(5) (2013) 1947–1957.
- [15] Q. Jiang, X. Yan, Plant-wide process monitoring based on mutual information-multiblock principal component analysis, *ISA Transactions*, 53(5) (2014) 1516–1527.
- [16] Y. Zhang, S. Li, Z. Hu, Improved multi-scale kernel principal component analysis and its application for fault detection, *Chemical Engineering Research and Design*, 90(9) (2012) 1271–1280.
- [17] C. Lau, K. Ghosh, M.A. Hussain, C.R. Che Hassan, Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS, *Chemometrics and Intelligent Laboratory Systems*, 120 (2013) 1–14.
- [18] M. Žvokelj, S. Zupan, I. Prebil, Non-linear multivariate and multiscale monitoring and signal denoising strategy using kernel principal component analysis combined with ensemble empirical mode decomposition method, *Mechanical Systems and Signal Processing*, 25(7) (2011) 2631–2653.
- [19] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE Journal*, 37(2) (1991) 233–243.
- [20] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, 10(5) (1998) 1299–1319.
- [21] J.-M. Lee, C. Yoo, S.W. Choi, P.A. Vanrolleghem, I.-B. Lee, Nonlinear process monitoring using kernel principal component analysis, *Chemical Engineering Science*, 59(1) (2004) 223–234.
- [22] J.-M. Lee, C. Yoo, I.-B. Lee, Fault detection of batch processes using multiway kernel principal component analysis, *Computers & Chemical Engineering*, 28(9) (2004) 1837–1847.
- [23] Q. Jiang, X. Yan, Weighted kernel principal component analysis based on probability density estimation and moving window and its application in nonlinear chemical process monitoring, *Chemometrics and Intelligent Laboratory Systems*, 127 (2013) 121–131.
- [24] I.B. Khediri, M. Limam, C. Weihs, Variable window adaptive kernel principal component analysis for nonlinear nonstationary process monitoring, *Computers and Industrial Engineering*, 61(3) (2011) 437–446.
- [25] X. Deng, X. Tian, 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*, 127 (2013) 195–209.
- [26] S.W. Choi, C. Lee, J.-M. Lee, J.H. Park, I.-B. Lee, Fault detection and identification of nonlinear processes based on kernel PCA, *Chemometrics and Intelligent Laboratory Systems*, 75(1) (2005) 55–67.
- [27] X. Deng, X. Tian, Nonlinear process fault pattern recognition using statistics kernel PCA similarity factor, *Neurocomputing*, 121 (2013) 298–308.
- [28] J. Fan, S.J. Qin, Y. Wang, Online monitoring of nonlinear multivariate industrial processes using filtering KICA-PCA, *Control Engineering Practice*, 22 (2014) 205–216.
- [29] L. Cai, X. Tian, 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*, 28(1) (2017) 122–134.
- [30] L.H. Chiang, E.L. Russell, R.D. Braatz, Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis, *Chemometrics and Intelligent Laboratory Systems*, 50 (2000) 243–252.
- [31] L.H. Chiang, M.E. Kotanchek, A.K. Kordon, Fault diagnosis based on Fisher discriminant analysis and support vector machines, *Computers & Chemical Engineering*, 28(8) (2004) 1389–1401.

- [32] B. Jiang, X. Zhu, D. Huang, J.A. Paulson, R.D. Braatz, A combined canonical variate analysis and Fisher discriminant analysis (CVA-FDA) approach for fault diagnosis, *Computers & Chemical Engineering*, 77 (2015) 1–9.
- [33] X.B. He, Y.P. Yang, Y.H. Yang, Fault diagnosis based on variable-weighted kernel Fisher discriminant analysis, *Chemometrics and Intelligent Laboratory Systems*, 93(1) (2008) 27–33.
- [34] S. Zhong, Q. Wen, Z. Ge, Semi-supervised Fisher discriminant analysis model for fault classification in industrial processes, *Chemometrics and Intelligent Laboratory Systems*, 138 (2014) 203–211.
- [35] Z.-B. Zhu, Z.-H. Song, A novel fault diagnosis system using pattern classification on kernel FDA subspace, *Expert Systems With Applications*, 38(6) (2011) 6895–6905.
- [36] J. Feng, J. Wang, H. Zhang, Z. Han, Fault diagnosis method of joint Fisher discriminant analysis based on the local and global manifold learning and its kernel version, *IEEE Trans. Automation Science and Engineering*, 13(1) (2016) 122–133.
- [37] J. Yu, Nonlinear bioprocess monitoring using multiway kernel localized Fisher discriminant analysis, *Industrial and Engineering Chemistry Research*, 50(6) (2011) 3390–3402.
- [38] G. Rong, S.-Y. Liu, and J.-D. Shao, Fault diagnosis by locality preserving discriminant analysis and its kernel variation, *Computers and Chemical Engineering*, 49 (2013) 105–113.
- [39] J. Yu, Local and nonlocal preserving projection for bearing defect classification and performance assessment, *IEEE Trans. Industrial Electronics*, 59(5) (2012) 2363–2376.
- [40] W. Shao, X. Tian, P. Wang, Supervised local and non-local structure preserving projections with application to just-in-time learning for adaptive soft sensor, *Chinese J. Chemical Engineering*, 23(12) (2015) 1925–1934.
- [41] J. Yu, X. Lu, Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis, *IEEE Trans. Semiconductor Manufacturing*, 29(1) (2016) 33–43.
- [42] X. He, S. Yan, Y. Hu, P. Niyogi, H.J. Zhang, Face recognition using Laplacianfaces, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(3) (2005) 328–340.
- [43] J. Yang, D. Zhang, J.-Y. Yang, Non-locality preserving projection and its application to palmprint recognition, *Proceedings of ICARCV’06 (Singapore) 2006*, pp. 1–4.
- [44] J.-M. Lee, C. Yoo, I.-B. Lee, Statistical process monitoring with independent component analysis, *J. Process Control*, 14(5) (2004) 467–485.
- [45] X. Hong, S. Chen, C.J. Harris, A forward-constrained regression algorithm for sparse kernel density estimation, *IEEE Trans. Neural Networks*, 19(1) (2008) 193–198.
- [46] X. Hong, J. Gao, S. Chen, T. Zia, Sparse density estimation on the multinomial manifold, *IEEE Trans. Neural Networks and Learning Systems*, 26(11) (2015) 2972–2977.
- [47] Z. Ge, Z. Song, Performance-driven ensemble learning ICA model for improved non-Gaussian process monitoring, *Chemometrics and Intelligent Laboratory Systems*, 123 (2013) 1–8.
- [48] C. Tong, A. Palazoglu, X. Yan, Improved ICA for process monitoring based on ensemble learning and Bayesian inference, *Chemometrics and Intelligent Laboratory Systems*, 135 (2014) 141–149.
- [49] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proceedings of the International Joint Conference on Artificial Intelligence (Montreal, Canada) 1995*, pp. 1137–1143.
- [50] S. An, W. Liu, S. Venkatesh, Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression, *Pattern Recognition*, 40(8) (2007) 2154–2162.
- [51] M.C. Johannesmeyer, A. Singhal, D.E. Seborg, Pattern matching in historical data, *AIChE Journal*, 48(9) (2002) 2022–2038.
- [52] X. Deng, X. Tian, Sparse kernel locality preserving projection and its application in nonlinear process fault detection, *Chinese J. Chemical Engineering*, 21(2) (2013) 163–170.