IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 16, 2019, accepted June 28, 2019, date of publication July 5, 2019, date of current version July 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927013

Incipient Fault Detection Method Based on Stream Data Projection Transformation Analysis

YINGHUA YANG, YONGKANG PAN, LIPING ZHANG, AND XIAOZHI LIU
College of Information Science and Engineering, Northeastern University, 110819 Shenyang, China

Corresponding author: Ying-hua Yang (yhyang@mail.neu.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2017YFB0304202.

ABSTRACT Early detection of incipient faults is a challenging task in the field of chemical process monitoring. For this problem, this paper proposes a new data-driven process monitoring method called stream data projection transformation analysis (SDPTA). First, we determine a set of projection transformation vectors, orthogonal basis vectors, based on original data to solve the problem that the data space original basis vector has relevance. Then, we use a sliding window to project data onto the basis vectors to obtain the basis vector components which is defined as projection transform components (PTCs). In this way, the stream data local sequence information can be utilized effectively. Furthermore, each PTC represents the coverage of local data on the corresponding basis vector. The length of PTCs can reveal some important process features, implying that condition changes can be detected by monitoring the length of PTCs. Finally, the potential of the window-based SDPTA method in monitoring continuous processes is explored using two case studies (a numerical example and the challenging Tennessee Eastman process). The performance of the proposed method is compared with the existing MSPM methods, such as PCA, DPCA, and RTCSA. The monitoring results clearly demonstrate the superiority of our method.

INDEX TERMS Incipient fault detection, projection transformation analysis, sliding window, stream data.

I. INTRODUCTION

As modern industrial processes become very complicated, large-scale and highly invested, fault diagnosis technology shows its great value in ensuring process safety and improving product quality. In the past several decades, industrial process fault diagnosis methods have been discussed extensively by researchers [1], and existing methods are usually divided into model, knowledge, and data-based classes. Among these three classes, data-based fault diagnosis methods have become an increasingly hot topic in recent years, because a large amount of industrial process data can be collected and stored in the computer control system database in real time [2]. Some representative multivariate statistical process monitoring (MSPM) methods include principal component analysis (PCA) [3]–[5], partial least squares (PLS) [6]–[8], independent component analysis (ICA) [9], [10]. As one of most well-known data-based fault diagnosis methods, PCA and its extensions have been studied in depth. JM Lee *et al.* proposed a nonlinear process monitoring method based on kernel principal component analysis (KPCA), which improved the monitoring performance of PCA for nonlinear processes [11]. Ku W *et al.* introduce dynamic linear model identification and time lag shift into statistical process monitoring (DPCA) [12]. SJ Qin proposed RPCA algorithm and applied it to a rapid thermal annealing process in semiconductor processing for adaptive monitoring [13]. The above-mentioned methods have proven their effectiveness in different cases. However, they are not focused on incipient fault detection tasks.

In practical cases, numerous abnormal conditions gradually evolve from incipient faults [14]. Incipient faults usually develop slowly, this implies that, if faults are detected in their incipient stages, abnormal conditions may be effectively avoided. However, the incipient fault which has a small magnitude including early changing and the slow developing may be buried by nonstationary trends resulting in low fault detection rate [15]. Therefore, incipient fault detection is a challenging work. In recent years, some researchers have done some research on issues related to incipient fault detection. Deng *et al.* [16] propose an enhanced SLKPCA method which monitors the process changes based on the residual

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

vectors computed by the statistical local approach, highlight the influence of the faulty residual samples. Zhang *et al.* [15] propose a two-layer fault detection method to detect the incipient fault for multiphase batch processes with limited batches. Ge *et al.* [17] combining the wavelet analysis with the residual evaluation was carried out for monitoring the incipient fault. Youssef A *et al.* [18] propose an incipient fault detection method based on the combination of Kullback-Leibler Divergence (KLD) and Principal Component Analysis (PCA). Pilario and Cao [19] presented a canonical variate dissimilarity analysis (CVDA) method for process incipient fault detection in nonlinear dynamic processes under varying operating conditions. For these methods, it is common that process variables are monitored to detect faults. Statistical models are built by sufficient training data based on correlations, which lead to the decomposition of the original measurement space [20]. During online monitoring, sample vectors are directly projected onto corresponding subspaces. This implies that the latest sample is projected separately without considering sequence information among samples. When detecting incipient faults, samples belonging to normal and abnormal conditions usually overlap to a large extent owing to their small fault magnitudes. As a result, conventional sample-wise MSPM methods easily lead to a significant number of missed detections.

One possible solution to reduce the missed detection rate is to utilize sequence information among measurements. Window-based monitoring methods can partially alleviate data overlap. He and Wang [21] proposed statistics pattern analysis (SPA) to address the challenges encountered in semiconductor processes, which was also extended to continuous process monitoring [22]. Instead of monitoring process variables, SPA monitors the statistics of process variables in sliding windows, demonstrating a superior performance over PCA and DPCA. However, SPA may not effectively detect some incipient faults with small magnitudes. J. Shang, M. Chen, H. Ji [23] proposed an MSPM method called recursive transformed component statistical analysis (RTCSA), which processes data in sliding windows to obtain orthogonal transformed components (TCs). In this way, some incipient faults can be detected by monitoring the statistics of TCs. These methods can provide better fault detection performance than the traditional MSPM methods. However, they do not take the effects of the original basis vector correlation of the data space into account. Therefore, how to eliminate the original basic vector correlation and how to reasonably extract and use the data local sequence information are still a valuable problem, which provides motivation for the research of this paper.

To provide better monitoring performance for incipient fault monitoring, we propose a new MSPM method called stream data projection transformation analysis (SDPTA). In this proposed method, the projection transform components (PTCs) extracted through the sliding window, which represents the coverage of local data on the corresponding basis vector. That is, PTCs can reflect some important process features. In addition, PTCs are orthogonal, and the sample length of PTCs are equivalent to the eigenvalues of the sample covariance matrix of the normalized window-data. This makes monitoring the length of PTCs more convenient than monitoring the PTC.

The remainder of the paper is organized as follows. In section 2, the traditional PCA-based monitoring method is briefly reviewed. In Section 3, the algorithm of SDPTA is introduced in detail, including projection transformation, basis vector selection, projection length quantization and its fault detection flow. In section 4, two simulation examples are used to demonstrate the performance of the proposed method: a numerical example and the Tennessee Eastman process (TEP). Conclusions are given in section 5.

II. PRINCIPAL COMPONENT ANALYSIS

Consider the original measurement matrix $X \in \mathrm{R}^{n\times m}$ where n and m denote the number of samples and measured variables, respectively. First, normalize the data matrix to get X^*, as follows.

$$X^* = (X - I_n^{\mathrm{T}}\bar{X})\Sigma^{-1} \tag{1}$$

$$\bar{X} = \frac{1}{n} \times I_n \times X \tag{2}$$

where $\bar{X} \in \mathrm{R}^m$ is column mean vector of X; the diagonal matrix $\Sigma \in \mathrm{R}^{m\times m}$ denotes the standard deviation for each column of X; $I_n = [1, 1, \cdots, 1] \in \mathrm{R}^n$.

By the SVD algorithm, the matrix X^* is decomposed as follows:

$$X^* = T_q P_q^{\mathrm{T}} + E_{m-q} = T_q P_q^{\mathrm{T}} + \tilde{T}_{m-q}\tilde{P}_{\mathrm{m}-q}^{\mathrm{T}} = [T\tilde{T}][P\tilde{P}] \tag{3}$$

where $T \in \mathrm{R}^{n\times q}$ and $P \in \mathrm{R}^{m\times q}$ are the score and loading matrices, respectively. In general, we use Cumulative Percent Variance (*CPV*) criterion to extract q principal components, that is

$$CPV_q = \sum_{i=1}^{q} \lambda_i / \sum_{i=1}^{m} \lambda_i \geq 0.85 \tag{4}$$

The PCA projection reduces the original set of m variables to q principal components (PCs), where λ_i $(i = 1, \cdots, m)$ is the eigenvalue of the covariance matrix $[1/(n-1)](X^*)^{\mathrm{T}}X^*$ and satisfies $\lambda_1 > \lambda_2 > \cdots \lambda_m$.

For fault monitoring on a new sample vector x, Hotelling's T^2 and *SPE* charts are commonly used. Hotelling's T^2 is a measure of the variation in relative principal component subspace (PCS):

$$T^2 = xP\Lambda^{-1}P^{\mathrm{T}}x^T \tag{5}$$

where Λ is the diagonal matrix containing the first q eigenvalues of λ_i $(i = 1, \cdots, m)$ in descending. For a given significance level α, the process is considered normal if $T^2 \leq T_\alpha^2$, where the upper control limit T_α^2 can be obtained using the F-distribution. The SPE statistic indicates how well

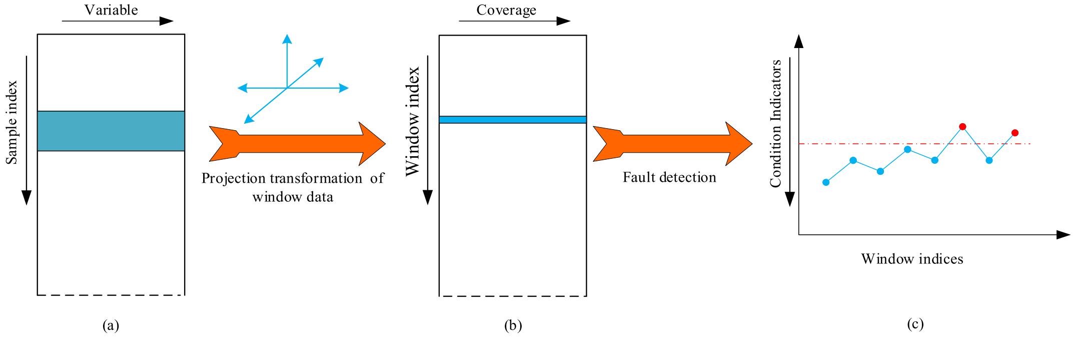


FIGURE 1. Schematic of the window-based SDPTA method for monitoring continuous processes: (a) original process data; (b) computed the length of PTCs; (c) fault detection. A block or a window of process variables shaded in (a) is used to generate a Projection length shaded in (b), which is then used to generate a point of dissimilarity measure shaded in (c) to perform fault detection.

each sample conforms to the model, measured by the projection of the sample vector onto the residual subspace (RS):

$$SPE = \|\tilde{x}\|^2 = \left\|(I - PP^{\mathrm{T}})x\right\|^2 \quad (6)$$

The process is considered normal if $SPE \leq \delta_\alpha^2$, where δ_α^2 denotes the upper control limit for SPE with a significance level α. δ_α^2 can be computed by assuming that the process data follow a normal distribution.

III. STREAM DATA PROJECTION TRANSFORMATION ANALYSIS

For most MSPM methods, during online monitoring, sample vectors are directly projected onto corresponding subspaces. This implies that the latest sample is projected separately without considering sequence information among samples. To address this issue, we introduce streaming data processing methods into process monitoring. Stream data consists of data continuously generated by stream sources, e.g., sensors, enriched with process information [24], [25]. Hence, in this paper, we regard the process data collected at uniform sampling frequency as streaming data. This method selects data in order by sliding the window, and processes window data to obtain critical sequence information. It also has good real-time performance while utilizing data sequence information. On the other hand, in actual industrial production, due to the influence of noise and sensor measurement random error, we cannot use one sampling information as the base for determining the fault state. By considering the local sequence relationship of the data and extracting the features of the local data, better fault monitoring results can be obtained.

The basic idea of the SDPTA based process monitoring is that the length of PTCs under abnormal conditions will show some deviation from the distribution of the length of PTCs under normal operation. Therefore, in the stream data projection transformation analysis method, the process behavior is characterized by length of PTCs of the process variables instead of by the process variables themselves. In other words, the SDPTA fault detection method monitors the variance covariance structure of the length of PTCs instead of the variance-covariance structure of the process variables.

As shown in Fig. 1, two steps are involved in the SDPTA based monitoring for continuous processes: calculation of projection length and dissimilarity discrimination. For a continuous process, the length of PTCs is the projected length of the window (or segment) of the process measurement on basis vector. It should be noted that the length of PTCs is related to the width of the window. After PTCs being calculated from the original data, the dissimilarity between the training PTCs are quantized to determine an upper control limit of the detection index. In this work, we applied PCA to quantify the differences between training PTCs and defined two test indicators similar to Hotelling's T^2 and SPE. When a new measurement block is available for fault detection, the window moves one or more samples forward and calculates a new PTC; then its dissimilarity to the training PTCs is quantified and compared with the threshold to perform fault detection.

A. PROJECTION TRANSFORM COMPONENTS

Consider the original measurement matrix $X_{ori} \in \mathrm{R}^{n\times m}$, where n and m denote the number of samples and measured variables, respectively. X^* is normalized data matrix of X_{ori}.

$$X_{ori} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_m(1) \\ x_1(2) & x_2(2) & \cdots & x_m(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(n) & x_2(n) & \cdots & x_m(n) \end{bmatrix} \quad (7)$$

$$X^* = \frac{X_{ori} - I_n \mu_{ori}^{\mathrm{T}}}{\Sigma_{ori}} \quad (8)$$

where $\mu_{ori} \in \mathrm{R}^{1\times m}$ is the column mean vector of the sample matrix X_{ori};$\Sigma_{ori} \in \mathrm{R}^{m\times m}$ is the standard deviation matrix of the sample matrix X_{ori};$I_n = [1, 1, \cdots, 1] \in \mathrm{R}^n$.

As shown in Fig. 2, the relationship between the data sample X^* and the basis vector is shown. Where, the blue dot approximation represents the row (observation point) information of the matrix X; The vector (black axis) approximation represents the column (variable) information of the matrix X^*; The point-to-vector distance approximation

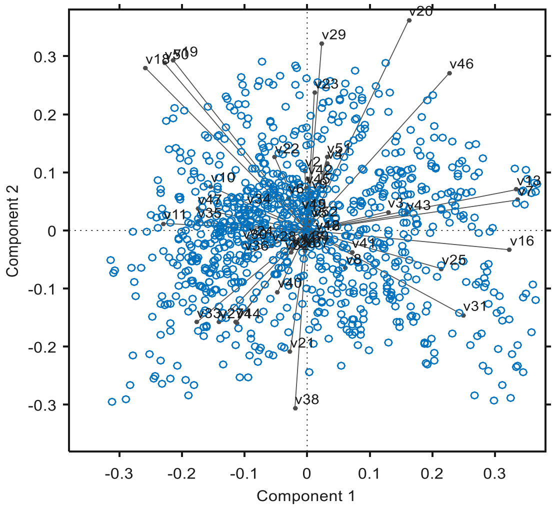


FIGURE 2. Biplot View of the data subspace. (Blue dots represent high dimensional data samples; black axis represents the basis vector).

represents the value of the sample point in the normalized matrix under this column index.

To eliminate the effects of the correlation between the original measured variables, we need to re-determine a set of independent basis vectors as the data projection transformation vectors. Considering the SVD decomposition of the data matrix can automatically provide us with such a set of basis vectors. Therefore, we use the covariance matrix of X^* to find a set of basis vectors. R_{X^*} is covariance matrix of X^* given by

$$R_{X^*} = \frac{(X^*)^{\mathrm{T}} X^*}{n-1} \tag{9}$$

As the covariance matrix R_{X^*} is Hermitian matrix, which can be decomposed as

$$R_{X^*} = P \Lambda P^{\mathrm{T}} \tag{10}$$

where the diagonal matrix $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_m)$ denotes the eigenvalues of R_{X^*}, and $P \in \mathrm{R}^{m \times m}$ denotes the corresponding eigenvectors. The matrix P consisting of m eigenvectors of the covariance matrix R_{X^*} is the basis vectors we are looking for. Then, the data matrix X^* can be transformed into the following forms:

$$T = X^* P \tag{11a}$$

$$\Leftrightarrow \begin{bmatrix} t_1 & t_2 & \cdots & t_m \end{bmatrix} = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix} \begin{bmatrix} p_1 & p_2 & \cdots & p_m \end{bmatrix} \tag{11b}$$

$$t_j = X^* \times p_j \tag{11c}$$

where $T \in \mathrm{R}^{n \times m}$ is the projection of data matrix X^* under the basis vectors. Each row of T represents the new coordinates of a normalized sample point. t_j is the jth column of T, which represents the coverage of the data X^* on the jth projection vector p_j. In the traditional PCA method, P denotes the loading matrix, and T denotes the score matrix.

After determining the projection transformation vectors, we need to perform a projection transformation on process measurements data. Construct a one-step sliding window $X_k \in \mathrm{R}^{w \times m}$ to select normalized process measurements data, where w is window width.

$$X_k = \begin{bmatrix} x_1(k-w+1) & x_2(k-w+1) & \cdots & x_m(k-w+1) \\ x_1(k-w+2) & x_2(k-w+2) & \cdots & x_m(k-w+2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(k) & x_2(k) & \cdots & x_m(k) \end{bmatrix} \tag{12}$$

By projecting the selected data X_k onto the basis vectors P, we can obtain the projection transform components (PTCs) of the window-data which is recorded as T_k^*. Similar to formula (11), the projection of the window data on the basis vectors can be defined as

$$T_k^* = X_k P \tag{13}$$

Similarly, T_k^* represents the coverage of the kth window-data X_k on the basis vector.

According to the relevant properties of the matrix, we can get two conclusions: first, PTCs are orthogonal (or T_k^* are orthogonal). Second, the sample length of PTCs are equivalent to the eigenvalues of the sample covariance matrix of the normalized data X_k (*See "Appendix 1" for detailed proof*). Considering the orthogonal PTCs, the quantization of PTCs can be converted into the following form:

$$\frac{1}{w-1} T_k^{*\mathrm{T}} T_k^* = \Lambda_{new} \tag{14}$$

That is, the length of PTCs corresponds to the elements on the diagonal of Λ_{new}.Where $1\big/(w-1)$ is scale factor which can eliminates the effect of window width on PTCs quantization.

After the movement of the window on the data matrix X^*, we can get a series of values that represent the coverage of the data on the load vector (the elements on the Λ_{new} diagonal), and write these values into a new matrix X_{train}, where the kth row element of X_{train} represents the measure of the data X_k on the basis vector.

Remark 1: When using the original data to establish the principal component model, we determine the main elements according to the size of the score vector T_q, refer to the form of equation (3). The size of the score vectors reflects the coverage of the data on the corresponding load vector. It is hard to remark that the score vector with a small coverage on the load vector belongs to the measurement error space, or is independent of the system state estimation. More importantly, these discarded loading vectors may also contain important system information. Therefore, we select all the loading vectors obtained from the original data as the basis vectors. We implement projection transformation of data without space division to obtain more data information. When we take $q = m$ in equation (3), the measurement error of the output matrix X is also taken into account in the estimated observation state. Although this will include the measurement error information, the established model can include

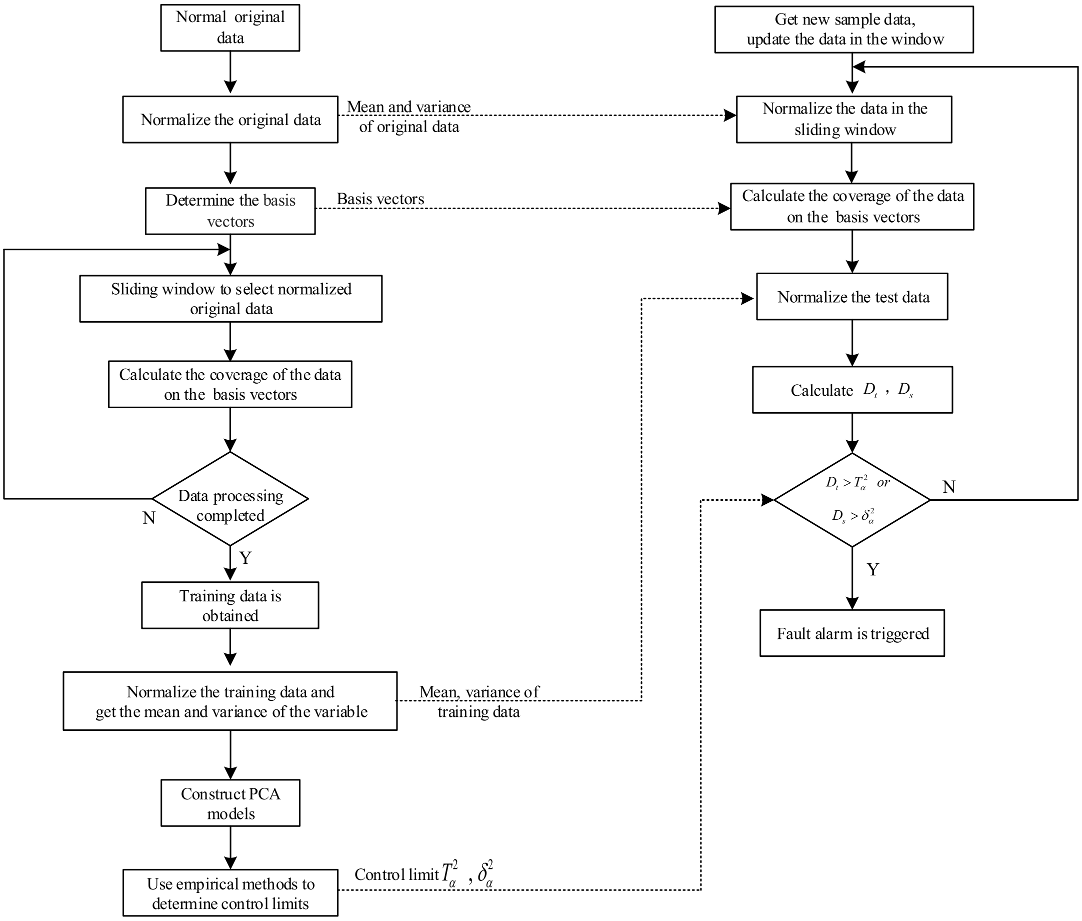

Normal original data
Normalize the original data
Mean and variance of original data
Determine the basis vectors
Basis vectors
Sliding window to select normalized original data
Calculate the coverage of the data on the basis vectors
Data processing completed
N
Y
Training data is obtained
Normalize the training data and get the mean and variance of the variable
Mean, variance of training data
Construct PCA models
Use empirical methods to determine control limits
Control limit T_α^2 , δ_α^2
Get new sample data, update the data in the window
Normalize the data in the sliding window
Calculate the coverage of the data on the basis vectors
Normalize the test data
Calculate D_t , D_s
$D_t > T_\alpha^2$ or $D_s > \delta_\alpha^2$
N
Y
Fault alarm is triggered

FIGURE 3. Flowchart of SDPTA-based fault monitoring.

the measurement error by obtaining the system measurement data for a long time. It should be noted that when we collect system normal state operation data, we should collect a long uptime to obtain a more comprehensive status description of the system. In other words, if we collect less running state data, the original data we get will not reflect the state of the system more.

B. DISSIMILARITY QUANTIFICATION AND FAULT DETECTION

We use SDPTA method to improve the separability between normal data and fault data. In this section, we also need a method to measure the dissimilarity of data.

Once the training data are obtained for normal original data, the next step is to quantify the dissimilarities among them and determine an upper control limit for the normal operation. Dissimilarities between different objects are usually quantified by distance-based or angle-based metrics. In fact, most fault detection methods such as the PCA-based methods adopt distance-based metrics.

In this work, we use PCA to assess the dissimilarity among training data obtained from normal operation data. In other words, we perform PCA on the training data to determine the upper control limits. To distinguish the SDPTA-based fault detection indices from the traditional PCA-based fault detection indices, we use D_t and D_s to denote the T^2 and SPE in SDPTA method. The process is considered normal if the dissimilarity indices are below the thresholds, i.e. $D_t \leq T_\alpha^2$ and $D_s \leq \delta_\alpha^2$, where T_α^2 and δ_α^2 denote the upper control limits for dissimilarity index in PCS and RS with a significance level α. It is worth noting that PCA is just one way to determine the similarities or dissimilarities among different samples; other methods can be implemented to obtain distance-based or angle-based similarity indices [26]. T_α^2 and δ_α^2 in SDPTA method can be obtained in empirical method, which based on the calibration or validation data under normal operation conditions [21], [23], [27]. For example, a 98% confidence upper control limit can be determined as D_t and D_s value below which 98% of the calibration samples are located.

C. MODEL DEVELOPMENT AND THE ONLINE MONITORING STRATEGY

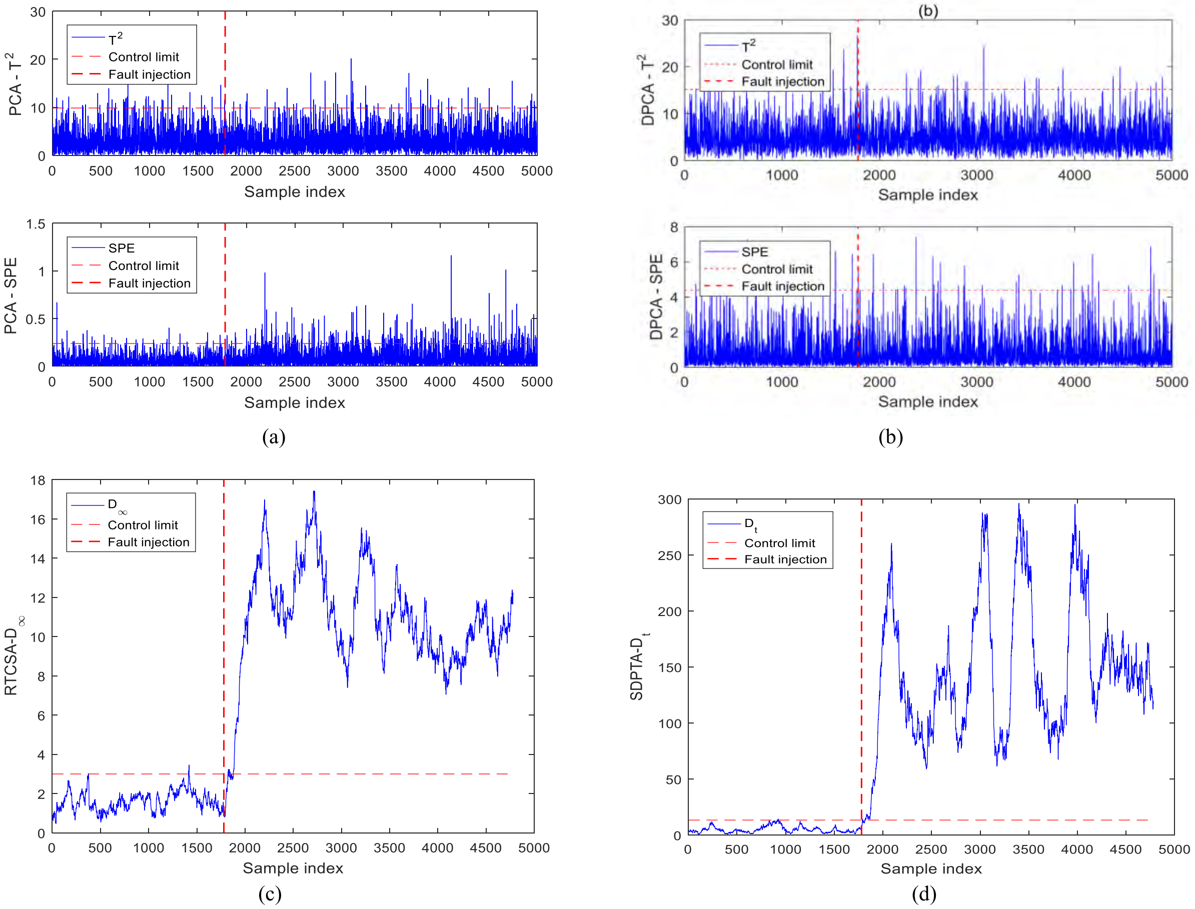
The scheme of the SDPTA-based fault monitoring is outlined in Fig. 3.

Offline Modeling:

(1) Acquire the normal original data set $X_{ori} \in \mathrm{R}^{n \times m}$ from the normal operating condition and normalize the data using

TABLE 1. Detection rates (%) of different MSPM methods for the numerical example.

Fault index	Type	Parameter	PCA		DPCA		RTCSA	SDPTA
			T^2	SPE	T^2	SPE	D_∞	D_t
1	Sensor constant bias	$f = 0.06$	2	5.1	1.5	2.4	94.6	96.23
2	Sensor gain degradation	$\eta = 0.93$	2.7	5.6	1.7	1.5	92.9	96.1
3	Sensor precision degradation	$\sigma_e = 0.08$	2.3	4.2	1.5	1.5	98	98.9
4	Additive process fault	$f = 0.67$	5.2	2	2.9	9.2	94.1	97.5



(a)

(b)

(c)

(d)

FIGURE 4. Detection performances of (a) PCA, (b) DPCA, (c) RTCSA and (d) SDPTA for fault 3 occurring at x1.

the mean and standard deviation of each variable, where m is the number of variables and n is the sample number. We can get the standardized data X^*.

(2) Perform covariance-based SVD decomposition on X^*, we can determine the projection transformation vectors $p_i \in P(i = 1, \cdots, m)$.

(3) Use sliding window to select data from X^*. Projecting the window data X_k (window width set as w) onto the basis vectors and calculate the coverage of the data on the basis vectors.

(4) Repeat step (3) until all original data has been processed, we can get training data $X_{train} \in R^{(n-w+1)\times m}$, where m is the number of variables and $(n - w + 1)$ is the sample number.

(5) Normalize data X_{train} using the mean and standard deviation of each variable.

(6) Perform PCA decomposition on training data to quantify the difference among normalized training data.

(7) Calculate the monitoring statistics (D_t and D_s) of the normalized training data. Determine the control limits of D_t and D_s statistics based on a specified significance level α.

Online monitoring:

(1) Collect and update data in the window. Normalize the new data stream X_k with the mean and standard deviation values of normal original data X_{ori}.

(2) Projecting the data stream X_k onto the basis vectors and calculate the coverage of the data on the basis vectors.

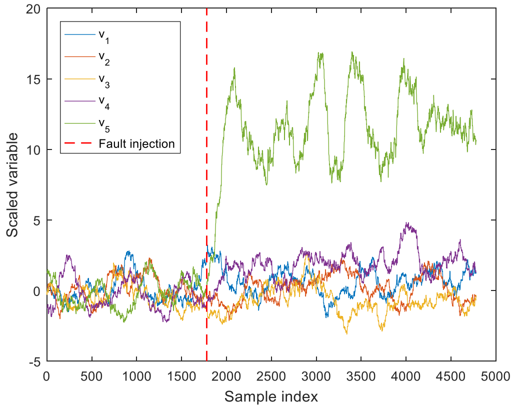


FIGURE 5. Relative variations of PTCs in SDPTA for fault 3 occurring at x1.

(3) Get the new test data x and normalize it with the mean and standard deviation values of the training data X_{train}.

(4) Calculate the monitoring statistics (D_t and D_s) of the data x.

(5) Monitor whether D_t or D_s exceeds its control limit α.

IV. CASE STUDIES

In this section, the proposed SDPTA monitoring method is applied to a numerical example and Tennessee Eastman process to evaluate the method's performance. The change of data sampling rate in practical application may affect the correct acquisition of data and the experimental results [28]. In this paper, the experimental data have been collected on uniform data rates. Therefore, the effect of data sampling rate and reliability on the experimental results is not considered in the paper.

A. CASE STUDY OF A NUMERICAL EXAMPLE

Consider a general multivariate process [29] represented by

$$x = As + e \tag{15}$$

where the measurement vector $x \in \mathrm{R}^m$ has m variables, the coefficient matrix $A \in \mathrm{R}^{m\times r}$ is assumed to be column full rank, $s \in \mathrm{R}^r$ denotes r independent data sources ($r < m$), with each sample i.i.d., $e \in \mathrm{R}^m$ denotes Gaussian white noises. Refer to the work of Shang *et al.* [23] to write the equation specifically as follows:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0.2183 & -0.1693 & 0.2063 \\ -0.1972 & 0.2376 & 0.1736 \\ 0.9037 & -0.1530 & 0.6373 \\ 0.1146 & 0.9528 & -0.2624 \\ 0.4173 & -0.2458 & 0.8325 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}$$

where s represents independent Gaussian distributed data source with mean $[2.3,\ 1.7,\ 3.1]^{\mathrm{T}}$ and unit standard deviation, respectively, and e denotes Gaussian white noises with standard deviation $[0.061, 0.063, 0.198, 0.176, 0.170]^{\mathrm{T}}$. Four types of incipient faults are given as follows:

(1) Sensor constant bias: $x = x^* + f$,

(2) Sensor gain degradation: $x = \eta x^*$,

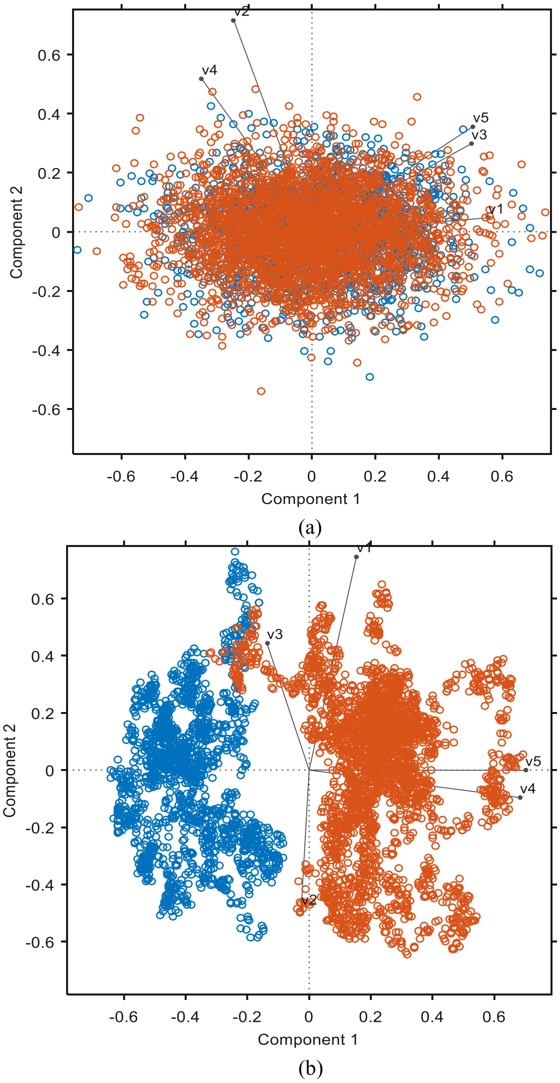


FIGURE 6. The visualization of the fault 3 data., (a) and (b) represent the results before and after being processed by the SDPTA method, respectively.

(3) Sensor precision degradation: $x = x^* + e$,

(4) Additive process fault: $s = s^* + f$.

The training and testing datasets, respectively, have 20000 and 5000 samples. All the faults are introduced at the 2001st sample index for convenience, assume sensor faults occur at x_1, whose standard deviation is 0.345, and process faults occur at s_1.

The detection performance of SDPTA and other MSPM methods including PCA, DPCA and RTCSA are evaluated. For comparison, we adopt the same confidence level for different methods. Parameter settings of different MSPM methods give as follows: For PCA and DPCA, CPV (90%) is used for determining the number of PCs; For DPCA, maximum time lag is selected as 2; For RTCSA, select all optional statistics (see Shang& chen [21], for details), and ℓ_∞ norm is used as scalarization; The window widths of RTCSA and SDPTA are all set as 220 and the window shifting step

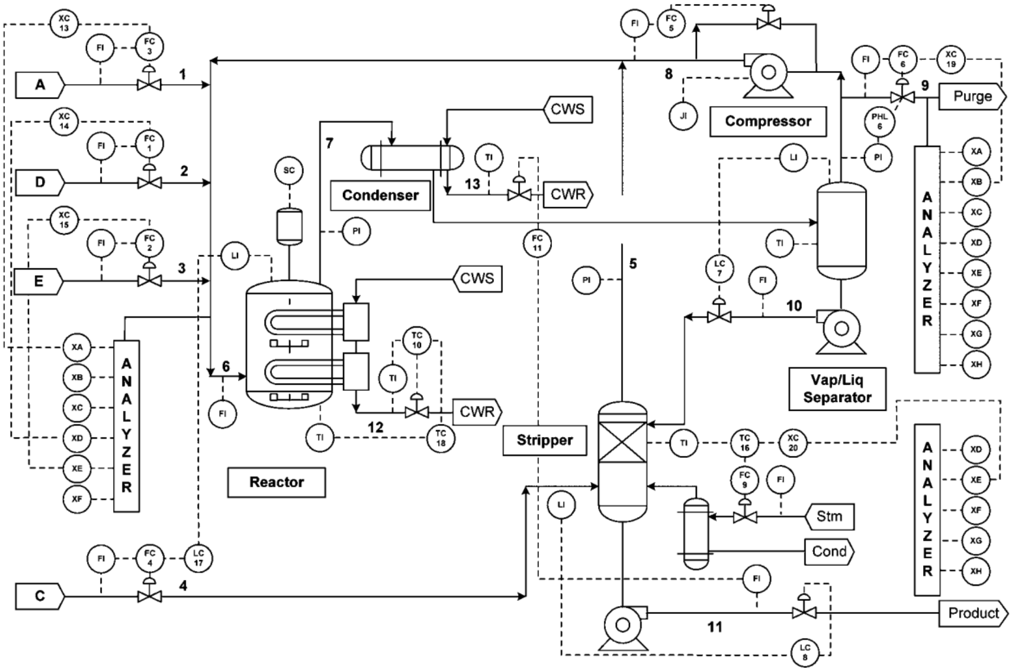


FIGURE 7. Process layout of the Tennessee Eastman process.

is set as 1. The significance level is set as 1%. Their fault detection rates are listed in Table 1, which illustrates that SDPTA is more sensitive to incipient faults in most cases.

For fault 3, the detection performances of different MSPM methods are illustrated in Fig. 4. Note that PCA fails to detect this type of fault, with detection rates close to the significance level. A possible reason is that PCA separately projects each sample into the subspace, without utilizing sequence information among measurements. If the fault magnitude is small enough, it is easily masked by the variation of process variables and measurement noises. In this time-invariant process, DPCA has a limited performance; hence, it almost has the same detection rate with PCA. SDPTA and RTCSA can effectively detect this fault.

Fig. 5 illustrates the relative changes of data projection length in SDPTA based process monitoring for fault 3. It is obvious that variable 5 relative changes most intensively after the fault occurs while the others maintain their original fluctuation levels, then the fault is precisely detected.

Fig. 6(a) shows the visualization of the fault 3 original data. Fig. 6(b) shows the visualization of the fault 3 data processed by the SDPTA method. By comparison, it can be found that the SDPTA method can significantly improve the separability between normal data and abnormal data.

The Biplot diagram in Fig. 6 is a PCA-based high-dimensional data visualization method, which only plays a role in data display and does not affect the separability of data. The components in Fig. 6 are the load vectors obtained by processing the data through PCA. Each load vector is a linear combination of the original data vectors. Here, the load vector that well represents the direction of data change is selected as cmponent1, cmponent2. On the other hand, the components in Fig. 6 (a) and (b) are different. The components in Fig. 6 (a) are obtained by performing PCA decomposition on the original data. then, we process the original data through the SDPTA method to get the new data. The components in Fig. 6(b) are obtained by performing PCA decomposition on the new data.

Remark 2: It should be noted that the PCA method is a way that data dimension reduction or data clustering. It does not have the ability of data classification. The normal/abnormal data can be separated because the data itself is separable. In other words, we just use the PCA method as a tool to show results. The improvement of data separability is the result of data be processed by SDPTA method.

Remark 3: It is worth noting that, as the SDPTA method is a window-based method, the monitoring performance will be affected by the window width, and detection delay may be introduced. However, our experience shows that delay is not always associated with the SDPTA method, and there is no simple correlation between the window width and monitoring performance. This may be due to the fact that with a wider window, the data in the sliding window will have smaller variance which lead to more stable PTCs and may improve the monitoring performance; At the same time, the contribution of the new measurement will be reduced, which may reduce the monitoring performance.

Hence, the window width is a trade-off between detectability and detection delay. Without a priori knowledge, it may be

TABLE 2. Monitored variables in the tennessee eastman process.

No	variable description	No	variable description	No	variable description
1	A feed (stream 1)	19	stripper steam flow	37	component D (stream 6)
2	D feed (stream 2)	20	compressor work	38	component E (stream 6)
3	E feed (stream 3)	21	reactor cooling water outlet temperature	39	component F (stream 6)
4	total feed (stream 4)	22	separator cooling water outlet temperature	40	component A (stream 9)
5	recycle flow (stream 8)	23	D feed flow valve (stream 2)	41	component B (stream 9)
6	reactor feed rate (stream 6)	24	E feed flow valve (stream 3)	42	component C (stream 9)
7	reactor pressure	25	A feed flow valve (stream 1)	43	component D (stream 9)
8	reactor level	26	total feed flow valve (stream 4)	44	component E (stream 9)
9	reactor temperature	27	compressor recycle valve	45	component F (stream 9)
10	purge rate (stream 9)	28	purge valve (stream 9)	46	component G (stream 9)
11	product separator temperature	29	separator pot liquid flow valve (stream 10)	47	component H (stream 9)
12	product separator level	30	stripper liquid product flow valve (stream 11)	48	component D (stream 11)
13	product separator pressure	31	stripper steam valve	49	component E (stream 11)
14	product separator underflow (stream 10)	32	reactor cooling water flow	50	component F (stream 11)
15	stripper level	33	condenser cooling water flow	51	component G (stream 11)
16	stripper pressure	34	component A (stream 6)	52	component H (stream 11)
17	stripper underflow (stream 11)	35	component B (stream 6)		
18	stripper temperature	36	component C (stream 6)		

TABLE 3. Process faults for the tennessee eastman process.

fault index	description	type
1	A/C feed ratio, B composition constant(stream 4)	step
2	B composition, A/C ratio constant (stream 4)	step
3	D feed temperature (stream 2)	step
4	reactor cooling water inlet temperature	step
5	condenser cooling water inlet temperature	step
6	A feed loss (stream 1)	step
7	C header pressure loss-reduced availability(steam 4)	step
8	A, B, C feed composition (stream 4)	random variation
9	D feed temperature (stream 2)	random variation
10	C feed temperature (stream 4)	random variation
11	reactor cooling water inlet temperature	random variation
12	condenser cooing water inlet temperature	random variation
13	reaction kinetics	slow drift
14	reactor cooling water valve	sticking
15	condenser cooling water valve	sticking
16-20	unknown	

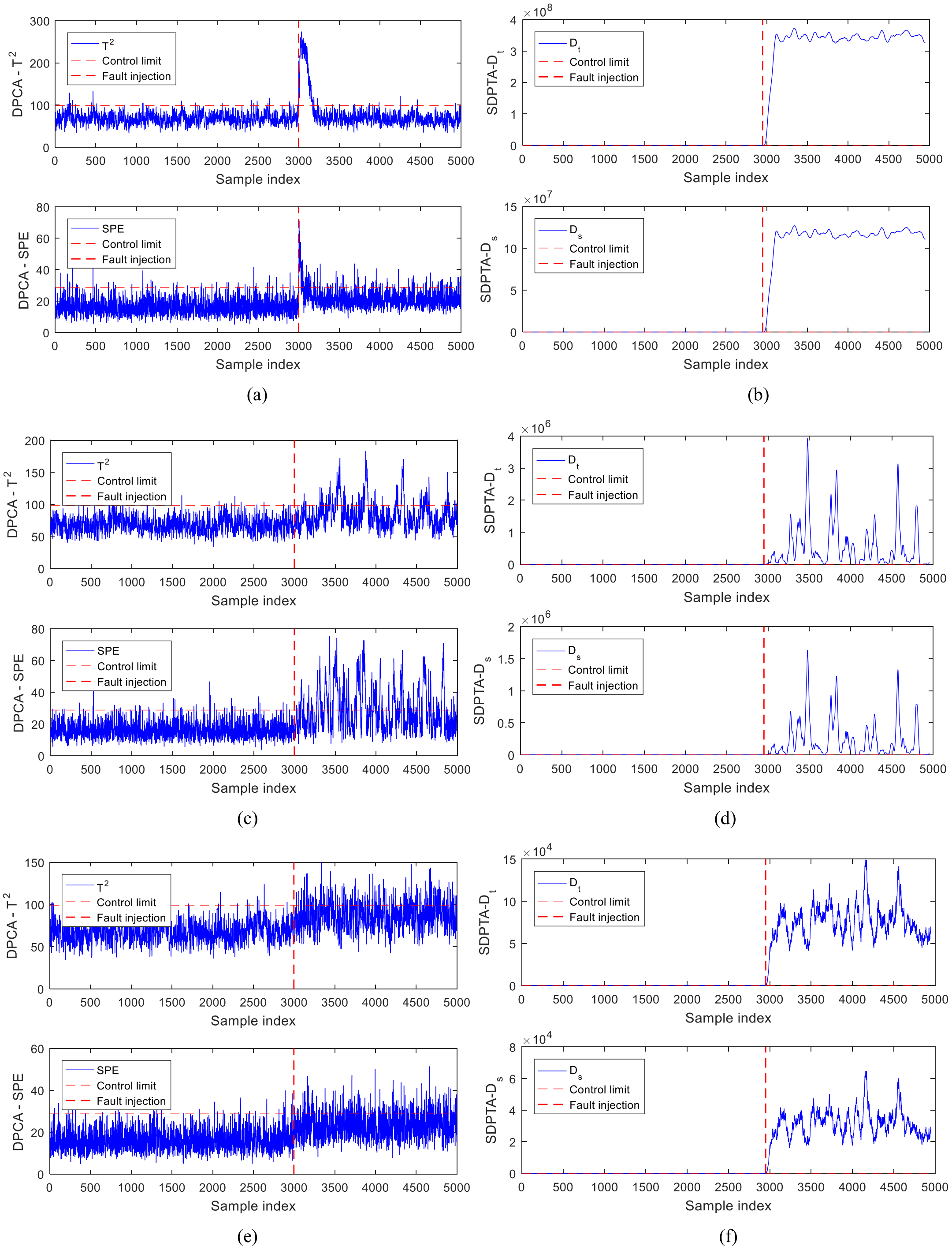
difficult to design a universal window width selection rule for systems with different characteristics.

B. CASE STUDY OF THE TE BENCHMARK PROCESS

Tennessee Eastman Process (TEP) created by the Eastman Chemical Company is designed to provide an actual industrial process for evaluating process control strategies [30]. This well-established benchmark has been widely used as data sources for evaluating various process monitoring approaches. We adopt the closed loop simulated process data developed by Braatz [31] to illustrate the detection performance of SDPTA for dynamic data. The process consists of five major unit operations: a reactor, a product condenser, a vapor-liquid separator, a recycle compressor, and a product stripper. Four reactants A, C, D, and E plus the inert B are fed to the reactor to generate products G and H, as well as byproduct F through two exothermic reactions. The diagram of the process is given in Fig. 7. The complete list of variables is given in Table 2. In this work, all 52 variables are used for monitoring.

Here, the simulation time is set as 250 h and the sampling time is set as 3 min. The process data include 11 manipulated variables, 22 continuous process measurements, and 19 composition measurements which are sampled less frequently. The information on the programmed faults are listed in Table 3. In this method, 5000 normal samples are used to build the model, another 5000 normal samples are used for validation, and each fault data set contains 5000 samples with the fault introduced after sample 2000.

In Table 4, it can be observed that these MSPM methods can effectively detect some obvious faults (such as fault 1, 2, 6, 14 and18) without excessive missed detection rates. On the



(a)

(b)

(c)

(d)

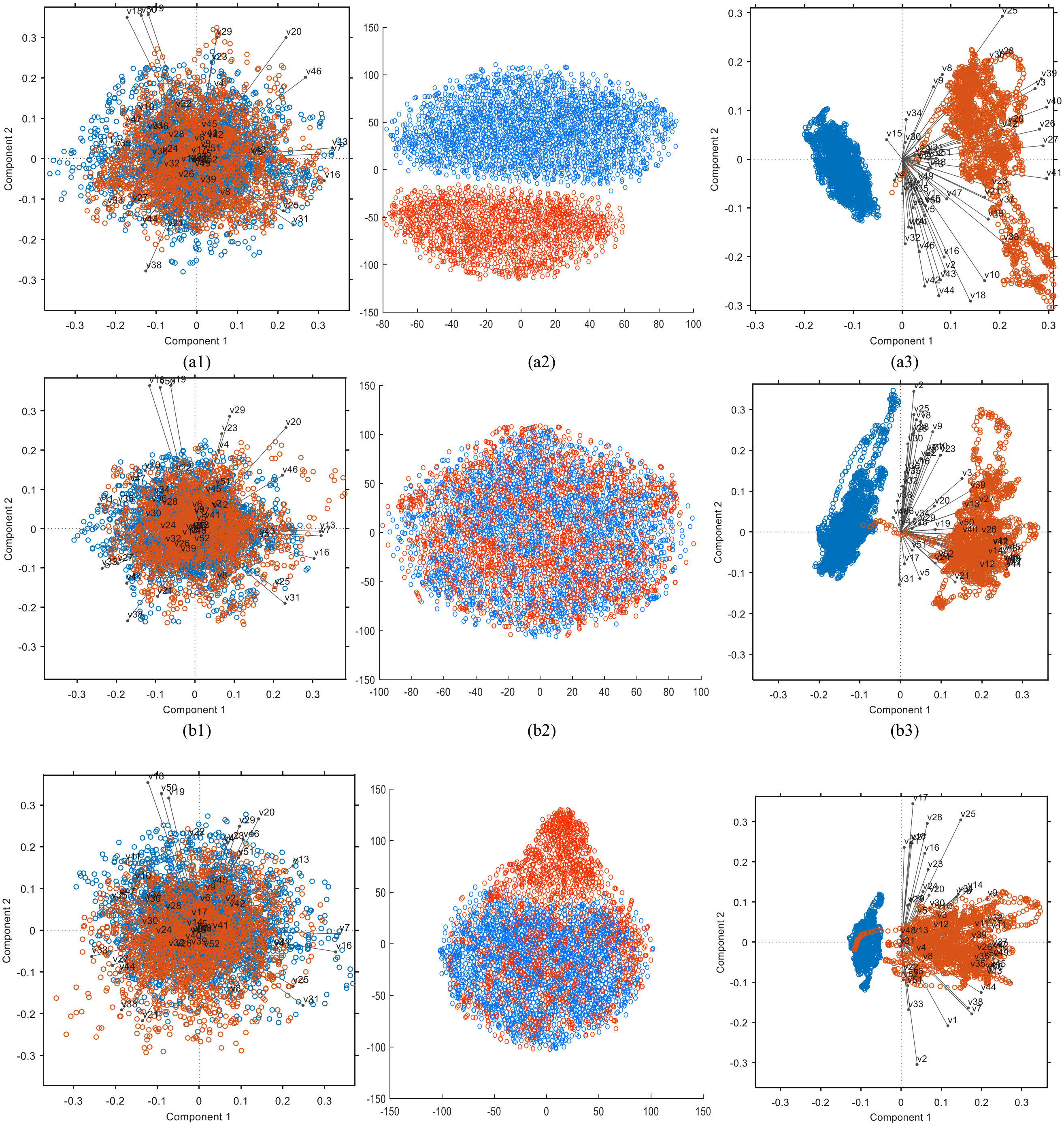
(e)

(f)

FIGURE 8. Detection performance of DPCA and SDPTA for fault 5 ((a) DPCA; (b) SDPTA), fault 10 ((c) DPCA; (d)SDPTA), and fault 19 ((e) DPCA; (f) SDPTA). The fault-injection lines indicate the fault onsets.

other hand, it is difficult to detect fault 3, 9 and 15 without a priori fault information [21], [23], [25], [32]. PCA and DPCA have difficulties in consistently detecting five faults (faults 5, 10, 16, 19 and 20), with detection rates less than 50% in most of the cases. On the other hand, the SDPTA method was able to detect all 17 faults consistently, with detection rates higher than 97%. At the same time, the SDPTA method has a smaller computational complexity while the detection rate is slightly better than the RTCSA method. The monitoring results for faults 5, 10, and 19 from DPCA and SDPTA are shown in Fig. 8.

In order to show the separability of data more intuitively, we use biplot, a classical data visualization algorithm, and t-SNE, a popular data visualization algorithm in recent



(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

(b) (d) (f)

FIGURE 9. Visual comparison of fault data separability, for fault 4 ((a1), (a2), (a3)), fault 19 ((b1), (b2), (b3)) and fault 20 ((c1), (c2), (c3)), respectively. (Blue dots represent normal data samples; Red dots represent fault data samples).

years [33], [34], to display the data. The specific form is shown in Figure 9. The biplot of original data are shown in Figure 9 (a1), (b1) and (c1), which represent fault (4), (19) and (20) respectively. At the same time, as a comparison, we introduce t-SNE method to visualize the original data (Parameters are default parameters.), which are show in Figure 9 (a2), (b2) and (c2). Biplot is used to visualize the data processed by SDPTA, and the results shown in

TABLE 4. Fault detection rates (percentage) of PCA, DPCA, TCSA and SDPTA for the TEP.

Fault index	PCA		DPCA		RTCSA	SDPTA	
	T^2	SPE	T^2	SPE	$D_{\propto}$	D_{t}	D_s
1	99.75	99.9	99.55	99.65	99.45	99.7	99.7
2	99.75	99.45	99.4	98.95	98.65	99.05	99.05
4	47.95	99.95	6.15	99.8	96.85	99.65	99.7
5	10.2	10.8	9.85	11.75	99.8	99.85	99.9
6	99.6	100	99.5	99.8	100	99.95	99.95
7	100	100	99.8	99.8	99.9	99.9	99.9
8	98.9	93.45	97.2	90.2	98.3	98.9	98.9
10	19.2	44.05	20.45	39.15	94	99	99
11	52.65	67.5	31	93.9	97.1	99.55	99.6
12	98.25	96.2	98.65	93.6	99.2	99.55	99.6
13	97.35	97.9	97.6	96.35	97.6	97.6	97.7
14	99.65	99.9	99.75	99.75	99.2	99.8	99.9
16	8.25	32.8	5.2	37	98.7	98	98.5
17	77.9	94.2	81.4	95.9	97.8	97.3	97.5
18	98.45	98.6	98.3	98.4	94.8	98.3	98.3
19	16.1	10.9	24.4	18.6	98.5	99.9	99.9
20	25.65	55.6	34.7	59.8	96.8	99.5	99.5

Figure 9 (a3), (b3) and (c3), which represent fault (4), (19) and (20) respectively. These results clearly show that the SDPTA method has a good detectability.

V. CONCLUSION

In this paper, a new data-driven process monitoring method called SDPTA is proposed for incipient fault detection. In the proposed approach, considering the sequence information of the data, the process measurement vector is converted into PTC in each sliding window. Instead of monitoring the process variable, SDPTA monitors the size of the process variable in the sliding window projected onto the basis vectors. Due to the consideration of the sequence information among measurements and basis vectors Without space partition, the SDPTA method can capture more process characteristics than traditional MSPM methods. In online monitoring, only multiplication of matrices is involved, which is low computational complexity and easy to use. In addition, the rationality of the SDPTA method is proved mathematically. To examine the performance of the proposed SDPTA method, it is applied to monitor a numerical example and the Tennessee Eastman process, and compared with the traditional PCA and DPCA methods. These examples demonstrate that the SDPTA method detects various faults more efficiently than the PCA and DPCA methods. In particular, it is able to handle incipient fault of the process, to detect changes in the system eigenstructure, and to detect subtle changes in various dynamic systems. Although the detection performance of SDPTA depends on the suitable selection of sliding window width, it generally outperforms other MSPM methods for the same or smaller window width in most cases. In the future work, we will continue to study fault detection methods of incipient faults, and further study the relationship between window width and fault detection rate.

APPENDIX 1

A The coverage of the window data X_k on the basis vectors $p_i \in P(i = 1, \cdots, m)$ can be represented by the eigenvalues of its covariance matrix, which given as follows:

$$R_{X_k} = \frac{1}{w-1} X_k^T X_k \tag{A1}$$

According to formula (10), $X^* \epsilon R^{n\times m}$ can be decomposed as

$$X^* = TP^T \tag{A2a}$$

$$T = X^* P \tag{A2b}$$

where P is a matrix composed of basic vectors, or named loading matrix; T is the score matrix; p_i is defined as the ith basis vector; $t_j \in T(j = 1, \cdots, m)$ is the jth score vector, which means the coverage of the data X^* on the jth basis vector.

Similarly, $X_k \in \mathrm{R}^{w\times m}$ can be decomposed as

$$X_k = T_k P_k^T \tag{A3a}$$

$$T_k = X_k P_k \tag{A3b}$$

where P_k represents the basis vectors obtained from the window data, and T_k represents the degree of coverage of the data in the window on the basis vectors. Note that X^* and X_k have the same data dimension, so the basis vectors P and P_k

are determined by formula (A2a) and (A3a), with the same dimensions. In other words, P and P_k are two different sets basis vectors in the same vector space.

Next, the coverage of X_k on P as T_k^* can be calculated as follows:

$$T_k^* = X_k P \tag{A4}$$

Since $t_j(j = 1, \cdot\cdot\cdot, m)$ are orthogonal, the length of t_j is equivalent to the eigenvalues of R_{X^*}, given as

$$\frac{1}{n-1} T^{\mathrm{T}} T = \Lambda \tag{A5}$$

where the diagonal matrix Λ denotes the eigenvalues of R_{X^*}. Similarly, for T_k and T_k^*, it can be obtained that

$$\frac{1}{w-1} T_k^{\mathrm{T}} T_k = \Lambda_k \tag{A6}$$

$$T_k^{*\mathrm{T}} T_k^* = P^{\mathrm{T}} X_k^{\mathrm{T}} X_k P = P^{\mathrm{T}} P_k T_k^{\mathrm{T}} T_k P_k^{\mathrm{T}} P \tag{A7}$$

From Formula (A6) and (A7), we have

$$\frac{1}{w-1} T_k^{*\mathrm{T}} T_k^* = P^{\mathrm{T}} P_k \Lambda_k P_k^{\mathrm{T}} P \tag{A8}$$

where the diagonal matrix Λ_k denotes the eigenvalues of R_{X_k}.

In formula (A8), P and P_k are two sets of standard orthogonal basis vectors. Using the linear space transformation theorem, we can get the following relationship:

$$P = P_k C \tag{A9}$$

$$P^{\mathrm{T}} P = P_k^{\mathrm{T}} P_k = I \tag{A10}$$

$$P^{\mathrm{T}} P = C^{\mathrm{T}} P_k^{\mathrm{T}} P_k C = C^{\mathrm{T}} C = I \tag{A11}$$

where C is the transition matrix of the basis P converted to the basis P_k. Specially, the transition matrix C is standard orthogonal.

Combining formula (A8) with (A9)-(A11), it can be derived that

$$\begin{aligned} T_k^{*\mathrm{T}} T_k^* &= (w-1) P^{\mathrm{T}} P_k \Lambda_k P_k^{\mathrm{T}} P \\ &= (w-1) C^{\mathrm{T}} P_k^{\mathrm{T}} P_k \Lambda_k P_k^{\mathrm{T}} P_k C \\ &= (w-1) C^{\mathrm{T}} \Lambda_k C \\ &= (w-1) \Lambda_{new} \end{aligned} \tag{A12}$$

Formula (A12) shows that the coverage of X_k in each basis vectors direction of p_i can be measured by the eigenvalue of R_{X_k}. Λ_{new} is the representation of Λ_k under the new basis. At the same time, we can get $t_j^* \in T_k^*(j = 1, \cdot\cdot\cdot, m)$ are orthogonal.

REFERENCES

[1] X. Deng, X. Tian, S. Chen, and C. J. Harris, "Nonlinear process fault diagnosis based on serial principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 560–572, Mar. 2018.

[2] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[3] C. Zhao and F. Gao, "Fault-relevant principal component analysis (FPCA) method for multivariate statistical modeling and process monitoring," *Chemometrics Intell. Lab. Syst.*, vol. 133, pp. 1–16, Apr. 2014.

[4] J. J. Hong, J. Zhang, and J. Morris, "Progressive multi-block modelling for enhanced fault isolation in batch processes," *J. Process Control*, vol. 24, no. 1, pp. 13–26, Jan. 2014.

[5] Q. Li, F. Pan, Z. Zhao, and J. Yu, "Process modeling and monitoring with incomplete data based on robust probabilistic partial least square method," *IEEE Access*, vol. 6, pp. 10160–10168, 2018.

[6] S. X. Ding, S. Yin, K. Peng, H. Hao, and B. Shen, "A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2239–2247, Nov. 2013.

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

[8] Y. Wang, Q. Jiang, and J. Fu, "Data-driven optimized distributed dynamic PCA for efficient monitoring of large-scale dynamic processes," *IEEE Access*, vol. 5, pp. 18325–18333, 2017.

[9] C. Tong, A. Palazoglu, and X. Yan, "Improved ICA for process monitoring based on ensemble learning and Bayesian inference," *Chemometrics Intell. Lab. Syst.*, vol. 135, pp. 141–149, Jul. 2014.

[10] 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.

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

[12] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 30, no. 1, pp. 179–196, 1995.

[13] S. J. Qin, W. Li, and H. H. Yue, "Recursive PCA for adaptive process monitoring," *IFAC Proc. Volumes*, vol. 32, no. 2, pp. 6686–6691, 1999.

[14] K. Watanabe, I. Matsuura, M. Abe, M. Kubota, and D. M. Himmelblau, "Incipient fault diagnosis of chemical processes via artificial neural networks," *AIChE J.*, vol. 35, no. 11, pp. 1803–1812, 1989.

[15] S. Zhang, C. Zhao, and F. Gao, "Incipient fault detection for multiphase batch processes with limited batches," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 103–117, Jan. 2019.

[16] X. Deng and J. Deng, "Incipient fault detection for chemical processes using two-dimensional weighted SLKPCA," *Ind. Eng. Chem. Res.*, vol. 58, no. 6, pp. 2280–2295, Jan. 2019.

[17] W. S. Ge, J. Wang, J. L. Zhou, H. Wu, and Q. B. Jin, "Incipient fault detection based on fault extraction and residual evaluation," *Ind. Eng. Chem. Res.*, vol. 54, no. 14, pp. 3664–3677, 2015.

[18] A. Youssef, J. Harmouche, C. Delpha, and D. Diallo, "Capability evaluation of incipient fault detection in noisy environment: A theoretical Kullback–Leibler divergence-based approach for diagnosis," in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2013, pp. 7364–7369.

[19] K. E. S. Pilario and Y. Cao, "Canonical variate dissimilarity analysis for process incipient fault detection," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5308–5315, Dec. 2018.

[20] S. J. Qin, "Statistical process monitoring: Basics and beyond," *J. Chemometrics, J. Chemometrics Soc.*, vol. 17, nos. 8–9, pp. 480–502, 2003.

[21] J. Wang and Q. P. He, "Multivariate statistical process monitoring based on statistics pattern analysis," *Ind. Eng. Chem. Res.*, vol. 49, no. 17, pp. 7858–7869, 2010.

[22] Q. P. He and J. Wang, "Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes," *AIChE J.*, vol. 57, no. 1, pp. 107–121, 2011.

[23] J. Shang, M. Chen, H. Ji, and D. Zhou, "Recursive transformed component statistical analysis for incipient fault detection," *Automatica*, vol. 80, pp. 313–327, Jun. 2017.

[24] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. De Carvalho, and J. Gama, "Data stream clustering: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, p. 13, 2013.

[25] D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth, "A native and adaptive approach for unified processing of linked streams and linked data," in *Proc. Int. Semantic Web Conf.* Berlin, Germany: Springer, 2011, pp. 370–388.

[26] A. Singhal and D. E. Seborg, "Pattern matching in multivariate time series databases using a moving-window approach," *Ind. Eng. Chem. Res.*, vol. 41, no. 16, pp. 3822–3838, 2002.

[27] E. L. Russell, L. H. Chiang, and R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 51, no. 1, pp. 81–93, 2000.

[28] R. F. Olanrewaju, B. U. I. Khan, A. R. Khan, M. Yaacob, and M. M. Alam, "Efficient cache replacement policy for minimising error rate in L2-STT-MRAM caches," *Int. J. Grid Utility Comput.*, vol. 9, no. 4, pp. 307–321, 2018.

[29] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring," *Automatica*, vol. 45, no. 7, pp. 1593–1600, Jul. 2009.

[30] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.

[31] L. H. Chiang, R. D. Braatz, and E. L. Russell, *Fault Detection and Diagnosis in Industrial Systems*. London, U.K.: Springer, 2001.

[32] J.-M. Lee, S. J. Qin, and I.-B. Lee, "Fault detection and diagnosis based on modified independent component analysis," *AIChE J.*, vol. 52, no. 10, pp. 3501–3514, Oct. 2006.

[33] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[34] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*. [Online]. Available: https://arxiv.org/abs/1802.03426

YINGHUA YANG received the B.S. degree from Northeastern University, Shenyang, China, in 1991 and the M.S. and Ph.D. degrees in control theory and control engineering from Northeastern University, Shenyang, China, in 1994 and 2002, respectively. He is an Associate Professor with the College of Information Science and Engineering, Northeastern University, Shenyang. His research interests include process monitoring, fault diagnosis and modeling method based on data driven, and its application on smart manufacturing.

YONGKANG PAN received the bachelor's degree in 2016 from Zhengzhou university of Aeronautics, Zhengzhou, China. He is currently pursuing the master degree with Northeastern University. His research interests include machine learning, process monitoring, and fault diagnosis.

LIPING ZHANG received the bachelor's degree from Nanjing Institute of Technology, Nanjing, China, in 2017. He is currently pursuing the master degree with Northeastern University. His research interests include machine learning, process monitoring, and fault diagnosis.

XIAOZHI LIU received the B.S. degree from Shenyang University of Technology, Shenyang, China, in 1990 and the M.S. and Ph.D. degrees in control theory and control engineering from Northeastern University, Shenyang, China, in 1995 and 2005, respectively. She is an Associate Professor with the College of Information Science and Engineering, Northeastern University in Shenyang. Her research interests include signal processing theory, process monitoring, and fault diagnosis.