Korean J. Chem. Eng., 34(2), 275-286 (2017)
DOI: 10.1007/s11814-016-0201-8

pISSN: 0256-1115
eISSN: 1975-7220

Fault detection based on polygon area statistics of transformation matrix identified from combined moving window data

Bei Wang, Xuefeng Yan†, and Yongfei Jin

Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai 200237, P. R. China
(*Received 15 January 2016 • accepted 8 July 2016*)

Abstract−Principal component analysis (PCA) has been widely used in monitoring industrial processes, but it is still necessary to make improvements in having a timely and effective access to variation information. It is known that the transformation matrix generated from real-time PCA model indicates inner relations between original variables and new produced components, so this matrix would be different when modeling data deviate due to the change of the operating condition. Based on this theory, this paper proposes a novel real-time monitoring approach which utilizes polygon area method to measure the variation degree of the transformation matrices and then constructs a statistic for monitoring purpose. The on-line data are collected through a combined moving window (CMW), containing both normal and monitored data. To evaluate the performance of the proposed method, a simple numerical simulation, the CSTR process and the classic Tennessee Eastman process are employed for illustration, with some PCA-based methods used for comparison.

Keywords: Principal Component Analysis, Combined Moving Window, Polygon Area, Real-time Process Monitoring, Fault Detection

INTRODUCTION

Thanks to the progress of data processing techniques, massive amounts of data produced from industrial systems can be gathered and stored, such that the state of operating processes can be observed by capturing significant information from these data. Multivariate statistical process monitoring (MSPM) is an efficient method for extracting characteristic information from original data [1-5].

Among various MSPM approaches, principal component analysis (PCA) is regarded as the most fundamental technology and is extensively used in industrial process monitoring [6,7]. This method estimates the main inner correlations of variables to reduce dimensionality and explores latent data factors to provide a precise description of the process [8]. To deal with various situations, PCA is continuously modified to satisfy various requirements. For a nonlinear process, Schölkopf introduced the kernel function principle to PCA and developed kernel PCA [9]. Based on this theory, follow-up improvements have been continuously proposed to achieve improved performance [10-12]. To handle data following the non-Gaussian distribution, PCA is extended to independent component analysis method [13,14]. Ge and Song then merged these two methods to generate a global monitoring model in order to obtain both Gaussian and non-Gaussian information simultaneously [15, 16]. Additionally, to monitor the large-scale industrial process with complex system structure and complicated variable relations, multi-block strategy is employed to construct multi-block PCA, which simplifies the monitoring processes through dividing the process, original variable spaces, or extracted component spaces [17-19]. When addressing a batch process, multi-way PCA is developed to extract information in the multivariate trajectory data and then to track the progress of new batch runs [20,21]. Many other PCA-based approaches have also been proposed to solve different problems [22-25]. However, some issues still exist: (1) online process information cannot be captured timely when only one PCA model is built; (2) the comparison between normal and abnormal data is ignored; and (3) effective information extraction from PCA model remains an open issue.

Real-time process monitoring is effective in fault detection because it can help timely fault determination and normalize the operating process. To monitor a dynamic process, time series models, such as the AP model [26,27] and the IMA model [28], are introduced. Negiz and Cinar established a state space model with data from normal conditions and then monitored the prediction of state vector [29]. Simoglou et al. captured the system dynamics and correlation structure of the process data to create a state space representation [30]. Ku et al. used the "time lag shift" method to consider dynamic behavior in the PCA model; thus, dynamic PCA (DPCA) is produced [31]. Evidently, constructing the PCA model based on time series can help obtain timely and accurate information from dynamic processes.

Among various works on monitoring multivariate dynamic processes, moving window (MW) is an available and frequently used approach to collect online data from operating processes [31-33]. MW updates the PCA model with the newest samples in the window that slides along the data. This strategy can be considered as a recursive technique that constantly constructs previous models

†To whom correspondence should be addressed.
E-mail: xfyan@ecust.edu.cn, qchjiang@mail.ecust.edu.cn

rather than constructing only one model at the beginning of the process [22,34]; thus, new data characteristics are continuously added to the model and the changing information can be explored more timely. However, the data trend would become flat but abnormal when the operating process is compensated by the loop system with time going. Therefore, the way of data collection should be modified to overcome this weakness.

In PCA model, the loadings and the corresponding eigenvalues are generated by decomposing the data with singular value decomposition (SVD). All the loading vectors are orthogonal to one another and represent new directions of the principal component (PC) space. The loading matrix transfers the original space to a new one with maximum variance, and the eigenvalues respectively corresponding to the loading vectors scale the generated components to variance of 1 in new space [9]. To reduce dimensionality, many methods have been proposed to select PCs, so that the main data features can be conserved and redundant information are removed. The most widely used strategy is cumulative percent variance (CPV), which selects the first several PCs according to percentage of the cumulative variance [22,35]. Other methods, such as the cross-validation technique [36] and variance of reconstruction error [37], are also applied in some cases. Jiang and Yan suggested to collect sensitive PCs on the basis of the change rate of statistics, so that to highly useful information can be obtained (PCA-SVDD). However, the PCs are only selected at the beginning of the process [38] and do not change again with the operation of the processes. Jiang and Yan modified this method by introducing the just-in-time (JIT) strategy and developed JIT-PCA-SVDD [39] to choose sensitive PCs based on kernel density estimation (KDE) [40] at every point. Obviously, online information extraction improves the monitoring performance, but the calculation of KDE for all PCs at every point is difficult and the number of selected PCs is uncertain. Therefore, a concise and serviceable way to extract online data characteristics is necessary for real-time data, particularly for plant-wide processes.

It is evident that the loadings and eigenvalues satisfy lower dimension and simpler construction, compared with the original data. Moreover, they indicate the inner connections between variables and new components. When the modeling data changed from the normal to the faulty ones, the two matrices would inevitably change in some degree. Based on this theory, Kano et al. [41] suggested the DISSM method, which measures dissimilar distribution between normal and present datasets in MW based on the Karhunen-Loeve expansion. The variation information in real-time loadings and eigenvalues can reveal the condition of the dynamic process. However, few related researches focus on this topic later on.

We propose a novel process monitoring method that employs combined moving window (CMW) strategy to collect online data for online modeling and then extracts variation information of the online data from the generated transformation matrix (combined with a loading matrix and a diagonal matrix structured by eigenvalues). This approach is termed as TPCA-CMW. In operating processes, fault data will reach a new balance level when the industrial system has a compensation from control loop; thus the CMW containing both normal data and online data is utilized for model construction, in order to highlight the difference of the faulty data. Then the variation information can be revealed through observing the changing of the real-time generated transformation matrix. To measure the extent of variation, a mathematical method (polygon area method) [42,43] is employed to estimate the distribution change of the key elements in transformation matrix, and a novel statistic is constructed for process monitoring.

The remainder of this article is organized as follows: Section 2 briefly introduces the basic PCA theory and uses a motivational example to illustrate the inspiration of the proposed method. After presenting the CMW strategy, Section 3 presents the detailed description and procedures of the TPCA-CMW method. Section 4 analyzes the effectiveness and veracity of this method through the classic Tennessee Eastman process, and some previous methods are listed for comparison, as well as to demonstrate the superiority of TPCA-CMW. Finally, Section 5 discusses the conclusions.

PRELIMINARIES

This section briefly reviews the basic PCA model theory, and the motivation of the proposed method is illustrated through a simple numerical example.

1. Principal Component Analysis

PCA aims to decompose high-dimensional data and generate a low-dimensional representation. After scaling the data that follows the Gaussian distribution, the loading matrix $\mathbf{P}=[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M]\in R^{M\times M}$ can be obtained from the scaled data $\mathbf{X}\in R^{N\times M}$ through SVD as follows:

$$\mathbf{S}=\mathbf{X}^T\mathbf{X}/(N-1)=\mathbf{P}\Lambda\mathbf{P}^T, \tag{1}$$

where the eigenvalues ($\lambda_1, \lambda_2, \ldots, \lambda_M$) are in the diagonal matrix Λ and are arranged in descending order; and N, M are the number of samples and variables, respectively. To optimally capture the data variation and minimize the effect of random noise, the loading vectors that correspond to the first a eigenvalues with maximum values are chosen to describe the data characteristics, as expressed by

$$\mathbf{X}=\mathbf{T}\hat{\mathbf{P}}^T+\mathbf{E}, \tag{2}$$

where $\mathbf{T}\in R^{N\times a}$ is the score matrix, $\hat{\mathbf{P}}\in R^{M\times a}$ is the selected loading matrix, and $\mathbf{E}\in R^{N\times M}$ is the residual matrix. Here, the parameter a, i.e., the number of PCs, is determined by the CPV principle, as introduced in the following:

$$\sum_{i=1}^{a}\lambda_i\Big/\sum_{i=1}^{M}\lambda_i\times 100\%\geq 85\%, \tag{3}$$

When the accumulation of percent variances is equal to or greater than 85%, the corresponding components are selected as PCs. To observe the sample $\mathbf{X}_r\in R^{M\times 1}$, the statistic T^2 is constructed for monitoring purposes, as presented in the following:

$$T^2=\mathbf{X}_r^T\hat{\mathbf{P}}\Lambda_a^{-1}\hat{\mathbf{P}}^T\mathbf{X}_r, \tag{4}$$

The statistic monitors the data variation and sequentially reflects the operating process condition. Since the following study involve around the constructed space, therefore the part of SPE statistic is not discussed here.

2. Problem Statement and Motivational Example

Given that the data collected from the system represent the monitoring process condition, the PCA model constructs a new space and projects data to extract data feature for monitoring purposes. In the PCA model, the loading matrix and the eigenvalues are two significance matrices that transform the original space to a new one. The loadings rotate the principal axes to a new space where new components have maximum variance, and the corresponding eigenvalues scale components to the standard variance of 1 [8]. Extending the PCA model to dynamic processes, w−1 observation samples before current time k are provided to supply information as follows:

$$\mathbf{X}(k)=[\mathbf{x}(k)\ \ \mathbf{x}(k-1)\ \ldots\ \mathbf{x}(k-w+1)]^T, \tag{5}$$

where $\mathbf{x}(k)=[x_{1,k}\ \ x_{2,k}\ \ \ldots\ \ x_{M,k}]$ is the M-dimensional observation vector. Obviously, the data collected in this way update information of the operating process in a timely manner, and the variation of the monitored process can be revealed accurately. When constructing the PCA model with these online data, the generated loadings and eigenvalues would inevitably differ from those generated from the normal PCA model. To explain this theory, a simple numerical example is listed for illustration as follows:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 1.31 & 0.95 \\ 0.87 & 1.15 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix}+\begin{bmatrix} e \\ e \end{bmatrix}, \tag{6}$$

where r_1 and r_2 satisfy the Gaussian distribution within [0, 2], and e are the Gaussian distributed noise of N(0, 0.1). This simple numerical process produces two variables for the following study. A total of 200 samples under normal conditions are collected, and then the mean and variance are generated for normalization. Three most common fault types are created as follows:

Fault 1: A step change of 4 is added to x_1 at the 101st sample.

Fault 2: A step change of 3 is introduced to x_2 from sample 101 to the end.

Fault 3: A ramp change of x_1 by adding 0.5×(i−100) from sample 101.

Fault 4: An overturn change of x_1 changing to $-x_1$ from sample 101 to the end.

Each fault case generates 200 samples, as presented in Figs. 1(a)-1(d), where both normal and faulty data are displayed for comparison. The distribution of the faulty data is very different from that of original data. If using faulty data to construct PCA model, which produces new directions and scales the new components to variance of 1, the generated transformation matrix must be totally different. Here, the transformation matrix is defined as follows:

$$\mathrm{tran}=\mathbf{p}_i/\sqrt{\lambda_i}. \tag{7}$$

The above four faulty datasets and one normal dataset are used to

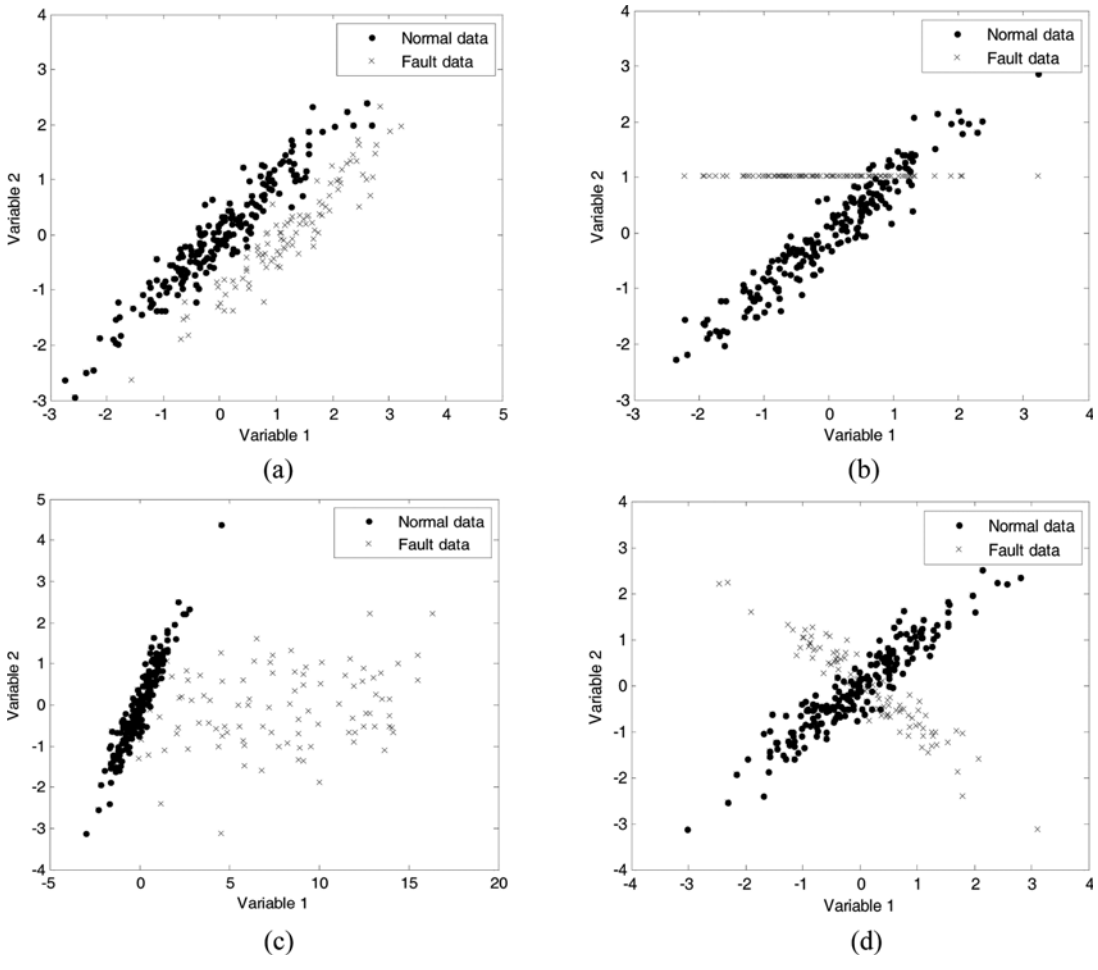


Fig. 1. Distribution map of normal data and fault data from (a) case 1; (b) case 2; (c) case 3; (d) case 4.

Table 1. *tran* values calculated with the normal data and the three fault data

	Normal data		Fault case 1		Fault case 2		Fault case 3		Fault case 4	
Column	1	2	1	2	1	2	1	2	1	2
Transformation	0.506	−3.280	0.718	−0.913	0.526	−1.365	0.198	−0.043	0.667	0.756
Matrix	0.506	3.280	0.505	1.297	0.433	1.659	0.008	1.021	0.673	−0.749

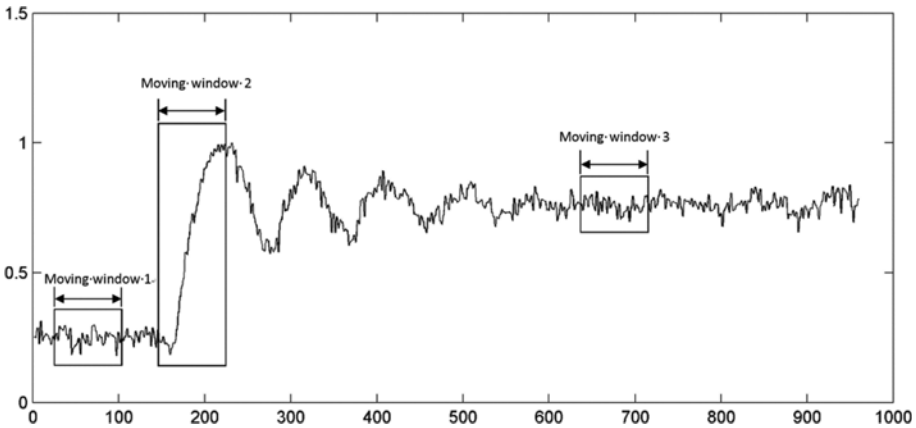


Fig. 2. Conditions of collected data in moving window at different time.

make models and then the transformation matrices are generated for observation, listed in Table 1. As can be seen, the values of two columns, especially the second column, have changed obviously, because PCA always constructs new space and scales the new components to variance of 1. When the area of the data distribution rises, the elements in the transformation matrix have to be smaller to scale the data used to make model. If this kind of variation can be described in some way, the operating process can be monitored online through measuring the variation of transformation matrix.

3. Monitoring Based on the Variation of the Transformation Matrix

In this section, a novel data collection method that highlights change information is primarily introduced, and then details on the proposed TPCA-CMW method are presented, with some characteristics discussed.

3-1. Combined Moving Window

The MW strategy is applied to collect online data from the monitoring process. Usually, this strategy simply employs several successive samples before current time to supply information for the present sample. However, the variables would achieve a new level and maintain steady condition with the compensation of the control loop. As the example shown in Fig. 2, initially, the variable is going steady smoothly, as well as collected data in MW1. Later, when a certain fault happens, there is a sharp increase in the figure and the corresponding data in MW2 reveal this change. However, as time passes, the variable arrives at a new balance and the collected data in MW3 tend to stabilization again, although a fault still exists in the monitoring process.

To address this problem, the CMW strategy is developed here to distinguish fault data from normal data, shown in Fig. 3. The CMW first selects w_1 samples from normal data as benchmark data, and then the rest of the parts are filled with online data gathered from the monitoring process as Eq. (5). w_1 normal data and w_2 monitored data are selected for CMW with a window size of $w=w_1+w_2$. Therefore, fault data can be recognized by comparing with normal data, although the variable returns to a steady state. Real-time information can be obtained and fault information is distinguished in CMW.

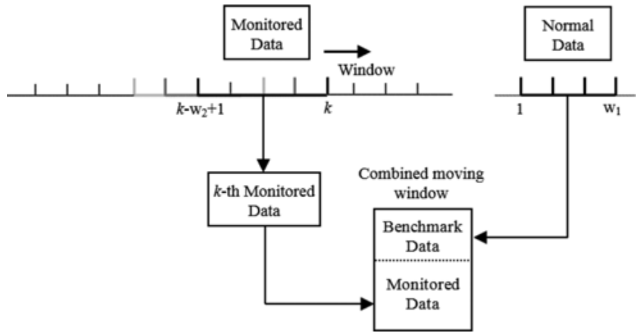


Fig. 3. Illustration of the CMW strategy for PCA model construction.

3-2. Polygon Area Statistics of Transformation Matrix

As analyzed above, the transformation matrix *tran* would be distinctly different when modeling with fault data. In PCA model, the eigenvalues are arranged at a decreasing order, and there is a one-to-one relationship between each column of the loading matrix and the eigenvalues. The loadings corresponding to bigger eigenvalues obviously have richer variation when a certain fault occurs. Therefore, after studying and testing, the first two columns of transformation matrix are selected in this paper for extracting the variation and constructing monitoring statistic.

The reasons for choosing two columns of transformation matrix are that the first columns vary more significantly when model data

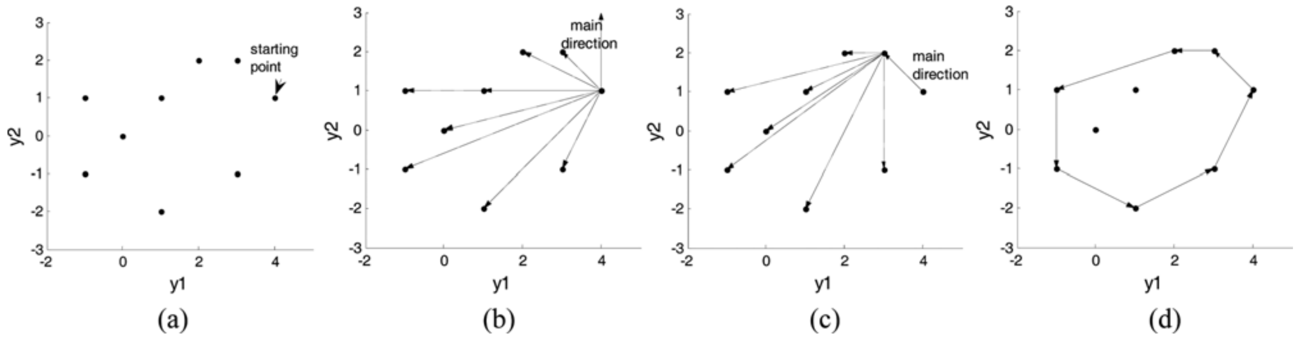


Fig. 4. Illustration of the generation of polygon (a) selecting the starting point; (b) calculating their included angles; (c) selecting the second point; (d) generating the polygon.

change, and a 2-dimensional space is easy to observe. As described in section 2.2, the data would deviate from original distribution area after the system is abnormal, so the variance of the data would increase. The elements in transformation matrix have to be smaller to rotate the variables and scale them to variance of 1 in new space. Thus, if projecting these elements a on the plane, they would concentrate after fault occurrence. To measure the variation quantitatively, a method that estimates the data distribution state is developed to observe data separation and aggregation. Assume that two datasets $\mathbf{y}1=(2, 1, 0, 1, -1, -1, 3, 3, 4)^T$ and $\mathbf{y}2=(2, 1, 0, -2, -1, 1, -1, 2, 1)^T$ exist with randomly selected values. A two-dimensional space is constructed, and these points are mapped, as shown by the data distribution in Fig. 4(a). The developed method aims to calculate the minimum area that covers all the points. The following strategy is introduced first to generate a polygon using a line to surround the peripheral points.

Step 1: The point on the far right is chosen as the starting point, and the axis perpendicular to y1 is defined as the main direction.

Step 2: The vectors that start at the chosen point and end in the other points are set, as shown in Fig. 4(b), and the included angles of these vectors with the main direction are calculated.

Step 3: The included angles are compared, and the point that has the minimum angle with the main direction is selected as the next starting point.

Step 4: The main direction is replaced with a new generated direction, as shown in Fig. 4(c), and then Step 2 is repeated until the vector pointing at the original starting point is achieved. If two points exist in the same direction, the far point is chosen as the next boundary point.

Based on the above iteration, the generated polygon covers all the points, as shown in Fig. 4(d), and the vertices are saved as $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_r, \beta_r)$ in this order, where r is the number of boundary points. Here, the problem has been converted into calculating the polygon area through the following formula [42,43]:

$$\mathrm{S}=\frac{1}{2}\left(\begin{vmatrix}\alpha_1 & \alpha_2\\ \beta_1 & \beta_2\end{vmatrix}+\begin{vmatrix}\alpha_2 & \alpha_3\\ \beta_2 & \beta_3\end{vmatrix}+\ldots+\begin{vmatrix}\alpha_r & \alpha_1\\ \beta_r & \beta_1\end{vmatrix}\right). \tag{8}$$

The area is the sum of the area of the triangle $[\mathbf{0}\,\mathrm{A}_t\,\mathrm{A}_{t+1}]$, where $\mathrm{A}_t=(\alpha_t, \beta_t)$ is the vertex. The area is regarded as a statistic to reflect a change in the monitoring process. When the process is under normal condition, the constructed statistic is steady. Once a fault happens in the operating system, causing data to deviate from original area, the transformation matrix will be totally different and the constructed statistics will decrease. Thus, the trend of statistic S indicates the monitoring process condition. In addition, KDE is used to calculate the corresponding confidence limit for judging the operating process condition.

Remark When online data participate in modeling, their variation is easy to be reflected by transformation matrix, which contains inner relations and has low-dimensional construction. Here, the monitoring statistic is constructed based on the change information of transformation matrix, and the polygon area method is the approach to measure this kind of variation. Once some fault occurs in the process, the transformation matrix from the model constructed with fault data would change more obviously, especially the first several columns. Choosing the first two columns is because their variation is more significant and projecting onto plane is easy to capture their changing characteristic. The following monitoring results in this article also can determine the well function of the proposed method. However, this way still causes information loss more or less;; therefore, the study about n-dimensional hyper-volume to capture variation from transformation matrix is undergoing.

4. Implementation

The detailed steps of the method (TPCA-CMW) are given below.

Offline modeling

Step 1: Specify the width w of CMW $\mathbf{X}_c=[\mathbf{X}_a; \mathbf{X}_b]$, where $\mathbf{X}_a \in \mathrm{R}^{w_1 \times M}$ is the benchmark data and $\mathbf{X}_b \in \mathrm{R}^{w_2 \times M}$ is the monitored data.

Step 2: Choose normal data as the benchmark data $\mathbf{X}_a$ and select monitored data $\mathbf{X}_b$ from normal monitoring process.

Step 3: Construct PCA model with the data in CMW, and generate online transformation matrix.

Step 4: Project the first two columns of transformation matrix onto the plane, and generate the minimum polygon to cover the elements.

Step 5: Calculate the statistic S according to Eq. (8).

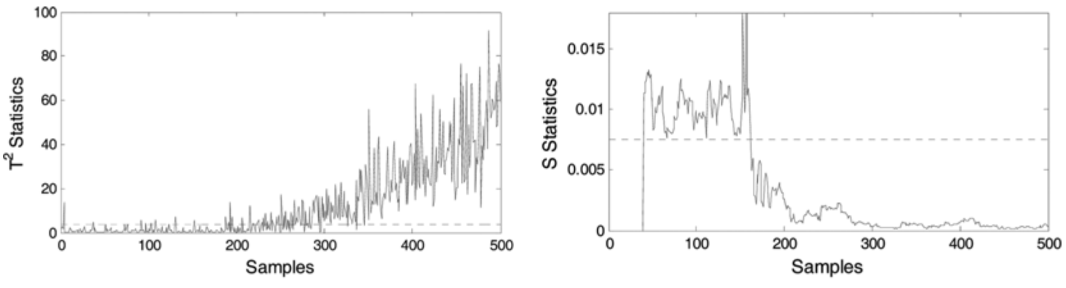
Step 6: Select monitored data $\mathbf{X}_b$ for next moment and turn to Step 3 until n statistics are determined.

Step 7: The confidence limit with n statistics is estimated using KDE.

Online monitoring

Step 1: $\mathbf{X}_b$ is collected at current time k from the operating process, while $\mathbf{X}_a$ remain unchanged.

Step 2: Build the real-time PCA model with the data in CMW,



(a) (b)

Fig. 5. Process monitoring results of case 1. (a) PCA; (b) TPCA-CMW.

and generate the transformation matrix *tran*.

Step 3: Define the area that covers the elements, and calculate the monitoring statistic S according to Eq. (8).

Step 4: The condition of the operation process is assessed. If S still goes below the confidence limit, the monitoring process maintains some fault. Otherwise, the process is deemed as normal, and Step 1 is repeated for further monitoring.

5. Applications

In this section, the proposed TPAC-CMW method is applied in a numerical system and the TE benchmark process. A comparison with conventional PCA and other PCA-based methods is presented.

5-1. A Numerical Example

To illustrate the efficiency of the proposed method, a simple five-variable system is constructed as follows:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0.568 & 0.358 \\ 0.845 & 0.476 \\ 0.365 & 0.458 \\ 0.756 & 0.647 \\ 0.498 & 0.548 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix} \quad (9)$$

where $[s_1s_2]^T$ satisfy Gaussian distribution with zero mean and standard deviation of 0.9; the noises $[e_1e_2e_3e_4e_5]^T$ are zero-mean white noises with standard deviation 0.02. To simulate the large-scale process, three operating conditions are constructed for study. First 200 data samples under normal condition are collected directly from Eq. (9). Then the conventional PCA is constructed using the normal data with 99% confidence limit set for fair judgment. After modeling, the first two PCs, occupying 88.36% cumulative variance, are deemed to be dominant components for process monitoring based on CPV rule. Then two faults are programmed into the system, respectively, to produce testing dataset as follows:

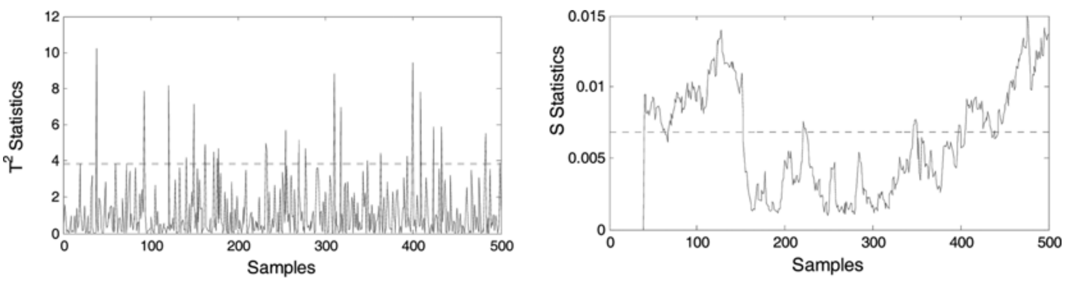
Case 1: A ramp change of $0.1\times(i-150)$ is added to x_1 from sample 151 to the end.

Case 2: A step change 3 is introduced to x_2 from sample 151 to 350.

Each case generates 500 samples for testing the function of methods. The proposed TPCA-CMW is constructed with 10 normal data samples and 30 online data samples in CMW. The monitoring results for case 1 of both PCA and TPCA-CMW are shown in Fig. 5(a) and (b). As can be seen, the statistics in PCA method cannot exceed the confident limit until around sample 300, but TPCA-CMW can detect the fault effectively and timely. Herein, as described above, the trend of S statistic for fault detection is different from that of T^2 statistic. When the condition is normal, the S statistics stay above the confidence limit, and when a certain fault happens in the process, the S statistics would be below it. For case 2, Fig. 6 displays the monitoring performance of these two methods for case 2. Due to the shortcoming of PC selection, this fault cannot be detected by conventional PCA in Fig. 6(a). However, if constructing an online model, the generated transformation matrix can reflect this variation timely and accurately, shown in Fig. 6(b), where the statistics go down the confidence limit after sample 151 and return to initial level after 350. Here, there is a delay due to the employment of a moving window.

5-2. Application to CSTR Process

The continuous-stirred-tank-reactor (CSTR) process used to simulate dynamic process is employed here for the purpose of function testing. A detailed diagram of the process is shown in Fig. 7.



(a) (b)

Fig. 6. Process monitoring results of case 2. (a) PCA; (b) TPCA-CMW.

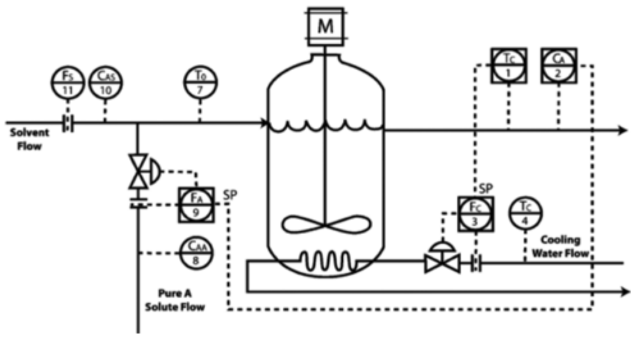


Fig. 7. The diagram of the CSTR process.

There are nine variables in this system: the inlet templerature T_0, cooling water temperature T_c, inlet concentration C_{AA} and C_{AS}, cooling water flow F_C, solvent flow F_S, outlet concentration C_A and temperature Ts, and reactant flow F_A. The nine variables are sampled every minute, and 200 samples are collected under nomal conditions considered as a training dataset. More details and the parameters about the CSTR process can be found in the references [39, 44-46].

For the purpose of comparison, the tradtional PCA model with PCs occupying >85% CPV colected is constructed here. In addition, the size of moving window is set as 40, with 15 normal samples from histrical dataset and 25 samples colected online. To test the function of the methods, three different faults are introduced into the system from the 501st sample:

Fault 1: a bias of 1.5 (K) is added to the sensors of the inlet temperature T_0.

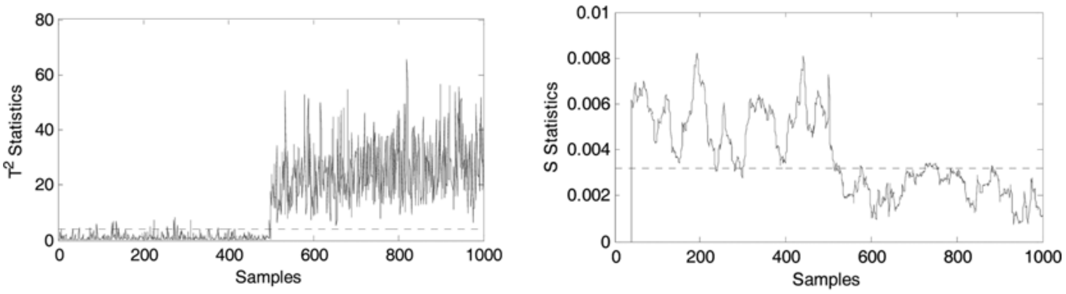
Fault 2: a drift (dC_{AA}/dt=0.02 (kmol/(m^{3}min))) is introduced into the sensor of C_{AA}.

Fault 3: a form of an exponential degradation of the reaction rate caused by catalyst poisoning happens in the reaction kinetics, with the reaction rate coefficient changing with time as k0(t+1)=0.996 * k0(t) [47].

In the monitoring of fault 1, the deviation of the variable T_0 can be detected after sample 501 by the listed two methods PCA and TPCA-CMW, shown in Fig. 8(a)-(b). When monitoring the Fault 2, both PCA and TPCA-CMW show good performance in detection, with monitoring results displayed in Fig. 9(a)-(b). As can be seen, this fault can be detected in a timely way after its occurrence and they keep their corresponding monitoring statistics being far away from the confidence limit. Fault 3 is a complex fault which causes the change of servel variables, such as the output temperature T, concentration C_A, and the cooling water flow F_C. Detecting such a complex fault, the PCA do not perform well, with slight change in monitoring statistics, shown in Fig. 10(a). On the contrary, the proposed TPCA-CMW shows better detection ability in this fault, given in Fig. 10(b). The monitoring statistics go below the confidence limit to indicate the occurence of the fault, and the number of nondetections is close to zero. Thus, the proposed method TPCA-CMW shows better ability of fault detection in dynamic process based on the comparison of the monitoring results.

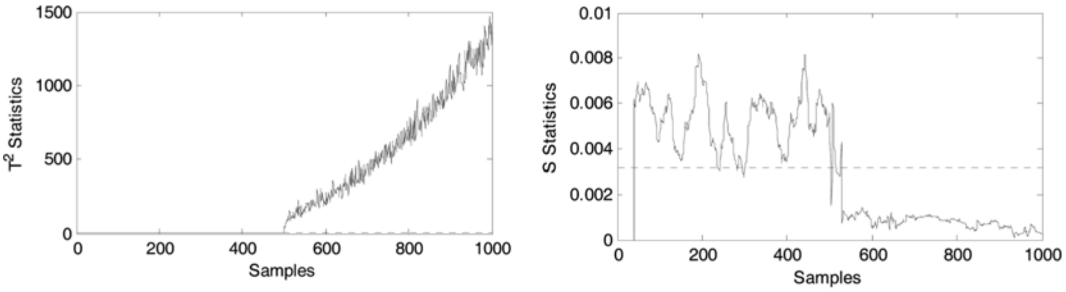
5-3. Application to TE Process

To test the function of various monitoring methods, Downs and Vogel [48] established a classic process, the Tennessee Eastman process, to simulate the actual industrial process, shown in Fig. 11. Five



(a) (b)

Fig. 8. Process monitoring results of Fault 1 in CSTR process. (a) PCA; (b) TPCA-CMW.



(a) (b)

Fig. 9. Process monitoring results of fault 2 in CSTR process. (a) PCA; (b) TPCA-CMW.

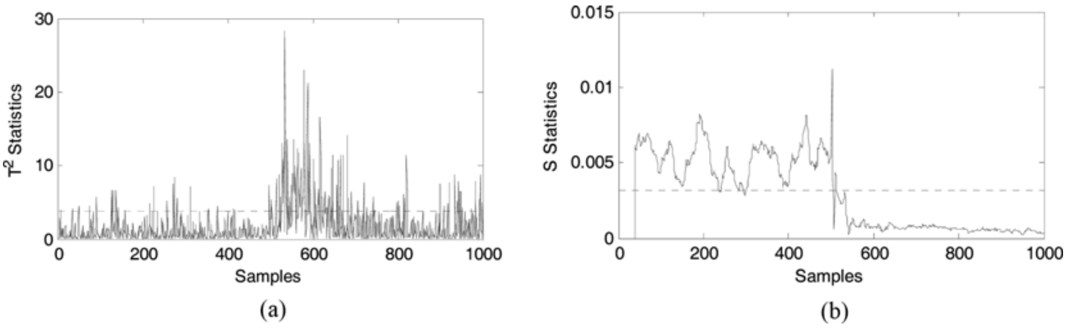


Fig. 10. Process monitoring results of Fault 3 in CSTR process. (a) PCA; (b) TPCA-CMW.

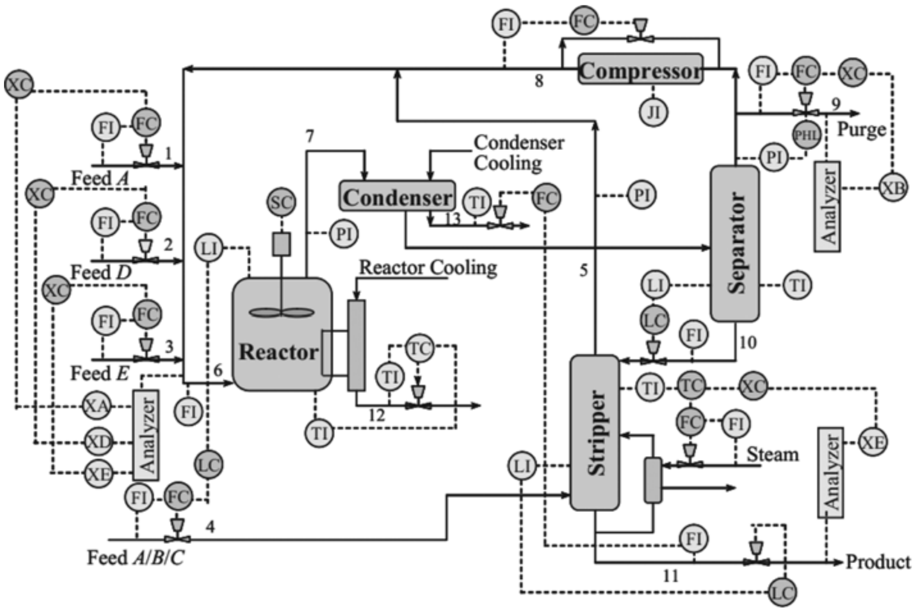


Fig. 11. Control system of the tennessee eastman process.

major unit operations exist in this simulator: a reactor, a product condenser, a vapor-liquid separator, a recycle compressor, and a product stripper. Forty-one measured variables and 12 manipulated variables exist in this system, but only 33 main variables are usually maintained to generate data for discussion, as shown in [1]. A normal dataset is composed of 500 observation samples collected from the TE process as training data. For testing purposes, 21 different preprogrammed faults [49] are introduced into the simulation process from sample 161, and 960 samples are collected from each fault case for analysis. In addition, fault 0, which represents the normal operating state of the TE process, is generated as the testing dataset. More details can be found in Refs. [8,48] with the simulation code downloaded from http://web.mit.edu/braatzgroup/links.html.

In TPCA-CMW, online samples are collected from the TE process and a real-time TPCA-CMW model is created to estimate the novel statistic. The confidence limit is set at 95%, which means 95% statistics remains above the threshold under normal conditions. The first step in TPCA-CMW is identifying the CMW size. Here 10 normal samples from historical data and 30 monitored samples from the operating process combine the CMW. In the TE simulation process, 21 faults are arranged, where faults 1, 2, 6, 7, 8, 12, 13, 14, and 18 can be detected easily by most monitoring methods. By contrast, faults 3, 9, and 15 slightly affect the process; thus, these faults are not listed in this paper for discussion. The other faults are selected to analyze the influence of CMW on the proposed TPCA-CMW method. The missed detection rate is defined to judge the performance of the monitoring method, as expressed by

$$\theta = n_{fault}/n_{total}, \tag{10}$$

where n_{total} represents the total number of samples after a fault occurs, and n_{fault} represents the number of fault samples that go over the confidence limit.

Other PCA-based methods are employed for comparison to prove the superior performance of the proposed TPCA-CMW method. First is the traditional PCA, in which 15 PCs occupying 86.49% cumulative variance are selected on the basis of the CPV principle [8,50]. Next, the extension of PCA to dynamic process, that is, DPCA, is discussed to highlight the satisfactory performance of TPCA-CMW in terms of dealing with dynamic process. Ge and Song proposed a linear subspace and the Bayesian inference-based PCA method [18]. Tong et al. suggested a four-subspace construc-

Table 2. Missed detection rates of the different monitoring methods in the TE process

Fault number	PCA T^2	DPCA T^2	BSPCA BIC_T^2	FSCB $BIC_{(D)}$	KL-MBPCA BIC_T^2	PCA-SVDD	JIR-PCA-SVDD	TPCA-MW	TPCA-CMW
1	0.008	0.006	0.008	0.003	0.003	0.008	0.001	0.546	0.008
2	0.018	0.019	0.015	0.018	0.014	0.016	0.010	0.665	0.023
4	0.688	0.939	0.849	**0**	0.669	0.569	**0**	0.715	**0**
5	0.720	0.758	0.769	**0**	0.718	0.719	0.678	0.285	0.234
6	0.006	0.013	**0**	**0**	0.006	0.006	0.005	0.175	0.001
7	**0**	0.159	**0**	**0**	**0**	**0**	**0**	0.493	**0**
8	0.026	0.028	0.029	0.021	0.023	0.026	0.014	0.029	0.025
10	0.543	0.580	0.659	0.186	0.564	0.521	0.441	0.275	0.236
11	0.516	0.801	0.570	0.280	0.509	0.493	0.176	0.018	**0.009**
12	0.015	0.010	0.011	0.003	0.014	0.014	0.009	0.005	**0.001**
13	0.058	0.049	0.058	0.053	0.053	0.056	0.043	0.015	0.015
14	0.005	0.061	**0**	0.001	**0**	0.004	**0**	0.001	0.001
16	0.696	0.783	0.750	0.135	0.758	0.673	0.558	0.733	0.608
17	0.198	0.240	0.110	0.056	0.100	0.199	0.028	0.020	0.028
18	0.101	0.111	0.106	0.102	0.101	0.099	0.088	0.348	**0.068**
19	0.854	0.993	0.850	0.168	0.895	0.908	0.769	0.068	**0.063**
20	0.571	0.644	0.728	0.196	0.493	0.548	0.311	0.114	**0.054**
21	0.594	0.644	0.611	0.528	0.555	0.574	0.390	1	0.560

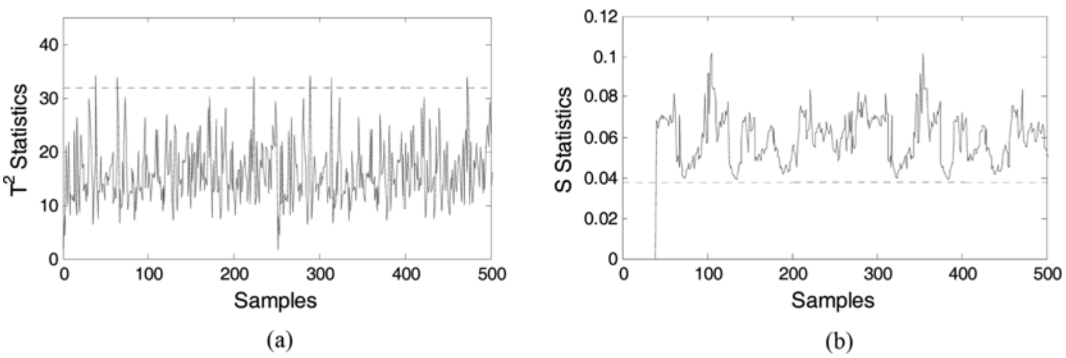


Fig. 12. Process monitoring results of fault 0 in the TE process. (a) PCA; (b) TPCA-CMW.

tion based on the relevance or irrelevance to the principal component and residual subspaces, and combined Bayesian inference, thus producing the FSCB method [17]. Wang et al. used a multi-block strategy to divide the variables based on Kullback-Leibler divergence (KL-MBPCA) [51]. These three PCA-based methods that are used to handle huge data problem are presented in this paper for comparison. In addition, the following paper presents two methods, PCA-SVDD and JIR-PCA-SVDD, which select PCs offline and online, respectively, for the monitoring process. The missed detection rates of the aforementioned monitoring methods are all shown in Table 2. Here, to determine the importance of the reservation of the normal data in CMW, the results that use MW without any additional normal data are also listed for comparison. The results in the table indicate that TPCA-CMW outperforms the other methods, given the significant reduction in the missed detection rates. Meanwhile, the CMW strategy is important in the proposed method because TPCA-MW poorly performs even in many easy faults, such as fault 1.

The monitoring charts present the performance of the proposed method. First, Fig. 12 illustrates the normal process monitoring results of PCA and TPCA-CMW to show the trend of statistics under normal conditions. Subsequently, three process faults are selected to demonstrate the well function of the proposed method. Fault 4 is the fault case caused by a disturbance in the reactor cooling water inlet temperature. Traditional PCA behaves poorly in detecting this fault with statistics fluctuating around the confidence limit, as shown in Fig. 13(a). However, the S statistic immediately shrinks after sample 160, where a fault exists, and avoids the threshold, as shown in the monitoring results in Fig. 13(b). The monitoring chart of the proposed method using the traditional MW is presented for comparison. Evidently, the fault can be detected at the beginning with a statistic that goes below the confidence limit, but the calculated statistics normalize eventually and can no longer reveal fault information. Obviously, the CMW not only aids in capturing online data, but also highlights fault information when a fault occurs. This particular function contributes to the good per-

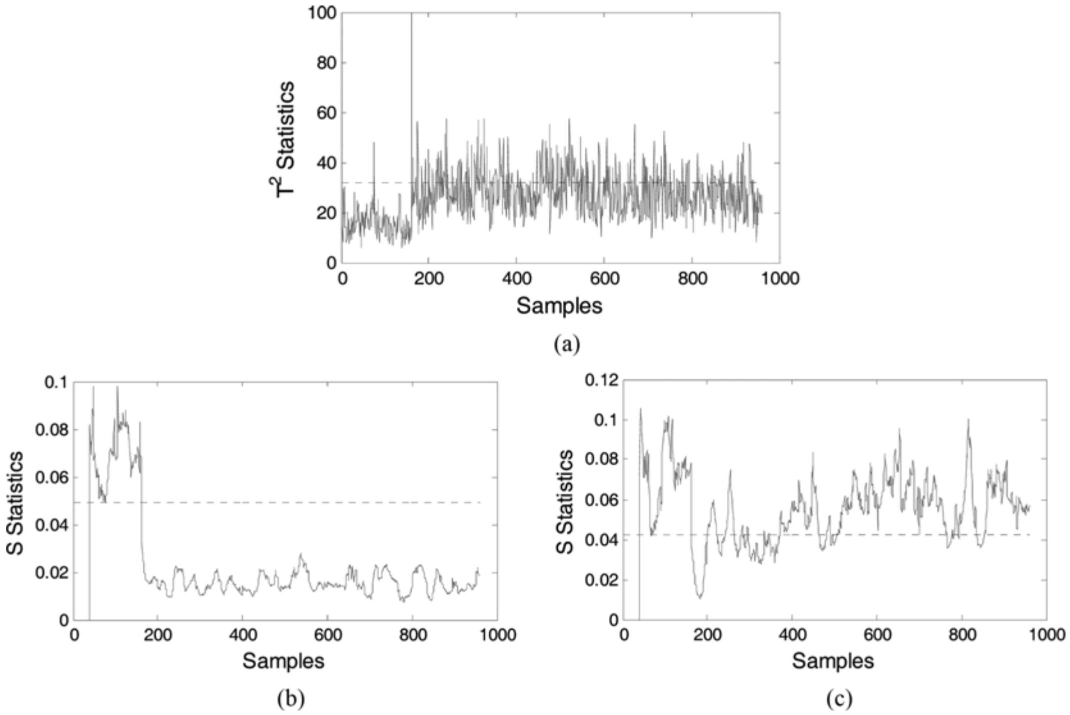


Fig. 13. Process monitoring results of fault 4 in the TE process. (a) PCA; (b) TPCA-CMW; (c) TPCA-MW.

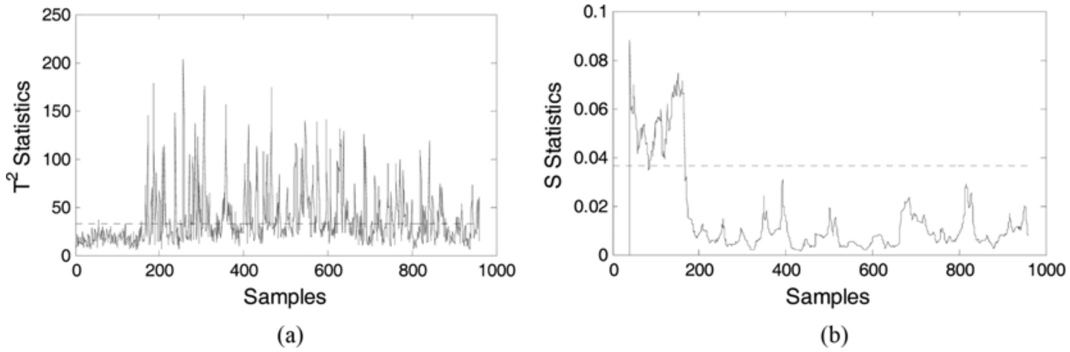


Fig. 14. Process monitoring results of fault 11 in the TE process. (a) PCA; (b) TPCA-CMW.

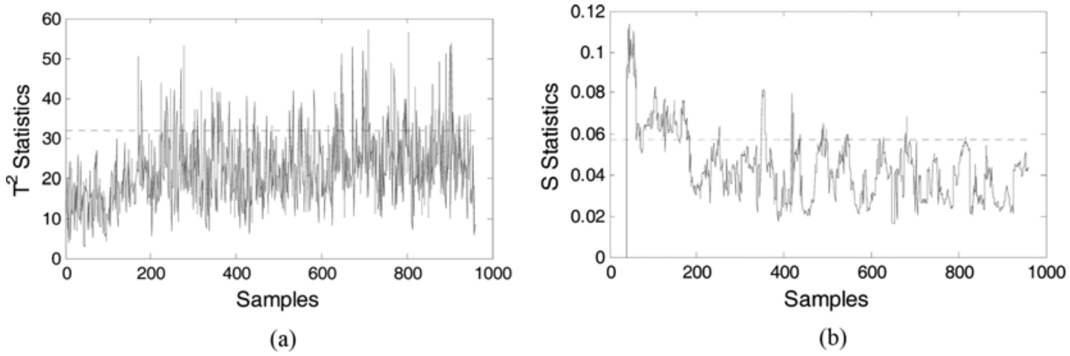


Fig. 15. Process monitoring results of fault 19 in the TE process (a) PCA; (b) TPCA-CMW.

formance of TPCA-CMW in detecting this fault.

The second fault is fault 11, which is a random variation in the reactor cooling water inlet temperature. Figs. 14(a)-(b) present the monitoring results of fault 11 using PCA and the proposed method, respectively. A considerable percentage of statistics below the confidence limit exists in Fig. 14(a), whereas almost all the statistics in

Fig. 14(b) exceed the confidence limit when a fault occurs. The corresponding missed detection rates in Table 2, particularly 0.516 with PCA and 0.009 with TPCA-CMW, also confirm the superior performance of the proposed method. Given that the data would fluctuate significantly after sample 160 and cannot be compensated by the loop system, the function of CMW is not significantly obvious.

The last fault for illustration is fault 19, which is an unknown fault in the TE process. When monitoring this fault using PCA, most statistics still maintain the same state as before fault occurs, as shown in Fig. 15(a). By contrast, the new statistic S shown in Fig. 15(b) can exceed the confidence limit after sample 160, thus improving monitoring performance extensively. The listed monitoring results of these three fault cases all indicate that the proposed TPCA-CMW method utilizes the changing information of the data and presents high sensitivity and accuracy in fault detection.

CONCLUSIONS

A novel real-time process monitoring method that fully extracts variation information from transformation matrix of online PCA model is proposed to improve the monitoring performance of the dynamic process. The transformation matrix in TPCA-CMW model is sensitive to the data used to construct model, and the first several columns always show comparatively significant variation, so these key variation characteristics are measured by polygon area method to reflect the condition of the operating process. In addition, the CMW strategy for data collection containing both normal and online data at the same time can distinguish faulty data timely to help fault detection. The constructed S statistic displays satisfactory monitoring performance, evaluated by two case studies and a comparison with other PCA-based methods. The monitoring results all indicate that TPCA-CMW provides superior performance in dealing with dynamic process and outperforms other PCA-based methods. However, only the first two columns of transformation may lose some information more or less. To achieve comprehensive information extraction, studies uncovering useful information from the remaining parts are in progress. In addition, some techniques are being investigated to aid in suitably capturing the variation information in real-time models.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support from the following foundations: 973 project of China (2013CB733600), National Natural Science Foundation of China (21176073) and the Fundamental Research Funds for the Central Universities.

REFERENCES

1. J.-M. Lee, C. Yoo and I.-B. Lee, *J. Process Control*, **14**, 467 (2004).
2. A. Nowicki, M. Grochowski and K. Duzinkiewicz, *International J. Appl. Mathematics Computer Sci.*, **22**, 939 (2012).
3. Q. Jiang, X. Yan, Z. Lv and M. Guo, *Korean J. Chem. Eng.*, **30**, 1181 (2013).
4. S. S. Kolluri, I. J. Esfahani, P. S. N. Garikiparthy and C. Yoo, *Korean J. Chem. Eng.*, **32**(8), 1486 (2015).
5. C. Lu, S. Xiao and X. Gu, *Korean J. Chem. Eng.*, **31**, 1943 (2014).
6. I. Jolliffe, *Principal component analysis*, Wiley Online Library (2005).
7. R. Wehrens, in Principal component analysis, pp. 43-66, Springer, (2011).
8. L. H. Chiang, R. D. Braatz and E. L. Russell, *Fault detection and diagnosis in industrial systems*, Springer Science & Business Media (2001).
9. B. Schölkopf, A. Smola and K.-R. Müller, *Neural Computation*, **10**, 1299 (1998).
10. J.-D. Shao, G. Rong and J. M. Lee, *Chem. Eng. Res. Design*, **87**, 1471 (2009).
11. C.-Y. Cheng, C.-C. Hsu and M.-C. Chen, *Ind. Eng. Chem. Res.*, **49**, 2254 (2010).
12. Q. Jiang and X. Yan, *Korean J. Chem. Eng.*, **31**, 1935 (2014).
13. A. Hyvärinen and E. Oja, *Neural Networks*, **13**, 411 (2000).
14. A. Hyvärinen, J. Karhunen and E. Oja, *Independent component analysis*, Wiley (2004).
15. Z. Ge and Z. Song, *Korean J. Chem. Eng.*, **26**, 1467 (2009).
16. Z. Ge and Z. Song, *Ind. Eng. Chem. Res.*, **46**, 2054 (2007).
17. C. Tong, Y. Song and X. Yan, *Ind. Eng. Chem. Res.*, **52**, 9897 (2013).
18. Z. Ge, M. Zhang and Z. Song, *J. Process Control*, **20**, 676 (2010).
19. B. Wang, X. Yan, Q. Jiang and Z. Lv, *J. Chemometrics*, **29**(3), 165 (2014).
20. P. Nomikos and J. F. MacGregor, *AIChE J.*, **40**, 1361 (1994).
21. N. A. A. Majid, M. P. Taylor, J. J. Chen, M. A. Stam, A. Mulder and B. R. Young, *Control Eng. Practice*, **19**, 367 (2011).
22. W. Li, H. H. Yue, S. Valle-Cervantes and S. J. Qin, *J. Process Control*, **10**, 471 (2000).
23. C. Cheng and M.-S. Chiu, *Chemometrics and Intelligent Laboratory Systems*, **76**, 1 (2005).
24. T. Korenius, J. Laurikkala and M. Juhola, *Information Sciences*, **177**, 4893 (2007).
25. C. Lau, K. Ghosh, M. Hussain and C. Che Hassan, *Chemometrics and Intelligent Laboratory Systems*, **120**, 1 (2013).
26. L. C. Alwan and H. V. Roberts, *J. Business Economic Statistics*, **6**, 87 (1988).
27. D. Montgomery, C. Mastrangelo, F. W. Faltin, W. H. Woodall, J. F. MacGregor and T. P. Ryan, *J. Quality Technol.*, **23**, 179 (1991).
28. S. Vander Wiel, *Technometrics*, **38**, 139 (1996).
29. A. Negiz and A. Çlinar, *AIChE J.*, **43**, 2002 (1997).
30. A. Simoglou, E. Martin and A. Morris, *Computers Chem. Eng.*, **26**, 909 (2002).
31. W. Ku, R. H. Storer and C. Georgakis, *Chemometrics and Intelligent Laboratory Systems*, **30**, 179 (1995).
32. W. Jingyuan, S. Zhijiang, J. Peng, Y. Ketian and C. Zhiqiang, *Computers Appl. Chem.*, **1**, 2 (2010).
33. S. R. Ryu, I. Noda and Y. M. Jung, *Bull. Korean Chem. Soc.*, **32**, 2232 (2011).
34. X. Wang, U. Kruger and G. W. Irwin, *Ind. Eng. Chem. Res.*, **44**, 5691 (2005).
35. E. R. Malinowski, Factor analysis in chemistry, Wiley (2002).
36. J. Camacho and A. Ferrer, *J. Chemometrics*, **26**, 361 (2012).
37. S. Valle, W. Li and S. J. Qin, *Ind. Eng. Chem. Res.*, **38**, 4389 (1999).
38. Q. Jiang, X. Yan and W. Zhao, *Ind. Eng. Chem. Res.*, **52**, 1635 (2013).
39. Q. Jiang and X. Yan, *AIChE J.*, **60**, 949 (2014).

40. C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, Springer New York (2006).
41. M. Kano, S. Hasebe, I. Hashimoto and H. Ohno, *AIChE J.*, **48**, 1231 (2002).
42. D. P. Robbins, *Discrete Computational Geometry*, **12**, 223 (1994).
43. I. Pak, *Advances in Appl. Mathematics*, **34**, 690 (2005).
44. S. Yoon and J. F. MacGregor, *J. Process Control*, **11**, 387 (2001).
45. J.-H. Cho, J.-M. Lee, S. W. Choi, D. Lee and I.-B. Lee, *Chem. Eng. Sci.*, **60**, 279 (2005).
46. M. Mansouri, M. Nounou, H. Nounou and N. Karim, *J. Loss Prevent. Proc.*, **40**, 334 (2016).
47. C. F. Alcala and S. J. Qin, *Ind. Eng. Chem. Res.*, **49**, 7849 (2010).
48. J. J. Downs and E. F. Vogel, *Comput. Chem. Eng.*, **17**, 245 (1993).
49. Q. Jiang, X. Yan and B. Huang, *IEEE T. Ind. Electron.*, **63**, 377 (2015).
50. K.-L. Du and M. Swamy, Neural Networks and Statistical Learning, 355, Springer (2014).
51. B. Wang, Q. Jiang and X. Yan, *Korean J. Chem. Eng.*, **31**, 930 (2014).