

Sparse Kernel Locality Preserving Projection and Its Application in Nonlinear Process Fault Detection*

DENG Xiaogang (邓晓刚)** and TIAN Xuemin (田学民)

College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China

Abstract Locality preserving projection (LPP) is a newly emerging fault detection method which can discover local manifold structure of a data set to be analyzed, but its linear assumption may lead to monitoring performance degradation for complicated nonlinear industrial processes. In this paper, an improved LPP method, referred to as sparse kernel locality preserving projection (SKLPP) is proposed for nonlinear process fault detection. Based on the LPP model, kernel trick is applied to construct nonlinear kernel model. Furthermore, for reducing the computational complexity of kernel model, feature samples selection technique is adopted to make the kernel LPP model sparse. Lastly, two monitoring statistics of SKLPP model are built to detect process faults. Simulations on a continuous stirred tank reactor (CSTR) system show that SKLPP is more effective than LPP in terms of fault detection performance.

Keywords nonlinear locality preserving projection, kernel trick, sparse model, fault detection

1 INTRODUCTION

With the increased demands on plant safety and quality consistency, fault detection technique has been one of the most fascinating topics in industrial process control field. In the past years several fault detection methods have been developed which can be classified into three major categories: analytical methods, knowledge-based methods, data-driven methods. Particularly, data-driven methods have attracted great attentions from both academic researchers and process engineers for large amounts of data are available for modern computer control systems [1, 2]. Principal component analysis (PCA) and partial least squares (PLS) are the most classical data-driven methods and many extended methods have been developed including neural PCA [3], multiway PCA [4], kernel PLS [5], multiscale kernel PCA [6]. Although PCA, PLS and their extended methods have been widely used in the fault detection field, these methods are based on the global structure analysis of the process data but omit the local structure information between neighbor data points which is very important for process monitoring.

Recently, some manifold learning methods such as locality preserving projection (LPP) [7], locally linear embedding (LLE) [8], and local tangent space alignment (LTSA) [9] have been presented to analyze the local manifold structure of data set. Among these methods, locality preserving projection (LPP) is a well-known local structure analysis method for it can be simply used onto new data. Now, LPP has been introduced for process monitoring and fault detection. Hu *et al.* [10] firstly proposed multiway LPP (MLPP) method for batch process fault detection. Yu [11] used LPP for bearing performance degradation assessment. Zhang *et al.* [12] combined LPP and PCA for process monitoring. In these studies, LPP has been thought as

a promising fault detection method. However, LPP is in nature a linear transformation, which may lead to poor monitoring performance for complicated nonlinear industrial process.

In this paper, we propose an improved nonlinear LPP method called sparse kernel LPP (SKLPP) for effective nonlinear process monitoring. Based on the LPP model, the kernel trick is used to perform nonlinear mapping for the construction of kernel LPP model. Furthermore, in order to reduce the complexity of kernel model computation, feature samples selection technique is applied to build sparse kernel LPP model. Lastly, a simulated continuous stirred tank reactor (CSTR) system is used to demonstrate the effectiveness of the proposed method.

2 LOCALITY PRESERVING PROJECTION

Locality preserving projection (LPP) is a novel linear dimensionality reduction (DR) technique proposed by He and Niyogi in the manifold learning field [6]. Different from traditional DR methods such as principal component analysis (PCA) which aims at capturing the global structure of the data, LPP optimally preserves the neighborhood structure and reveals the intrinsic manifold structure of the data more effectively.

The optimization procedure of LPP includes two parts: firstly an adjacency graph is built to describe the neighborhood relationship between the data points, and then LPP projection vectors are optimized such that the neighboring points in original space remain close to each other in the low-dimensional embedding space. The details of LPP algorithm is shown as follows.

Given the high-dimensional input data matrix $X = [x_1 \ x_2 \ \cdots \ x_n]^T \in \mathbf{R}^{n \times m}$, which represents n samples

Received 2012-05-07, accepted 2012-07-16.

* Supported by the National Natural Science Foundation of China (61273160), the Natural Science Foundation of Shandong Province of China (ZR2011FM014) and the Fundamental Research Funds for the Central Universities (10CX04046A).

** To whom correspondence should be addressed. E-mail: dengxiaogang@upc.edu.cn

of m measured variables, its adjacency graph and corresponding weight matrix are firstly built. In the adjacency graph, if nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighborhood points, they are connected by an edge and the corresponding weight value on this edge is noted as w_{ij} . Contrarily if two nodes are not neighborhood points, there is no edge between them and the corresponding weight value is zero. The definition of "neighborhood" can be expressed by K nearest neighbors (KNN). KNN puts a neighborhood edge between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\mathbf{x}_i$ is among K nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among K nearest neighbors of $\mathbf{x}_i$. With the neighborhood information from the adjacency graph, a usual way to select weight parameter w_{ij} can be given as [7]

$$w_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ is connected} \\ 0, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ is not connected} \end{cases} \quad (1)$$

The adjacency graph with the weight values can be viewed as a geometric description of data set. Based on the adjacency graph, LPP seeks a linear transformation $\mathbf{y}_i = \mathbf{x}_i^T \mathbf{a}$ to project $\mathbf{X}$ into a low-dimensional embedding subspace $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_n]^T \in \mathbf{R}^{n \times 1}$ so that the local neighborhood structure between data points is preserved. A reasonable criterion for choosing $\mathbf{a}$ is to minimize the next objective function [13]:

$$\min_{\mathbf{a}} \sum_{i=1}^n \sum_{j=1}^n (y_i - y_j)^2 w_{ij} \quad (2)$$

For a clearer matrix formulation, Eq. (2) can be reformulated as

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^n (y_i - y_j)^2 w_{ij} &= \sum_{i=1}^n \sum_{j=1}^n (\mathbf{x}_i^T \mathbf{a} - \mathbf{x}_j^T \mathbf{a})^2 w_{ij} \\ &= \mathbf{a}^T \mathbf{X}^T (\mathbf{D} - \mathbf{W}) \mathbf{X} \mathbf{a} \\ &= \mathbf{a}^T \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{a} \end{aligned} \quad (3)$$

where $\mathbf{W}$ is the weight matrix whose (i, j) th element is

w_{ij} , $\mathbf{D}$ is a diagonal matrix as $d_{ii} = \sum_{j=1}^n w_{ij}$ whose en-

tries are column sum of $\mathbf{W}$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is usually called the Laplacian matrix. To avoid the trivial solution, a constraint is imposed as

$$\mathbf{a}^T \mathbf{X}^T \mathbf{D} \mathbf{X} \mathbf{a} = 1 \quad (4)$$

Combining Eqs. (3) and (4), the optimization problem in LPP is expressed as

$$\min_{\mathbf{a}} \mathbf{a}^T \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{a} \quad \text{s.t.} \quad \mathbf{a}^T \mathbf{X}^T \mathbf{D} \mathbf{X} \mathbf{a} = 1 \quad (5)$$

The above problem can be solved by the generalized

eigenvalue decomposition as

$$\mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{a} = \lambda \mathbf{X}^T \mathbf{D} \mathbf{X} \mathbf{a} \quad (6)$$

The solutions of Eq. (6) bring a series of eigenvectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ ordered by their corresponding eigenvalues $\lambda_1 < \lambda_2 < \dots < \lambda_m$. The dimensionality of the problem can be reduced by only retaining the first p ($p \leq m$) eigenvectors. For any test vector $\mathbf{x}_t$, the project vectors can be computed by

$$y_{t,j} = \mathbf{x}_t^T \mathbf{a}_j, \quad j = 1, 2, \dots, m \quad (7)$$

More detailed descriptions about the LPP algorithm can be found in [7, 13].

3 SPARSE KERNEL LPP

LPP is in essence a linear method and may not effectively capture the nonlinear relationships in realistic complex processes. To tackle nonlinear monitoring problem, in this section we apply kernel trick and feature samples selection technique to construct a nonlinear LPP, referred to as sparse kernel LPP (SKLPP).

3.1 Kernel LPP

In order to build a nonlinear LPP algorithm, we suppose that a nonlinear mapping $\Phi: \mathbf{x} \in \mathbf{R}^{m \times 1} \rightarrow \Phi(\mathbf{x}) \in \mathbf{F}$ maps the training data onto a linear feature space $\mathbf{F}$. In the feature space, we apply LPP to build the next objective function

$$\min_{\mathbf{a}} \mathbf{a}^T \Phi(\mathbf{X})^T \mathbf{L} \Phi(\mathbf{X}) \mathbf{a} \quad \text{s.t.} \quad \mathbf{a}^T \Phi(\mathbf{X})^T \mathbf{D} \Phi(\mathbf{X}) \mathbf{a} = 1 \quad (8)$$

The projection vector $\mathbf{a}$ can be spanned using training samples as [14, 15]

$$\mathbf{a} = \sum_{j=1}^n v_j \Phi(\mathbf{x}_j) = \Phi(\mathbf{X})^T \mathbf{v} \quad (9)$$

where $\mathbf{v}$ is the coefficient vector.

Combining Eqs. (8) and (9), we obtain the generalized eigenvalue problem:

$$\begin{aligned} \Phi(\mathbf{X}) \Phi(\mathbf{X})^T \mathbf{L} \Phi(\mathbf{X}) \Phi(\mathbf{X})^T \mathbf{v} = \\ \lambda \Phi(\mathbf{X}) \Phi(\mathbf{X})^T \mathbf{D} \Phi(\mathbf{X}) \Phi(\mathbf{X})^T \mathbf{v} \end{aligned} \quad (10)$$

To avoid the difficulty of explicitly defining the nonlinear mapping, the well-known kernel trick is applied [16]. With the introduction of a kernel function, the vector dot products in a feature space can be computed by the kernel function in the original space as $\Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j)^T = \mathbf{K}$, where $\mathbf{K}$ is the kernel matrix and its element is calculated with kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$. There exist a number of kernel functions that can be utilized for this purpose,

including polynomial kernel, sigmoid kernel and radial basis kernel. In this application a radial basis function $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)$ is selected as the kernel function.

With the help of kernel trick, we rewrite the eigenvalue problem Eq. (10) as

$$KLK\mathbf{v} = \lambda KDK\mathbf{v} \quad (11)$$

By solving Eq. (11) we obtain the eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ corresponding to their eigenvalues $\lambda_1 < \lambda_2 < \dots < \lambda_n$. For any test vector $\mathbf{x}_t$, the projection is extracted as

$$y_{t,j} = \Phi(\mathbf{x}_t)^T \mathbf{a}_j = \mathbf{K}_t \mathbf{v}_j, \quad j = 1, 2, \dots, n \quad (12)$$

where $\mathbf{K}_t = [k(\mathbf{x}_t, \mathbf{x}_1) \ k(\mathbf{x}_t, \mathbf{x}_2) \ \dots \ k(\mathbf{x}_t, \mathbf{x}_n)]$.

3.2 Sparse kernel model representation

The training stage of KLPP requires storing and manipulating the kernel matrix $\mathbf{K}$, the size of which is the square of the sample number. When the sample number n becomes large, the calculation of eigenvalues and eigenvectors will be time consuming and lead to great computational cost. In order to reduce the computational complexity, a feature samples selection (FSS) scheme is adopted to build sparse kernel model in this section.

In fact, a subset of training samples set $\mathbf{X}_s = [\mathbf{x}_{s,1}, \mathbf{x}_{s,2}, \dots, \mathbf{x}_{s,n_s}]^T$ is sufficient to construct the basis space of the projection vectors in Eq. (9), where $n_s (n_s \leq n)$ is the number of selected samples [17]. The samples in subset are often called as feature samples. Based on feature samples, Eq. (9) is rewritten as

$$\mathbf{a} = \sum_{j=1}^{n_s} v_j \Phi(\mathbf{x}_{s,j}) = \Phi(\mathbf{X}_s)^T \mathbf{v} \quad (13)$$

With application of samples subset, Eqs. (10), (11), (12) are redefined as

$$\begin{aligned} \Phi(\mathbf{X}_s) \Phi(\mathbf{X})^T L \Phi(\mathbf{X}) \Phi(\mathbf{X}_s)^T \mathbf{v} = \\ \lambda \Phi(\mathbf{X}_s) \Phi(\mathbf{X})^T D \Phi(\mathbf{X}) \Phi(\mathbf{X}_s)^T \mathbf{v} \end{aligned} \quad (14)$$

$$\mathbf{K}_s^T L \mathbf{K}_s \mathbf{v} = \lambda \mathbf{K}_s^T D \mathbf{K}_s \mathbf{v} \quad (15)$$

$$y_{t,j} = \Phi(\mathbf{x}_t)^T \mathbf{a}_j = \mathbf{K}_{st} \mathbf{v}_j, \quad j = 1, 2, \dots, n_s \quad (16)$$

where $\mathbf{K}_s$ is a kernel matrix defined as $\mathbf{K}_s = \Phi(\mathbf{X}) \Phi(\mathbf{X}_s)^T$, $\mathbf{K}_{st}$ is a kernel vector denoted as $\mathbf{K}_{st} = [k(\mathbf{x}_t, \mathbf{x}_{s,1}) \ k(\mathbf{x}_t, \mathbf{x}_{s,2}) \ \dots \ k(\mathbf{x}_t, \mathbf{x}_{s,n_s})]$.

According to Eq. (15), the matrix dimension for eigenvalue decomposition is reduced to $n_s \times n_s$, which

is less than $n \times n$. The dimension of $\mathbf{K}_{st}$ in Eq. (16) is $1 \times n_s$ which is also less than the dimension of $\mathbf{K}_t$ in Eq. (12). So kernel LPP model becomes sparse with use of feature samples, which leads to a lower computation cost.

The feature samples can be selected using FSS as follows. Given subset $\mathbf{X}_s$, any vector $\Phi(\mathbf{x}_i), i = 1, \dots, n$ could be estimated as a linear combination of $\Phi(\mathbf{X}_s)$ as

$$\hat{\phi}_i = \Phi_s^T \beta_i \quad (17)$$

where β_i is the coefficient vector denoted as $\beta_i = [\beta_{i,1} \ \beta_{i,2} \ \dots \ \beta_{i,n_s}]^T$, Φ_s is the abbreviation of $\Phi(\mathbf{X}_s)$, $\hat{\phi}_i$ is the estimation of $\Phi(\mathbf{x}_i)$.

FSS is to find the coefficients β_i such that the estimated mapping $\hat{\phi}_i$ is as close as possible to the real mapping $\Phi(\mathbf{x}_i)$, which is also abbreviated as ϕ_i . This leads to a problem of minimizing the normalized Euclidean distance of the following form:

$$\delta_i = \frac{\|\phi_i - \hat{\phi}_i\|^2}{\|\phi_i\|^2} \quad (18)$$

According to $(\partial \delta_i / \partial \beta_i) = 0$ we can compute the optimal coefficient as

$$\beta_i = (\Phi_s^T \Phi_s)^{-1} \Phi_s^T \phi_i \quad (19)$$

So the minimized δ_i can be listed as

$$\delta_{i,\min} = 1 - \frac{\mathbf{K}_{si}^T \mathbf{K}_s^{-1} \mathbf{K}_{si}}{k_{ii}} \quad (20)$$

where $\mathbf{K}_s$ is a square kernel matrix for the selected samples, $\mathbf{K}_{si}$ is a column vector of dot product between $\Phi(\mathbf{x}_i)$ and the selected vector set $\Phi(\mathbf{X}_s)$, k_{ii} is the vector of dot product between $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_i)$.

For all training samples, Eq. (20) can be also transformed as a maximization optimization expressed by

$$J_{\max} = \max_{\mathbf{X}_s} \left(\sum_{\mathbf{x}_i \in \mathbf{X}} \frac{\mathbf{K}_{si}^T \mathbf{K}_{ss}^{-1} \mathbf{K}_{si}}{k_{ii}} \right) \quad (21)$$

An iterative sequential forward selection process is often designed to solve Eq. (21) for obtaining a feature samples subset. At each step, a sample vector is selected such that it gives the maximum fitness function $J_{\max}$ combining with the previously selected vectors. The selection process stops when $\mathbf{K}_{ss}$ is no longer invertible or a predefined number of selected vectors is reached. The detailed FSS steps can be found in [18].

4 PROCESS MONITORING BASED ON SKLPP

In order to detect process faults using SKLPP, two monitoring statistics are built which include T^2 and Q statistics. For any test vector $\mathbf{x}_t$, the T^2 statistic is applied to measure the dominating data variation in a low-dimensional embedding subspace, which is defined as

$$T^2 = \mathbf{y}_{tp}^T \mathbf{S}^{-1} \mathbf{y}_{tp} \quad (22)$$

where $\mathbf{y}_{tp} = [y_{t,1} \ y_{t,2} \ \cdots \ y_{t,p}]$ is obtained from Eq. (16) corresponding to the first p eigenvalues and $\mathbf{S}$ is the covariance matrix of $\mathbf{y}_{tp}$ in the training dataset. The Q statistic is used to measure data variability in residual space, which is defined as

$$Q = \|\mathbf{y}_t\|^2 - \|\mathbf{y}_{tp}\|^2 = \sum_{j=1}^{n_s} y_{t,j}^2 - \sum_{j=1}^p y_{t,j}^2 \quad (23)$$

where $\mathbf{y}_t = [y_{t,1} \ y_{t,2} \ \cdots \ y_{t,n_s}]$ is calculated from Eq. (16).

After monitoring statistics are obtained, the confidence limit is calculated to determine whether process is in control. As no prior distribution knowledge is available for SKLPP, the confidence limit of monitoring statistics for nominal operating regions is determined by the well-known kernel density estimation method [19, 20].

The process monitoring procedure based on SKLPP includes two stages: offline modeling stage and online detection stage. The detailed steps for process monitoring can be shown as follows.

Offline modeling stage:

(1) Acquire normal operating data matrix $\mathbf{X} \in \mathbf{R}^{n \times m}$ and normalize the data using the mean and variance of each variable.

(2) Construct adjacency graph and compute weight matrix $\mathbf{L}$ and $\mathbf{D}$.

(3) Compute the kernel matrix $\mathbf{K}$ and scale it so that the mapped data $\Phi(\mathbf{x})$ in the feature space should be mean centred and variance scaled [16].

(4) Apply the feature samples selection technique in Section 3.2 to select a sample subset and construct

the sparse kernel matrix $\mathbf{K}_s$.

(5) Solve generalized eigenvalue problem Eq. (15) to obtain the eigenvalues and corresponding eigenvectors.

(6) Calculate the T^2 and Q statistics of the normal operating data and establish the control limits of monitoring statistics by kernel density estimation.

Online detection stage:

(1) Obtain the new data $\mathbf{x}_{new}$, and scale it with mean and variance from step (1) of offline modeling stage.

(2) Compute the kernel vector $\mathbf{k}_{new} = [k(\mathbf{x}_{new}, \mathbf{x}_{s,1}) \ k(\mathbf{x}_{new}, \mathbf{x}_{s,2}) \ \cdots \ k(\mathbf{x}_{new}, \mathbf{x}_{s,n_s})]$ and apply mean centering and variance scaling.

(3) Obtain the projections of the new data according to Eq. (16) and compute the two monitoring statistics T_{new}^2 and Q_{new} according to Eqs. (22) and (23), respectively.

(4) Compare the two statistics of the new observed data with the corresponding confidence limits. If the upper control limits are exceeded, the abnormal behavior of the process is detected.

5 CASE STUDIES

The proposed SKLPP process monitoring method was tested with a simulated continuous stirred tank reactor (CSTR) system [21, 22]. It is assumed that a classic first order irreversible reaction, $A \rightarrow B$, happens in CSTR. The flow of solvent and reactant A into a reactor produces a single component B as an outlet stream. Heat from the exothermic reaction is removed through cooling flow of jacket. The temperature of reactor is controlled by manipulating the coolant flow. The level is controlled by manipulating the outlet flow. A schematic diagram of the CSTR with feedback control system is shown in Fig. 1.

Based on the mass, energy and component balances, the CSTR system dynamic model can be built as

$$\frac{dc_A}{dt} = -k_0 e^{-E/RT} + \frac{Q_F c_{AF} - Q_C c_A}{Ah} \quad (24)$$

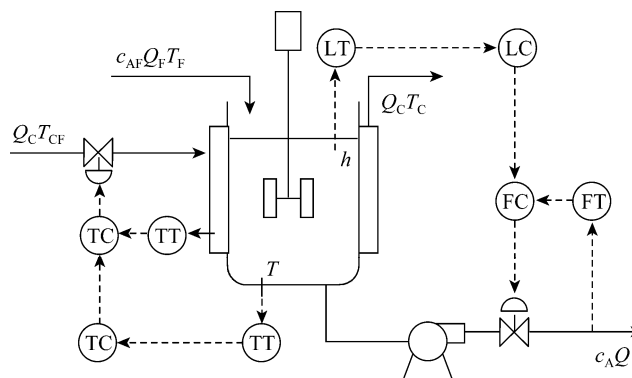


Figure 1 Diagram of the CSTR system

$$\frac{dT}{dt} = \frac{k_0 e^{-E/RT} c_A (-\Delta H)}{\rho C_p} + \frac{Q_F T_F - Q_C T}{Ah} + \frac{UA_C (T_C - T)}{\rho C_p Ah} \quad (25)$$

$$\frac{dT_C}{dt} = \frac{Q_C (T_{CF} - T_C)}{V_C} + \frac{UA_C (T - T_C)}{\rho_C C_{pC} V_C} \quad (26)$$

$$\frac{dh}{dt} = \frac{Q_F - Q}{A} \quad (27)$$

where A is reactor cross-sectional area, c_A is concentration of species A in the reactor, c_{AF} is concentration of species A in reactor feed stream, C_p is heat capacity of reactor contents, C_{pC} is heat capacity of the coolant, E is activation energy, h is reactor liquid level, k_0 is preexponential factor, Q_F is reactor feed flow rate, Q_C is coolant flow rate, R is universal gas constant, T is reactor temperature, T_C is temperature of coolant in the cooling jacket, T_{CF} is temperature of coolant feed, T_F is temperature of reactor feed stream, U is heat-transfer coefficient, A_C is total heat transfer area, ΔH is reaction heat, ρ is density of reactor contents, ρ_C is density of coolant. The detailed model parameters and nominal operating conditions can be found in Johannesmeyer *et al* [22].

The data of normal operating and faulty conditions were generated by simulating the CSTR process. Ten process variables were recorded and Gaussian noise was added to all measurements in simulation procedure. Table 1 listed the measured variables and the corresponding noise standard deviation values. The simulation data were sampled every 10 s. Normal operating data with 300 samples were generated for building statistical models. Five kinds of fault pattern

Table 1 Measured variables and noise standard deviation values

Measured variable	Noise standard deviation values
c_A	0.0024 mol·L ⁻¹
c_{AF}	0.0024 mol·L ⁻¹
T	0.71 K
T_F	0.71 K
T_C	0.63 K
T_{CF}	0.63 K
h	0.004 m
Q	0.71 L·min ⁻¹
Q_F	0.71 L·min ⁻¹
Q_C	0.10 L·min ⁻¹

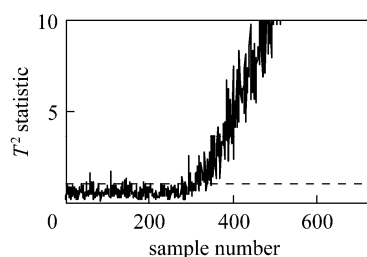
data were simulated with 720 samples to test fault detection methods. For each fault pattern data, fault was introduced after the 240th sample. The simulated fault patterns were shown in Table 2, which contained operation condition change, process parameter change and sensor bias.

In this section LPP and SKLPP were both applied to detect all five faults. The 95% confidence limits are adopted as the alarming threshold. For convenience of methods comparison, all monitoring statistics are divided by their respective confidence limit when the monitoring charts are plotted so that the alarm limits in all plots were equal to 1. In all monitoring charts, we consider a fault is detected if 8 continuous samples exceed confidence limit.

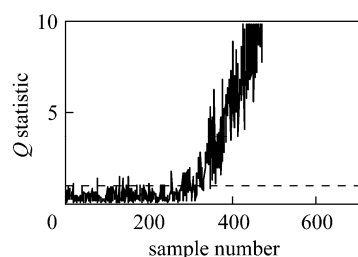
Fault F2 are firstly illustrated for methods comparison. The monitoring results of LPP and SKLPP are shown in Figs. 2 and 3 respectively. As statistical

Table 2 Fault patterns for CSTR system

Fault	Description	Variable	Value
F1	coolant temperature ramps down	T_{CF}	-0.1 K·min ⁻¹
F2	feed concentration ramps up	c_{AF}	+6×10 ⁻⁴ mol·L ⁻¹ ·min ⁻¹
F3	feed temperature ramps up	T_F	+0.05 K·min ⁻¹
F4	heat transfer coefficient ramps down	UA_C	-80 J·min ⁻² ·K ⁻¹
F5	coolant temperature sensor has a bias	T_C	+3 K



(a) T^2 chart



(b) Q chart

Figure 2 LPP monitoring charts for fault F2

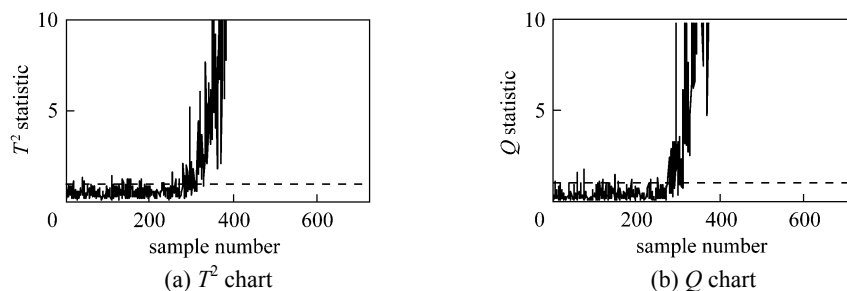


Figure 3 SKLPP monitoring charts for fault F2

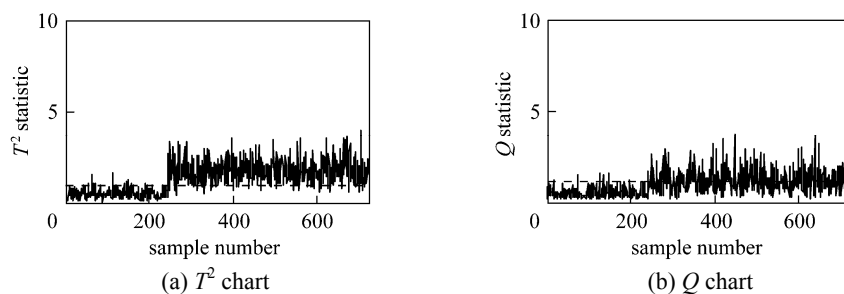


Figure 4 LPP monitoring charts for fault F5

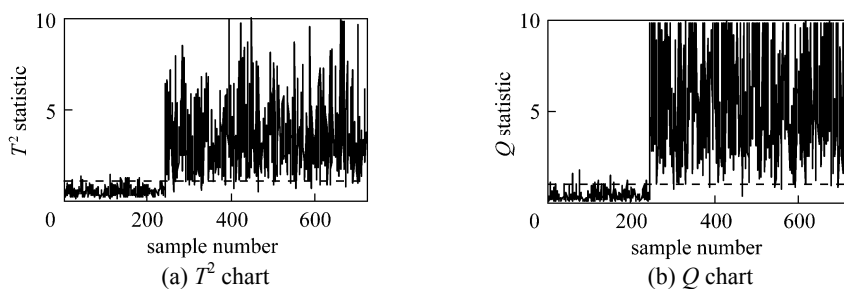


Figure 5 SKLPP monitoring charts for fault F5

monitoring methods, both LPP and SKLPP may bring some false alarming samples during normal operation before the 240th sample. According to Fig. 2, there is a false alarming rate of 6.67% for LPP T^2 chart and 3.75% for LPP Q chart. When SKLPP is used, the false alarming rates of T^2 and Q charts are 5.0% and 4.17% respectively. Under normal operation, LPP and SKLPP have similar false alarming rates.

When fault F2 is introduced after the 240th sample, it can be detected by the LPP T^2 chart at the 325th sample, while the LPP Q chart performs worse and gives alarming signal at the 329th sample. As the non-linear version of LPP, SKLPP improves the monitoring performance and its T^2 and Q statistics detect fault F2 at the 311th sample simultaneously. So for faulty operation, the nonlinear SKLPP method does better than LPP for monitoring fault F2.

Fault F5 is also used to test the methods. When fault F5 happens which is associated with the coolant temperature sensor bias, both LPP and SKLPP can detect the fault (see Figs. 4 and 5). The LPP T^2 statistic gives an alarming at the 244th sample, and the Q chart of LPP detects fault until the 428th sample. Two

charts of SKLPP can indicate the fault quickly at the 241th sample simultaneously, which are 3 samples earlier than LPP T^2 chart. With further analysis, two methods have different fault detection rates. When LPP is applied, fault detection rates of LPP are 90.63% and 47.08% for the T^2 and Q statistics respectively. Although the T^2 statistic of LPP gives a good performance, the Q statistic is not satisfactory. Fig. 5 shows the monitoring results of SKLPP, whose fault alarming rates for T^2 and Q are 95.42%, and 98.12% respectively. Both monitoring statistics of SKLPP exceed confidence limits very clearly, which also demonstrates the advantages of SKLPP in fault detection.

An overall fault detection performance comparison between SKLPP and LPP is investigated in Table 3. This table lists the fault detection rate of two methods on all five fault scenarios. It can be seen that SKLPP-based monitoring method has higher fault detection rate and performs better than LPP method.

In chemical processes, many faults are caused by the slowly-varying parameters, such as faults F1 to F4 in Table 2. This kind of fault is often called ramp fault or gradual fault. Here fault F4 is used as an example to

Table 3 Comparison of fault detection rate (%)

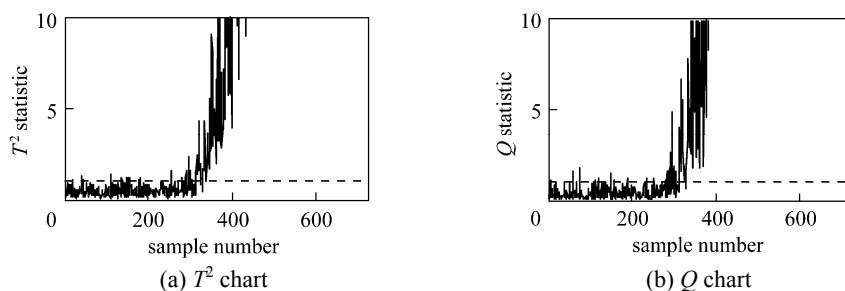
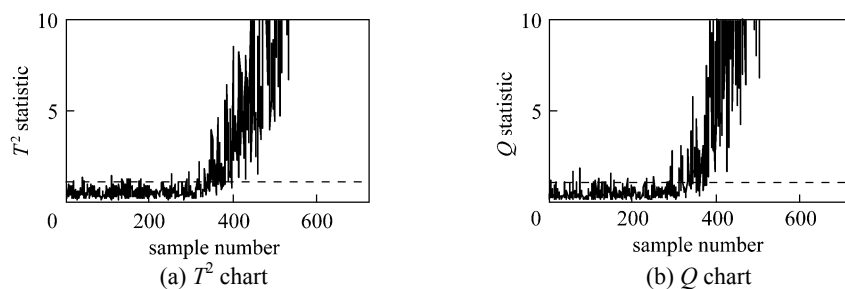
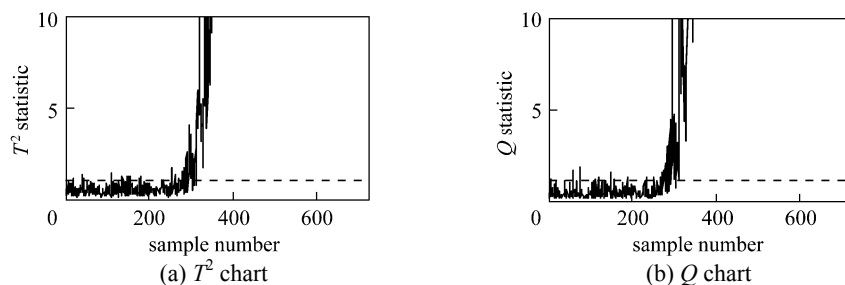
Fault	LPP		SKLPP	
	T^2	Q	T^2	Q
F1	78.54	89.17	85.21	90.63
F2	85.83	87.29	88.96	89.79
F3	70.63	64.58	74.38	76.25
F4	80.63	85.83	86.88	87.71
F5	90.63	47.08	95.42	98.12

demonstrate the influence of the varying rate of faulty parameters on fault detection time delay. Originally, in fault F4 the ramp rate of UA_C is set as $-80 \text{ J} \cdot \text{min}^{-2} \cdot \text{K}^{-1}$. For comparison, we modify the ramp rate as $-50 \text{ J} \cdot \text{min}^{-2} \cdot \text{K}^{-1}$ to generate a more slowly varying fault noted as fault F4a. Then we modify the ramp rate of UA_C as $-120 \text{ J} \cdot \text{min}^{-2} \cdot \text{K}^{-1}$ to simulate a faster fault noted as fault F4b. SKLPP is applied for detecting fault F4, F4a and F4b. The monitoring results are shown in Figs. 6, 7 and 8 respectively. The SKLPP T^2 chart detects fault F4 at the 311th sample and Q chart

gives alarming signal at the 313th sample. When fault F4a is simulated, it can be seen that the fault is detected at the 381th sample by the T^2 statistic and at the 378th sample by the Q statistic. Because fault F4a represents a slower change in heat transfer coefficient, the fault detection time delay is increased. In contrast, fault F4b represents a quick process parameter change and the monitoring results of SKLPP are shown in Fig. 8. Two statistics gives fault alarming signal at the 293th sample simultaneously, which displays an earlier detection. According to above analysis, it can be concluded that it is difficult to detect the ramp fault with very slow parameter change, and it is very valuable to study how to improve the sensitivity of ramp fault detection.

6 CONCLUSIONS

This paper proposed a novel nonlinear process fault detection method based on sparse kernel locality preserving projection (SKLPP). The proposed method is motivated by nonlinear generalization of LPP. Kernel trick is introduced to deal with nonlinear transformation

**Figure 6** SKLPP monitoring charts for fault F4**Figure 7** SKLPP monitoring charts for fault F4a**Figure 8** SKLPP monitoring charts for fault F4b

so that kernel version of LPP model is built. Feature samples selection technique is applied to reduce the model complexity of kernel LPP, which results in sparse kernel LPP model. A simulation example of CSTR is used to demonstrate that the proposed method can effectively detect process faults and its performance outperforms LPP.

NOMENCLATURE

a	linear transformation vector
D	weight matrix
F	feature space
$J_{\max}$	maximum fitness function
K	kernel matrix
K_{si}	kernel vector between $\Phi(x_i)$ and the sample subset
K_{ss}	square kernel matrix for the selected samples
K_t	kernel vector corresponding to testing vector
k	kernel function
L	Laplacian matrix
m	number of measured variable
n	number of training samples
n_s	number of select samples in FSS
p	dimension of low-dimensional embedding subspace
Q	statistic for monitoring residual space
R	original data space
S	covariance matrix
T	statistic for monitoring dominating data variation
v	coefficient vector
W	weight matrix
w_{ij}	weight value
X	training data matrix
X_s	training sample subset
x_i	measured data vector
x_t	testing vector
y	projection of input data matrix
y_i	projection of measured data vector
β_i	coefficient vector
δ_i	normalized Euclidean distance
λ	eigenvalue
Φ	nonlinear mapping

REFERENCES

- Liang, J., Qian, J., "Multivariate statistical process monitoring and control: recent developments and applications to chemical industry", *Chin. J. Chem. Eng.*, **11** (2), 191–203 (2003).
- Ye, L., Shi, X., Liang J., "A multi-level approach for complex fault isolation based on structured residuals", *Chin. J. Chem. Eng.*, **19** (3), 462–472 (2011).
- Liu, X., Li, K., McAfee, M., Irwin, G.W., "Improved nonlinear PCA process monitoring using support vector data description", *Journal of Process Control*, **21** (9), 1306–1317 (2011).
- Wang, Z., Yuan, J., "Online supervision of penicillin cultivations based on rolling MPC", *Chin. J. Chem. Eng.*, **15** (1), 92–96 (2007).
- Zhang, Y., Ma, C., "Fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS", *Chem. Eng. Sci.*, **66** (1), 64–72 (2011).
- Deng, X., Tian, X., "Multivariate statistical process monitoring using multiscale kernel principal component analysis", In: Proceedings of the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Tsinghua University, Beijing, 108–112 (2006).
- He, X., Niyogi, P., "Locality preserving projections", *Adv. Neural Infor. Process. Systems*, **16**, 153–160 (2003).
- Roweis, S.T., Saul, L.K., "Nonlinear dimensionality reduction by locally linear embedding", *Science*, **290** (5500), 2323–2326 (2000).
- Zhang, T., Yang, J., Zhao, D., Ge, X., "Linear local tangent space alignment and application to face recognition", *Neurocomputing*, **70** (7–9), 1547–1553 (2007).
- Hu, K., Yuan, J., "Multivariate statistical process control based on multiway locality preserving projections", *Journal of Process Control*, **18** (7–8), 797–807 (2008).
- Yu, J. B., "Bearing performance degradation assessment using locality preserving projections", *Expert Systems Application*, **38** (6), 7440–7450 (2011).
- Zhang, M., Ge, Z., Song, Z., "Global-local structure analysis model and its application for fault detection and identification", *Ind. Eng. Chem. Res.*, **50** (11), 6837–6848 (2011).
- He, X., Cai, D., Min, W., "Statistical and computational analysis of locality preserving projection", In: Proceedings of the 22th International Conference on Machine Learning, Bonn, Germany, 281–288 (2005).
- Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Smola, A., Muller, K.R., "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces", *IEEE Transactions on Pattern Anal. Machine Intelligent*, **25** (5), 623–628 (2003).
- Melzer, T., Reiter, M., Bischof, H., "Appearance models based on kernel canonical correlation analysis", *Pattern Recognition*, **36** (9), 1961–1971 (2003).
- Choi, S.W., Lee, C., Lee, J.M., Park, J.H., Lee, I.B., "Fault detection and identification of nonlinear processes based on kernel PCA", *Chemometrics and Intelligent Laboratory Systems*, **75** (1), 55–67 (2005).
- Sumana, C., Bhushan, M., Venkateswarlu, C., Gudi, R.D., "Improved nonlinear process monitoring using KPCA with sample vector selection and combined index", *Asia-Pacific J. Chem. Eng.*, **6** (3), 460–469 (2011).
- Baudat, G., Anouar, F., "Feature vector selection and projection using kernels", *Neurocomputing*, **55** (1–2), 21–38 (2003).
- Lee, J.M., Yoo, C.K., Lee, I.B., "Statistical process monitoring with independent component analysis", *Journal of Process Control*, **14** (5), 467–485 (2004).
- Tian, X., Zhang, X., Deng, X., Chen, S., "Multiway kernel independent component analysis based on feature samples for batch process monitoring", *Neurocomputing*, **72** (7–9), 1584–1596 (2009).
- Singhal, A., Seborg, D.E., "Effect of data compression on pattern matching in historical data", *Ind. Eng. Chem. Res.*, **44** (9), 3203–3212 (2005).
- Johannesmeyer, M.C., Singhal, A., Seborg, D.E., "Pattern matching in historical data", *AIChE J.*, **48** (9), 2022–2038 (2002).