IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 26, 2020, accepted June 10, 2020, date of publication June 24, 2020, date of current version July 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004564

Nonlinear Fault Detection of Batch Processes Using Functional Local Kernel Principal Component Analysis

FEI HE AND ZHIYAN ZHANG
Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, Beijing 100083, China

Corresponding author: Fei He (hefei@ustb.edu.cn)

This work was supported in part by the National Key Technology Research and Development Program of the 12th Five-year Plan of China under Grant 2015BAF30B01, in part by the Open Foundation from the State Key Laboratory of rolling and automation, Northeastern University under Grant 2018RALKFKT003, in part by the Fundamental Research Funds for the Central Universities under Grant FRF-BR-17-030A, and in part by the University of Science and Technology Beijing—Taipei University of Technology Joint Research Program under Grant TW2019013.

ABSTRACT In order to guarantee and improve the product quality, the data-driven fault detection technique has been widely used in industry. For three-way datasets of batch process in industry process (i.e., batch × variable × time), a novel method named functional local kernel principal component analysis (FLKPCA) is proposed. Since the variables' trajectories often show functional nature and can be considered as smooth functions rather than just vectors. Firstly, the variables' trajectory is expressed as the combination of smooth basis functions using functional data analysis (FDA), which means that the datasets of batches process would be transformed from the three-ways array into two-ways function matrix. Then, kernel locality preserving projections (LKPCA) is used to perform dimensionality reduction on two-way function matrix directly. Different from kernel principal component analysis (KPCA). LKPCA aims at preserving the both local and global structure of the data in a new optimization objective. Consequently, FLKPCA could more effectively seek the potential information that hidden in the three-ways datasets. Lastly, the effectiveness of the proposed approach is illustrated by the benchmark of fed-batch penicillin fermentation process and the hot strip rolling process.

INDEX TERMS Kernel functional local principal component analysis, process monitoring, fault detection, function data analysis.

I. INTRODUCTION

With the development of industry process, the customers put forward higher demands for production quality, so the fault detection of the industrial process has been research hotspot. In the past few years, fault detection methods could be classified as being associated with one or more approaches: data-driven, analytical and knowledge-based [1]. Since the advancement of statistical algorithms and computer technology, the fault detection methods based on data-driven approaches have been widely applied to industrial systems. At the beginning, the fault detection is just performed on single variable, such as statistical process control (SPC). The SPC charts including Shewhart, cumulative sum (CUSUM) and exponentially weighted moving average (EWMA) charts have been used to monitor processes and improve product quality. Those methods are only suitable for the small industrial systems, because they maybe lead to misleading judgments to monitor multiple process characteristics independently [2], [3]. The shortcoming of single variable monitoring promotes the rapid development of the multivariate statistical process monitoring (MSPM) technique, and many famous data-driven fault diagnosis methods have been proposed including principal component analysis (PCA), canonical variate analysis (CVA), independent component analysis (ICA) and partial least squares (PLS) [4]–[9]. The common methods are to reduce the dimension of the data, which try to extract the most important features from the original process data. The methods of dimension reduction will directly affect the monitoring performance [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Luo.

PCA, PLS and their extended methods only consider the global structure of dataset and ignore the local information of dataset. In order to preserve the local structure, the manifold learning has been proposed and developed rapidly, including isometric feature mapping (Isomap), locally linear embedding (LLE) [11], Laplacian Eigen-maps (LE) [12], locality preserving projections (LPP) [13], etc.

Considering both local and global data information for dimension reduction, some new algorithms have been proposed, including a global-local structure analysis (GLSA) [10], local and global principal component analysis (LGPCA) [14], orthogonal multi-manifold projection (OMMP) [15] and local KPCA (LKPCA) [16]. For LKPCA algorithm, a new optimization target is constructed by integrating local and global structure information based on KPCA. LKPCA has more effective feature extraction than KPCA and KLPP method, which considers both global and local data information.

With the rapid development of the batch process in the manufacturing industry, batch processes are great importance in the manufacturing of many high-quality and high value-added products such as bio-chemicals, food and semiconductors [17]. The fault detection of batch processes has become the research focus to guarantee and improve the product quality. Different from the continuous process, the batch process has the following distinct features: (a) operation duration is finite; and (b) repeats itself until the specified amount of products has been made [18]. The dataset of batch process can be expressed as the three-way data array (batch × variable × time). The traditional analytical method of three-way data array mainly includes bilinear and trilinear methods. Bilinear methods try to unfold the three-ways data array into a matrix before constructing monitoring models, such as multiway partial least squares (MPLS), multiway locality preserving projections (MLPP), multiway principal component analysis (MPCA), multiway independent component analysis (MICA), and the related nonlinear methods [19]–[25]. The trilinear methods that the three-way data is dealt directly with tensor decomposition, including parallel factor analysis (PARAFAC), tensor locality preserving projections (TLPP), trilinear decomposition (TLD), GTucker2 and Tucker3 decomposition [26]–[30].

Both the bilinear and trilinear methods require the original dataset at the same sampling duration for each batch and same sampling rate for each variable, but these constraints are often not satisfied in the actual production process. Functional data analysis (FDA) [31], which uses the basis function to represent the trajectory of the original data, can deal with varying length in batch runs. Besides, FDA can reduce the variable dimension, since the number of basis functions is often much smaller than that of sampling time [31]. Meanwhile, functional data analysis (FDA) is capable of pinpointing the subtle differences in the variable trajectories between normal and faulty batches by removing the random noise. Many FDA-related methods have been proposed, such as functional data analysis and modeling [32], [33], function space analysis based PCA (FSPCA) [34], multivariable functional kernel principal component analysis (MFKPCA) [31], functional partial least squares (FPLS) [35] and functional kernel locality preserving projections (FKLPP) [36].

In the paper, a novel method of fault detection has been proposed called functional local kernel principal component analysis (FLKPCA) to enhance the fault detection in batch process. For the three-way dataset of batch process, FDA is used to transform the original dataset into functional matrix firstly. FDA can effectively reduce the noise and extract the effective information from trajectories of original data. In order to obtain local-global structure information and deal with nonlinear features in the data, LKPCA based on the functional matrix is used to build the process monitoring model. Then, the related statistic T^2 and *SPE* are used as metrics index to judge whether the batch is abnormal.

The organization of the rest content is as follows, Section II introduces the functional data representation and Section III shows the derivative process of the FLKPCA. Section IV presents how to use the FLKPCA for fault detection and introduce two cases of fault detection in Section V. Section VI summarizes the proposed method and research results.

II. FUNCTIONAL DATA REPRESENTATION

For the batch process, the dataset can be expressed as three-ways array $\boldsymbol{Y}$ ($I\times J\times K$) where the element $y_{i,j,k}$ represents the *i-th* batch of variable *j-th* in the *k-th* observation and the corresponding time is $t_{i,j,k}$, where $i = 1, \ldots, I, j = 1, \ldots, J$, $k = 1, \ldots, K$. Here, the relationship of observation $y_{i,j,k}$ and sampling times $t_{i,j,k}$ satisfy the following mode:

$$y_{i,j,k} = x_{i,j}(t_{i,j,k}) + \varepsilon_{i,j,k} \tag{1}$$

where the function $x_{i,j}(t)$ lies in the Hilbert space $L^2(T)$ of squared integrable functions for a compact domain T. Meanwhile, assuming that the measurement errors $\varepsilon_{i,j,k}$ conform to the independent normal distribution. For the i-th batch sample, the batch sample can be expressed as a J-dimensional vector-valued function $\boldsymbol{x}_i(t) = [x_{i,1}(t), \cdots, x_{i,J}(t)]^{\mathrm{T}}$, and the entire three-ways dataset is transformed into an $I \times J$-dimensional functional matrix $\boldsymbol{X}(t)$

$$\boldsymbol{X}(t) = \begin{bmatrix} x_{1,1}(t) & \cdots & x_{1,J}(t) \\ \vdots & \ddots & \vdots \\ x_{I,1}(t) & \cdots & x_{I,J}(t) \end{bmatrix} \tag{2}$$

The functions matrix in (2) can be expressed as the combination of the basis functions, such as the Fourier basis functions, B-splines, wavelets, etc. [36] In general, fitting the periodic data can use the Fourier basis functions or wavelet. Spline functions are the most common choice of approximation system for non-periodic functional data or parameters. [31] Variable j is expressed by the basis functions $\{\varphi_{j,d}(t),\ d = 1, \cdots, D_j\}$, where d and D_j are respectively the sequence and total number of the basic functions used for variable j, then

$$x_{i,j}(t) = \sum_{d=1}^{D_j} c_{i,j,d}\varphi_{j,d}(t) = \boldsymbol{c}_{i,j}^{\mathrm{T}}\boldsymbol{\varphi}_j(t) \tag{3}$$

where $\boldsymbol{\varphi}_j(t) = [\varphi_{j,1}(t), \cdots, \varphi_{j,D_j}(t)]^{\mathrm{T}}$ is the basis functions vector of variable j and $\boldsymbol{c}_{i,j} = [c_{i,j,1}, \cdots, c_{i,j,D_j}]^{\mathrm{T}}$ is the coefficient vector. In order to estimate the coefficient vector $\boldsymbol{c}_{i,j}$, the least squares can be used as follow:

$$\min_{\boldsymbol{c}_{i,j}} \sum_{k=1}^{K_{i,j}} [y_{i,j,k} - x_{i,j}(t_{i,j,k})]^2 = \min_{\boldsymbol{c}_{i,j}} (\boldsymbol{y}_{i,j} - \boldsymbol{\psi}_{i,j}\mathbf{c}_{i,j})^{\mathrm{T}}(\boldsymbol{y}_{i,j} - \boldsymbol{\psi}_{i,j}\mathbf{c}_{i,j}) \quad (4)$$

where the basis function matrix $\boldsymbol{\psi}_{i,j}$ is

$$\boldsymbol{\psi}_{i,j} = \begin{bmatrix} \varphi_{j,1}(t_{i,j,1}) & \cdots & \varphi_{j,D_j}(t_{i,j,1}) \\ \vdots & \ddots & \vdots \\ \varphi_{j,1}(t_{i,j,K_{i,j}}) & \cdots & \varphi_{j,D_j}(t_{i,j,K_{i,j}}) \end{bmatrix}_{K_{i,j}\times D_j} \quad (5)$$

and the vector $\boldsymbol{y}_{i,j} = [y_{i,j,1}, y_{i,j,2} \cdots, y_{i,j,K_{i,j}}]$, the $K_{i,j}$ is the time point number of variable *j-th* in the *i-th* batch, the coefficient matrix $\hat{\boldsymbol{c}}_{i,j}$ can be calculated by solving the (4)

$$\hat{\boldsymbol{c}}_{i,j} = (\boldsymbol{\psi}_{i,j}^{\mathrm{T}}\boldsymbol{\psi}_{i,j})^{-1}\boldsymbol{\psi}_{i,j}^{\mathrm{T}}\boldsymbol{y}_{i,j} \quad (6)$$

In order to control the smoothness of the functions, a roughness penalty function $pen(x_{i,j}(t)) = \int_T [L(x_{i,j}(t))]^2 dt$ is often added, where $L(\cdot)$ is a linear differential operator defined as

$$L(x_{i,j}(t)) = \omega_0 + \omega_1 \frac{dx_{i,j}(t)}{dt} + \cdots + \omega_n \frac{d^n x_{i,j}(t)}{dt^n} \quad (7)$$

Since the roughness penalty function is to smooth the curve, the second derivative $[\boldsymbol{d^2x(t)/dt}]$ is usually chosen as the measure of a function's roughness. Then, the optimization problem becomes

$$\min_{\boldsymbol{c}_{i,j}} \sum_{k=1}^{K_{i,j}} [y_{i,j,k} - x_{i,j}(t_{i,j,k})]^2 + \eta \times pen(x_{i,j}(t)) \Rightarrow \min_{\boldsymbol{c}_{i,j}} (\boldsymbol{y}_{i,j} - \boldsymbol{\psi}_{i,j}\boldsymbol{c}_{i,j})^{\mathrm{T}}(\boldsymbol{y}_{i,j} - \boldsymbol{\psi}_{i,j}\boldsymbol{c}_{i,j}) + \eta c_{i,j}^{\mathrm{T}}\boldsymbol{R}_j\boldsymbol{c}_{i,j} \quad (8)$$

where $\boldsymbol{R}_j = \int_T L(\varphi_j(t))L(\varphi_j(t))^{\mathrm{T}}dt$ is a $D_j \times D_j$ symmetric matrix by using a numerical quadrature scheme. The smoothing parameter is a trade-off between the error of fitting and the smoothness of functions, which can be chosen by the generalized cross-validation approach. Thus, the estimated coefficient vector can be solved according to

$$\hat{\boldsymbol{c}}_{i,j} = (\boldsymbol{\psi}_{i,j}^{\mathrm{T}}\boldsymbol{\psi}_{i,j} + \eta\boldsymbol{R}_j)^{-1}\boldsymbol{\psi}_{i,j}^{\mathrm{T}}\boldsymbol{y}_{i,j} \quad (9)$$

and the corresponding fitting function is

$$\hat{x}_{i,j}(t) = \hat{\boldsymbol{c}}_{i,j}^{\mathrm{T}}\boldsymbol{\varphi}_j(t) \quad (10)$$

Then, the $\tilde{x}_{i,j}(t)$ is normalized and expressed as $\hat{x}_{i,j}(t)$. The standardization of $\hat{x}_{i,j}(t)$ can be expressed as:

$$\tilde{x}_{i,j}(t) = (\hat{x}_{i,j}(t) - \mu_j(t))/\delta_j \quad (11)$$

where the $\mu_j(t)$ and δ_j are the mean function and the scaling parameter of $\hat{x}_{i,j}(t)$, as follows:

$$\mu_j(t) = \frac{1}{I}\sum_{i=1}^{I} \hat{x}_{i,j}(t)$$
$$\delta_j = \sqrt{\frac{1}{I}\sum_{i=0}^{I} \frac{1}{L(T)} \int_T (\widehat{\mathrm{x}}_{i,j}(t) - \mu_j(t))^2} \quad (12)$$

where the parameter I is the number of the batch process and the $L(T)$ is the length of time variable T. According to the (12), the standardization function $\tilde{x}_{i,j}(t)$ can be expressed as

$$\begin{aligned} \tilde{x}_{i,j}(t) &= (\hat{\boldsymbol{c}}_{i,j}^{\mathrm{T}}\boldsymbol{\varphi}_j(t) - \frac{1}{I}\sum_{i=0}^{I} \hat{\boldsymbol{c}}_{i,j}^{\mathrm{T}}\boldsymbol{\varphi}_j(t))/\delta_j \\ &= [(\hat{\boldsymbol{c}}_{i,j}^{\mathrm{T}} - \frac{1}{I}\sum_{i=0}^{I} \hat{\boldsymbol{c}}_{i,j}^{\mathrm{T}})/\delta_j]\boldsymbol{\varphi}_j(t) \\ &= \tilde{\boldsymbol{c}}_{i,j}^{T}\boldsymbol{\varphi}_j(t) \end{aligned} \quad (13)$$

Therefore the *i-th* batch sample function vector is $\tilde{\boldsymbol{x}}_i(t) = [\tilde{x}_{i,1}(t), \cdots, \tilde{x}_{i,J}(t)]$, and the batch functional matrix can be expressed as $\tilde{\boldsymbol{X}}(t) = [\tilde{\boldsymbol{x}}_1(t), \cdots, \tilde{\boldsymbol{x}}_I(t)]^{\mathrm{T}}$.

III. FUNCTIONAL LOCAL KERNEL PRINCIPAL COMPONENT ANALYSIS

A. FUNCTIONAL KERNEL PRINCIPAL COMPONENT ANALYSIS

For batch processes whose samples are vector-valued functions $\tilde{\boldsymbol{x}}_i(t)$, the training dataset $\boldsymbol{X}(t)$ is expressed as $[\tilde{\boldsymbol{x}}_1(t), \tilde{\boldsymbol{x}}_2(t), \cdots, \tilde{\boldsymbol{x}}_I(t)]^T$ (dimension is $I \times J$). According to the kernel trick, the linear PCA in the high-dimensional feature space corresponds to a nonlinear PCA in the original input space [37], so the training data is mapped onto a high-dimensional space Γ and the nonlinear mapping is expressed as $\phi : \boldsymbol{x}_i(t) \rightarrow \phi(\boldsymbol{x}_i(t)) \in \Gamma$. Meanwhile, seek a loading functional vector $\boldsymbol{p}(t)$ to project the high-dimensional matrix $\phi(\boldsymbol{X}(t))$ into low-dimensional subspace vector $\boldsymbol{z}$ as follows

$$\boldsymbol{z} = < \tilde{\phi}(\boldsymbol{X}(t))^T, \boldsymbol{p}(t) > \quad (14)$$

where the input matrix $\tilde{\phi}(\boldsymbol{X}(t))$ has been standardized, and the inner product of $\tilde{\phi}(\boldsymbol{X}(t))$ and $\boldsymbol{p}(t)$ can be solved as the form

$$< \tilde{\boldsymbol{\phi}}(\boldsymbol{X}(t))^T, \boldsymbol{p}(t) > = \int_T \tilde{\boldsymbol{\phi}}(\boldsymbol{X}(t))\boldsymbol{p}(t)dt \quad (15)$$

According to the PCA algorithm, the objective is to maximize variance of the vector $\boldsymbol{z}$. Then, this expression can be formulated as

$$\begin{aligned} &\max J(\boldsymbol{p}(t)) = \max \frac{1}{I}\boldsymbol{z}^T\boldsymbol{z} \\ &= \max \frac{1}{I}\left\langle \tilde{\boldsymbol{\phi}}(\boldsymbol{X}(t))^T, \boldsymbol{p}(t) \right\rangle^T \left\langle \tilde{\boldsymbol{\phi}}(\boldsymbol{X}(t))^T, \boldsymbol{p}(t) \right\rangle \\ &s.t\ \langle \boldsymbol{p}(t), \boldsymbol{p}(t) \rangle = 1 \end{aligned} \quad (16)$$

where the loading vector $\boldsymbol{p}(t)$ can be transformed to the linear combination of the feature functional matrix $\tilde{\phi}(\boldsymbol{X}(t))$, and the corresponding coefficients vector is $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_I]^T$ (dimension is $I \times 1$), such that the loading functional vector takes the form

$$\boldsymbol{p}(t) = \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha} \tag{17}$$

Substituting (17) into (16) leads to the form as follow

$$\begin{aligned} &\max J_{FKPCA}(\boldsymbol{p}(t)) \\ &\quad = \max \frac{1}{I} \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha} \right\rangle^T \\ &\quad \times \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha} \right\rangle \\ &\quad = \max \boldsymbol{\alpha}^T \frac{1}{I} \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \tilde{\phi}(\boldsymbol{X}(t))^T \right\rangle^T \\ &\quad \times \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \tilde{\phi}(\boldsymbol{X}(t))^T \right\rangle \boldsymbol{\alpha} \\ &s.t \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha}, \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha} \right\rangle = 1 \end{aligned} \tag{18}$$

The kernel trick has been applied to simplify calculations of the nonlinear high-dimensional mapping. Therefore, the inner product of high-dimension matrix $\tilde{\phi}(\boldsymbol{x}_i(t))$, $\tilde{\phi}(\boldsymbol{x}_j(t))$ can be expressed by the following kernel function:

$$k(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) = \left\langle \tilde{\phi}(\boldsymbol{x}_i(t))^T, \tilde{\phi}(\boldsymbol{x}_j(t))^T \right\rangle \tag{19}$$

where the kernel function k can be used in many forms including polynomial kernel, Gaussian kernel and radial basis kernel etc. Based on (19), the corresponding kernel matrix can be expressed as follows:

$$\begin{aligned} \boldsymbol{K} &= k(\boldsymbol{X}(t), \boldsymbol{X}(t)) \\ &= \begin{bmatrix} k(\boldsymbol{x}_1(t), \boldsymbol{x}_1(t)) & \cdots & k(\boldsymbol{x}_1(t), \boldsymbol{x}_I(t)) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_I(t), \boldsymbol{x}_1(t)) & \cdots & k(\boldsymbol{x}_I(t), \boldsymbol{x}_I(t)) \end{bmatrix}_{I \times I} \end{aligned} \tag{20}$$

the matrix $\tilde{\boldsymbol{K}}$ is expressed as the corresponding centered kernel matrix $\boldsymbol{K}$ and the form is expressed as follows:

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \boldsymbol{K} - 1_{I \times I} \boldsymbol{K} - \boldsymbol{K} 1_{I \times I} + 1_{I \times I} \boldsymbol{K} 1_{I \times I} \\ \text{and } 1_{I \times I} &= \frac{1}{I} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{I \times I} \end{aligned} \tag{21}$$

The $\tilde{\boldsymbol{K}}$ is a symmetric matrix. Substituting kernel matrix into (18).

$$\begin{aligned} &\max \ J_{FKPCA}(\boldsymbol{\alpha}) \\ &\quad = \max \frac{1}{n-1} \boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \tilde{\boldsymbol{K}} \boldsymbol{\alpha} \\ &s.t. \ \boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \alpha = 1 \end{aligned} \tag{22}$$

According to the optimization in (22). The optimal solution of $\boldsymbol{\alpha}$ can be solved by the eigenvector problem:

$$\tilde{\boldsymbol{K}} \tilde{\boldsymbol{K}} \boldsymbol{\alpha} = (n-1) \lambda \tilde{\boldsymbol{K}} \boldsymbol{\alpha} \tag{23}$$

with the constraint condition $\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \alpha = \boldsymbol{1}$ in (22). The optimal coefficients vector $\boldsymbol{\alpha}$ of (22) can be obtained by solving the equivalent problem:

$$\tilde{\boldsymbol{K}} \boldsymbol{\alpha} = (n-1) \lambda \boldsymbol{\alpha} \tag{24}$$

When solving the eigenvalue problem in (24), it will yield the nonzero eigenvalues $\lambda^1 \geq \lambda^2 \geq \cdots \lambda^n > 0$ with the corresponding eigenvector $\boldsymbol{\alpha}^i$, $1 \leq i \leq n$. Then, the eigenfunction in the feature space satisfies

$$\begin{aligned} z_i &= \left\langle \tilde{\phi}(\boldsymbol{x}_i(\mathrm{t}))^T, \boldsymbol{p}(t) \right\rangle \\ &= \left\langle \tilde{\phi}(\boldsymbol{x}_i(\mathrm{t}))^T, \tilde{\phi}(\boldsymbol{X}(\mathrm{t})^T \boldsymbol{\alpha} \right\rangle = \tilde{\boldsymbol{k}}_i^T \boldsymbol{\alpha} \end{aligned} \tag{25}$$

where $\tilde{\boldsymbol{k}}_i = [\tilde{k}(\boldsymbol{x}_i(t), \boldsymbol{x}_1(t)), \cdots, \tilde{k}(\boldsymbol{x}(t)_i, \boldsymbol{x}_I(t))]^T$.

B. FUNCTIONAL LOCAL KERNEL PRINCIPAL COMPONENT ANALYSIS

The KPCA is viewed as a global structure data analysis technique, which just considers the outer relationship of the whole dataset. In order to preserve the inner relationship among neighborhood samples, the manifold learning methods have been proposed. Both local and global structure information in the data can reflect the potential characteristics of the data, which motivates us to combine the local structure analysis with the FKPCA. In this section, local structure analysis based on the LPP is briefly introduced firstly, and then local structure is integrated with KPCA.

1) LOCAL STRUCTURE ANALYSIS

Based on the basic idea of lpp which is to find the optimal linear approximation that makes the neighboring points to stay as close together as possible. Seek a transformation matrix $\boldsymbol{p}(t)$ to projects high-dimensional functional matrix $\tilde{\boldsymbol{X}}(t)$ into a low-dimensional numerical vector $\boldsymbol{z} = [z_1, z_2, \cdots, z_I]$ and the form can be expressed as $z_i = \langle \boldsymbol{p}^T(t), \tilde{\boldsymbol{x}}_i(t) \rangle$. Meanwhile, the values z_i, z_j are as close as possible when the corresponding high-dimensional vectors $\tilde{\boldsymbol{x}}_i(t), \tilde{\boldsymbol{x}}_j(t)$ are close. According to the optimization goal, the minimize objective function is expressed as follow

$$\min J_{LSA} = \min \sum_{ij} \left\| z_i - z_j \right\|^2 W_{ij} \tag{26}$$

where the weight coefficient W_{ij} represents a heavy penalty, if the neighboring points $\tilde{\boldsymbol{x}}_i(t), \tilde{\boldsymbol{x}}_j(t)$ have been mapped far away. Therefore, a possible way of defining of weight coefficient W_{ij} can be defined using the Heat kernel [36], which can be expressed as

$$W_{ij} = \begin{cases} e^{-\dfrac{\left\| \tilde{\boldsymbol{x}}_i(t) - \tilde{\boldsymbol{x}}_j(t) \right\|^2}{\sigma}} \\ \quad if \ \boldsymbol{x}_j \ and \ \boldsymbol{x}_i \ is \ the \ k-nearest \ neighbor \\ 0 \\ \quad otherwise \end{cases} \tag{27}$$

where σ is an empirical parameter, which is selected as the mean value of the K-nearest neighbors. The detail can be

found in [38]. Meanwhile define the $d_i = \sum_{i=1}^{n} W_{ij}$ and substitute into (28) that can be rearranged as:

$$\begin{aligned}\min J_{LSA}(\boldsymbol{p}) &= \min \sum_{ij} (\left\langle \boldsymbol{p}^T(t), \tilde{\boldsymbol{x}}_i(t) \right\rangle - \left\langle \boldsymbol{p}^T(t), \tilde{\boldsymbol{x}}_j(t) \right\rangle) W_{ij} \\ &= \min \left[\sum_{i=1}^{n} \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle^T d_i \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle \right. \\ &\quad \left. - \sum_{i,j=1}^{n} \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle^T w_{i,j} \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle \right] \\ &= \min \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle^T (\boldsymbol{D} - W) \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle \\ &= \min \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle^T L \left\langle \tilde{\boldsymbol{x}}_i(t)^T, \boldsymbol{p}(t) \right\rangle \end{aligned} \tag{28}$$

where $\boldsymbol{D} = diag\{d_1, d_2, \cdots, d_n\}$ denotes the diagonal matrix and sparse symmetric matrix $\boldsymbol{W}$ is a weight matrix consisted of weight coefficient W_{ij}, $L = W - D$ is Laplacian matrix. Then, the functional matrix is mapped to a high-dimensional matrix $\tilde{\phi}(\boldsymbol{X}(t))$ by the kernel trick to deal with the nonlinear characteristic. The optimization functional (28) can be rewritten as:

$$\begin{aligned}&\min J_{LSA}(\boldsymbol{p}) \\ &\quad = \min \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \boldsymbol{p}(t) \right\rangle^T \boldsymbol{L} \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \boldsymbol{p}(t) \right\rangle \end{aligned} \tag{29}$$

Similar to KPCA algorithm, the projection vector $\boldsymbol{p}(t)$ takes the form of the linear combination of training samples as $\boldsymbol{p}(t) = \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha}$, the above optimization function in (29) becomes

$$\begin{aligned}&\min J_{LSA}(\boldsymbol{\alpha}) \\ &\quad = \min \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha} \right\rangle^T \\ &\qquad \boldsymbol{L} \left\langle \tilde{\phi}(\boldsymbol{X}(t))^T, \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha} \right\rangle \\ &\quad = \min \boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \boldsymbol{L} \tilde{\boldsymbol{K}} \alpha \end{aligned} \tag{30}$$

2) FUNCTIONAL LOCAL KERNEL PRINCIPAL COMPONENT ANALYSIS

For the FKPCA, its optimization objective that maximize the variance in the projection direction to extract the global structure information. And the FKLPP tends to preserves the neighborhood structure between pre- and post-feature extraction. Both the local and global structure information of dataset are useful for data analysis in actual industrial process, so the novel method called FLKPCA has introduced the local structure projection into FKPCA. As shown in (22) and (30), the optimization objective of the FLKPCA can be described as maximize $\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \tilde{\boldsymbol{K}} \boldsymbol{\alpha}$ (to extract maximize variance) and to minimize $\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \boldsymbol{L} \tilde{\boldsymbol{K}} \boldsymbol{\alpha}$ (to preserve local structure). The optimization objective is expressed as follows:

$$\begin{aligned}&\max J_{FLKPCA}(\boldsymbol{\alpha}) = \max \frac{\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \tilde{\boldsymbol{K}} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \boldsymbol{L} \tilde{\boldsymbol{K}} \boldsymbol{\alpha}} \\ &s.t.\ \boldsymbol{\alpha}^T \tilde{\boldsymbol{K}} \boldsymbol{\alpha} = 1. \end{aligned} \tag{31}$$

The coefficient vector $\boldsymbol{\alpha}$ that maximize the objective function is given by solving the generalized eigenvector problem:

$$\tilde{\boldsymbol{K}} \tilde{\boldsymbol{K}} \boldsymbol{\alpha} = \lambda \tilde{\boldsymbol{K}} \boldsymbol{L} \tilde{\boldsymbol{K}} \boldsymbol{\alpha} \tag{32}$$

In order to ensure a nonsingular problem, the regularization method can be used by substituting $\tilde{\boldsymbol{K}} \boldsymbol{L} \tilde{\boldsymbol{K}}$ with $\tilde{\boldsymbol{K}} \boldsymbol{L} \tilde{\boldsymbol{K}} + \delta \boldsymbol{I}_n$ in (34), where δ is a small positive regularization parameter and $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix [16]. A set of coefficient vector $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^n$ could be solved by (32). Then, the first l coefficient vector as $\boldsymbol{\alpha}_l$ could be preserved to reduce the dimensionality of variables. For a new sample $\boldsymbol{x}_{new}(t)$, the new transform value z_{new} can be computed according to

$$z_{new} = \sum_{j=1}^{m} \boldsymbol{\alpha}_j (\tilde{\phi}(\boldsymbol{x}_{new}(\mathrm{t}))^T \tilde{\phi}(\boldsymbol{x}_j(t)) = \tilde{\boldsymbol{k}}_{new}^T \boldsymbol{\alpha}_m \tag{33}$$

where $\tilde{\boldsymbol{k}}_{new} = [\tilde{k}(\boldsymbol{x}_{new}(t), \boldsymbol{x}_1(t)), \cdots, \tilde{k}(\boldsymbol{x}_{new}(t), \boldsymbol{x}_n(t))]^T$. The widely used radial basis kernel is the following form:

$$k(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) = \exp\left(- \left\| \boldsymbol{x}_i(t) - \boldsymbol{x}_j(t) \right\|^2 \Big/ S^2 \right) \tag{34}$$

where the kernel parameter S is pre-specified by the user, and the squared L^2-norm of (34) is expressed as follow:

$$\begin{aligned}&\left\| \tilde{\boldsymbol{x}}_i(t) - \tilde{\boldsymbol{x}}_j(t) \right\|^2 \\ &\quad = \sum_{k=1}^{J} \int_T (\tilde{\boldsymbol{x}}_{i,k}(t) - \tilde{\boldsymbol{x}}_{j,k}(t))^2 dt \\ &\quad = \sum_{k=1}^{J} (\tilde{\boldsymbol{c}}_{i,k} - \tilde{\boldsymbol{c}}_{j,k})^{\mathrm{T}} \int_T \varphi_j(t) \varphi_j(t)^{\mathrm{T}} dt (\tilde{\boldsymbol{c}}_{i,k} - \tilde{\boldsymbol{c}}_{j,k}) \\ &\quad = \sum_{k=1}^{J} (\tilde{\boldsymbol{c}}_{i,k} - \tilde{\boldsymbol{c}}_{j,k}) \boldsymbol{R}_j^0 dt (\tilde{\boldsymbol{c}}_{i,k} - \tilde{\boldsymbol{c}}_{j,k}) \end{aligned} \tag{35}$$

Meanwhile, the corresponding kernel matrix $\boldsymbol{K}$ also can be obtained via substituting (35) into (34).

IV. FLKPCA FOR FAULT DETECTION OF BATCH PROCESSES

A. CONTROL CHARTING SCHEME

For fault detection of PCA, the control charts based on statistical variables T^2 and squared prediction error (*SPE*) have been widely used in real industry, but the roles of *SPE* and T^2 are not symmetric. The T^2 is used to measure the magnitude of variations in the principal component subspace (PCS), while *SPE* calculates the variability based on the residual subspace (RS). For fault detection, statistic T^2 and *SPE* are also used as metrics index. The T^2 statistic of one sample $\boldsymbol{x}_i(t)$ can be computed as follows

$$T_i^2 = z_i^T \boldsymbol{\Lambda}^{-1} z_i \tag{36}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix composed of first l principal components, as $diag(\lambda_1, \lambda_2, \cdots, \lambda_l)$. The score vector

$z_i = [z_{i,1}, z_{i,2} \cdots z_{i,l}]^T$ and the score value $z_{i,l}$ of the l-th principal component can be expressed as below

$$\begin{aligned} z_{i,l} &= \left\langle \tilde{\phi}(\boldsymbol{x}_i(t)), \boldsymbol{p}_l(t) \right\rangle \\ &= \int_T \tilde{\phi}(\boldsymbol{x}_i(t)) \boldsymbol{p}_l(t) dt \\ &= \int_T \tilde{\phi}(\boldsymbol{x}_i(t)) \tilde{\phi}(\boldsymbol{X}(t))^T \boldsymbol{\alpha}_l dt = \tilde{K}_i(\boldsymbol{x}_i(t), \boldsymbol{X}(t)) \boldsymbol{\alpha}_l \end{aligned} \tag{37}$$

The *SPE* is calculated from the residual value, which can be expressed as follows:

$$SPE = \left\| \tilde{\phi}(\boldsymbol{x}_i(t)) - \tilde{\phi}^p(\boldsymbol{x}_i(t)) \right\|^2 \tag{38}$$

where $\tilde{\phi}^p(\boldsymbol{x}_i(t))$ is the refactoring functional data matrix and can be expressed as the form:

$$\tilde{\phi}^p(\boldsymbol{x}_i(t)) = \sum_{j=1}^{l} z_{i,j} \boldsymbol{p}_j(t)^T = \boldsymbol{z}_i \boldsymbol{P}(t)^T \tag{39}$$

where the projection function matrix of first l principal components can be computed as follows

$$\begin{aligned} \boldsymbol{P}(t) &= [\boldsymbol{p}_1, \boldsymbol{p}_2 \cdots \boldsymbol{p}_l] \\ &= \tilde{\phi}(\boldsymbol{X}(t))^T [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots \boldsymbol{\alpha}_l] \\ &= \tilde{\phi}(\boldsymbol{X}(t))^T \Upsilon \end{aligned} \tag{40}$$

Substitute (40) into (39), the form can be changed as:

$$\begin{aligned} SPE_i &= \left\| \tilde{\phi}(\boldsymbol{x}_i(t)) - \boldsymbol{z}_i \boldsymbol{P}(t)^T \right\|^2 \\ &= < \tilde{\phi}(\boldsymbol{x}_i(t)), \tilde{\phi}(\boldsymbol{x}_i(t)) > - \boldsymbol{z}_i \boldsymbol{P}(t)^T \boldsymbol{P}(t) \boldsymbol{z}_i^T \\ &= \tilde{k}(\boldsymbol{x}_i(t), \boldsymbol{x}_i(t)) - \boldsymbol{z}_i \boldsymbol{P}(t)^T \boldsymbol{P}(t) \boldsymbol{z}_i^T \\ &= \tilde{k}(\boldsymbol{x}_i(t), \boldsymbol{x}_i(t)) - \boldsymbol{z}_i \Upsilon^T \boldsymbol{K}(\boldsymbol{X}(t), \boldsymbol{X}(t)) \Upsilon \boldsymbol{z}_i^T \end{aligned} \tag{41}$$

The illustration of FLKPCA for fault detection is shown in Fig. 1.

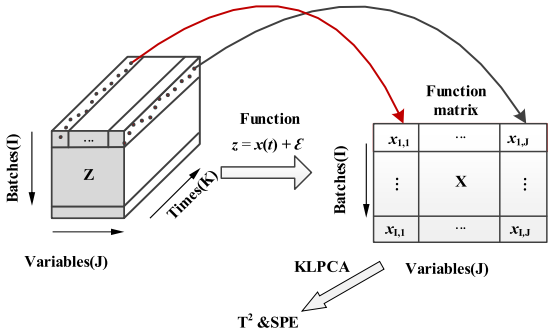

FIGURE 1. Illustration of FLKPCA for fault detection.

B. CONTROL LIMITS

As the previous section, the T^2 and *SPE* statistic have been employed to detect process fault. So the control limits need to be determined for judging whether the process is within control. Based on assumption that the distribution of process parameters are the multivariate normal, the T^2 and *SPE* statistics are following F and χ^2 distribution respectively. However, when dataset show sufficient evidence of a marked departure from multivariate normality, the control limits based on the χ^2-distribution and F-distribution are poor or even inaccurate. In order to avoid the initial assumptions, the kernel density estimation (KDE) is proposed to estimate the distribution of the T^2 and *SPE* statistic based on training samples [39]. For a given vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$, the distribution of $\boldsymbol{x}$ can be estimated as follow

$$\begin{aligned} \hat{f}_h(x) &= \frac{1}{n} \sum_{i=1}^{n} K \left[\frac{(x - x_i)}{h} \right] \\ \hat{f}_h(x) &> 0 \quad \text{and} \quad \int \hat{f}_h(x) dx = 1 \end{aligned} \tag{42}$$

where K is the kernel function whose bandwidth is h in KDE. The KDE approach is able to estimate the underlying probability density function (PDF) of the T^2 and *SPE* statistics. The confidence region is given by 99% in the present work, which means occupying the 99% area of these two density functions. The 99% confidence limit C_α with the significance level $\alpha = 0.01$ can be determined by

$$\int_{-\infty}^{C_\alpha} \hat{f}_h(t) dt = 1 - \alpha \tag{43}$$

There are many kernel functions that can be used, such as normal, uniform, triangular, cosines, among others. Among all kernel function, the normal kernel function has been widely used which is also applied in the paper. Meanwhile, the bandwidth is also important to fault detection. When the bandwidth value is not suitable, the density estimator will not able to estimate the accurate distribution of the variable [36]. To determine the right bandwidth, an adaptive kernel density estimation method based on the smoothing properties of linear diffusion process is used to compute the control limits [40]. The method is considered as "nonparametric" method which does not require any normality assumption for the dataset.

This adaptive kernel density estimation method can be divided into three steps as follows:

(1) Calculate T^2 or *SPE* statistics based on the training data;
(2) Get an adaptive bandwidth by the diffusion method based on [41];
(3) Estimate the probability density of variables using (42), and calculate the control limit C_α by (43).

C. MONITORING PROCEDURES

The inference process of FLKPCA can be summarized as shown in Fig 2. According to the role for fault detection, the FLKPCA-based fault detection procedure can be divided into two stages including: offline modelling stage and fault detection stage. The specific steps for each stage are shown as below:

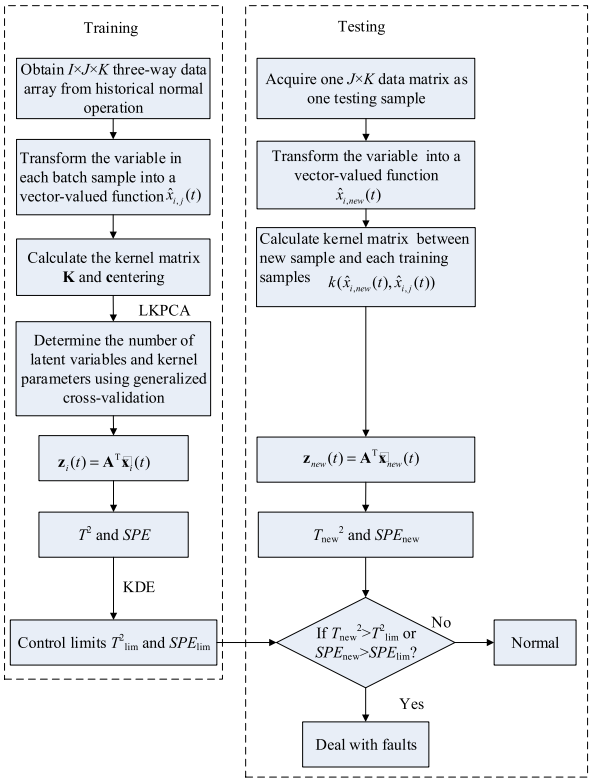

FIGURE 2. Formal procedure of the FLKPCA process monitoring.

For the offline modelling stage:

(1) Get the three-way training dataset ($I\times J\times K$) of reference batches within in-control;
(2) Transform the three-way dataset into functional matrix $\tilde{\boldsymbol{X}}(t)$ via functional data analysis (FDA) using (13);
(3) Calculate the kernel matrix $\boldsymbol{K}$ by (34), and centralize the kernel matrix as $\tilde{\boldsymbol{K}}$ using (21);
(4) Determine the hyper-parameter values of latent variables P and kernel parameters S via the generalized cross-validation approach;
(5) Solve the generalized eigenvector problem in (32) to obtain the coefficient vector $\boldsymbol{\alpha}$;
(6) Transform the training dataset into T^2 and SPE statistics based on (36) and (41), and calculate the control limits of fault detection according to (42).

For fault detection stage:

(1) Get a new test sample data matrix $\boldsymbol{X}_{new}$ (The dimension is $J\times K$);
(2) Transform the test dataset into functional vector $\tilde{\boldsymbol{x}}_{i,new}(t)(i=1,\ldots,I)$ by the (13);
(3) Calculate the kernel vector $\tilde{\boldsymbol{k}}(x_{i,new}(t),x_{i,j}(t))$ between the new sample and each training sample, and centralize the kernel vector;
(4) Calculate the T^2 and SPE statistics of the test sample based the coefficient vector $\boldsymbol{\alpha}$, which is calculated by the offline modelling stage, and compare with the control limits to determine whether the process is abnormal.

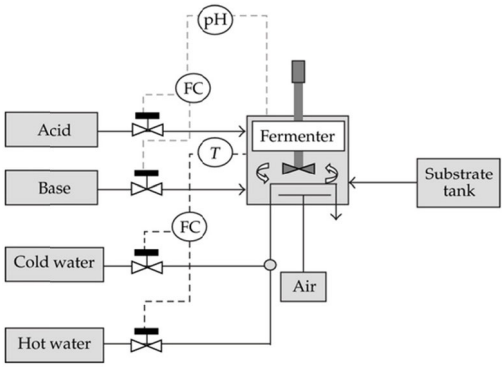

FIGURE 3. Flowchart of the penicillin process.

In order to measure the accuracy of monitoring results, the false alarm rate (FAR) and fault detection rate (FDR) are used as the measurement indexs, which can be described as follows:

$$FAR=\frac{NN}{TNN} \tag{44}$$

$$FDR=\frac{AN}{TAN} \tag{45}$$

where NN is the number that normal is identified abnormal; TNN is the total number of the normal batches; AN is the number that abnormal is identified abnormal; TAN is the total number of the abnormal batches.

V. RESULTS AND DISCUSSION

In this section, the fault detection procedure of FLKPCA is demonstrated by following actual cases, including the Fed-batch penicillin fermentation process and head width shrinkage of hot rolled strip process.

A. FED-BATCH PENICILLIN FERMENTATION PROCESS

In the section, the simulated fed-batch fermentation process for penicillin production is used to test the fault detection capability of FLKPCA. Such a process is a biochemical batch benchmark process with nonlinear dynamics and multiphase characteristics, which has been widely used for testing batch process monitoring based on multivariate statistical methods and fault diagnosis [41]–[44]. The simulation tool of this process has been released in Web at: http://simulator.iit.edu/web/pensim/simul.html. Fig 3 shows the whole simulation process of penicillin production. At first a certain amount of biomass and substrate need to be added to fermenter and the substrate need continue to be added with the reaction progresses to make sure the reaction works. Meanwhile, the reaction rate is affected by HP of the reaction environment and fermenter temperature. Detailed descriptions of the penicillin production and its applications are presented in Birol *et al.* [41]. In this study, a total of 17 process variables have been measured which related to the production process and used for fault detection. These 17 process variables are shown in TABLE 1.

TABLE 1. Overview of available measurements.

Number	Name	Unit
x_1	Fermentation volume	m^3
x_2	Dissolved oxygen	mg/L
x_3	Dissolved CO_2	mg/L
x_4	Reactor temperature	K
x_5	pH	–
x_6	Reaction heat	cal
x_7	Feed rate	L/h
x_8	Feed temperature	K
x_9	Aeration rate	L/h
x_{10}	Agitator power	W
x_{11}	Water flow rate	L/h
x_{12}	Cold water temperature	K
x_{13}	Hot water temperature	K
x_{14}	Base flow rate	mL/h
x_{15}	Acid flow rate	mL/h
x_{16}	Cumulative base flow	mL
x_{17}	Cumulative acid flow	mL

TABLE 2. Five types of faulty batches.

Fault No.	Variable that disturbance is introduced
1	Sudden change in feed substrate concentration -5%
2	Gradual change of feed rate (saturating at 0.04/0.08L/h for negative/positive drifts) -0.15%/h
3	ph sensor drift (saturating at +2) +0.01/h
4	Reactor temperature sensor drift (saturating at -5/+5℃ for negative/positive drifts) -0.10℃/h
5	Contamination (drift of $Y_{p/s}$ with -0.05/h to indicated level) -0.2

For the simulated model involved in the section, the normal feed rate is set as 0.06 L/h. A batch is terminated when total of 25 L of substrate have been added, initial conditions are randomly determined to introduce additional batch-to-batch variability. The volume decrease is used as indicator during the first (batch) phase, and used to resample the full measurement data in this phase every 0.5% decrease for a total of 201 data points. In the second (fed-batch) phase, the total amount of added substrate is used to resample all batches every 2.5 L added, resulting in 1000 data points in this phase, or a total batch length of 1201 samples. Based on the model, a total of 400 normal batches have been produced and array size of three-way normal batch datasets is $400 \times 17 \times 1201$. Five types of faulty batches are investigated for fault detection as shown in TABLE 2. For the fault batches, the feeding process is divided into 4-time interval and the fault will start in any time interval. In order to ensure statistical representability of monitoring results, each combination of time interval and fault type will be repeated 50 times, so each fault type have 200 batches. The training dataset consists of 200 normal batches which dimension is $200 \times 17 \times 1201$. The test dataset is divided into five groups according to the fault type and each group is constituted with the other 200 normal batches and 200 batches per fault type, the dimension of which is $400 \times 17 \times 1201$.

Since the trajectory of all variables do not have obvious periodic characteristic, the B-splines basis function is used to construct the trajectory function. In the work, the 50 B-splines of the fourth order are selected as basic functions and the integrated squared functions is chosen as the roughness with the smoothing parameter $\eta = 1$. Other hyper-parameters of algorithms are chosen via the generalized cross-validation approach that uses the mean squared error measure as loss function. Here, the hyper-parameters of FLKPCA, FKPCA and FKLPP methods include number of principal components $P \in \{1,2,\ldots,p\}$ (p is the total variables number), and the kernel parameter of the radial basis kernel $S \in \{10^{-3},10^{-2},\ldots,10^{7}\}$. Then evaluate all possible combinations of parameters in grid space and get the best parameters combination based on the generalized cross-validation approach. The optimal parameters and detection results of FLKPCA, FKPCA and FKLPP for the penicillin cultivation process are shown in TABLE 3. The results show that the performance of FLKPCA is superior to the FKPCA and FKLPP in TABLE 3. Although FLKPCA has a higher false alarm rate than FKPCA and FKLPP, its false alarm rate is within a reasonable range. Meanwhile, both the control charts

TABLE 3. The optimal parameters and detection result of FLKPCA, FKPCA, FKLPP for the penicillin cultivation (%).

Fault	FLKPCA				FKPCA				FKLPP			
Kernel parameter	10^6				10^5				10^5			
Latent Number	44				3				2			
	T^2		SPE		T^2		SPE		T^2		SPE	
	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR
1	2	90	6	100	0.5	1	0.5	91	2.5	99.5	0.5	68.5
2	2	100	6	100	0.5	1	0.5	89.5	2.5	98.5	0.5	68
3	2	100	6	100	0.5	4	0.5	49.5	2.5	58	0.5	21.5
4	2	100	6	100	0.5	2.5	0.5	79	2.5	80	0.5	2.5
5	2	93.5	6	100	0.5	0.5	0.5	92	2.5	99	0.5	68

TABLE 4. The optimal parameters and detection result of ICA, TPCR, LKPCA for the penicillin cultivation (%).

Fault	ICA						TPCR				LKPCA			
Kernel parameter	/						/				10^7			
Latent Number	5						3				10			
	I^2		Ie^2		SPE		T2y		T2o		T^2		SPE	
	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR
1	1	84	3	88	9	92.5	2	84.5	5.5	82.5	1	76	1.5	81
2	1	88	3	89	9	100	2	92	5.5	84	1	83.5	1.5	94
3	1	69	3	96.5	9	100	2	98	5.5	94	1	89	1.5	99
4	1	69.5	3	76.5	9	100	2	76.5	5.5	59	1	51.5	1.5	80.5
5	1	86.5	3	90	9	93	2	88	5.5	82.5	1	80	1.5	81.5

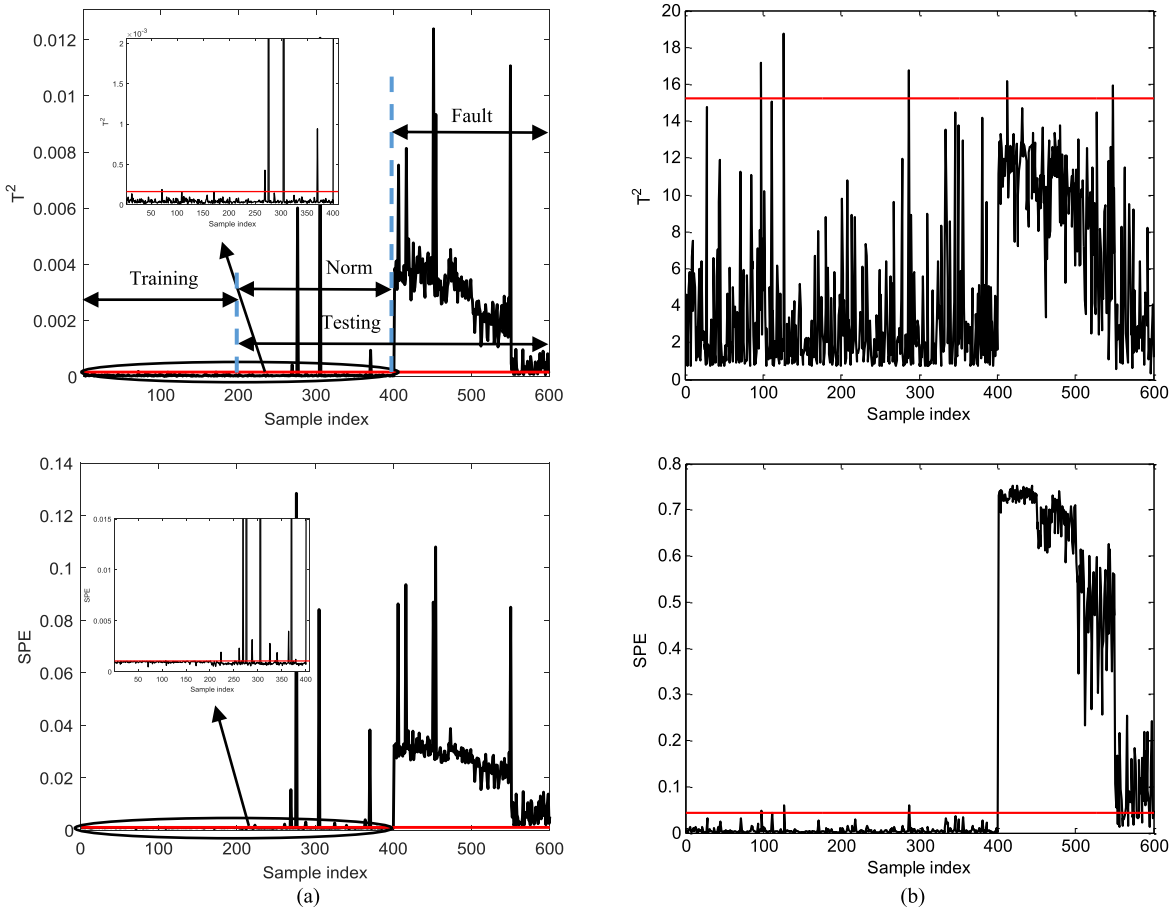

FIGURE 4. Control charts with 95% confidence limits based on FLKPCA and FKPCA for fault 1:(a) FKLPCA;(b) FKPCA.

of T^2 and *SPE* based on FLKPCA can clearly distinguish the normal and abnormal batch, whereas FKPCA and FKLPP can only get good results in one aspect. To summarize, LKPCA that considering both local and global data information show better performance of feature extraction. Besides, the results of FKLPP and FLKPCA which consider the local structure are superior to FKPCA which only consider the globe structure in all fault types.

Furthermore, compared with those algorithms which unfold three-way tensor into a two-way matrix, such as total principle component regression (TPCR) [45], independent component analysis (ICA) and LKPCA, the testing results are shown in TABLE 4. According to TABLE 4, the ICA, TPCR and LKPCA can distinguish most of the fault samples. But the performances of ICA, TPCR and LKPCA for all fault types are worse compared with FLKPCA. In a word, FDA can be considered as an effective tool to deal with batch process data.

In the Fig 4, the T^2 and *SPE* statistics control charts of FKPCA and FLKPCA for fault 1 have been exhibited. From the result of fault 1, the best monitoring result of FLKPCA

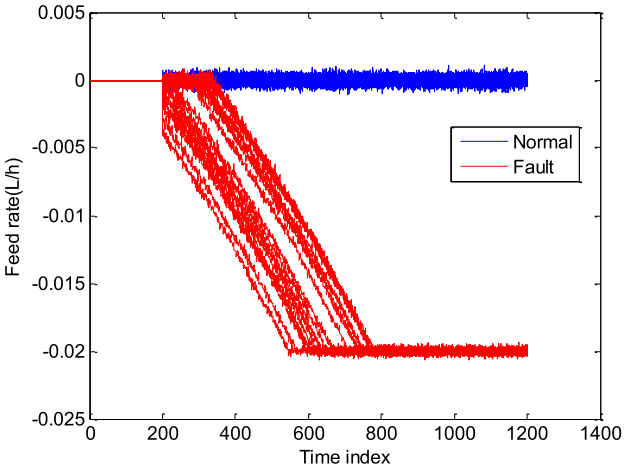

FIGURE 5. The variable trajectory of fault 2.

is the FDR of 100%, which is better than the best of FKPCA and FKLPP. In order to show the fault pattern pictorially, the trajectories of feed rate based on fault 2 are shown in Fig 5 where the blue lines represent the normal batches and the red represent the abnormal batches. In Fig. 5, the feed rate of fault batches is significantly different from the normal batches and the change starts at between 200 and 400 of sample. Likewise, the control charts of fault 2 are shown in Fig 6, and the monitoring results of FLKPCA are also better than FKPCA and FKLPP. In order to effectively evaluate

TABLE 5. Variables table of head width narrow of hot rolled strip.

	Type	Parameters names	Number
Process parameters	Temperature	Finishing entry, exit Temp	1~2
	Measured velocity	Measured velocity of Finishing mill F1~F7	3~9
	Looper value	Looper value of Number 1-6	10~15
	Roll force	① Set value of Finishing mill F1~F7 ② Measured Working side rolling force value of Finishing mill F1~F7 ③ Measured Drive side rolling force value of Finishing mill F1~F7	16~36
	Gap value	Gap value of Finishing mill F1~F7	37~43
	Speed difference	Speed difference of every two adjacent mills	44~49
	Looper height	Looper height of number 1~6	50~55
Quality		Width deviation	

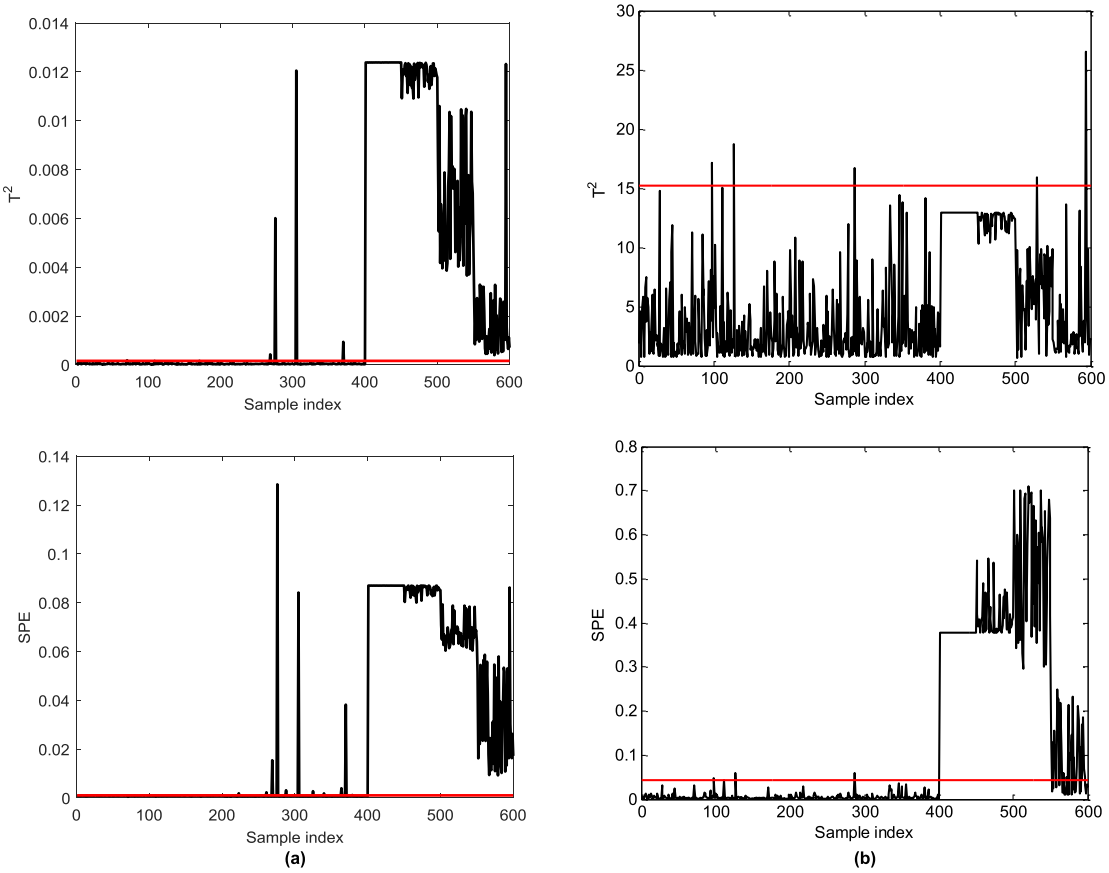

FIGURE 6. Control charts with 95% confidence limits based on FLKPCA and FKPCA for fault 2:(a) FKLPCA; (b) FKPCA.

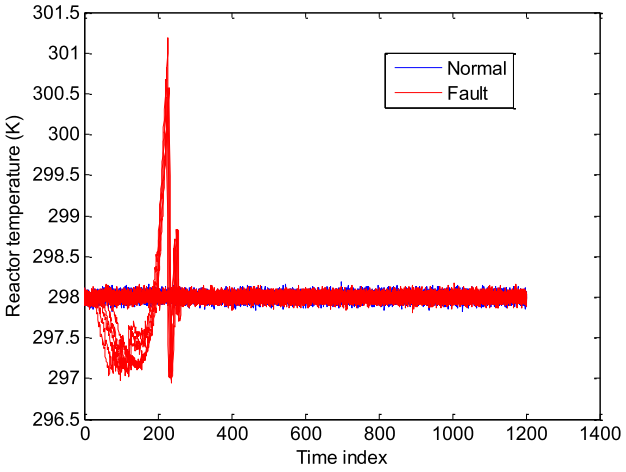

FIGURE 7. The variable trajectory of fault 4.

capabilities of fault detection, the fault 4 (reactor temperature sensor drift) is analyzed in detail here. According to those trajectories of the reactor temperature that is shown in Fig 7, among all the test batches, only 6 abnormal batches of data were significantly different from normal batches, while other abnormal have little change compared to the normal. Base on fault 4, the FLKPCA still has good performance of fault detection and can clearly distinguish between abnormal and normal batch. The testing dataset detection charts of fault 4 and 5 based on FLKPCA are shown in Fig 8. Meanwhile, the testing dataset detection charts of ICA, TPCR and LKPCA have been show in Fig 9.

B. HEAD WIDTH SHRINKAGE OF HOT ROLLED STRIP

During hot rolling, the steel slabs are heated to the temperature above recrystallization and rolled to the required thickness. The hot rolling line can be divided into the following main parts, including heating furnaces, a roughing mill, several finishing mills, cooling area, and coilers. The flowchart of hot rolling process which is involved in the section is shown as Fig 10. In this production line, the steel slabs, as raw material, are reheated target temperature about 1200∼1300 °C in the heating furnace firstly. Then, the slabs will pass through the roughing mills and the finishing mills in turn. In the roughing mill, the reheated slabs are reduced to a thickness of 25–50 mm, then the resulting sheet slabs are transported to the finishing mill, where it is further reduced to the final thickness.

Afterwards, the rolled strip is cooled continuously in the cooling area based on the process specification to meet the microstructure properties requirements. Lastly, the strip is packed into steel coils by the coiler for the convenience of transportation. In the hot strip rolling process, the increasing demands on product quality give different kinds of challenges in optimal control and process monitoring areas, particularly

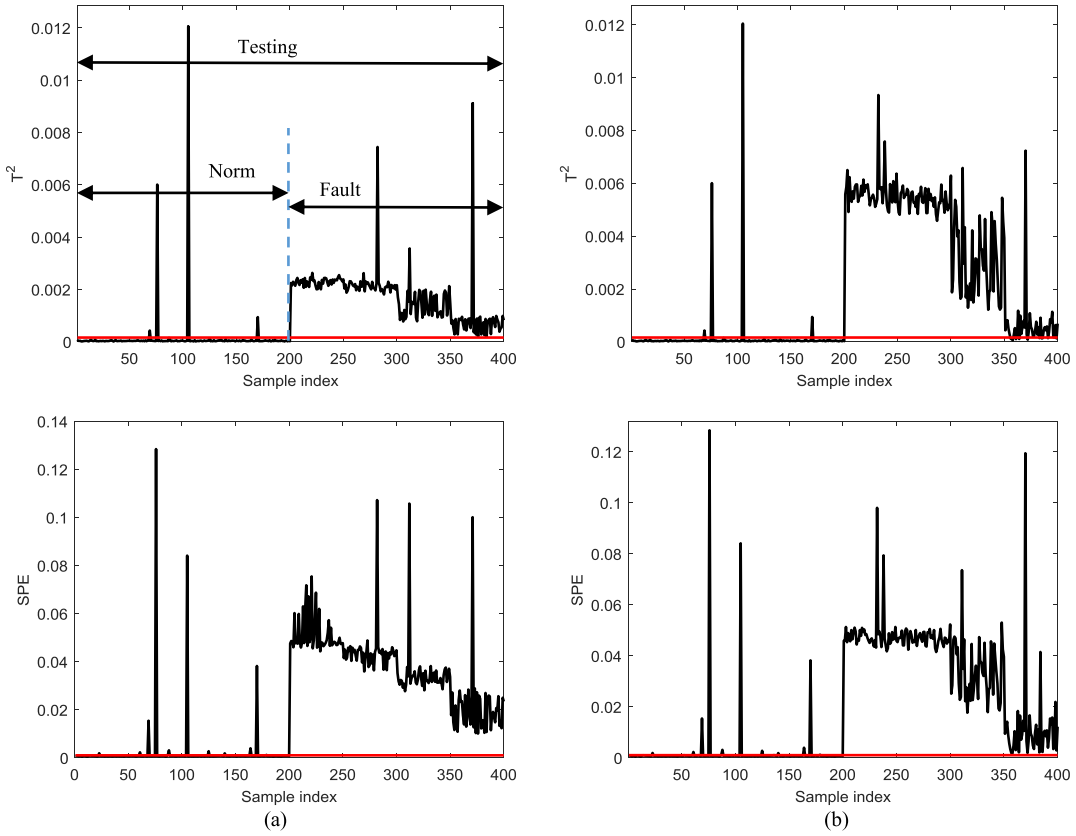

FIGURE 8. Control charts with 95% confidence limits based on FLKPCA for fault 4 and fault 5: (a) Fault 4; (b) Fault 5.

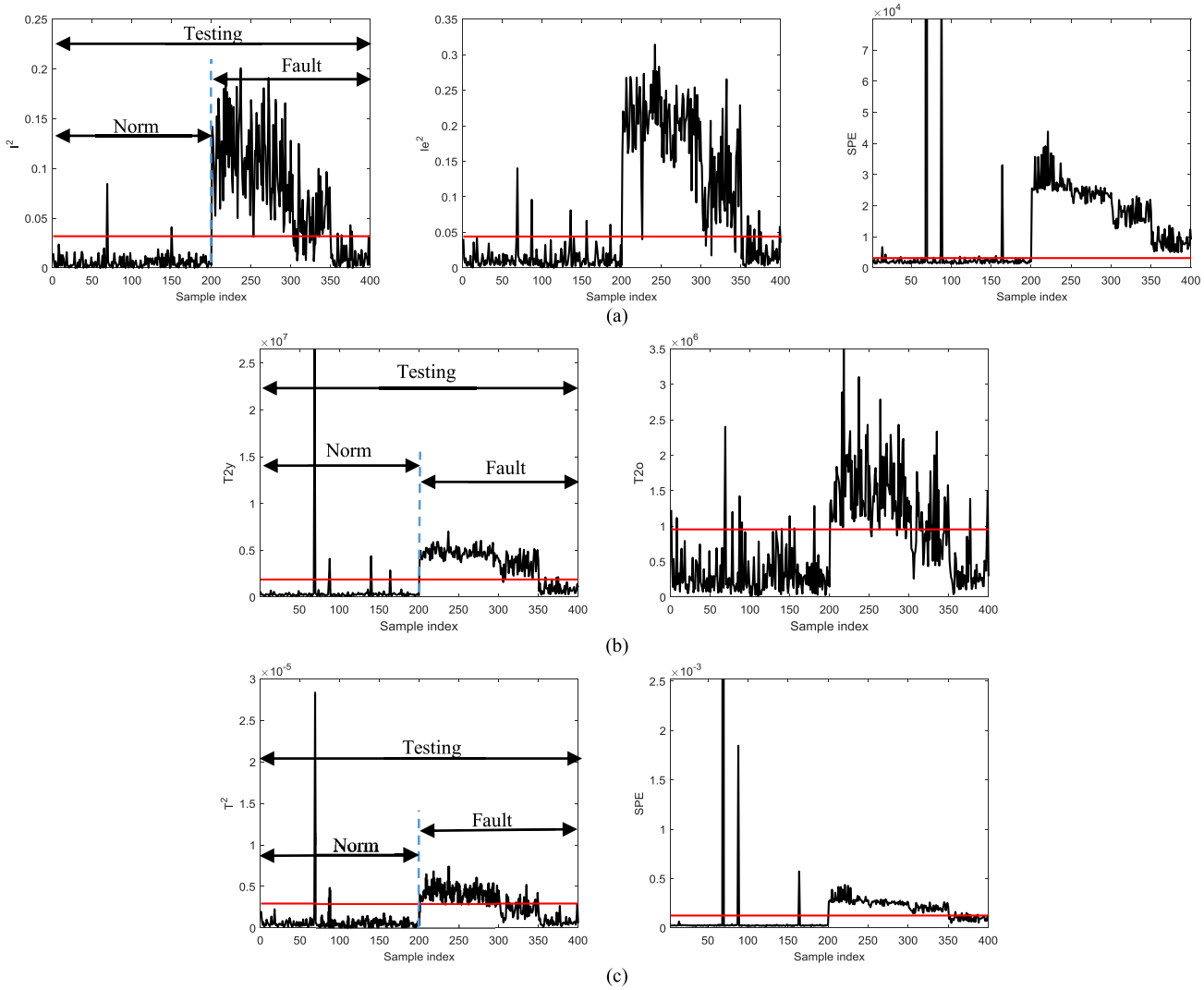

FIGURE 9. The test-set control charts of ICA, TPCR and LKPCA for fault 4: (a) ICA; (b) TPCR; (c) LKPCA.

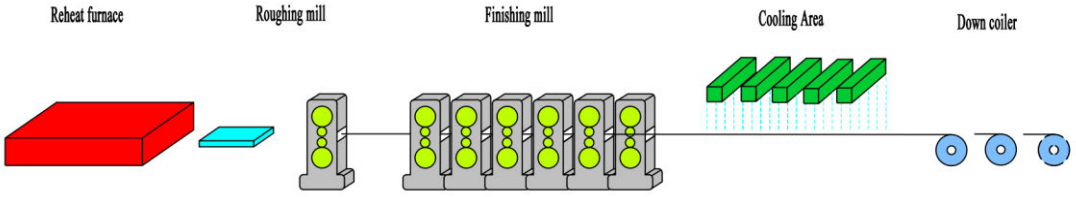

FIGURE 10. Layout drawing of hot rolling process.

the along the length, *et.al* [46]. One of the most important quality is the width accuracy because width deviation needs a larger width tolerance margin to be trimmed along the full length of the coil. The causes of head width shrinkage include: tapered cast slab along length, local temperature deviation of slabs during reheating, width shortage in mill operation, inter-stand tension jump during finishing rolling, and width contraction during the coiling process [47]. It is critical to fault detection of width shrinkage, because the width of steel strips cannot be measured online and the shape control only depends on the mathematical model. In this study, the width shrinkage is only considered due to mill operation and other reasons have been ignored. Base on expert knowledge, a total of 55 process variables which may impact the width of steel strips has been selected to construct model and shown in TABLE 5.

TABLE 6. The optimal parameters and detection result of FLKPCA, FKPCA, FKLPP for head width narrow (%).

Fault	FLKPCA				FKPCA				FKLPP			
Kernel parameter	10^7				10^5				10^5			
Latent Number	30				7				2			
	T^2		SPE		T^2		SPE		T^2		SPE	
	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR
1	0	8.7	21.88	90.43	0	0	0	5.22	21.88	81.74	25	89.57

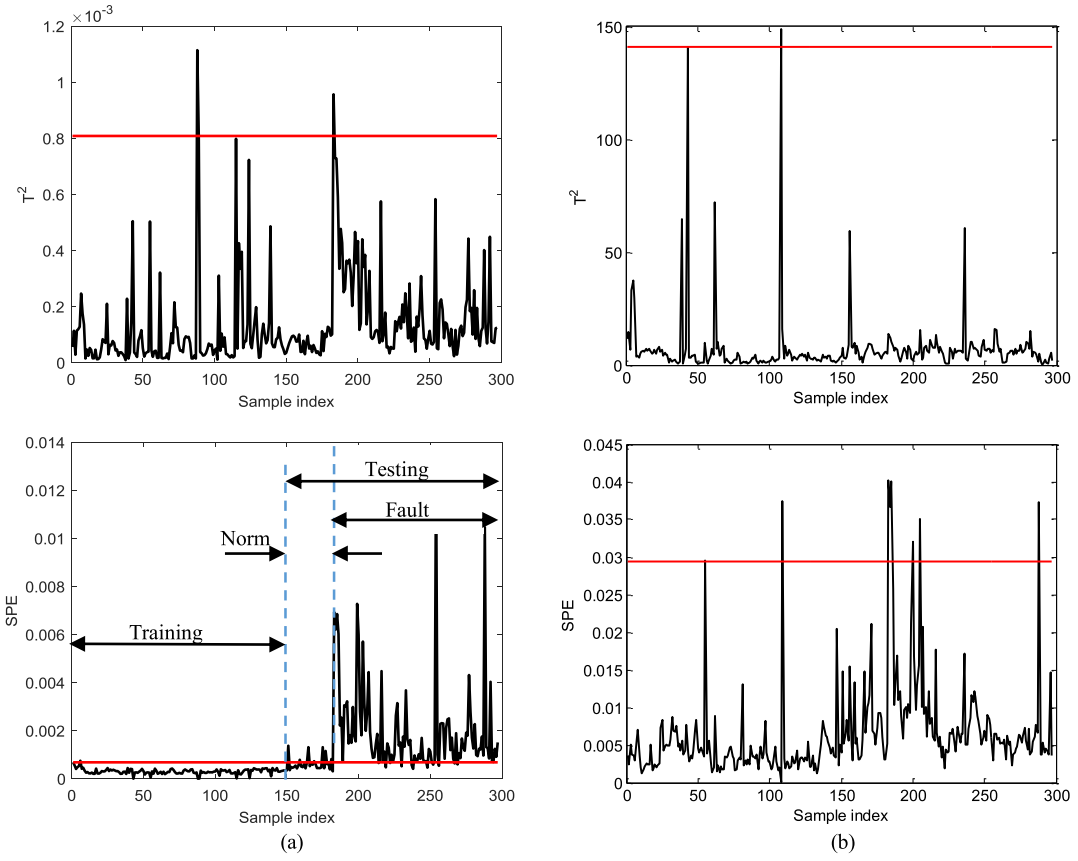

FIGURE 11. Control charts based on FKLPP and FKPCA for hot strip rolling process: (a) FKLPCA; (b) FKPCA.

The hot rolled strip data are collected from the actual production process of one steel mill in China. Here, the training dataset consists of normal data with 150 batches. Besides, the testing dataset includes 115 fault batches and 32 normal batches. Since the width shrinkage of steel strip appears on the head area, the first 1500 samples in each strip would be selected as model data to fault detection. In sum, the training dataset is a three-way data array with dimensions of 150 × 55× 1500, and the size of test dataset is 147 × 55× 1500.

Similar to the section 4, the mesh parameters can be organized by the number of principal component P∈{1,2,…, p} (p is the total variables number) and the RBF kernel parameter S∈{10^{-3},10^{-2},…,10^7}, then the best parameters combination has been selected via the generalized cross-validation approach. The optimal parameters and detection results of FLKPCA, FKPCA and FKLPP for the head width shrinkage are shown in TABLE 6. As shown in TABLE 6, it can be easily found that both the control chart based on T^2 and *SPE* statistics of FKPCA can hardly distinguish between normal and fault batches. Comparing the results of FKPCA, FLKPCA and FKLPP, the detection performance of FLKPCA is superior to FKLPP and FKPCA, where FLKPCA has a smaller false alarm rate (minimum of FAR is 28.13%) and a larger fault detection rate (maximum of FDR is 90.47%). And the control charts of FLKPCA and FKPCA have been shown in Fig. 11. Further, ICA, TPCR and LKPCA are also used for fault detection and the results are exhibited in TABLE 7. From the monitoring results, ICA has poorly fault detection result in head width shrinkage. That means that the common linear methods cannot deal with the complex relationship in

TABLE 7. The optimal parameters and detection result of ICA, TPCR, LKPCA for head width narrow (%).

`	ICA						TPCR				LKPCA			
Kernel parameter	/						/				10^7			
Latent Number	6						5				28			
	I^2		Ie^2		SPE		T2y		T2o		T^2		SPE	
	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR
1	0	35.65	3.125	32.17	0	27.83	93.75	99.13	34.38	92.17	0	0	46.87	93.04

actual production. Although LPCA and TPCR have a high fault detection rate (the FDR of LPCA and TPCR are 93.04% and 92.17%), the high false alarm rate of them (the FAR of LPCA and TPCR are 46.87% and 34.38%) would make them difficult to distinguish normal from fault. Meanwhile, it should be noted that TPCR requires the quality data, which are difficult to obtain in many industrial processes. In conclusion, the proposed FLKPCA-based monitoring method is more reliable for batch process monitoring.

VI. CONCLUSION

In this paper, a novel method, called functional local kernel principal component analysis, is proposed to fault detection in batch process. Different from the traditional batch process monitoring methods which unfold the three-ways datasets into matrix, the novel proposed method could transform the three-ways datasets into two-way function matrixes. LKPCA based on functional matrixes which integrates the local structure analysis with kernel principal component analysis is proposed to build the fault detection model. Meanwhile the Hotelling's T^2 and *SPE* have been used to construct the control charts for batch process monitoring, and control limits can be obtained based on the kernel density estimation (KDE). The performance of the proposed FKLPCA method has been demonstrated through the process monitoring of the fed-batch penicillin fermentation process and the hot strip rolling process. The results show that FLKPCA has better fault detection capability than FKPCA and FKLPP.

Lastly, there are still some open problems to be solved for future study in the paper. As for the selection of optimal parameters, the grid search method can be limited by grid spacing and parameter range, which leads to trap in local optimum. Here, we mainly focus on the problem of end-of-batch fault detection. How to deal with on-line monitoring is still an interesting topic in the future. There may be problems such as unequal length data and detection lag. Further research is still needed.

REFERENCES

[1] Z. Chen, S. X. Ding, H. Luo, and K. Zhang, "An alternative data-driven fault detection scheme for dynamic processes with deterministic disturbances," *J. Franklin Inst.*, vol. 354, no. 1, pp. 556–570, Jan. 2017.
[2] Y. Zhang, Y. Fan, and N. Yang, "Fault diagnosis of multimode processes based on similarities," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2606–2614, Apr. 2016.
[3] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20590–20616, 2017.
[4] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometric Intell. Lab. Syst.*, vol. 171, pp. 16–25, Dec. 2017.
[5] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.
[6] T. Lan, C. Tong, X. Chen, X. Shi, and Y. Chen, "KPI relevant and irrelevant fault monitoring with neighborhood component analysis and two-level PLS," *J. Franklin Inst.*, vol. 355, no. 16, pp. 8049–8064, Nov. 2018.
[7] J.-M. Lee, C. Yoo, and I.-B. Lee, "Statistical process monitoring with independent component analysis," *J. Process Control*, vol. 14, no. 5, pp. 467–485, Aug. 2004.
[8] J.-M. Lee, S. J. Qin, and I.-B. Lee, "Fault detection of non-linear processes using kernel independent component analysis," *Can. J. Chem. Eng.*, vol. 85, no. 4, pp. 526–536, May 2008.
[9] S.-K.-S. Fan, Y. Lin, C.-M. Fan, and Y.-Y. Wang, "Process identification using a new component analysis model and particle swarm optimization," *Chemometric Intell. Lab. Syst.*, vol. 99, no. 1, pp. 19–29, Nov. 2009.
[10] M. Zhang, Z. Ge, Z. Song, and R. Fu, "Global–local structure analysis model and its application for fault detection and identification," *Ind. Eng. Chem. Res.*, vol. 50, no. 11, pp. 6837–6848, Jun. 2011.
[11] Z. Su, B. Tang, J. Ma, and L. Deng, "Fault diagnosis method based on incremental enhanced supervised locally linear embedding and adaptive nearest neighbor classifier," *Measurement*, vol. 48, pp. 136–148, Feb. 2014.
[12] P. Arena, L. Patanè, and A. G. Spinosa, "Data-based analysis of Laplacian Eigenmaps for manifold reduction in supervised liquid state classifiers," *Inf. Sci.*, vol. 478, pp. 28–39, Apr. 2019.
[13] Y. Zhang, D. Ye, and Y. Liu, "Robust locally linear embedding algorithm for machinery fault diagnosis," *Neurocomputing*, vol. 273, no. 17, pp. 323–332, Jan. 2018.
[14] J. Yu, "Local and global principal component analysis for process monitoring," *J. Process Control*, vol. 22, no. 7, pp. 1358–1373, Aug. 2012.
[15] C. Tong, X. Shi, and T. Lan, "Statistical process monitoring based on orthogonal multi-manifold projections and a novel variable contribution analysis," *ISA Trans.*, vol. 65, pp. 407–417, Nov. 2016.
[16] X. Deng, X. Tian, and S. Chen, "Modified kernel principal component analysis based on local structure analysis and its application to nonlinear process fault diagnosis," *Chemometric Intell. Lab. Syst.*, vol. 127, pp. 195–209, Aug. 2013.
[17] H. Zhang, X. Tian, and X. Deng, "Batch process monitoring based on multiway global preserving kernel slow feature analysis," *IEEE Access*, vol. 5, pp. 2696–2710, 2017.
[18] Y. Yang, S. Mo, and F. Gao, "Batch process control, from traditional approaches to 2D control," in *Proc. 12th Int. Conf. Control, Autom. Syst.*, Dec. 2012, pp. 529–532.
[19] P. Nomikos and J. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE J.*, vol. 40, no. 8, pp. 1361–1375, Aug. 1994.
[20] P. Nomikos and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometric Intell. Lab. Syst.*, vol. 30, no. 1, pp. 97–108, Nov. 1995.
[21] F. Liu and Z.-G. Zhao, "On-Line batch process monitoring using multiway kernel independent component analysis," in *Proc. Int. Symp. Neural Netw.* Chengdu, China: Springer, 2006, pp. 951–956.
[22] K. Hu and J. Yuan, "Multivariate statistical process control based on multiway locality preserving projections," *J. Process Control*, vol. 18, nos. 7–8, pp. 797–807, Aug. 2008.
[23] J.-M. Lee, C. Yoo, and I.-B. Lee, "Fault detection of batch processes using multiway kernel principal component analysis," *Comput. Chem. Eng.*, vol. 28, no. 9, pp. 1837–1847, Aug. 2004.

[24] X. Tian, X. Zhang, X. Deng, and S. Chen, "Multiway kernel independent component analysis based on feature samples for batch process monitoring," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1584–1596, Mar. 2009.
[25] D.-S. Cao, Y.-Z. Liang, Q.-S. Xu, Q.-N. Hu, L.-X. Zhang, and G.-H. Fu, "Exploring nonlinear relationships in chemical data using kernel-based methods," *Chemometric Intell. Lab. Syst.*, vol. 107, no. 1, pp. 106–115, May 2011.
[26] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G. G. Barna, "A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process," *J. Chemometrics*, vol. 13, nos. 3–4, pp. 379–396, May 1999.
[27] X. Meng, A. J. Morris, and E. B. Martin, "On-line monitoring of batch processes using a PARAFAC representation," *J. Chemometrics*, vol. 17, no. 1, pp. 65–81, 2003.
[28] D. J. Louwerse and A. K. Smilde, "Multivariate statistical process control of batch processes based on three-way models," *Chem. Eng. Sci.*, vol. 55, no. 7, pp. 1225–1235, Apr. 2000.
[29] L. Luo, S. Bao, Z. Gao, and J. Yuan, "Batch process monitoring with GTucker2 model," *Ind. Eng. Chem. Res.*, vol. 53, no. 39, pp. 15101–15110, Oct. 2014.
[30] K. Hu and J. Yuan, "Batch process monitoring with tensor factorization," *J. Process Control*, vol. 19, no. 2, pp. 288–296, Feb. 2009.
[31] J. Ramsay, *Functional Data Analysis*. New York, NY, USA: Springer, 2005.
[32] F. Ferraty and P. Vieu, *Nonparametric Functional Data Analysis*. New York, NY, USA: Springer, 2006.
[33] H. Wang and M. Yao, "Fault detection of batch processes based on multivariate functional kernel principal component analysis," *Chemometric Intell. Lab. Syst.*, vol. 149, pp. 78–89, Dec. 2015.
[34] J. Chen and J. Liu, "Derivation of function space analysis based PCA control charts for batch process monitoring," *Chem. Eng. Sci.*, vol. 56, no. 10, pp. 3289–3304, May 2001.
[35] A. M. Aguilera, M. C. Aguilera-Morillo, and C. Preda, "Penalized versions of functional PLS regression," *Chemometric Intell. Lab. Syst.*, vol. 154, pp. 80–92, May 2016.
[36] F. He, C. Wang, and S.-K.-S. Fan, "Nonlinear fault detection of batch processes based on functional kernel locality preserving projections," *Chemometric Intell. Lab. Syst.*, vol. 183, pp. 79–89, Dec. 2018.
[37] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.
[38] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, vol. 14, no. 6, pp. 585–591.
[39] Y.-M. Chou, R. L. Mason, and J. C. Young, "The control chart for individual observations from a multivariate non-normal distribution," *Commun. Statist.-Theory Methods*, vol. 30, nos. 8–9, pp. 1937–1949, Jul. 2001.
[40] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Ann. Statist.*, vol. 38, no. 5, pp. 2916–2957, Oct. 2010.
[41] G. Birol, C. Ündey, and A. Çinar, "A modular simulation package for fed-batch fermentation: Penicillin production," *Comput. Chem. Eng.*, vol. 26, no. 11, pp. 1553–1565, Nov. 2002.
[42] M. Aimin, L. Peng, and Y. Lingjian, "Neighborhood preserving regression embedding based data regression and its applications on soft sensor modeling," *Chemometric Intell. Lab. Syst.*, vol. 147, pp. 86–94, Oct. 2015.
[43] J. Van Impe and G. Gins, "An extensive reference dataset for fault detection and identification in batch processes," *Chemometric Intell. Lab. Syst.*, vol. 148, pp. 20–31, Nov. 2015.
[44] X. Yuan, Z. Ge, and Z. Song, "Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes," *Ind. Eng. Chem. Res.*, vol. 53, no. 35, pp. 13736–13749, Sep. 2014.
[45] Y. Jiang and S. Yin, "Recursive total principle component regression based fault detection and its application to vehicular cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1415–1423, Apr. 2018.
[46] K. Peng, K. Zhang, B. You, J. Dong, and Z. Wang, "A quality-based nonlinear fault diagnosis framework focusing on industrial multimode batch processes," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2615–2624, Apr. 2016.
[47] K. Peng, K. Zhang, J. Dong, and B. You, "Quality-relevant fault detection and diagnosis for hot strip mill process with multi-specification and multi-batch measurements," *J. Franklin Inst.*, vol. 352, no. 3, pp. 987–1006, Mar. 2015.

FEI HE received the B.S. and Ph.D. degrees in mechanical engineering from the University of Science and Technology Beijing, Beijing, China, in 2004 and in 2010, respectively.

From 2010 to 2016, he was an Assistant Professor with the National Engineering Research Center of Flat Rolling Equipment, University of Science and Technology Beijing. Since 2016, he has been an Associate Professor with the Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing. He is the author of two books, more than 40 articles, and more than ten inventions. His research interests include industrial data analysis, quality measurement, diagnosis and control, and steel intelligent manufacturing. He was a recipient of the Beijing Youth Talents of University, in 2013.

ZHIYAN ZHANG received the B.S. degree in mechanical engineering from the Taiyuan University of Science and Technology, Taiyuan, China, in 2014 and 2018, respectively. He is currently pursuing the master's degree with the Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, Beijing, China. His research interests include statistical signal processing, machine learning theory, and industrial process monitoring.