Chapter 10

Process Monitoring Using Data-Based Fault Detection Techniques: Comparative Studies

Mohammed Ziyan Sheriff, Chiranjivi Botre,
Majdi Mansouri, Hazem Nounou,
Mohamed Nounou and Mohammad Nazmul Karim

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67347

Abstract

Data based monitoring methods are often utilized to carry out fault detection (FD) when process models may not necessarily be available. The partial least square (PLS) and principle component analysis (PCA) are two basic types of multivariate FD methods, however, both of them can only be used to monitor linear processes. Among these extended data based methods, the kernel PCA (KPCA) and kernel PLS (KPLS) are the most well-known and widely adopted. KPCA and KPLS models have several advantages, since, they do not require nonlinear optimization, and only the solution of an eigenvalue problem is required. Also, they provide a better understanding of what kind of nonlinear features are extracted: the number of the principal components (PCs) in a feature space is fixed a priori by selecting the appropriate kernel function. Therefore, the objective of this work is to use KPCA and KPLS techniques to monitor nonlinear data. The improved FD performance of KPCA and KPLS is illustrated through two simulated examples, one using synthetic data and the other using simulated continuously stirred tank reactor (CSTR) data. The results demonstrate that both KPCA and KPLS methods are able to provide better detection compared to the linear versions.

Keywords: principal component analysis, partial least squares, kernels, fault detection, process monitoring

1. Introduction

Process monitoring is an essential aspect of nearly all industrial processes, often required both to ensure safe operation and to maintain product quality. Process monitoring is generally carried out in two phases: detection and diagnosis. This chapter focuses only on the fault detection aspect. Fault detection methods can be categorized using a number of different methodologies. One popular method of categorization is into quantitative model-based methods, qualitative model-based methods, and data (process history)-based methods [1–3]. **Figure 1** illustrates a general schematic of fault detection phase.

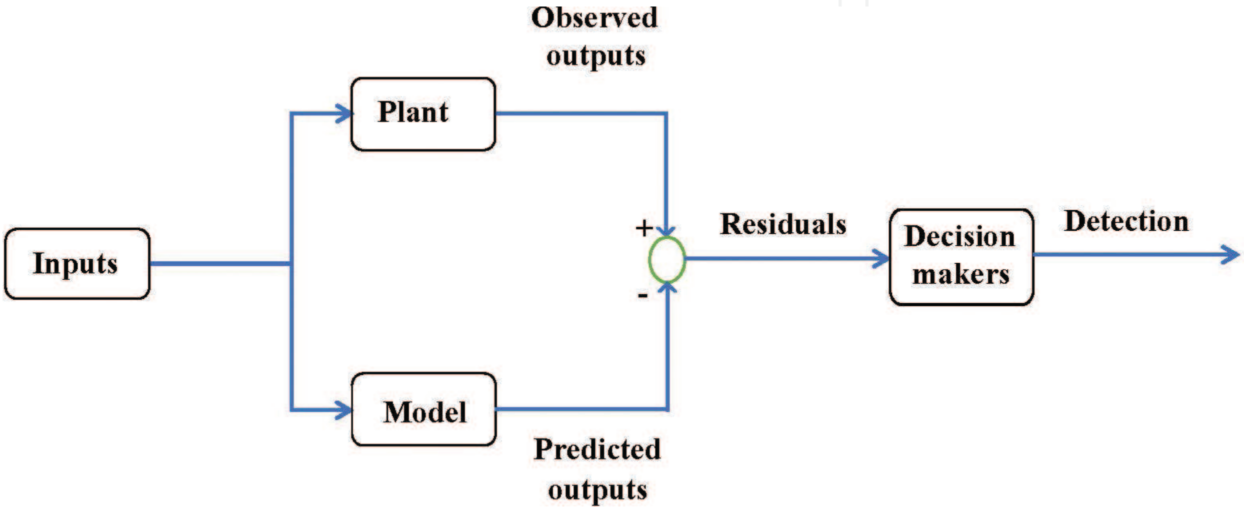

Figure 1. Schematic illustration of detection phase.

Quantitative model-based methods require knowledge of the process model, while qualitative model-based methods require expert knowledge of the given process. Hence, data-based methods are often used as they require neither prior knowledge of the process model nor expert knowledge of the process [4].

Data-based monitoring methods can be further classified into input model-based methods and input-output model-based methods. Input model-based methods only require the data matrix of the input process variables, while input-output model-based methods require both the input and output data matrices in order to formulate a model and carry out fault detection [5]. Input model-based methods are sometimes utilized when the input-output models cannot be formed due to the high dimensionality and complexity of a system being monitored [6]. However, input-output model-based methods do have the added advantage of being able to detect faults in both the process and the variables [5].

Principal component analysis (PCA) is a widely used input model-based method that has been used for monitoring a number of processes including air quality [7], water treatment [8], and semiconductor manufacturing [9]. On the other hand, partial least squares (PLS) are an input-output model-based method that has been applied in chemical processes to monitor online measurement variables and also to monitor and predict the output quality variable [10]. PLS has been applied for the monitoring of distillation columns, batch reactors [11], continuous polymerization processes [12], and other similar industrial processes, which are described by input-output models. However, both PCA and PLS are fault detection techniques that only work reasonably well with linear data. PCA and PLS have been extended to handle nonlinear data by utilizing kernels to transform the data to a higher dimensional space, where linear relationships between variables can be drawn. The extensions kernel principal component analysis (KPCA) and kernel partial least squares (KPLS) have both shown improved performance over the conventional PCA and PLS techniques when handling nonlinear data [5, 13]. T^2 and Q charts are commonly used as fault detection statistics. In the literature, it has been seen that T^2 test is less effective fault detection technique compared to Q statistic; this is because T^2 test can only represent variation of the data in the principle component and not in residue of the model [14].

In our previous works [5, 13, 15], we addressed the problem of fault detection using linear and nonlinear input models (PCA and kernel PCA) and input-output model (PLS and kernel PLS)-based generalized likelihood ratio test (GLRT), in which PCA, kernel PCA, PLS, and kernel PLS methods are used for modeling and the univariate GLRT chart is used for fault detection. In the current work, we propose to use the PCA, kernel PCA, PLS, and kernel PLS methods for multivariate fault detection through their multivariate charts Q and T^2. The fault detection performance is evaluated using two examples, one using simulated synthetic data and the other utilizing a simulated continuous stirred tank reactor (CSTR) model.

The remainder of this chapter is organized as follows. Section 1 introduces linear PCA and PLS, along with the fault detection indices used for these methods. Section 2 then describes the idea of using kernels for nonlinear transformation of data, along with the kernel fault detection extensions: KPCA and KPLS. In Section 3, two illustrative examples are presented, one using simulated synthetic data and the other utilizing a simulated continuous stirred tank reactor. At the end, the conclusions are presented in Section 4.

2. Conventional linear fault detection methods

Before constructing either the PCA or PLS models, data are generally preprocessed to ensure that all process variables in the data matrix are scaled to zero mean and unit variance. This step is essential as different process variables are usually measured with varying standard deviations and means and often using different units.

2.1. Principal component analysis (PCA)

Consider the following input data matrix, $X \in R^{n \times m}$, where m and n represent the number of process variables and the number of observations, respectively. After preprocessing the data, single value decomposition (SVD) can be utilized to express the input data matrix as follows:

$$X = TP^T \tag{1}$$

where $T = [t_1, t_2, t_3 \ldots t_m] \in R^{n \times m}$ is a matrix of the transformed variables, where each column represents the score vectors or the transformed variables, and $P = [p_1, p_2, p_3 \ldots p_m] \in R^{m \times m}$ is a matrix of the orthogonal vectors, where each column is also known as loading vectors, and these are eigenvectors that are associated with the covariance matrix of the input data matrix X. The covariance matrix can be computed as follows [13]:

$$\sum = \frac{1}{n-1} X^T X = P \Lambda P^T \text{ with } PP^T = P^T P = I_m \tag{2}$$

where $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_m)$ is a diagonal matrix that contains the eigenvalues that are related to the m principal components ($\lambda_1 > \lambda_2 > \ldots > \lambda_m$), and I_m is the identity matrix [16]. It should be noted that the model built by PCA uses the same number of principal components as the original number of process variables in the input data matrix (m). However, since many industrial processes may contain process variables that are highly correlated, a smaller number of principal components can be utilized to capture the variation in the process data [6]. The quality of the model built by PCA is dictated by the number of principal components obtained. Overestimating the number could introduce noise that may mask important features in the data, while underestimating the number could decrease the prediction ability of the model [17].

Therefore, selection of the number of principal components is vital, and several methods have been developed for this purpose. A few popular techniques are cumulative percent variance (CPV) [13], scree plot and profile likelihood [18], and cross validation [19]. CPV is commonly utilized due to its computational simplicity and because it provides a good estimate of the number of principal components that need to be retained for most practical applications. CPV can be computed as follows [13]:

$$CPV(l) = \frac{\sum_{i=1}^{l} \lambda_i}{trace(\Sigma)} \times 100 \tag{3}$$

CPV is used to select the smallest number of principal components that represents a certain percentage of the total variance (e.g., 99%). Once the number of principal components to retain is determined, the input data matrix can then be expressed as [13]:

$$X = TP = \left[\widehat{T}\ \tilde{T}\right]\left[\widehat{P}\ \tilde{P}\right]^T \tag{4}$$

where $\widehat{T} \in R^{n \times l}$ and $\tilde{T} = \in R^{n \times m-l}$ represent the matrices containing the l retained principal components and the ignored $(m - l)$ principal components, respectively. Likewise, the matrices that contain the l retained eigenvectors and the ignored $(m - l)$ eigenvectors are represented by $\widehat{P} \in R^{m \times l}$ and $\tilde{P} \in R^{m \times m-l}$, respectively.

After expansion X can be expressed as [13]

$$X = \widehat{T}\widehat{P}^T + \tilde{P}\tilde{T}^T = \overbrace{X\widehat{P}\widehat{P}^T}^{\widehat{X}} + \overbrace{X\left(I_m - \widehat{P}\widehat{P}^T\right)}^{E} \tag{5}$$

where matrix $\widehat{X}$ is the modeled variation of X computed utilizing only the l retained principal components, while matrix E represents the residual space formed by variations that correspond to process noise.

The PCA model can be illustrated as shown in **Figure 2**.

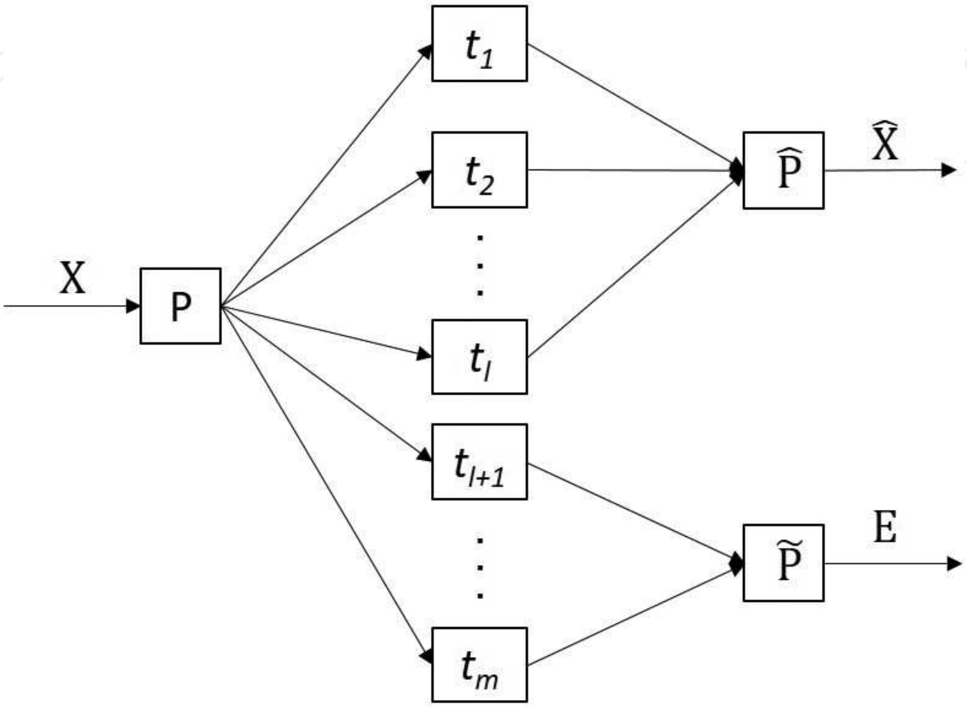

Figure 2. Schematic illustration of PCA.

2.2. Partial least squares (PLS)

PLS is a popular input-output technique used for modeling, regression and as a classification tool, which has been extended to fault detection purpose [20]. PLS includes process variables ($X \in R^{n \times m}$) and the quality variables ($Y \in R^{n \times p}$) with a linear relationship between input and output score vectors. Nonlinear iterative partial least square (NIPALS) algorithm developed by Word et al. is used to compute score matrices and loading vectors [21]:

$$X = TP^T + E = \sum_{i=1}^{M} t_i p_i + E$$
$$Y = UQ^T + F = \sum_{j=1}^{M} u_j q_j^t + F \tag{6}$$

where $E \in R^{n \times m}$ and $F \in R^{n \times p}$ are the PLS model residues; $T \in R^{n \times M}$ and $U \in R^{n \times M}$ are the orthonormal input and output score matrix, respectively; P and Q are the loading vectors of the input (X) and output (Y) matrices, respectively; m and n are the number of process variables and observations in input (X) matrix; p is the number of quality variables in output (Y) matrix; and M is the total number of latent variables extracted. NIPALS method is shown in

Algorithm 1; X and Y matrixes are first standardized by mean centering and unit variance. NIPALS algorithm is initialized by assigning one of the columns of output matrix (Y) as output score vector (u); at each iteration t, u, p, and q are computed and stored; M latent variables are extracted.

Another modification to NIPALS algorithm has been published in the literature [22]. In other work, different variations of PLS technique have been stated. Qin et al. have presented recursive PLS model [23], where PLS model is updated with new training data set; MacGregor et al. [24] have developed multiblock PLS model to monitor subsection of process variables. While for process monitoring of batch processes, PLS has been extended to multiway PLS technique [25] to incorporate past batches in training data set.

PLS being an input-output type model can also be used as a regression tool, to predict quality variable (Y) from online measurement variable (X). From PLS, model input and output matrices are related by

$$Y = BX + G \tag{7}$$

The regression coefficient B is computed as shown in Eq. (8):

$$B = W\left(P^T W\right)^{-1} C^T \tag{8}$$

Substituting weights, loading vector P and constant C from Algorithm 1:

$$B = X^T U\left(T^T X X^T Y\right)^{-1} T^T Y \tag{9}$$

From Eqs. (7) and (9), the output matrix (Y) is predicted as

$$Y_{new} = X_{new} X^T U\left(T^T X X^T U\right)^{-1} T^T Y + G \tag{10}$$

Algorithm 1: Modified NIPALS algorithm

```
1. Initialized output score: $u = y_i$.                                  (11)
2. Weights regressed on X: $w = u^T X / u^T u$.                          (12)
3. Normalizing weight: $w = w / \| w \|$.                                (13)
4. Weights R: $r_1 = w_1$                                                (14)
   $r_j = \prod_{i=1}^{j-1} \left(I_{m\times m} - w_i p_j^T\right) w_j, j > 1.$
5. Input Score vector: $t = Xr / r^T r$.                                 (15)
6. Input loading vector: $p = X t^T / t^T t$.                            (16)
7. Output loading vector: $q = Y t^T / t^T t$.                           (17)
```

8. Output score vector: $u = Yq/q^Tq$. (18)

9. Normalizing weights, loading vectors and scores:

$$p = \frac{p}{norm(p)}, w = w \times norm(p), t = t \times norm(p). \tag{19}$$

10. Input and output matrices are deflated:

$$\begin{aligned} X &= X - tp^T \\ Y &= Y - tq^T \end{aligned} \tag{20}$$

11. Store latent score vectors in T and U, loading vectors in P and Q

12. Repeat steps 2 to 11 until M latent variables are computed

2.3. Fault detection indices

Different fault detection indices can be used for the linear PCA and PLS techniques. The two most popular indices are the T^2 and Q statistics. T^2 measures the variation of the model, while the Q statistic measures the variation in the residual space, and these statistics will be described next.

2.3.1. T^2 *statistic*

The T^2 statistic measures the variation in the principal components at different time samples and is defined as follows [26, 27]:

$$T^2 = X^{\mathbf{T}}\widehat{\mathbf{P}}\widehat{\Lambda}\widehat{\mathbf{P}}^{\mathbf{T}}X \tag{21}$$

where $\widehat{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_l)$ is the diagonal matrix that contains the eigenvalues that are associated with the retained principal components. For testing data, a fault is declared when the T^2 value exceeds the value of the threshold as follows:

$$T^2 \geq T^2_\alpha = \frac{(n^2-1)l}{n(n-l)}F(l, n-l) \tag{22}$$

where α is the level of significance, generally assigned a value between 90 and 99%, and $F(a,\ n-a)$ is the critical value of the Fisher-Snedecor distribution with n and $n{-}a$ degrees of freedom.

2.3.2. *Q statistic*

The Q statistic measures the projection of the data on to the residual subspace and allows the user to measure how well the data fit the PCA model. The Q statistic is defined as follows [16]:

$$Q = \|\tilde{X}\| = \left\|\left(I - \widehat{P}\widehat{P}T\right)X\right\|^2 \tag{23}$$

For testing data, a fault is declared when the threshold value is violated as follows [16]:

$$Q \geq Q_\alpha = \varphi_1 \frac{h_0 c_\alpha \sqrt{2\varphi_2}}{\varphi_1} + 1 + \frac{\varphi_2 h_0 (h_0 - 1)}{\varphi_1^2} \tag{24}$$

where $\varphi_i = \sum_{j=l+1}^{m} \lambda_j^i \;\; i = 1, 2, 3$, $h_0 = 1 - \frac{2\varphi_1\varphi_3}{3\varphi_2^2}$, where c_α is the value obtained from the normal distribution of significance α.

3. Nonlinear fault detection methods using kernel transformations

A popular nonlinear version of PCA and PLS is the projection of nonlinear data to a high-dimensional feature space, where the linear fault detection method is applied in the features space, *F*. The authors in Ref. [28] used projection of *X* for PLS response surface modeling using the quadratic function as the mapping function:

$$\Phi : \chi = R^2 \rightarrow F = R^3 \tag{25}$$

However, it is difficult to know the accurate nonlinear transformation function for nonlinear data matrix to be linear in the feature space. According to Mercer's theorem, orthogonal semi-positive definite function can be used to map the data into the feature space instead of knowing the explicit nonlinear function. This nonlinear function is called the kernel function and is defined as the dot product of the mapped data in the feature space:

$$k\left(X_i, X_j\right) = \Phi(X_i)\Phi\left(X_j\right) \tag{26}$$

Thus, kernel-based multivariate methods can be defined as nonlinear fault detection methods in which the input data matrix is mapped into high-dimensional feature space and developed linear models can be applied in the feature space for fault detection purposes.

Commonly used kernel functions are given below [29]:

Radial basis function (RBF):

$$K(X, Y) = \exp\left(\frac{-\|X - Y\|^2}{c}\right) \tag{27}$$

Polynomial function:

$$K(X, Y) = \langle X, Y \rangle^d \tag{28}$$

Sigmoid function:

$$K(X, Y) = \tanh\left(\beta_0 \langle X, Y \rangle + \beta_1\right) \tag{29}$$

The next section describes the methodology of utilizing kernel transformations to extend linear PCA and PLS to the hyperdimensional space in order to carry out fault detection of nonlinear data.

3.1. Kernel principal component analysis (KPCA)

While PCA seeks to find the principal components by minimizing the data information loss in the input space, KPCA does this in the feature space (F). For KPCA learning using training data, $X_1, X_2, \ldots, X_n \in R^m$, nonlinear mapping gives Φ: $X \in \mathfrak{R}^m \rightarrow Z \in \mathfrak{R}^h$, where input data are extended into the hyperdimensional feature space, where the dimension can be very large and possibly infinite [30].

The covariance in the feature space can be computed as follows [31]:

$$\mathbf{C}^F = \frac{1}{n}\sum_{j=1}^{n}\Phi(X_j)\Phi(X_j)^T \tag{30}$$

Similar to PCA, the principal components in the feature space can be found by diagonalizing the covariance matrix. In order to diagonalize the covariance matrix, it would be necessary to solve the following eigenvalue problem in the feature space [31]:

$$\lambda\mathbf{v} = \mathbf{C}^F\mathbf{v} \tag{31}$$

where $\lambda \geq 0$ and represents the eigenvalues.

In order to solve the eigenvalue problem, the following equation is derived [32]:

$$n\lambda\alpha = K\alpha \tag{32}$$

where K and α are the $n \times n$ kernel matrix and eigenvectors, respectively.

For test vector X, the principal components (t) are extracted projecting $\Phi(X)$ onto the eigenvectors $\mathbf{v}_k$ in the feature space where $k = 1,\ldots,l$:

$$\mathbf{t}_k = \langle \mathbf{v}_k, \Phi(X)\rangle = \sum_{i=1}^{N}\alpha_i^k\langle\Phi(X_i), \Phi(X)\rangle \tag{33}$$

It is important to note that before carrying out KPCA, it is necessary to mean center the data in the high-dimensional space. This can be accomplished by replacing the kernel matrix K with the following [32]:

$$\mathbf{K} = \mathbf{K} - 1_n\mathbf{K} - \mathbf{K}1_n + 1_n\mathbf{K}1_n \tag{34}$$

where $1_n = \frac{1}{n}\begin{vmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{vmatrix} \in R^{n\times n}$.

3.1.1. T^2 *statistic for KPCA*

Variation in the KPCA model can be found using T^2 statistic, which is the sum of normalized squared scores, computed as follows [31]:

$$T^2 = [\mathbf{t}_1, \ldots, \mathbf{t}_l]\Lambda^{-1}[\mathbf{t}_1, \ldots, \mathbf{t}_l]^T \tag{35}$$

where $\mathbf{t}_k$ is obtained from Eq. (33).

The confidence limit is computed as follows [31]:

$$T^2_{l,n,\alpha} \sim \frac{l(n-1)}{n-l} F_{l,n-l,\alpha} \tag{36}$$

3.1.2. Q statistic for KPCA

In order to compute the Q statistic, the feature vector $\Phi(X)$ needs to be reconstructed. This is done by projecting $\mathbf{t}_k$ into the feature space using $\mathbf{v}_k$ as follows [31]:

$$\widehat{\Phi}_n(X) = \sum_{k=1}^{n} \mathbf{t_k v_k} \tag{37}$$

The Q statistic in the feature space can now be computed as [31]

$$Q = \left\|\Phi(X) - \widehat{\Phi}_l(X)\right\|^2 \tag{38}$$

The confidence limit of the Q statistic can then be computed using the following equation [31]:

$$Q_\alpha \sim g\chi_h^2 \tag{39}$$

This limit is based on Box's equation, obtained by fitting the reference distribution obtained using training data, to a weighted distribution. Parameter g is the weight assigned to account for the magnitude of the Q statistic, and h represents the degree of freedom. Considering a and b the estimated mean and variance of the Q statistic, g and h are approximated using $g = b/2a$ and $h = 2a^2/b$.

3.2. Kernel partial least square (KPLS)

The KPLS methodology works by mapping the data matrices into the feature space and then applying the nonlinear partial least square algorithm and computing the loading and score vectors.

The mapped data points are given as

$$\begin{gathered}\Phi : \chi \rightarrow F \\ \chi \rightarrow \left(\sqrt{\lambda_1}\varphi_1(X), \sqrt{\lambda_2}\varphi_2(X), \ldots, \sqrt{\lambda_n}\varphi_n(X)\right)\end{gathered} \tag{40}$$

The kernel gram function can be used to map the data into the feature space instead explicitly using the nonlinear mapping function; this is called the kernel trick. Kernel gram function is defined as the dot product of the mapping function:

$$k\left(X_i, X_j\right) = \Phi(X_i)\Phi\left(X_j\right) \tag{41}$$

As with the KPCA algorithm, the kernel matrix has to be mean centered before applying the NIPALS algorithm using Eq. (34).

The input score matrix and weights are computed as

$$\begin{aligned} t &= \overline{\phi}(X)^T R \\ R &= \Phi^T U\left(T^T \overline{K} U\right)^{-1} \end{aligned} \tag{42}$$

Thus, the score matrix is given as [33]

$$t = K_{tt} Z \tag{43}$$

where $Z = U\left(T^T K U\right)^{-1}$.

Now, the relationship between the input and output score matrices can be derived by combing Eqs. (15), (17), and (18):

$$t = XX^T u \tag{44}$$

In the feature space, replace X by its image Φ:

$$t = \Phi\Phi^T u \tag{45}$$

Substituting the kernel gram function, $K = \Phi\Phi^T$, input and output scores are given by

$$\begin{aligned} t &= Ku \\ u &= Yt \end{aligned} \tag{46}$$

After every iteration, input kernel (K) and output matrix (Y) are deflated as

$$\Phi\Phi^T = \left(\Phi - tt^T\Phi\right)\left(\Phi - tt^T\Phi\right)^T \tag{47}$$

$\Phi\Phi^T$ dot product is replaced by kernel gram function K:

$$\begin{aligned} K &= K - tt^T K - Ktt^T + tt^T Ktt^T \\ Y &= Y - tt^T Y \end{aligned} \tag{48}$$

Let $\{X_i\}_{i=1}^n$ be the training data and $\left\{X_j\right\}_{j=1}^n$ be the testing data, $\Phi(X_i)$ is the mapped training data, and $\Phi(X_j)$ is the mapped testing data. Kernel functions for the testing data are given as

$$\begin{aligned} K_t &= K(X, X_t) = \sum_{i=1}^{n} \sqrt{\lambda_i}\varphi_i(X)\sqrt{\lambda_i}\varphi_i(X_t) = (\Phi(X) \cdot \Phi(X_t)) = \Phi(X)^T\Phi(X_t) \\ K_{tt} &= K(X_t, X_t) = \sum_{i=1}^{n} \sqrt{\lambda_i}\varphi_i(X_t)\sqrt{\lambda_i}\varphi_i(X_t) = (\Phi(X_t) \cdot \Phi(X_t)) = \Phi(X_t)^T\Phi(X_t) \end{aligned} \tag{49}$$

KPLS algorithm can also be used to predict output matrix Y from input matrix X as

$$Y_t = \Phi_t B \tag{50}$$

Φ_t is mapped testing data in feature space from $\{X_j\}_{j=1}^{n}$, and B is the regression coefficient which is given as [34]

$$B = \Phi^T Y \left(T^T K U\right)^{-1} T^T Y \tag{51}$$

Thus combining Eqs. (50) and (51), we get predicted output quality matrix:

$$Y_t = K_t U \left(T^T K U\right)^{-1} T^T Y \tag{52}$$

Algorithm 2: Kernel partial least square (KPLS) algorithm

1. Compute Kernel matrix: K.
2. Kernel matrix is mean centered using Eq. (34).
3. For first iteration, initialized score matrix: $u = y_i$.
4. Calculate scores t and u using Eq. (46).
5. Deflate K and Y, using Eq. (48)
6. Score vectors t and u are stored in cumulative matrix T and U
7. Repeat steps 1 to 6 to extract M latent variables.

3.2.1. T^2 statistic for KPLS

The T^2 statistic for KPLS can be computed as

$$T_t^2 = t^T \Lambda^{-1} t \tag{53}$$

where $\Lambda = (n-1)^{-1} T^T T$ and the score matrix being orthonormal matrix $T^T T = I$, leading to $\Lambda = (n-1)^{-1} I$. The score matrix is $t = K_{tt} Z$; hence the T^2 statistic is given by [33]

$$T_t^2 = (n-1) K_{tt} Z Z^T K_{tt} \tag{54}$$

The threshold value for T^2 statistic is computed using the f-inverse distribution and is given by [35]

$$T_{\alpha}^{2} = g \cdot \mathrm{finv}(\alpha, m, h) \tag{55}$$

where n and m are the total number of observations and variables in the input data matrix X, respectively, and

$$\begin{aligned} g &= \frac{m\left(n^{2}-1\right)}{n(n-m)} \\ h &= n-m \end{aligned} \tag{56}$$

3.2.2. Q statistic for KPLS

As with the other data-based models, the Q statistic computes the mean square error of the residue from the KPLS model:

$$\begin{aligned} Q &= \left\| \overline{\phi} - \overline{\overline{\tilde{\phi}}} \right\|^{2} \\ Q &= \overline{\phi}^{T}\overline{\phi} - 2\overline{\phi}^{T}\overline{\overline{\tilde{\phi}}} + \overline{\overline{\tilde{\phi}}}^{T}\overline{\overline{\tilde{\phi}}} \end{aligned} \tag{57}$$

Substituting the kernel gram functions as the dot product of mapped points:

$$Q = K_{tt} - 2K_{t}^{T}KZt + t^{T}T^{T}KTt \tag{58}$$

where $Z = U\left(T^{T}KU\right)^{-1}$.

The threshold value for the Q statistic under the significance level of α [36] is given by

$$Q_{\alpha} = g\chi_{\alpha}^{2}(h) \tag{59}$$

where g and h are given by

$$\begin{aligned} g &= \frac{variance(Q)}{2 \times mean(Q)} \\ h &= \frac{2 \times (mean(Q))^{2}}{variance(Q)} \end{aligned} \tag{60}$$

A fault is declared in the system if the Q statistic value is higher than threshold value (Q_{α}) for new data set.

The following section demonstrates the implementation of the fault detection methods described above and analyzes the effectiveness of all techniques.

4. Illustrative examples

The effectiveness of the kernel extensions of PCA and PLS for fault detection purposes will be demonstrated through two illustrative nonlinear examples, using a simulated synthetic data set and a simulated continuous stirred tank reactor (CSTR).

4.1. Simulated synthetic data

Synthetic nonlinear data can be simulated through the following model [37]:

$$\begin{aligned} x_1 &= u^2 + 0.3\sin(2\pi u) + \varepsilon_1 \\ x_2 &= u + \varepsilon_2 \\ x_3 &= u^3 + u + 1 + \varepsilon_3 \end{aligned} \tag{61}$$

where u is a variable that is defined between -1 and 1 and ε_i is a variable of independent white noise distributed uniformly between -0.1 and 0.1. Training and testing data sets of 401 observations each are generated using the model above. The performance of KPCA and KPLS techniques is illustrated and compared to the conventional PCA and PLS methods for two different cases. In the first case, the sensor measuring the first variable x_1 is assumed to be faulty with a single fault. In the second case, multiple faults are assumed to occur simultaneously in x_1, x_2, and x_3.

Figure 3 shows the generated data.

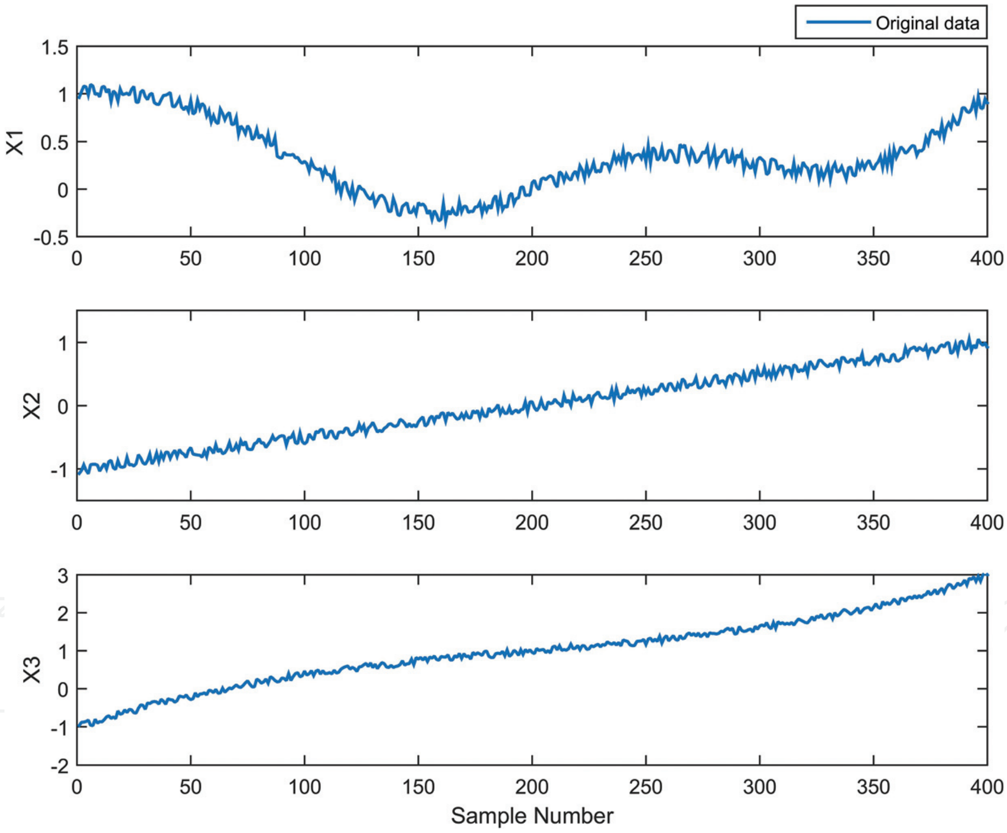

Figure 3. Generated data.

Case 1

In this case, a single fault of magnitude unity is introduced between observations 200 and 250 in x_1 in the testing data set. The Gaussian kernel was chosen to model the nonlinearity in the process data. The most common fault detection metrics used are the missed detection rate, the false alarm rate, and the out-of-control average run length (ARL_1). The missed detection rate is when a fault goes undetected in the faulty region, while the false alarm rate is when an observation is flagged as a fault in the non-faulty region. The false alarm and missed detection rates are also commonly referred to as Type I and Type II errors, respectively. ARL_1 is the number of observations, and it takes for a particular technique to flag a fault in faulty region and is used to assess the speed of a detection. The fault-free and faulty data are shown in **Figures 4** and **5**, respectively.

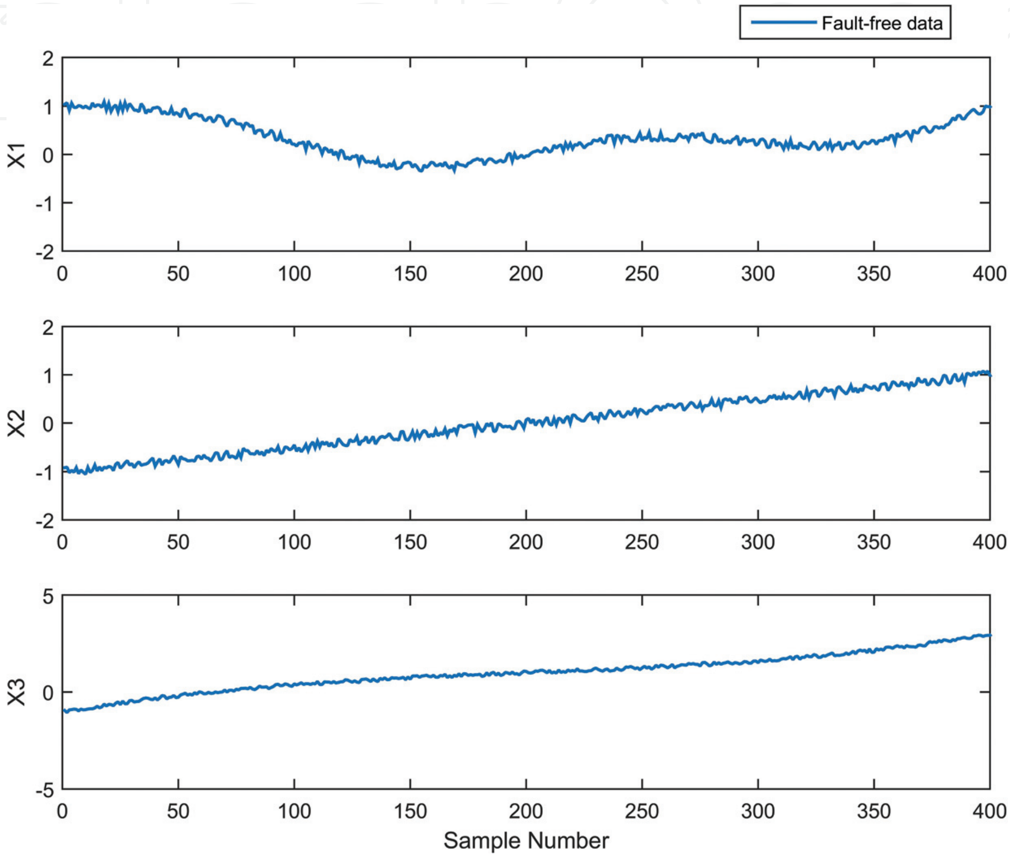

Figure 4. Fault-free data.

The fault detection (FD) performance of PCA-, KPCA-, PLS-, and KPLS-based Q methods is shown in **Figures 6** and **7** as well as **Table 1**. The results show that both KPCA and KPLS-based Q provide a better FD performance than the linear PCA- and PLS-based Q methods and are able to detect the faults with lower missed detection rates, false alarm rates, and ARL_1 values (see **Table 1**).

Case 2

In this case, a multiple faults of magnitude unity are introduced between observations 200 and 250 in x_1, 100 and 150 in x_2, and 385 and 401 in x_3 in the testing data set (as shown in **Figure 8**).

The FD performance of the kernel PCA and kernel PLS methods is illustrated and compared to that of the conventional PCA and PLS methods using the Q statistic. The Q statistic was chosen for analysis, since it is often better able to detect smaller faults using the residual space and for simplicity of analysis as well. The fault detection performance of a particular process monitoring technique can be monitored using multiple fault detection metrics.

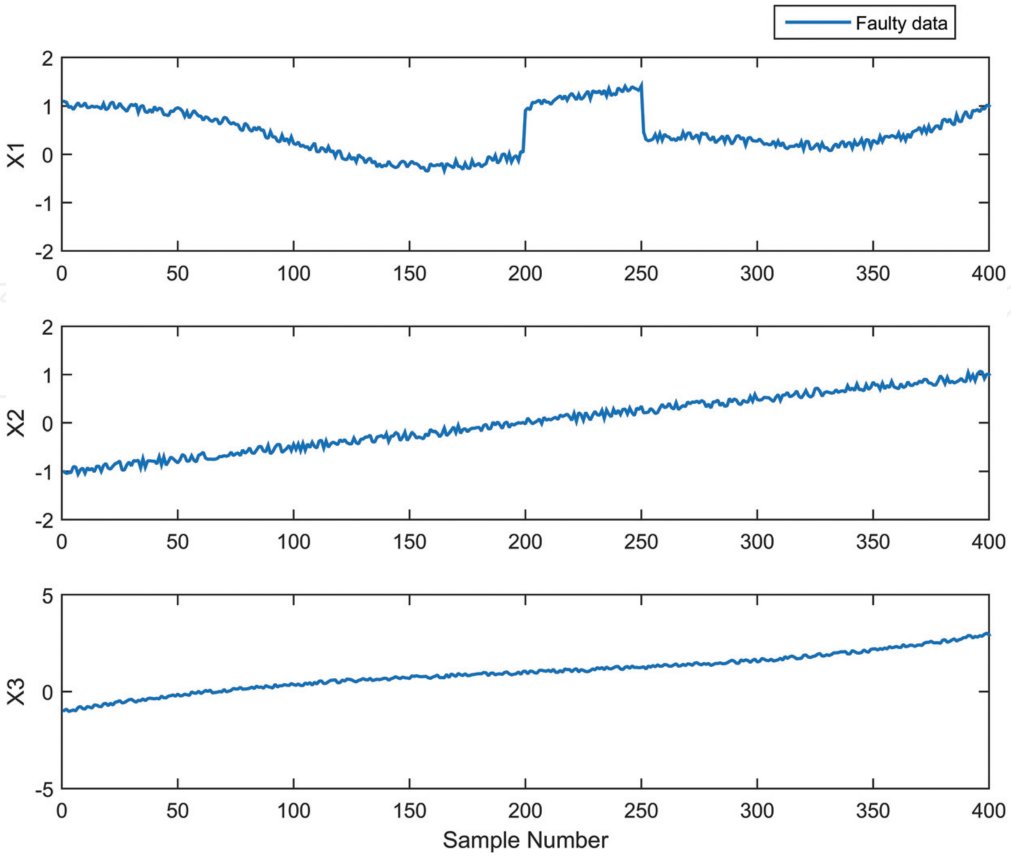

Figure 5. Faulty data in the presence of single fault in x_1.

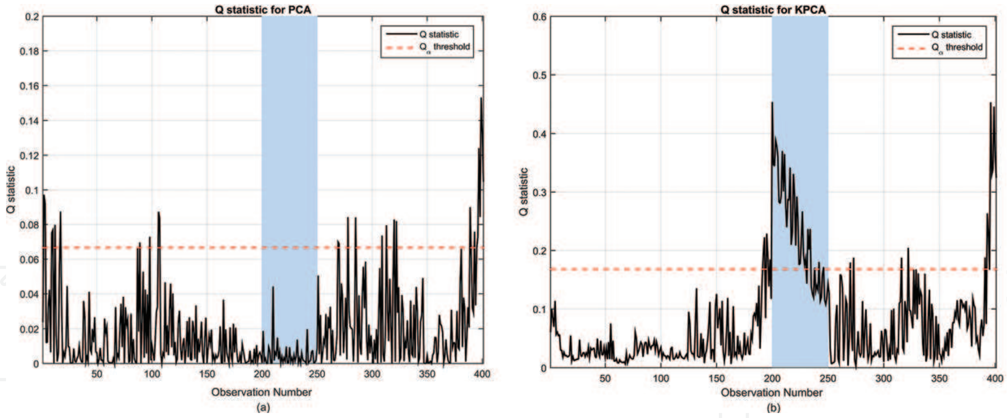

Figure 6. Monitoring single fault using PCA- and KPCA-based Q methods—case 1.

As can be seen through **Figures 9(a)** and **10(a)** and **Table 2**, the conventional linear PCA and PLS techniques are unable to effectively capture the nonlinearity present in the data set, which leads to entire sets of faults going undetected for both the linear PCA and PLS techniques. However, as demonstrated in **Figures 9(b)** and **10(b)**, the KPCA and KPLS-based Q techniques are better able to detect the faults with lower missed detection rates, false alarm rates, and ARL_1 values than the linear PCA and PLS methods (as shown in **Table 2**). These improved results can be attributed to the fact that the kernel techniques are able to capture the nonlinearity

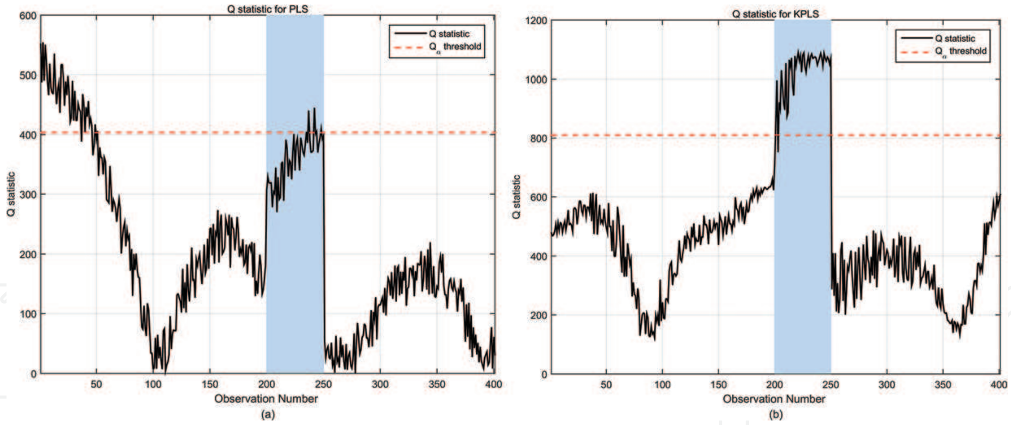

Figure 7. Monitoring single fault using PLS- and KPLS-based Q methods—case 1.

	Missed detection (%)	False alarm (%)	ARL_1
PLS-based Q statistic	90.1961	13.1429	36
KPLS-based Q statistic	3.9216	0	2
PCA-based Q statistic	100	7.4286	-
KPCA-based Q statistic	27.4510	5.1429	1

Table 1. Summary of missed detection (%), false alarms (%), and ARL1 for case 1.

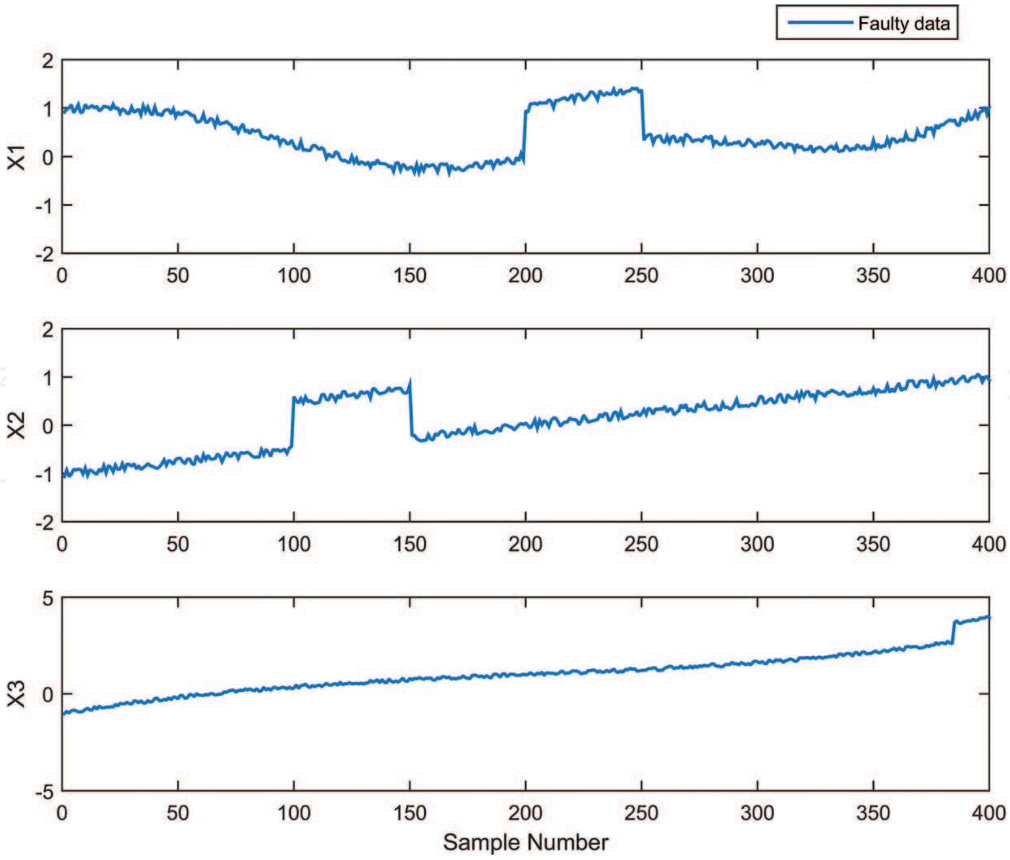

Figure 8. Faulty data in the presence of multiple faults in x_1, x_2, and x_3.

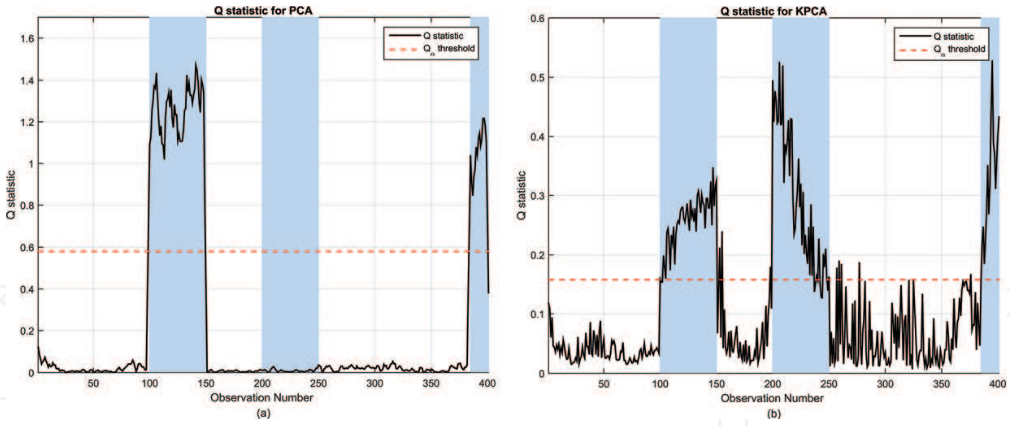

Figure 9. Monitoring multiple faults using PCA- and KPCA-based Q methods using simulated synthetic data—case 2.

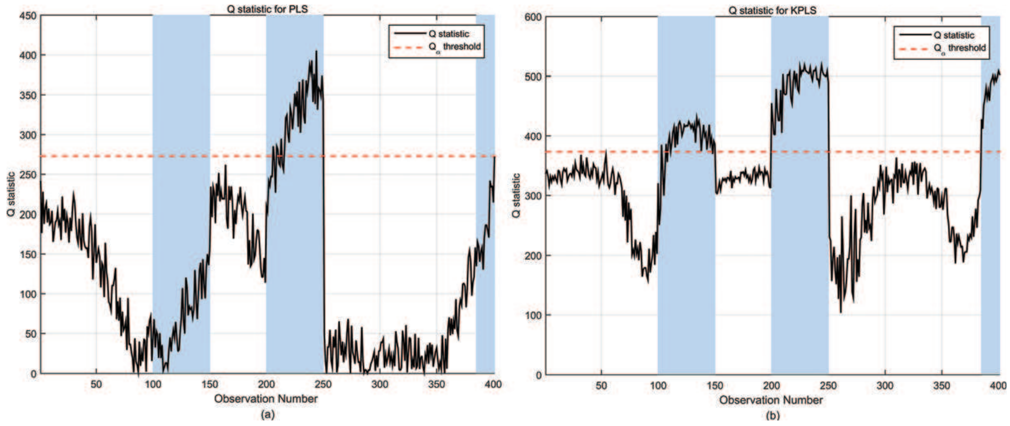

Figure 10. Monitoring multiple faults using PLS- and KPLS-based Q methods using simulated synthetic data—case 2.

	Missed detection (%)	False alarm (%)	ARL_1
PLS-based Q statistic	67.2269	0	58
KPLS-based Q statistic	6.7227	0	1
PCA-based Q statistic	42.8571	0	1
KPCA-based Q statistic	8.4034	2.8369	1

Table 2. Summary of missed detection (%), false alarms (%), and ARL1 for case 2.

in the hyperdimensional feature space, providing better detection especially in this case where there are multiple faults in the system.

4.2. Simulated CSTR model

In order to effectively assess the performance of the kernel PCA and kernel PLS techniques, it is also necessary to examine the performance of the techniques using an actual process application as well. A continuous stirred tank reactor model can be used to generate nonlinear data, and the fault detection charts can be applied to test their performance.

4.2.1. CSTR process description

The dynamic for the CSTR that was utilized for this simulated example is represented as follows [5]:

$$\begin{aligned}
&\frac{\partial C_A}{\partial t} = \frac{F}{V}(C_{A_0} - C_A) - k_0 e^{-E/RT} C_A \\
&\frac{\partial T}{\partial t} = \frac{F}{V}(T_0 - T) + \frac{(-\Delta H)}{\rho C_P} e^{-E/RT} C_A - \frac{q}{V \rho C_p} \\
&q = \frac{aF_c^{b+1}}{F_c + \left(\dfrac{aF_c^{b}}{2\rho_c C_{pc}}\right)} (T - T_{cin})
\end{aligned} \tag{62}$$

where k_0, E, F, and V represent the reaction rate constant, activation energy, flow rates (both inlet and outlet), and reactor volume, respectively. The concentration of A in the inlet stream and of B in the exit stream is represented by C_A and C_B, respectively. The temperatures of the inlet stream and of the cooling fluid in the jacket are T_i and T_j, respectively. ΔH, U, A, ρ, and C_p represent the heat of reaction, overall heat transfer coefficient, area through which the heat transfers to the cooling jacket, density, and heat transfer coefficient of all streams, respectively.

Using the described CSTR model, 1000 observations were generated, which was assumed to be initially noise-free. Zero-mean Gaussian noise with a signal-to-noise ratio of 20 was used to contaminate the noise-free process observations, in order to replicate reality. **Figure 11** shows

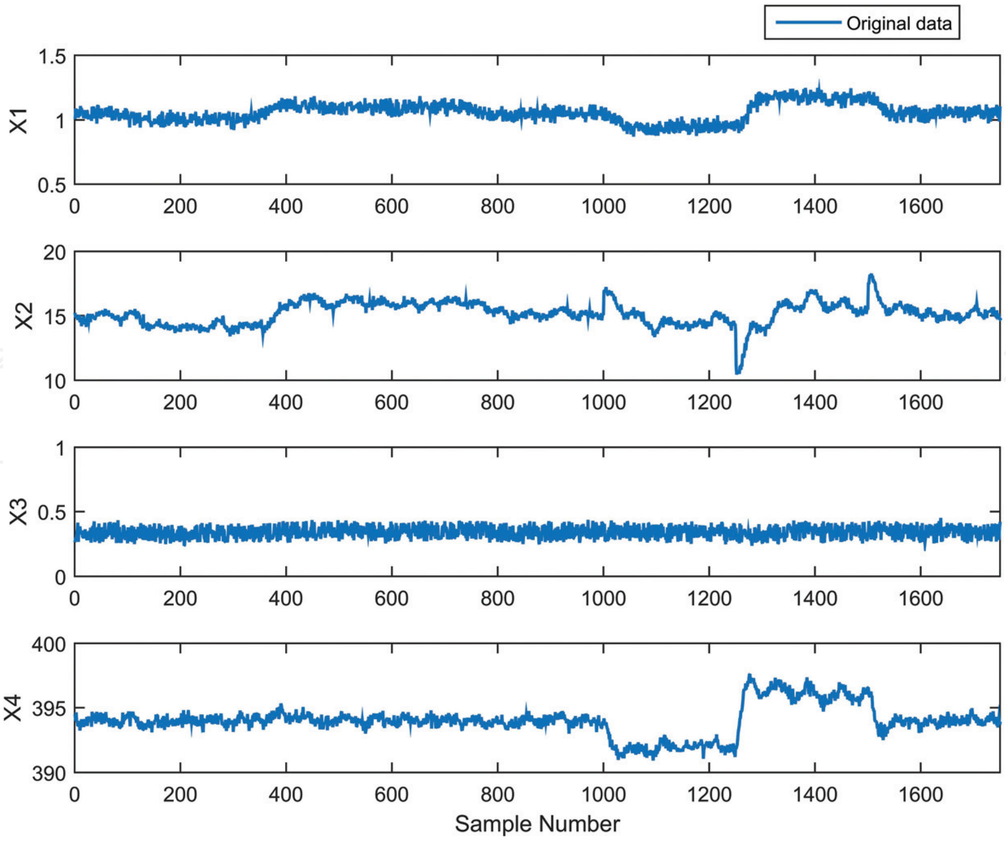

Figure 11. Generated continuously stirred tank reactor (CSTR) data.

the generated CSTR data. This data set was then split into training and testing data sets, of 500 observations each. Faults of magnitude 3σ were added to the temperature and concentration process variables in the testing data set, at three different locations: observations 101–150, 251–350, and 401–450. σ is the standard deviation of that particular process variables. **Figures 12** and **13** show the unfaulty and faulty data, respectively. Similar to the previous example, the performance of kernel PCA and kernel PLS methods is compared to the conventional linear PCA and PLS methods using the Q statistic.

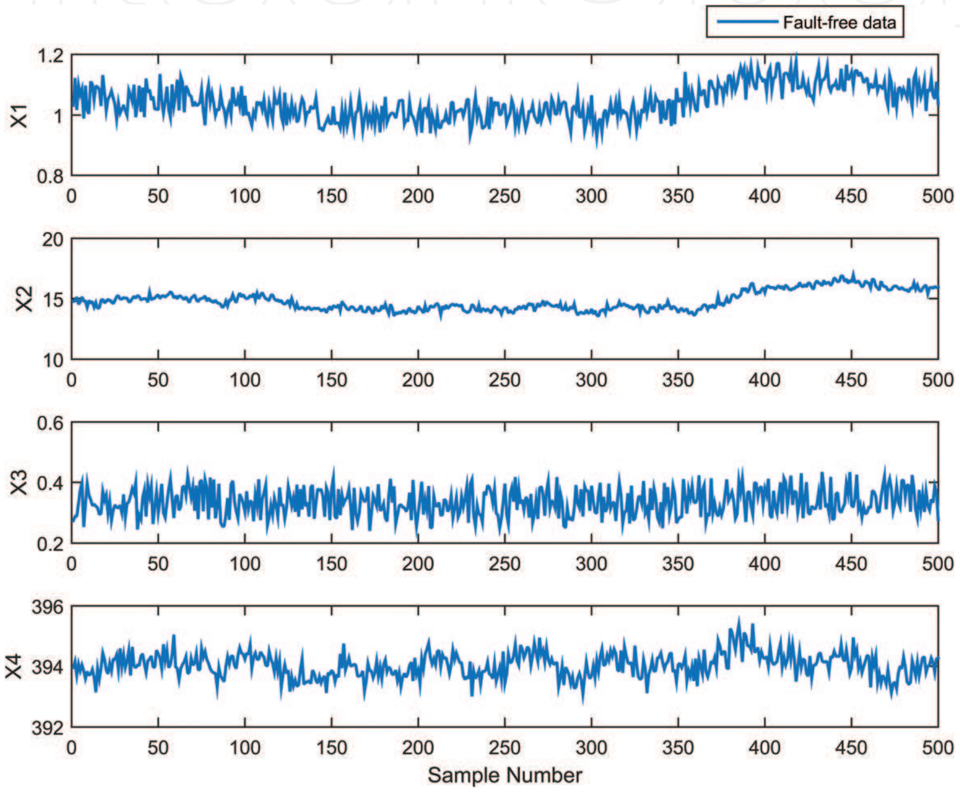

Figure 12. Fault-free data.

For this example, comparing the two conventional techniques, we can see that the PCA-based Q statistic is unable to all faults (see **Figure 14** (**a**)), while the PLS-based Q model is able to better detect the faults (see **Figure 15** (**a**)). However, the kernel PCA and kernel PLS-based Q techniques are able to provide result charts with lower missed detection rates, false alarm rates, and ARL_1 values than their corresponding conventional techniques (see **Figures 14(b)** and **15(b)**). These improved results can once again be attributed to the kernel techniques being able to effectively capture the nonlinearity of the data in the hyperdimensional feature space. The FD results using the two examples showed that the kernel PLS-based Q provides a relative performance compared to the kernel PCA Q. This is because kernel PCA is an input space model and cannot take into consideration outcome measures and most chemical processes or many of them are usually described by input-output space models.

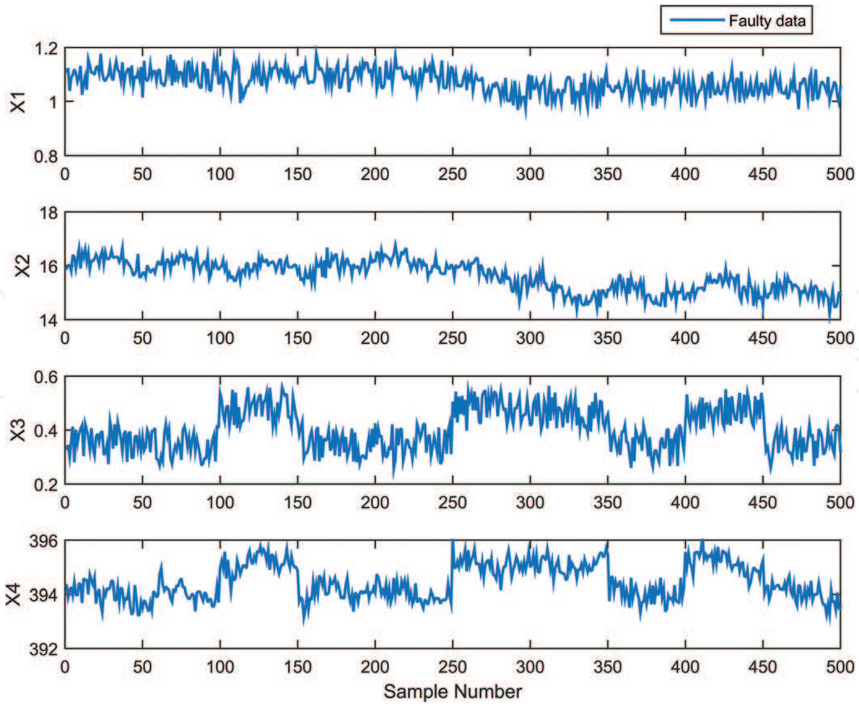

Figure 13. Faulty data in the presence of multiple faults in temperature and concentration.

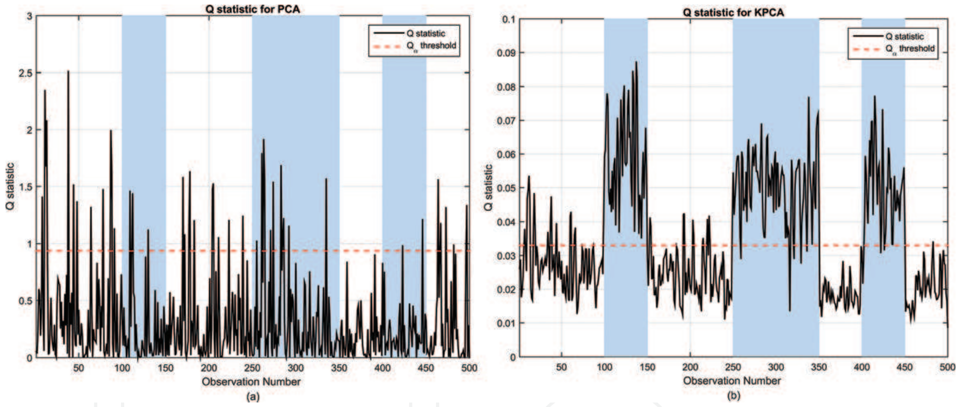

Figure 14. Fault detection using PCA- and kernel PCA-based Q methods using CSTR data.

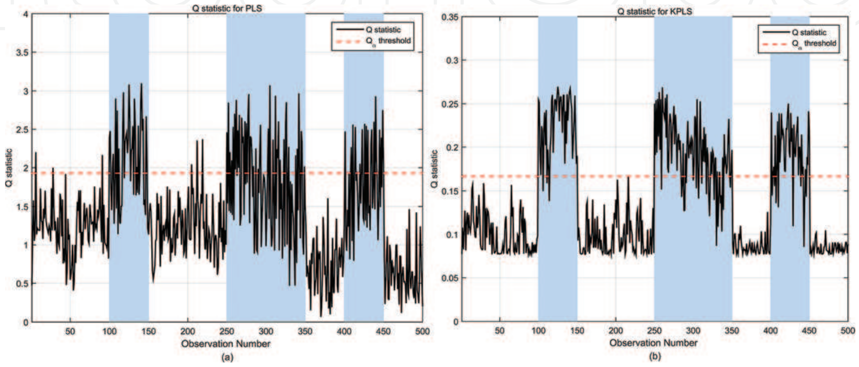

Figure 15. Fault detection using PLS- and kernel PLS-based Q methods using CSTR data.

5. Conclusion

In this chapter, a nonlinear multivariate statistical techniques are used for fault detection. Kernel PCA and kernel PLS have been widely used to monitor various nonlinear processes, such as distillation columns and reactors. Thus, in the current work, both kernel PCA and kernel PLS methods are used for nonlinear fault detection of chemical process. A commonly used fault detection index is Q-square statistic, and it is used to detect fault in the system. The fault detection performance using linear and nonlinear input models (PCA and kernel PCA) and input-output models (PLS and kernel PLS) is evaluated through two simulated examples, synthetic data set and continuous stirred tank reactor (CSTR). Missed detection rate, false alarm rate, and ARL_1 are the parameters used to compare the fault detection techniques. The results of the two case studies showed that the kernel PCA and kernel PLS-based Q provide improved fault detection performance compared to the conventional PCA- and PLS-based Q methods.

6. Acknowledgements

This work was made possible by NPRP grant NPRP7-1172-2-439 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Author details

Mohammed Ziyan Sheriff[1,2], Chiranjivi Botre[1], Majdi Mansouri[3], Hazem Nounou[3], Mohamed Nounou[2]* and Mohammad Nazmul Karim[1]

*Address all correspondence to: mohamed.nounou@qatar.tamu.edu

1 Artie McFerrin Department of Chemical Engineering, College Station, TX, USA

2 Chemical Engineering Program, Texas A&M University at Qatar, Doha, Qatar

3 Electrical and Computer Engineering Program, Texas A&M University at Qatar, Doha, Qatar

References

[1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, “A review of process fault detection and diagnosis part I: Quantitative model-based methods,” *Comput. Chem. Eng.*, vol. 27, pp. 293–311, 2003.

[2] V. Venkatasubramanian, R. Rengaswamy, and S. N. Ka, "A review of process fault detection and diagnosis part II: Qualitative models and search strategies," *Comput. Chem. Eng.*, vol. 27, pp. 313–326, 2003.

[3] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part III: Process history based methods," *Comput. Chem. Eng.*, vol. 27, pp. 327–346, 2003.

[4] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Ind. Eng. Chem. Res.*, vol. 52, no. 10, pp. 3543–3562, 2013.

[5] M. Mansouri, M. Nounou, H. Nounou, and N. Karim, "Kernel PCA-based GLRT for nonlinear fault detection of chemical processes," *J. Loss Prev. Process Ind.*, vol. 40, pp. 334–347, 2016.

[6] Jolliffe, Ian. *Principal component analysis*. John Wiley & Sons, Ltd., 2002.

[7] M. F. Harkat, G. Mourot, and J. Ragot, "An improved PCA scheme for sensor FDI: Application to an air quality monitoring network," *J. Process Control*, vol. 16, no. 6, pp. 625–634, 2006.

[8] J. P. George, Z. Chen, and P. Shaw, "Fault detection of drinking water treatment process using PCA and Hotelling's T 2 chart," pp. 970–975, 2009.

[9] J. Yu, "Fault detection using principal components-based gaussian mixture model for semiconductor," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 3, pp. 432–444, 2011.

[10] Y. Zhang, W. Du, Y. Fan, and L. Zhang, "Process fault detection using directional kernel partial least squares," *Ind. Eng. Chem. Res.*, vol. 54, no. 9, pp. 2509–2518, 2015.

[11] P. Nomikos and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 97–108, 1995.

[12] T. Kourti and J. F. J. F. MacGregor, "Process analysis, monitoring and diagnosis, using multivariate projection methods," *Chemom. Intell. Lab. Syst.*, vol. 28, pp. 3–21, 1995.

[13] M. Mansouri, M. Z. Sheriff, R. Baklouti, M. Nounou, H. Nounou, A. Ben Hamida, and M. N. Karim, "Statistical fault detection of chemical process - comparative studies," *J. Chem. Eng. Process Technol.*, vol. 7, no. 1, pp. 1–10, 2016.

[14] S. Joe Qin, "Statistical process monitoring: basics and beyond," *J. Chemom.*, vol. 17, no. 8–9, pp. 480–502, 2003.

[15] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associted with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[16] J. E. Jackson "Quality control methods for several related variables," *Technometrics*, vol. 1, no. 4, pp. 359–377, 1959.

[17] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Comput. Stat. Data Anal.*, vol. 51, no. 2, pp. 918–930, 2006.

[18] G. Diana and C. Tommasi, "Cross-validation methods in principal component analysis: A comparison," *Stat. Methods Appl.*, vol. 11, no. 1, pp. 71–82, 2002.

[19] R. Rosipal and L. J. Trejo, "Kernel partial least squares regression in reproducing kernel hilbert space," *J. Mach. Learn. Res.*, vol. 2, pp. 97–123, 2002.

[20] P. Geladi and B. R. Kowalski, "Partial least-squares regression: A tutorial," *Anal. Chim. Acta*, vol. 185, pp. 1–17, 1986.

[21] B. S. Dayal and J. F. MacGregor, "Recursive exponentially weighted PLS and its applications to adaptive control and prediction," *J. Process Control*, vol. 7, no. 3, pp. 169–179, 1997.

[22] S. J. Qin, "Recursive PLS algorithms for adaptive data modeling," *Comput. Chem. Eng.*, vol. 22, no. 4–5, pp. 503–514, 1998.

[23] J. F. MacGregor, C. Jaeckle, C. Kiparissides, and M. Koutoudi, "Process monitoring and diagnosis by multiblock PLS methods," *AIChE J.*, vol. 40, no. 5, pp. 826–838, 1994.

[24] T. Kourti, P. Nomikos, and J. F. MacGregor, "Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS," *J. Process Control*, vol. 5, no. 4, pp. 277–284, 1995.

[25] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441, 1933.

[26] U. Krüger and L. Xie, *Statistical monitoring of complex multivariate processes: With applications in industrial process control*. Chichester, West Sussex; Hoboken, N.J: Wiley, 2012.

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

[28] S. Wold, N. Kettaneh-Wold, and B. Skagerberg, "Nonlinear PLS modeling," *Chemom. Intell. Lab. Syst.*, vol. 7, no. 1–2, pp. 53–65, 1989.

[29] Rosipal, R. (2011). Nonlinear partial least squares: An overview. In Lodhi H. and Yamanishi Y. (Eds.), *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques,* pp. 169–189, 2011. ACCM, IGI Global. Retrieved from http://aiolos.um.savba.sk/~roman/Papers/npls_book11.pdf

[30] S. W. Choi, C. Lee, J. M. Lee, J. H. Park, and I. B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemom. Intell. Lab. Syst.*, vol. 75, no. 1, pp. 55–67, 2005.

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

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

[33] J. L. Godoy, D. A. Zumoffen, J. R. Vega, and J. L. Marchetti, "New contributions to non-linear process monitoring through kernel partial least squares," *Chemom. Intell. Lab. Syst.*, vol. 135, pp. 76–89, 2014.

[34] R. Rosipal, "Kernel partial least squares for nonlinear regression and discrimination," *Neural Netw. World*, vol. 13, no. 3, pp. 291–300, 2003.

[35] M.-F. Harkat, S. Djelel, N. Doghmane, and M. Benouaret, "Sensor fault detection, isolation and reconstruction using nonlinear principal component analysis," *Int. J. Autom. Comput*., vol. 4, no. 2, pp. 149–155, 2007.

[36] S. A. Vejtasa and R. A. Schmitz, "An experimental study of steady state multiplicity and stability in an adiabatic stirred reactor," *AIChE J*., vol. 16, no. 3, pp. 410–419, 1970.

[37] Botre, Chiranjivi, et al. "Kernel PLS-based GLRT method for fault detection of chemical processes," *J. Loss Prevent. Process Industries*, 43 (2016): 212–224.