Chinese Journal of Chemical Engineering 47 (2022) 54–70

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

journal homepage: www.elsevier.com/locate/CJChE

Full Length Article

Cycle temporal algorithm-based multivariate statistical methods for fault diagnosis in chemical processes

Jiaxin Zhang [1], Wenjia Luo [1,*], Yiyang Dai [2,*], Yuman Yao [1]

[1] School of Chemistry and Chemical Engineering, Southwest Petroleum University, Chengdu 610500, China
[2] School of Chemical Engineering, Sichuan University, Chengdu 610065, China

ARTICLE INFO

Article history:
Received 5 January 2021
Received in revised form 10 March 2021
Accepted 31 March 2021
Available online 29 July 2021

Keywords:
Cycle temporal algorithm
Fault diagnosis
Dynamic kernel principal component analysis
Multiway dynamic kernel principal component analysis
Reconstruction-based contribution

ABSTRACT

Multivariate statistical process monitoring methods are often used in chemical process fault diagnosis. In this article, (I) the cycle temporal algorithm (CTA) combined with the dynamic kernel principal component analysis (DKPCA) and the multiway dynamic kernel principal component analysis (MDKPCA) fault detection algorithms are proposed, which are used for continuous and batch process fault detections, respectively. In addition, (II) a fault variable identification model based on reconstructed-based contribution (RBC) model that paves the way for determining the cause of the fault are proposed. The proposed fault diagnosis model was applied to Tennessee Eastman (TE) process and penicillin fermentation process for fault diagnosis. And compare with other fault diagnosis methods. The results show that the proposed method has better detection effects than other methods. Finally, the reconstruction-based contribution (RBC) model method is used to accurately locate the root cause of the fault and determine the fault path.

1. Introduction

The scale of equipment and factories has increased with improvements in the data intelligence of chemical industry processes. Due to the flammable, explosive, toxic and corrosive nature of typical chemical processes, any accident can cause a major disaster, which in turn could cause huge environmental, social and economic losses. Effective and timely detection and location of fault locations is an important aspect of ensuring the safety of the current chemical industry process and product quality [1]. Process monitoring (PM) is a widely adopted tool for process safety and quality enhancement [2]. Generally, process monitoring can be divided into three categories: model-based methods, knowledge-based methods, and data-based methods. Data-based methods have been widely used in chemical processes in recent years [3–5].

According to different types of products produced, chemical processes can be divided into continuous processes and batch processes. Because the continuous process and batch process are multivariable control processes. Therefore, the multivariate statistical process monitoring (MSPM) method is one of the most commonly used methods in modern industrial processes. Among them, the principal component analysis (PCA) method in MSPM is the most widely used in chemical process [6–9]. By projecting the data into a low-dimensional subspace containing enough variance information for normal operating data, PCA can effectively process high-dimensional, noisy and highly correlated data. PCA application However, the PCA method has limitations for fault detection and fault identification in chemical processes with strong nonlinearity and dynamics. For the fault detection, Ku *et al.* proposed dynamic principal component analysis (DPCA) [10], Lee *et al.* [11] proposed kernel principal component analysis (KPCA) and Yang *et al.* [12] proposed dynamic kernel principal component analysis (DKPCA) for continuous process fault detection. Compared with continuous processes, batch processes have more operating stages, and the data patterns are usually high-dimensional. Before entering the monitoring model, data needs to be processed in multiway [13]. Therefore, multiway principal component analysis (MPCA) [14], multiway kernel principal component analysis (MKPCA) [15] and adjacent principal component analysis (ADPCA) [16] have been successively proposed. Using the above various PCA models for fault detection, the monitoring space is divided into two subspaces, called the principal component subspace and the residual subspace. By constructing T^2 statistics and SPE statistics to explain the mean and variance information of the processes in the two subspaces respectively [17], when one of the T^2 and SPE statistics exceeds the control limit, it indicates that the system has detected

* Corresponding authors.
E-mail addresses: luowenjia@swpu.edu.cn (W. Luo), daiyy@scu.edu.cn (Y. Dai).

https://doi.org/10.1016/j.cjche.2021.03.058
1004-9541/

a fault. For the fault identification part, the most widely used method based on the PCA model is the contribution of variable method [18]. The method is based on quantifying the contribution of each process variable to a single principal component score. The contribution of each process variable to the principal component in the out-of-control state is added, which is called variable contribution [19,20]. It is believed that the huge contribution of process variables may be the root cause of the fault. In recent years, many variable identification methods have been reported [21–23], but it is still difficult to identify faults, especially when the number of process variables is too large and the process is very complicated. The fault may be covered by the control loop. Therefore, it is very important to detect faults in time and capture the most useful information to identify the fault variable. The above-mentioned fault detection and identification based on the traditional PCA expansion method have some problems. For fault detection, when the amount of data is large and the matrix dimension calculated by the model is large, there are problems of computational redundancy and high fault detection delay. On the basis of the above problems, dynamic trend analysis (QTA) is one of the methods to solve the above problems. some researchers have also proposed some fault diagnosis methods based on QTA, such as: dynamic locus analysis (DLA) [24], dynamic time warping (DTW) [25], artificial immune system (AIS) [26], metric temporal logic (MTL) [27], metric interval temporal logic (MITL) [28] and signal temporal logic (STL) [29]. This article improves the traditional temporal logic, proposes a cycle temporal algorithm (CTA) and combines the DKPCA and MDKPCA fault detection methods to solve the above two problems. CTA not only has the strong qualitative ability of temporal logic, but also can effectively optimize the big data calculation redundancy and computational complexity of chemical process through the two parts of cycle segmentation and cycle merge, so as to improve the accuracy of fault detection and reduce the fault time delay. For fault identification, the main problem is that the existing fault identification methods do not effectively extract fault information, and then effectively identify fault variables. The reconstruction-based contribution (RBC) model to concentrate the effective fault information in the principal component and residual subspace, retain the fault information to the greatest extent, and determine the fault variable and the cause of the fault is used in this paper.

On the basis of the above discussion, this article proposes the CTA and combines it with the DKPCA model and MDKPCA model to form the CTA-DKPCA and CTA-MDKPCA fault diagnosis model, respectively, to monitor continuous and batch processes. The proposed model uses temporal logic to set the threshold, the starting point and ending point of segmentation, and perform piecewise linear fitting on the process data, thereby reducing the dimension of the calculation matrix and improving the calculation accuracy. The CTA-DKPCA model and CTA-MDKPCA model can effectively collect useful information in the subspace, deal with the loss of information, and greatly improve the detection rate. Combined with an enhanced fault variable identification algorithm, the RBC method is used to reconstruct the variable contributions, to visually display the contribution rate of the variable (expressed as a percentage), to emphasize the strong correlation between variables and to extract fault information. In the continuous process, 3 different types of faults are selected, and comparing the CTA-DKPCA model with similar derived algorithms in the Tennessee Eastman (TE) process indicates the advantages of the CTA-DKPCA method. Then, comparison with the 2-class SVM optimization algorithm proposed by Onel *et al.* [30] indicates the accuracy of the fault detection of the CTA-DKPCA model. For batch processes, choosing a typical fault and making comparisons with the MDKPCA method shows the advantages of the CTA-MDKPCA method. Finally, the RBC method is used to diagnose the key variables that cause the fault in the continuous process and the batch process, locate the root cause of the fault, and determine the path of the fault.

The rest of the paper is structured as follows. In Section 2, the cycle temporal algorithm is proposed, and combined with the DKPCA model and MDKPCA model, and the fault variable is diagnosed through the RBC method. In addition, the continuous and batch processes fault diagnosis models are obtained. In Section 3, the TE process and penicillin fermentation process are used to demonstrate the effectiveness of the CTA-DKPCA method and the CTA-MDKPCA model. Then, the RBC method is used to determine the root cause of the fault and the fault propagation path. In Section 4, we summarize our work.

2. Methods

The traditional PCA method has limitations for fault diagnosis of chemical continuous and batch processes with nonlinearity and dynamics. Therefore, there is a need for an algorithm that can perform calculations in batch and continuous processes, namely, the MDKPCA algorithm and DKPCA method. Specifically, the DKPCA is for continuous process fault diagnosis, and MDKPCA is for batch process fault diagnosis.

2.1. Dynamic kernel principal component analysis (DKPCA)

PCA is a popular dimension reduction technique to project the data onto a lower dimensional subspace that contains most of the variance in the data. However, PCA can only solve the linear correlation between data. For the strong nonlinear and dynamic characteristics of chemical process data, the PCA method has certain difficulties in calculating this type of data. Therefore, the KPCA method was proposed by Kumar *et al.* [31] through nonlinear improvement, and KPCA method is a kind of learning machine based on kernel. The data of original space is mapped to feature space by nonlinear mapping, and linear method is used to analyze in feature space. Thus, nonlinear problem of original space is transformed into linear problem of feature space. However, the dynamic capture of data requires the conversion of dynamic models, so a DKPCA model of non-linear and dynamic capture of data is formed.

Define the normal data set $\boldsymbol{X}$ contains m variables, and each variable has n observations, the vectors at time t and augmented matrix $\boldsymbol{X}_s$ containing the observations at the previous s time to reflect the relationship between the variables' dynamic relationship.

$$\boldsymbol{X}_s = \begin{bmatrix} x_t^T & \cdots & x_{t-s}^T \\ \vdots & \ddots & \vdots \\ x_{t+s-n}^T & \cdots & x_{t-n}^T \end{bmatrix} \tag{1}$$

Then use the dynamic matrix $\boldsymbol{X}_s$ to establish the DPCA through the PCA method, and then the dynamic characteristics can be analyzed.

$$\boldsymbol{X}_s = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \tag{2}$$

where $\boldsymbol{T}$ is the score matrix; $\boldsymbol{P}$ is the load matrix; $\boldsymbol{E}$ is the residual matrix, which is the projection of the sample in the residual space.

There is a nonlinear mapping $\Phi_i(t : t-s) : R^m \rightarrow F$, which maps the samples in the input space to the high-dimensional feature space F, and the covariance matrix of the samples in the feature space is calculated as follows:

$$\overline{\boldsymbol{C}} = \frac{1}{N}\sum_{i=1}^{N}\left(\Phi_i(x_i(t : t-s)) - m_{\Phi_i}\right)\left(\Phi_i(x_i(t : t-s)) - m_{\Phi_i}\right)^T \tag{3}$$

where $m_{\Phi_i} = \sum_{i=1}^{N} \Phi_i(x_i(t:t-s))/N$ is the mean value of the sample mapped in F space, let $\overline{\Phi}_i(x_i(t:t-s)) = \Phi_i(x_i(t:t-s)) - m_{\Phi_i}$. Then the principal component load vector v in the feature space can be obtained by solving the following eigenvalue problem:

$$\lambda v = \overline{\boldsymbol{C}} v = \frac{1}{N}\sum_{j=1}^{N}\left(\overline{\Phi}_j\left(x_j(t:t-s)\right)\cdot v\right)\overline{\Phi}_j\left(x_j(t:t-s)\right)^{\mathrm{T}} \tag{4}$$

where $\lambda > 0$ and $\upsilon \neq 0$, $''\cdot''$ means dot product. The solution υ of the characteristic Eq. (4) can be expressed as a linear combination of $\overline{\Phi}_1(x_1(t:t-s)), \overline{\Phi}_2(x_2(t:t-s)), \cdots, \overline{\Phi}_N(x_N(t:t-s))$, namely: $\upsilon = \sum_{j=1}^{N} \alpha_j \overline{\Phi}_j\left(x_j(t:t-s)\right)$.

Sitting on both sides of Eq. (4) as $\overline{\Phi}_j\left(x_j(t:t-s)\right)$, we obtain Eq. (5):

$$\lambda\left(\overline{\Phi}_j\left(x_j(t:t-s)\right)\cdot v\right) = \overline{\Phi}_j\left(x_j(t:t-s)\right)\left(\overline{\boldsymbol{C}} v\right) \tag{5}$$

Define the kernel matrix $\boldsymbol{K} = [K_{ij}]$, where $K_{ij} = (\Phi_i(x_i(t:t-s))\cdot \Phi_j(x_j(t:t-s)))$, $\overline{K} = [\overline{K}_{ij}]$, where $\overline{K}_{ij} = (\Phi_i(x_i(t:t-s))\cdot\Phi_j(x_j(t:t-s)))$, The eigenvalue problem of Eq. (5) can be simplified as:

$$\lambda\boldsymbol{\alpha} = \left(\frac{1}{N}\right)\bar{\boldsymbol{K}}\boldsymbol{\alpha} \tag{6}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_N]^{\mathrm{T}}$. In the above derivation, it is assumed that the sample is centralized in the feature space. However, because we can't give the specific form of mapping Φ_i under normal circumstances, we can't find m_{Φ_i}, so we can't directly calculate $\overline{\boldsymbol{K}}$. This article uses $\boldsymbol{K}$ to find the method of $\overline{\boldsymbol{K}}$, namely: $\overline{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{E} - \boldsymbol{E}\boldsymbol{K} + \boldsymbol{E}\boldsymbol{K}\boldsymbol{E}$, where $\boldsymbol{E} \in \boldsymbol{R}^{N\times N}$ and all components are $\frac{1}{N}$.

For feature vector normalization is necessary to satisfy Eq. (7):

$$\|\boldsymbol{\alpha}\|^2 = \frac{1}{N\lambda} \tag{7}$$

Let the first P eigenvalues satisfying Eq. (6) are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$, The corresponding eigenvectors form the load matrix $\boldsymbol{V} = [\upsilon_1 \cdots \upsilon_{\mathrm{p}}]$ of the principal component. Let the coefficient matrix of $\boldsymbol{V}$ is $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_p]$, Among them, the number of principal components p can be used based on the principle of minimum mean square reconstruction error, which is obtained through cross-validation.

Given a new sample $\boldsymbol{X}_{s,k}$, which maps $\Phi_k(x_k(t:t-s))$ in the feature space, the principal component score $\boldsymbol{t}_k = [t_{k1}, \cdots, t_{ki}, \cdots, t_{kp}]^{\mathrm{T}}$ in the feature space can be obtained by projecting to each load vector υ_i in the direction of $i = 1, \cdots, p$. Where t_{ki} represents the projection of $\Phi_k(x_k(t:t-s))$ in the v_i direction, namely:

$$t_{kj} = v_i \cdot \overline{\Phi}_k(x_k(t:t-s)) = \left(\alpha_i^{\mathrm{T}}\right)\overline{\boldsymbol{K}}_k \tag{8}$$

where $\overline{\boldsymbol{K}}_k$ and $\boldsymbol{K}_k$ are shown in Eq. (9) and Eq. (10).

$$\overline{\boldsymbol{K}}_k = \boldsymbol{K}_k - \boldsymbol{K}_k e - \boldsymbol{E}\boldsymbol{K}_k - \boldsymbol{E}\boldsymbol{K}_k e \tag{9}$$

$$\boldsymbol{K}_k = [K(x_1(t:t-s), x_{\mathrm{k}}(t:t-s))K(x_{\mathrm{N}}(t:t-s), x_k(t:t-s))]^{\mathrm{T}} \tag{10}$$

where $e \in \boldsymbol{R}^{N\times 1}$, $\boldsymbol{E} \in \boldsymbol{R}^{N\times N}$, each component of $\boldsymbol{e}$ and $\boldsymbol{E}$ are $\frac{1}{N}$.

Gaussian kernel function is selected here as KPCA method's kernel function, and the expression as follows:

$$K(x,y) = \exp\left(-\frac{||x-y||^2}{2\sigma}\right) \tag{11}$$

2.2. Multiway dynamic kernel principal component analysis

The batch process is a repetitive production process, and its data collection has more one-dimensional "batch" components than the continuous process data collection. A three-dimensional data matrix $\boldsymbol{X}(I\times J\times K)$, can be used to represent the process data collection. In the matrix, I is the number of batches, J is the number of variables, and K is the number of sampling points. MDKPCA cuts the three-dimensional data matrix into batch and variable data blocks $\boldsymbol{X}(I\times J)$, along the time axis, and each data block is arranged horizontally to the right to form a new two-dimensional data matrix $\boldsymbol{X}(I\times JK)$, as shown in Fig. 1. After the three-dimensional data matrix is expanded into a two-dimensional data matrix, the MDKPCA data processing and analysis process is equivalent to the DKPCA method.

The multiway method combined with the above DKPCA method can diagnose batch process faults.

2.3. Cycle temporal algorithm (CTA)

Fault diagnosis is particularly important for system safety, and qualitative trend analysis (QTA) [32] is an important technology for fault diagnosis. The main idea of QTA is to use the measurement signal as a trend sequence based on primitives, which are constant, rising and falling. However, the traditional temporal logic fault diagnosis method based on QTA has some limitations. That is, the system scale is small, the number of variables and sensors are small.

In this section, QTA is extended to the time constraints, and a new fault diagnosis method is proposed. An extended QTA method is proposed; the cycle temporal algorithm (CTA), which combines the aforementioned DKPCA and temporal logic. DKPCA was used to obtain the principal component (PC). The selected PC saves most of the process data variance, and each PC is a linear combination of all process variables. Compared with the traditional QTA, the proposed method uses temporal logic to describe the dynamic characteristics of the process in time. The temporal logic is statistically learned from the collected process data through the comprehensive understanding of the time characteristics of a single variable in the process variables to the linear correlation time characteristics between them. This helps to improve the accuracy of fault detection and recognition. The segmentation of the cycle temporal algorithm is brought into the model calculation, which also improves the calculation speed.

2.3.1. Temporal logic

The temporal logic equation is summarized and defined as follows:

$$\varphi := |\mathrm{T}|\overline{G}_{[b,e]}\varphi|\overline{F}_{[b,e]}\varphi| \dashv \varphi|\varphi_1 \wedge \varphi_2|\varphi_1\overline{U}_{[b,e]}\varphi_2 \tag{12}$$

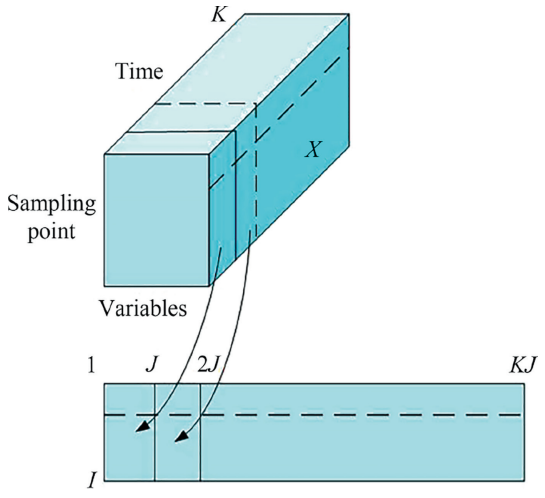

Fig. 1. MDKPCA data matrix decomposition.

where T as "always true", "negation" ($\dashv$) and "combination" ($\wedge$) are standard Boolean join operators. The time series operators $\overline{G}_{[b,e]}$, $\overline{F}_{[b,e]}$, and $\overline{U}_{[b,e]}$ represent the derived "always", the derived "final" and the derived "until", respectively and where $[b,e]$ represents the time interval, satisfying $b \le e$. Note that the derived $\overline{G}$, $\overline{F}$, and $\overline{U}$ are slightly different from the original G, F and U. Generally, $\overline{G}_{[b,e]}$ and $\overline{F}_{[b,e]}$ can be expressed as $\overline{G}_{[b,e]} = \dashv \overline{F}_{[b,e]} \dashv \varphi$ and $\overline{F}_{[b,e]} = \mathrm{T}\overline{U}_{[b,e]}\varphi$. In addition, based on the operator $\overline{U}$, define the "concatenation" $(\cdot)$operator, $\left(\overline{G}_{[0,b]}\varphi_1\right) \cdot \varphi_2 := \varphi_1 \overline{U}_{[b,e]}\varphi_2$, and $b = e$. The atomic predicate $u : R \to B$ represents a mapping from the real domain to the Boolean domain. This article defines $u(x[\tau]) := f(x[\tau]) \le 0$ and $f(x[\tau]) = (x[\tau] - k\tau - c)^2 - \delta_e$. δ_e is the preset threshold. Where $\delta_e = 4\sigma^2$, σ is the standard deviation of the data sequence. $x[\tau]$ represents the data point at time τ and k and c are constant parameters.

Given a data sequence of length n, $x = \langle x_1, x_2, \cdots x_n \rangle$. where x represents the data point at time τ. The defined temporal logic satisfies the following:

- $x[\tau] \models u$ if and only if $f(x[\tau]) > 0$.
- $x[\tau] \models \mathrm{T}$ if and only if "always true".
- $x[\tau] \models \overline{G}\varphi$ if and only if $\forall \tau' \in [1,\tau]$ satisfies $x[\tau'] \models \varphi$.
- $x[\tau] \models \overline{F}\varphi$ if and only if $\exists \tau' \in [1,\tau]$ satisfies $x[\tau'] \models \varphi$.
- $x[\tau] \models \dashv \varphi$ if and only if $\dashv (x[\tau] \models \varphi)$.
- $x[\tau] \models \varphi_1 \wedge \varphi_2$ if and only if $x[\tau] \models \varphi_1$ and $x[\tau] \models \varphi_2$.
- $x[\tau] \models \varphi_1 \overline{U}_{[b,e]}\varphi_2$ if and only if one of the following conditions:
  (1) If $\tau < b$, then $x[\tau] \models \mathrm{T}$.
  (2) If $\tau > e$, then $\exists \tau' \in [b,e]$ satisfies $x[\tau'] \models \varphi_2$ and $\forall \tau'' \in [1,\tau]$ satisfies $x[\tau''] \models \varphi_1$
  (3) If $\tau \in [b,e]$, then $\forall \tau' \in [1,\tau]$ satisfies $x[\tau'] \models \varphi_1$ or $\exists \tau' \le \tau$ satisfies $x[\tau'] \models \varphi_2$, and $\forall [\tau''] \in [1,\tau']$ satisfies $x[\tau''] \models \varphi_1$.

2.3.2. Construction of cycle temporal algorithm

To simplify the construction process of temporal logic, the calculation time is decreased and the optimal number of principal components is extracted. The CTA is improved based on temporal logic. The algorithm is divided into cycle segmentation process and cycle merge process as shown in Fig. 2.

First, determine the start point b_L and end point e_L of the data segment for the data that needs to be divided, and perform a segmentation process and linear fitting according to the entire data sequence $X = \langle X_1, X_2, \cdots X_n \rangle$. The two new end points generated by the division are set to $e_L/2$, The linear fitting equation is: $(\widehat{x}[\tau] = k\tau + c)$; k and c represent the slope and the y-axis intercept, respectively. Then calculate the size of the error generated by this linear fitting. If the linear fitting error err is greater than δ_e which calculated from the data segment equation above, the two sequences X_1 and X_2 after twice cycle are divided into two again. Then, as in the previous steps, linear fitting and error calculation are performed on the segmented data segments respectively, until the linear fitting errorerr is not greater than the predetermined error threshold δ_e. The linear fitting error err is calculated as shown in Eq. (13):

$$\mathrm{err} = \sum_{L=b_L}^{e_L} (x[\tau] - k_L - c_L)^2 / (e_L - b_L + 1) \tag{13}$$

If the err is less than the threshold but the data segment in the previous cycle also meets the condition, the length of the data segment is problematic. Therefore, the long data segment is re-circularly segmented at this time. Make the length of all data segments equal. Once a certain linear fit is obtained, according to the remaining data sequence, repeat the above steps until the entire data sequence is divided into data segment sequence, and each data segment in the sequence corresponds to a linear fit. The obtained segments are input into the model as input to the model for calculation, and after the segment output results are obtained, the merging process is executed. Merge two adjacent data segments into one data segment and perform linear fitting for the data segments to be combined. If the fitting error is not greater than the predetermined error threshold δ_e, merge the two data segments into one data segmentation; otherwise, it will not merge. Return the cycle to calculate the linear fitting error until the cycle to calculate err is greater than the threshold δ_e, and further judge whether the end point of the merged segment is the segment end point e_L. If the data points are not merged completely, return to continue to cycle to calculate the linear fitting error of the remaining data set, Until the merged data segment is merged into a complete data segment. It indicates that the start point b_L and the end point e_L are in the same data segment.

Based on the above segmented fitting method, cycle segmentation and cycle merge are performed for a given data sequence. A sequence of quaternions of degree L can be obtained:

$$T := \langle (k_1, c_1, b_1, e_1), (k_2, c_2, b_2, e_2), \cdots, (k_L, c_L, b_L, e_L) \rangle \tag{14}$$

In each quadruple, $(k_i, c_i, b_i, e_i)(i \in [1, L])$, k_i and c_i are the slope and y-axis intercept of the i-th segment of the linear fitting, respectively. The integers b_i and e_i represent the start and end times of the data segment, respectively, and the corresponding atomic predicate is shown in Eq. (15):

$$u_i := \left((x[\tau] - k_i\tau - c_i)^2 - \delta_e \le 0 \right) \tag{15}$$

Based on the above, the temporal logic is reconstructed as:

$$\varphi := \varphi_1 U_{[b_1,e_1]} \varphi_2 U_{[b_2,e_2]} \cdots \varphi_{L-1} U_{[b_{L-1},e_{L-1}]} \varphi_L \tag{16}$$

which means if $i \in [1, L-1]$, then $\varphi_i := u_i$.

Eventually, The CTA combines temporal logic to define the principal of the segmented cycle, determine the size of the cycle and the error threshold, and the final segment starting point and ending point. Complete the cycle segmentation part of the model data to reduce the dimensionality calculation of the matrix. The cycle merge part is the inverse process of the cycle segmentation part, which can merge the effective information obtained in the model. Through the effective merging and counting of the data, the calculation efficiency of the model is improved, the accuracy of the calculation result is improved, and the delay caused by the calculation is reduced.

2.4. Fault diagnosis for continuous and batch process

2.4.1. Continuous process fault detection

As in the previous section, for continuous process fault detection, $\widehat{\boldsymbol{P}}^{\boldsymbol{i}}$ is the single load matrix divided by the CTA. In addition, let the column vector $z(J \times 1)$ represent the current observation data point, and divide z into the corresponding L blocks based on the L variable block in the previous section, which is $z^i\left(J^i \times 1\right)(i = 1, 2, \cdots, L)$. Based on the T^2statistics and SPE statistics of the global DKPCA model constructed above, two statistics calculation equations of the segmented DKPCA model are obtained:

$$\mathrm{SPE}_{\mathrm{s,c},i} = \left(z^i\right)^{\mathrm{T}} \left(\boldsymbol{I}_{\boldsymbol{j_i}} - \widehat{\boldsymbol{P}}^{\boldsymbol{i}} \left(\widehat{\boldsymbol{P}}^{\boldsymbol{i}} \right)^{\mathrm{T}} \right) \quad z^i \le \mathrm{SPE}_{\mathrm{s,c},i,\mathrm{lim}} \tag{17}$$

$$T^2_{\mathrm{s,c},i} = \left(z^i\right)^{\mathrm{T}} \widehat{\boldsymbol{P}}^{\boldsymbol{i}} \left(\boldsymbol{\Lambda}^{\boldsymbol{i}} \right)^{-1} \left(\widehat{\boldsymbol{P}}^{\boldsymbol{i}} \right)^{\mathrm{T}} \quad z^i \le T^2_{\mathrm{s,c},i,\mathrm{lim}} \tag{18}$$

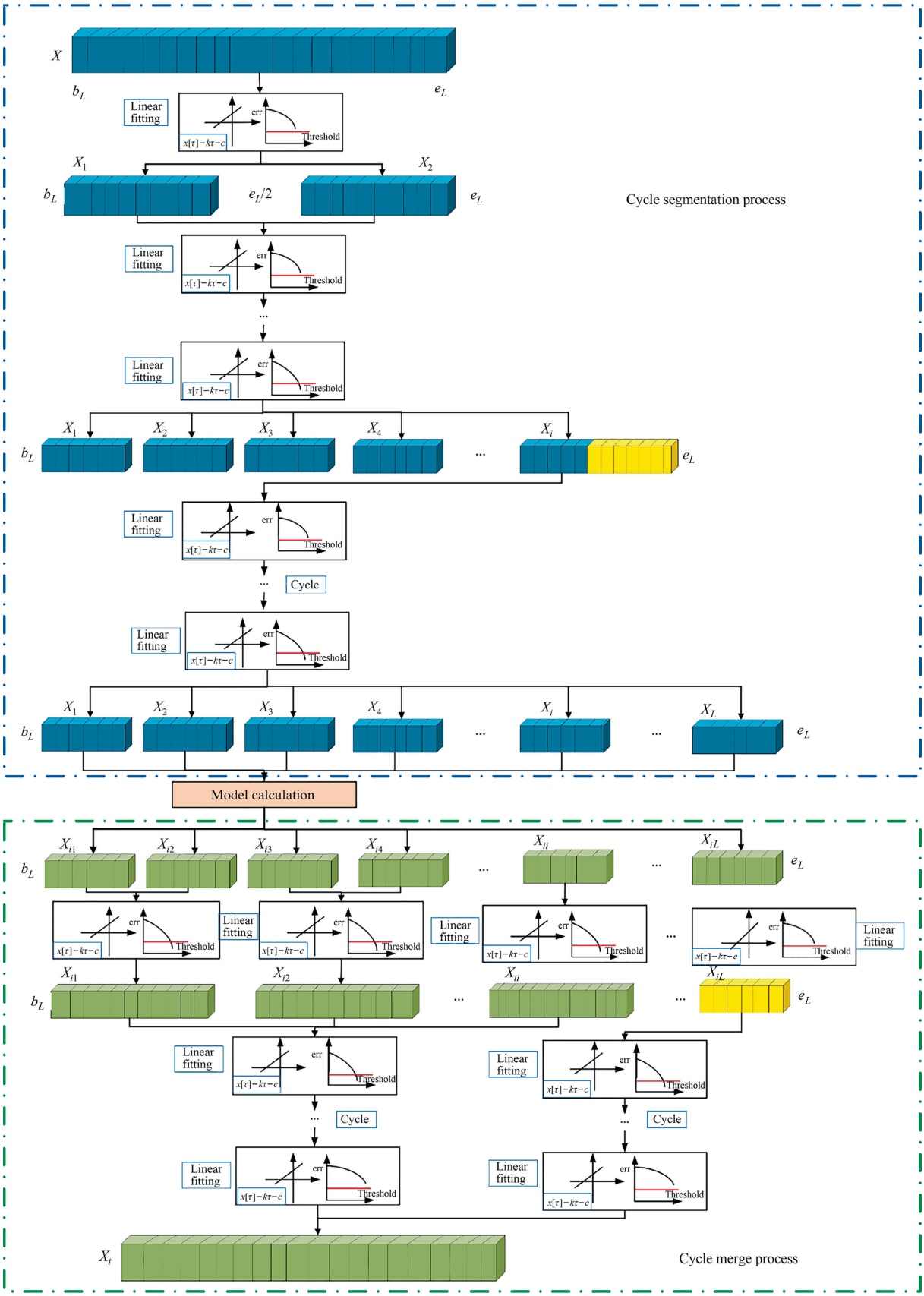

Fig. 2. The cycle temporal algorithm calculation process.

where $\Lambda^i = \left(\boldsymbol{T}_n^i\right)^{\mathrm{T}} \boldsymbol{T}_n^i/(N_n-1)$, $\widehat{\boldsymbol{P}}^i(J_i \times l_i)$ and $\boldsymbol{T}_n^i(N_n \times l_i)$ are the load matrix and score matrix of the$i$th DKPCA model, respectively. $\boldsymbol{I}_{\boldsymbol{J_i}}(J_i \times J_i)$ is the unit matrix, s stands for dynamic matrix, c stands for continuous process. Given confidence level α, these two control limits can be obtained from the χ distribution and the F distribution:

$$\mathrm{SPE}_{\mathrm{s,c},i,\mathrm{lim}} = \frac{v_i}{2m_i}\chi^2_{2m_i^2/v_i,\alpha} \tag{19}$$

$$T^2_{\mathrm{s,c},i,\mathrm{lim}} = \frac{l_i\left(J_i{}^2-1\right)}{J_i(J_i-l)}F_{l_iJ_i-l_i,\alpha} \tag{20}$$

where v_i and m_i^2 represent the mean and variance respectively.

If one of the above $2+2L$ statistics exceeds the corresponding control limit, the system is considered to have fault.

2.4.2. Batch process fault detection

The fault detection process of the batch process is similar to the continuous process, and the statistical data are $T^2$and SPE. Therefore, the $T^2_{\mathrm{s,b},i}$ statistic and the $\mathrm{SPE}_{\mathrm{s,b},i}$ statistic are calculated according to Eqs. (21) and (22):

$$T^2_{\mathrm{s,b},i} = t_i \boldsymbol{S_i}^{-1} t_i^{\mathrm{T}} \tag{21}$$

$$\mathrm{SPE}_{\mathrm{s,b},i} = e_i^{\mathrm{T}} e_i \tag{22}$$

where t_i refers to the score vector, $\boldsymbol{S_i}$ represents the covariance matrix, and e_i is representative of the residual vector, b means batch process. The control limits (control thresholds) of the two statistics are calculated according to the methods given in Eqs. (23) and (24):

$$T^2_{\mathrm{s,b,lim},i} \sim R\left(I_i^2-1\right)/I_i(I_i-R)F_{R,I_i-R,\alpha} \tag{23}$$

$$\mathrm{SPE}_{\mathrm{s,b,lim},i} \sim g\chi^2(h) = \frac{\left(\frac{v_i}{2m_i}\right)\chi^2 2m_i^2}{v_i},\alpha \tag{24}$$

In the Eq. (24), g is the weight, h is the degree of freedom, and the values of g and h are calculated by referring to the value of the SPE statistics at the ith moment of the model. The calculation method is $g = v/2m$ and $h = 2m^2/v$, where m and v are the mean and variance of the SPE statistics at the kth time in the reference model, respectively. The T^2 statistic is considered to approximately obey the F distribution.

2.4.3. Fault identification

Once the system detects a fault, then the numerical method of the RBC model is used to identify the variable contribution rate.

The corresponding variable direction is unit vector $\xi(J \times 1)$ with amplitude f_s; in the ith model, the corresponding model is unit vector $\xi_k^i(J_i \times 1)$ with amplitude f_k^i. ξ^i, ξ_k^i. Except for the first component being 1, all other components are 0, and the variable reconstruction value of data point z^i is:

$$z_k^i = z^i - \xi_k^i f_k^i \tag{25}$$

The statistical reconstruction index is defined as:

$$\psi\left(z_k^i\right) = \left(z_k^i\right)^{\mathrm{T}} M z_k^i \tag{26}$$

For the continuous process, based on Eqs. (17) and (18):

$$M_{\mathrm{c}}^i = \frac{I_{J_i} - \widehat{\boldsymbol{P}}^{\boldsymbol{i}}\left(\widehat{\boldsymbol{P}}^{\boldsymbol{i}}\right)^{\mathrm{T}}}{2\mathrm{SPE}_{i,\mathrm{lim}}} + \frac{\widehat{\boldsymbol{P}}^{\boldsymbol{i}}\left(\Lambda^{\boldsymbol{i}}\right)^{-1}\left(\widehat{\boldsymbol{P}}^{\boldsymbol{i}}\right)^{\mathrm{T}}}{2T^2_{i,\mathrm{lim}}} \tag{27}$$

For the batch process, based on Eqs. (21) and (22):

$$M_{\mathrm{b}}^i = \frac{I_{J_i} - \boldsymbol{S}^{\boldsymbol{i}}\left(\boldsymbol{S}^{\boldsymbol{i}}\right)^{\mathrm{T}}}{\mathrm{SPE}_{i,\mathrm{lim},\alpha}} + \frac{\boldsymbol{S}^{\boldsymbol{i}}\left(\boldsymbol{t}^{\boldsymbol{i}}\right)^{-1}\left(\boldsymbol{S}^{\boldsymbol{i}}\right)^{\mathrm{T}}}{T^2_{i,\mathrm{lim},\alpha}} \tag{28}$$

Minimize the reconstruction index f_k^i to obtain:

$$f_k^i = \left(\left(\xi_k^i\right)^{\mathrm{T}} M_{\mathrm{c/b}}^i \xi_k^i\right)^{-1}\left(\xi_k^i\right)^{\mathrm{T}} M_{\mathrm{c/b}}^i z^i \tag{29}$$

The contribution rate of the kth variable is defined as:

$$\begin{aligned}\psi\left(z_k^i\right) &= \left(z^i\right)^{\mathrm{T}} M_{\mathrm{c/b}}^i\left[I_{J_i} - \xi_k^i\left(\left(\xi_k^i\right)^{\mathrm{T}} M_{\mathrm{c/b}}^i \xi_k^i\right)^{-1}\left(\xi_k^i\right)^{\mathrm{T}} M_{\mathrm{c/b}}^i\right] z^i \\ &= \psi\left(z^i\right) - RBC_k{}^i\end{aligned} \tag{30}$$

It can be seen from the above two Equations that the larger the contribution rate $RBC_k{}^i$, the smaller the reconstruction index $\psi\left(z_k^i\right)$.

Integrating the above methods, the CTA-DKPCA and CTA-MDKPCA fault diagnosis models are obtained. The specific process is shown in Fig. 3.

The fault diagnosis model of the continuous and batch process based on CTA-DKPCA and CTA-MDKPCA is divided into four parts. In the first part, we obtained the required optimal segmentation data and segmentation data fitting threshold through the calculation temporal logic. The second part is the model calculation. Through the calculation of the offline and online stages in the model, it can be concluded whether the detection result of each piece of data exceeds the limit, and whether there is a fault. The third part is the segmented merge part, which linearly merges the segmented data after calculation to determine whether there are faulty data positions in the whole process. The fourth part is the variable identification part. In the process of fault detection, the contribution rate of each variable is calculated by RBC method, and the cause of the fault is comprehensively analyzed according to the detection result. The specific implementation steps are as follows:

(1) Cycle segmentation process:

① Initial process dataset $\boldsymbol{X}_{\mathrm{c/b}} = [x_{\mathrm{c/b}}(1), x_{\mathrm{c/b}}(2), x_{\mathrm{c/b}}(3), \cdots, x_{\mathrm{c/b}}(n)]$, set initial segment number $L = 1$. Where c represents continuous process, b represents batch process.

② Perform a linear fit to the initial process dataset $\boldsymbol{X}_{\mathrm{c/b}}$. The fitting parameters δ_{e}, b_1, and e_1 are set by Eq. (12).

③ The piecewise linear fitting error err is shown in Eq. (13). If the fitting error err is less than δ_{e}, the current segment is determined to be the final segment number L; Otherwise, divide $L = L+1$, divide the current segment into two halves, update the end value $e_L = e_L/2$ of the new segment, and then return to step 2 until err is less than the threshold δ_{e}. Go to step 4.

④ After err reaches the threshold δ_{e}, continue to judge that the set end time point e_{L}has reached the data length L. If it does not reach, it means that all data has not been trained. Return to step 3 to calculate the remain data in a cycle until the data is completely trained. Obtain new L segment data.

⑤ Input all the segmented data segments into the DKPCA/MDKPCA model for calculation.

(2) Model calculation

Off-line model:

① Collect normal segmentation data $\boldsymbol{X}_{\boldsymbol{c},\boldsymbol{i}}(N \times J)$ of the continuous process or the batch process $\boldsymbol{X}_{\boldsymbol{b},\boldsymbol{i}}(I \times J \times K)$. For continuous process data, N and J are the sampling points and the number of variables, respectively. For batch process data, I is the number of batch, J is the number of variables, and K is the number of sampling points.

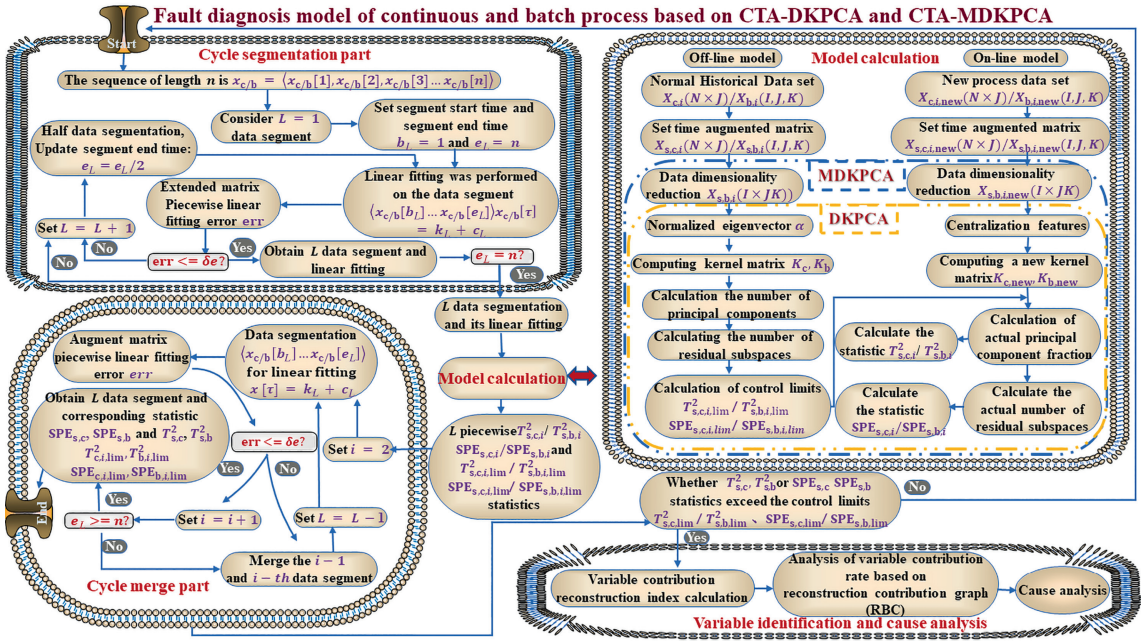

Fig. 3. The fault diagnosis model of continuous and batch processes based on CTA-DKPCA and CTA-MDKPCA.

② Set the dynamic time correlation of the collected data to obtain the matrix $\boldsymbol{X}_{\mathrm{s,c},i}(N\times J)$ and $\boldsymbol{X}_{\mathrm{s,b},i}(I\times J\times K)$ that capture the dynamics of the data.

③ The continuous process segmented data $\boldsymbol{X}_{\mathrm{s,c},i}(N\times J)$ enters step 4, and the batch process segmented data $\boldsymbol{X}_{\mathrm{s,b},i}(I\times J\times K)$ is reduced to two-dimensional data: $\boldsymbol{X}_{\mathrm{s,b},i}(I\times JK)$.

④ Standardize the segmented matrix $\boldsymbol{X}_{\mathrm{s,c},i}(N\times J)/\boldsymbol{X}_{\mathrm{s,b},i}(I\times JK)$ to have a zero mean and unit error, and the value of α_i is determined by Eq. (6): $\lambda\boldsymbol{\alpha}=\left(\frac{1}{N}\right)\overline{\boldsymbol{K}}\boldsymbol{\alpha}$.

⑤ Calculate the kernel matrix $\boldsymbol{K}_{\mathrm{c}}$ and $\boldsymbol{K}_{\mathrm{b}}$ of the each block matrix.

⑥ Calculate the number of principal component subspaces and residual subspaces.

⑦ After calculating the standard control limit $\mathrm{SPE}_{\mathrm{s,c},i,\mathrm{lim}}$, $T^2_{\mathrm{s,c},i,\mathrm{lim}}$/ $T^2_{\mathrm{s,b},i,\mathrm{lim}}\mathrm{SPE}_{\mathrm{s,b},i,\mathrm{lim}}$ of the no fault condition by Eqs. (17), (18), (21), (22), enter step 8 of the online detection section.

On-line model:

① Collect new segmented process data $\boldsymbol{X}_{\mathrm{c},i,\mathrm{new}}(N\times J)/\boldsymbol{X}_{\mathrm{b},i,\mathrm{new}}(I\times J\times K)$.

② Set the dynamic time correlation of the collected data to obtain the matrix $\boldsymbol{X}_{\mathrm{s,c},i,\mathrm{new}}(N\times J)$ and $\boldsymbol{X}_{\mathrm{s,b},i,\mathrm{new}}(I\times J\times K)$ that capture the dynamics of the data.

③ The continuous process segmented data $\boldsymbol{X}_{\mathrm{s,c},i,\mathrm{new}}(N\times J)$ enters step 4, and the batch process data $\boldsymbol{X}_{\mathrm{s,b},i,\mathrm{new}}(I\times J\times K)$ is reduced to two-dimensional data: $\boldsymbol{X}_{\mathrm{s,b},i,\mathrm{new}}(I\times JK)$.

④ Perform the standardization process

⑤ Calculate the new kernel matrix $\boldsymbol{K}_{\mathrm{c,new}}$ and $\boldsymbol{K}_{\mathrm{b,new}}$.

⑥ Centralize the kernel matrix by Eq. (9).

⑦ Calculate the nonlinear principal component by Eq. (8): $t_{kj}=v_i\cdot\overline{\Phi}_k(x_k(t:t-s))=(\alpha_i^{\mathrm{T}})\overline{\boldsymbol{K}}_k$.

⑧ Combining the reference value of the non-fault condition, calculate the new data statistics, check whether the $T^2_{\mathrm{s,c},i}$, $T^2_{\mathrm{s,b},i}$ or $\mathrm{SPE}_{\mathrm{s,c},i}$, $\mathrm{SPE}_{\mathrm{s,b},i}$ statistics exceed the control limit $T^2_{\mathrm{s,c},i,\mathrm{lim}}$, $T^2_{\mathrm{s,b},i,\mathrm{lim}}$, $\mathrm{SPE}_{\mathrm{s,c},i,\mathrm{lim}}$, $\mathrm{SPE}_{\mathrm{s,b},i,\mathrm{lim}}$. If one of the control limit is exceeded, it indicates that the segment has a fault condition; otherwise, it is identified as normal, and then return to step ① to continue to detect a new set of process datasets.

(3) Cycle merge part

① $T^2_{\mathrm{s,c},i}$, $T^2_{\mathrm{s,b},i}$ or $\mathrm{SPE}_{\mathrm{s,c},i}$, $\mathrm{SPE}_{\mathrm{s,b},i}$ statistics of L segments is obtained online.

② Starting from the number of merged sequences $i=2$, merge adjacent sequences and calculate the merged fitting error err.

③ Judge whether it exceeds the threshold δ_{e}; if yes, then $i=i+1$; go to step ④. if not, set $L=L-1$ and merge into the next group until err is greater than δ_{e}.

④ Whenerris greater than the threshold δ_{e}, it is further judged whether the current merging end point is the all data end point e_{L}, and if it is not reached, return to the step ③ cycle merge.

⑤ When the sequence is completed and all groups are equal merged, obtain $T^2_{\mathrm{s,c}}$, $T^2_{\mathrm{s,b}}$, $\mathrm{SPE}_{\mathrm{s,c}}$, $\mathrm{SPE}_{\mathrm{s,b}}$ and $T^2_{\mathrm{s,c,lim}}$, $T^2_{\mathrm{s,b,lim}}$, $\mathrm{SPE}_{\mathrm{s,c,lim}}$, $\mathrm{SPE}_{\mathrm{s,b,lim}}$, the value of i should be 1 greater than L, and then the merge is completed. Determine whether the total statistics exceed the control limit or exceed the control limit, and then perform variable identification and analysis. If it does not exceed, it is normal. Detect the next set of process sequence.

(4) Variable identification and cause analysis

(1) After the merged part, the $T^2_{\mathrm{s,c}}$, $T^2_{\mathrm{s,b}}$ or $\mathrm{SPE}_{\mathrm{s,c}}$, $\mathrm{SPE}_{\mathrm{s,b}}$ statistical information detection exceeds the limit, and variable identification is performed.

(2) According to the principle of variable reconstruction, statistical reconstruction indicators are determined, and the contribution rate $\psi(z_k)$ is calculated.

(3) According to the variable reconstruction and the contribution value of the detection result, when the detection result shows a fault, the reconstruction contribution rate and the contribution index of the cut variable are calculated to determine the cause of the fault and the path of the fault.

3. Case Studies of Continuous and Batch Processes

In this paper, the TE process and the penicillin fermentation process are selected to represent the continuous chemical process and the batch process, and the proposed fault diagnosis model is applied to the two chemical processes. Judge the validity of the proposed model.

3.1. Continuous process fault diagnosis

TE process was created by Downs and Vogel in 1993 [33] and is widely cited as a benchmark for studies in control and fault diagnosis. The flow chart of the TE benchmark process is presented in Fig. 4 [34].

The TE benchmark process consists of five operating units, including the reactor, product condenser, vapor liquid separator, recycle compressor, and product stripper, including 11 operating variables and 41 measured variables. In addition, the platform has 21 different types of faults, and its data are nonlinear, strong coupling, time varying, *etc*. The dataset of the TE process is divided into training data and test data. Both the training set and the test set contain 52 observation variables and 960 observation values, respectively, of which the first 160 in the test dataset are under normal working conditions, and the data from 161 to 960 are fault data. We take the TE process as the experimental object to evaluate the proposed algorithm. The proposed algorithm is devoted to chemical industry processes. TE benchmark process faults are shown in Table 1.

To verify the superiority of the CTA-DKPCA model for fault detection, the CTA-DKPCA model is used to analyze the causes of faults under different fault types.

3.1.1. Fault detection results

The number of PCs selected in this paper is calculated by cumulative variance percentage (CPV) in this paper. The calculation of CPV is shown in Eq. (31).

$$\mathrm{CPV} = \frac{\sum_{i=1}^{N}\lambda_i}{\sum_{i=1}^{j}\lambda_j} \tag{31}$$

where λ is the characteristic value of covariance, j is the number of variables, and N is the number of principal components. The number of PCs in the final selection is shown in Fig. 5.

The number of PCA and its derived algorithms is determined according to the cumulative method percentage greater than 85%. Finally, the number of PCA, KPCA, DKPCA and CTA-DKPCA is 9, 17, 22 and 12 respectively. In the calculation of the CTA, the relationship between the number of cycles and the error is shown in Fig. 6.

Table 1
Faults for the Tennessee Eastman process

Fault	Description	Type
01	A/C feed ratio, B composition constant (Stream4)	Step
02	B composition, A/C ratio constant (Stream4)	Step
03	D feed temperature (Stream2)	Step
04	Reactor cooling water inlet temperature	Step
05	Condenser cooling water inlet temperature	Step
06	A feed loss (Stream1)	Step
07	C header pressure loss-reduced availability (Stream4)	Step
08	A, B, C feed composition (Stream4)	Random
09	D feed temperature (Stream2)	Random
10	C feed temperature (Stream4)	Random
11	Reactor cooling water inlet temperature	Random
12	Condenser cooling water inlet temperature	Random
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16–20	Unknown	Unknown
21	Valve position constant (Stream4)	Constant position

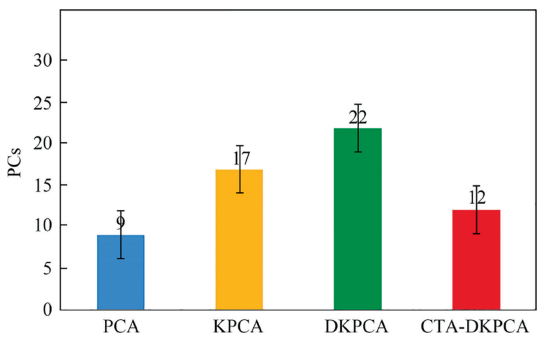

Fig. 5. The number of PCs calculated by different method.

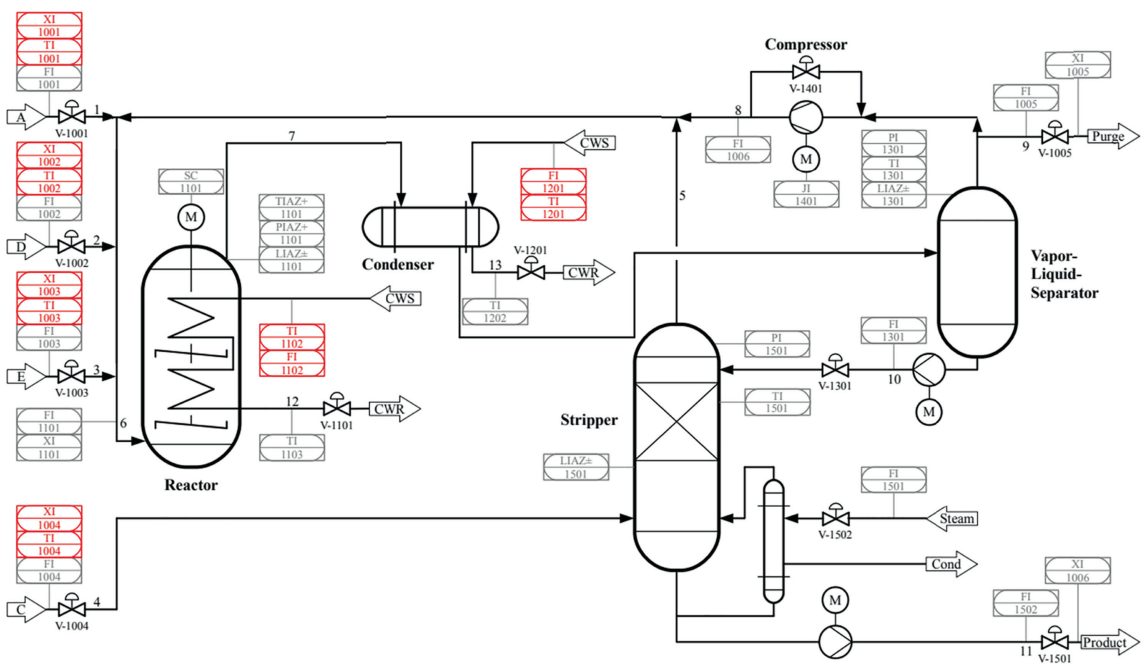

Fig. 4. TE benchmark process.

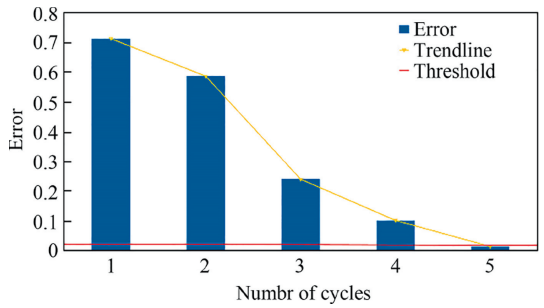

Fig. 6. The relationship between the number of cycles and the error.

It can be seen that for the TE process loop 5 times, 32 data segments are divided, and the error reaches the threshold requirement. Then, we compare the monitoring performance of the CTA-DKPCA algorithm with the PCA, KPCA and DKPCA algorithms. The normal samples are used to train the monitoring model. The kernel width of the radial basis function is set to 800, and the confidence is set to 95%. This paper uses the fault detection rate (FDR) to test all 18 faults in the TE process.

Table 2 lists the fault detection rate results of these 18 faults. It can be seen from the remaining 18 faults that CTA-DKPCA shows better fault detection performance under most faults.

The CTA-DKPCA method has a higher detection accuracy for most faults than the other PCA methods, In the T^2 subspace, FDR for faults 4, 5, 8, 10, and 21 reaches 100%, In the SPE subspace, FDR for faults 4, 5, 7, 18, 19 and 20 reaches 100%. Under other faults, the CTA-DKPCA fault detection algorithm proposed in this paper also performs better than other algorithms. This indicates that the CTA-DKPCA method is very sensitive to fault.

To further quantify the monitoring performance, two indicators are introduced, the fault alarm rate (FAR) and the time delay (TD). Calculate as follows:

$$\mathrm{FAR} = P(\text{control chart} > \text{Threshold}|F = 0) \times 100\% \tag{32}$$

$$\mathrm{TD} = t_\mathrm{d} - t_0 \tag{33}$$

where t_d is the fault detection time and t_0 is the fault occurrence time.

The FARs of T^2 and SPE for fault detection using the CTA-DKPCA method in the normal state of the TE process are 0.0012 and 0, respectively, indicating that the proposed CTA-DKPCA method has very low FAR.

The CTA-DKPCA method is compared with other fault detection methods, and the results are shown in Table 3.

The comparison with 2-class SVM method shows that the two methods are very similar in FDR. In terms of the two statistics of FAR and TD, the CTA-DKPCA method is better than the 2-class SVM under most fault conditions. Especially in fault 10, 13 and 18, the method proposed in this paper is far superior to the 2-class SVM method. The CTA-DKPCA algorithm solves the redundancy problem caused by the fault calculation process, which accelerates the calculation ability of the algorithm and improves the detection accuracy.

Select three different types of faults 7, 8, and 14 to show the detection effect,

Fault 7 is the pressure loss of material C, which is related to the step change; Fault 8 is composed of A, B and C, which is related to random changes. Fault 14 is the fault of the reactor cooling water valve, which is related to the check valve. They are representative to a certain extent. Fig. 7 shows the experimental results of the four algorithms corresponding to these three faults. The dotted lines corresponding to different colors indicate their control limits.

Based on the above experimental results, the following conclusions are drawn. Under the three fault conditions, the CTA-DKPCA algorithm is better than the other three methods. It is difficult for PCA to detect the fault correctly. KPCA introduces the nonlinear mapping of the kernel function to capture the nonlinear structure in the data, which has good separation abilities and good monitoring performance. However, due to the lack of dynamic data performance, DKPCA combines the advantages of KPCA and DPCA, which can express the dynamics of data through an augmented matrix, and the kernel function can also monitor the nonlinear process, but the problem of computational redundancy in the later stage of data calculation is larger and slightly weaker. CTA improves the DKPCA, uses temporal logic to perform cycle segmentation of data, eliminates the calculation redundancy caused by the large amount of data, and greatly improves the calculation accuracy and speed.

3.1.2. Fault diagnosis results

When a fault is detected, the RBC method described above is used to further diagnose the detected fault and determine the root cause of the fault to determine the cause and path of the overall

Table 2
Comparison of the FDR (%) of different methods in the TE process

Fault	PCA		KPCA		DKPCA		CTA-DKPCA	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
01	31.0	41.0	83.0	81.0	91.0	93.0	99.2	99.0
02	33.0	76.0	90.0	71.0	92.0	93.0	98.5	99.0
04	45.0	61.0	75.0	72.0	88.0	77.0	100.0	100.0
05	21.0	28.0	80.0	88.0	90.0	96.0	100.0	100.0
06	48.0	35.0	77.0	70.0	91.0	98.0	98.4	99.0
07	30.0	41.0	75.0	44.0	92.0	88.0	99.1	100.0
08	52.0	31.0	72.0	68.0	94.0	94.0	100.0	97.5
10	33.0	33.0	68.0	70.0	93.0	97.0	100.0	99.0
11	45.0	64.0	81.0	66.0	88.0	94.0	94.0	96.0
12	29.0	57.0	59.0	74.0	94.0	92.0	99.0	99.9
13	60.0	73.0	75.0	88.0	95.0	93.0	99.0	96.0
14	48.0	49.0	55.0	52.0	93.0	95.0	99.5	98.8
16	41.0	41.0	66.0	77.0	90.0	85.0	97.2	99.0
17	16.0	35.0	71.0	82.0	95.0	95.0	98.9	98.0
18	76.0	61.0	81.0	84.0	91.0	96.0	99.5	100.0
19	17.0	28.0	52.0	72.0	85.0	95.0	98.5	100.0
20	33.0	40.0	70.0	66.0	93.0	93.0	98.5	100.0
21	39.0	56.0	51.0	62.0	80.0	72.0	100.0	97.0
Average	38.7	47.2	71.2	71.5	90.8	91.5	98.63	98.74

Table 3
Comparisons with the Onel *et al.* (2-Class SVM) method

Fault	Onel *et al.* 2019 (2-Class SVM)			This study (CTA-DKPCA)		
	FDR/%	FAR/%	TD/s	FDR/%	FAR/%	TD/s
01	99.9	0.0	6.0	99.0	0.0	0.0
02	97.8	0.0	57.0	99.0	0.0	0.0
04	100.0	0.0	3.0	100.0	0.0	3.0
05	99.9	0.0	6.0	100.0	0.0	4.0
06	100.0	0.0	3.0	99.0	0.0	3.0
07	100.0	0.0	3.0	100.0	0.0	0.0
08	95.8	4.3	60.0	97.5	3.5	10.0
10	85.8	14.3	12.0	99.0	15.2	6.0
11	96.6	3.4	3.0	96.0	0.0	0.0
12	100.0	0.0	3.0	99.9	0.0	0.0
13	91.9	8.1	153.0	96.0	3.6	30.0
14	100.0	0.0	3.0	98.8	0.0	3.0
16	96.9	3.1	3.0	99.0	3.5	4.0
17	92.9	7.1	72.0	98.0	8.0	3.0
18	90.0	10.0	231.0	100.0	5.2	4.0
19	88.5	11.5	3.0	100.0	6.2	3.0
20	85.0	15.0	45.0	100.0	16.1	5.0
21	100.0	0.0	3.0	97.0	0.0	0.0

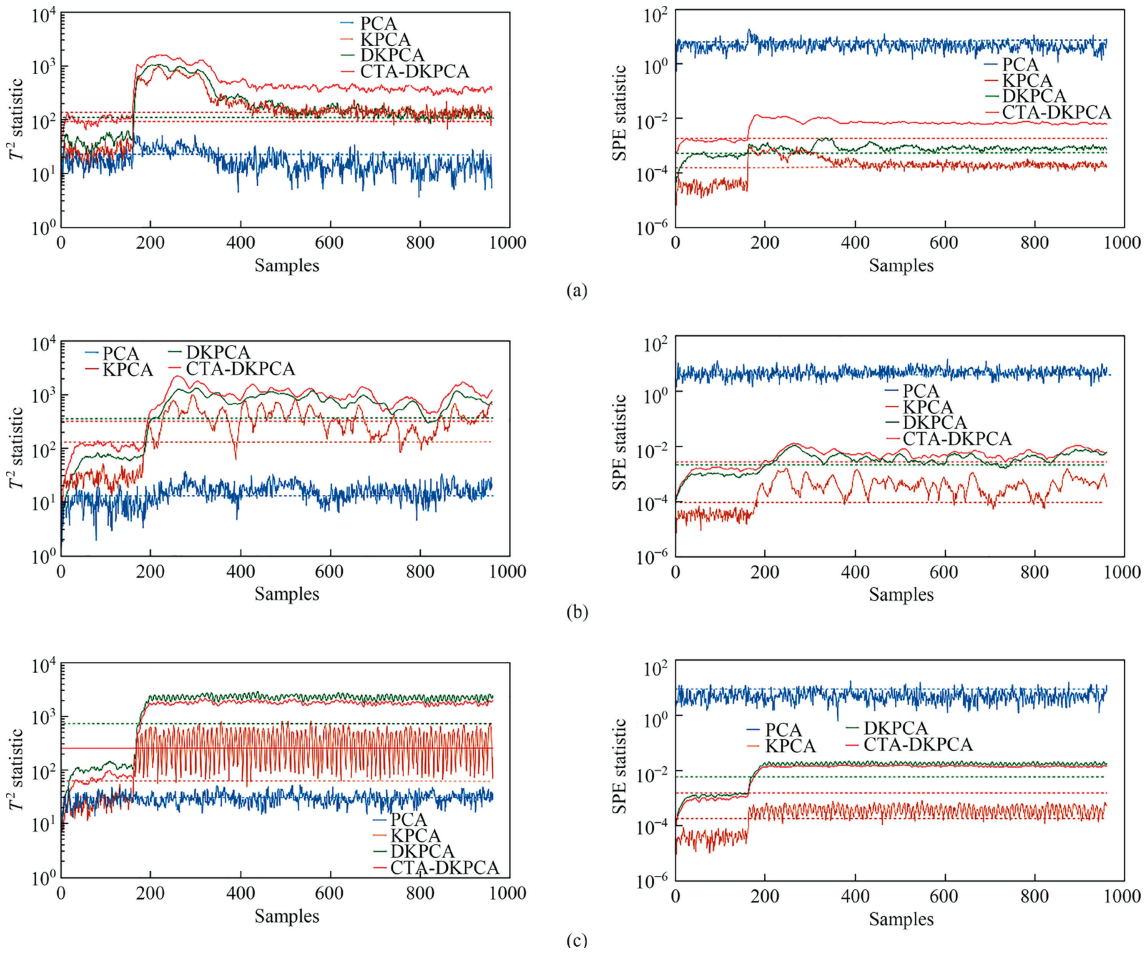

Fig. 7. The test results of four methods for three types of faults (a – Fault 7, b – Fault 8, c – Fault 14).

fault. For the TE process, 18 faults were diagnosed using the RBC method, and the root variable corresponding to each fault was obtained as shown in Table 4.

For different types of faults in the TE process, we are consistent with the three types of faults selected in the detection stage above. The three types of fault diagnosis results are shown in Fig. 8. Wherein a fault is shown in blue part of the contribution rate exceeds the average fault, the black line represents the normal range of fault contribution rate.

It can be shown from Fig. 8, the main variables that cause fault 7 are variable 45 (total feed volume), variable 7 (reactor pressure), and variable 13 (product separator pressure). From this, the cause

Table 4
The root fault variables of 18 faults in the TE process

Fault	Number of root fault variables	Root fault variables
1	2	16, 44
2	4	10, 28, 34, 41
4	1	51
5	4	11, 18, 50, 52
6	2	44, 1
7	3	45, 7, 13
8	5	7, 13, 16, 20, 47
10	4	50, 18, 19, 20
11	2	51, 9
12	7	50, 7, 11, 13, 16, 20, 38
13	7	7, 13, 18, 19, 20, 38, 51
14	3	51, 9, 21
16	2	18, 19, 50
17	1	21
18	7	7, 13, 16, 20, 43, 46, 51
19	2	46, 5
20	3	46, 20, 13
21	1	45

of fault 7 can be clarified as when the total feed flow rate on stream 4 changes, the process variable reactor pressure and the product separation pressure are measured separately. So, to compensate for the reduced C header pressure, the total feed flow rate is increased by adjusting the flow valve on stream 4, which in turn will affect the reactor pressure and product separation pressure in the process. The changes in each variable are shown in Fig. 9.

For fault 8, the diagnosis result shows that the main variable that causes fault 8 is process variable 47 (percentage of drain valve). When the drain valve discharge fails, the feed composition of A, B, C is disordered. The measured variables 7 (reactor pressure) and 13 (product separator pressure) will be affected, and then variable 16 (stripper pressure) will also be abnormal, and finally variable 20 (compression power) will be abnormal. Will be affected. The main variable changes are shown in Fig. 10.

The analysis of Table 4 shows that the key fault variables of fault 4, fault 11, and fault 14 overlap. Therefore, the three faults are distinguished separately.

For fault 4, the variable that caused the fault is process variable 51, which is the reactor temperature. The other variables still show a normal status. Therefore, if the diagnosis result shows that only control variable 51 is faulty, then fault 4 is due to a step increase in the temperature of the condensate water inlet of the reactor, and other variables in the reactor do not respond, so the controller responds first. The condensate flow of the condenser of the balance system rises rapidly to cool down the reactor, thereby diagnosing the fault. The cause of fault 11 is a random fault at the inlet temperature of the reactor, and the detection time of the fault is slower than that of fault 4. While the condenser cooling water flow increased, the reactor temperature also increased slightly. The way that fault 14 occurs is as follows. If the temperature of the reactor cooling water increases, thereby losing the cooling effect on the reactor, we will observe a direct increase in the reactor temperature (measurement variable 9). Then, the controller attempts to reduce the elevated reactor temperature by increasing the flow of the condenser cooling water (process variable 51). On the other hand, if the temperature of the reactor cooling water is reduced due to the valve being stuck, thereby increasing the cooling effect on the reactor, we will notice a drop in the reactor temperature. In this case, the controller will reduce the condenser cooling water flow to achieve equilibrium. In addition, the third key process variable is the measured measurement variable 21, that is, the reactor

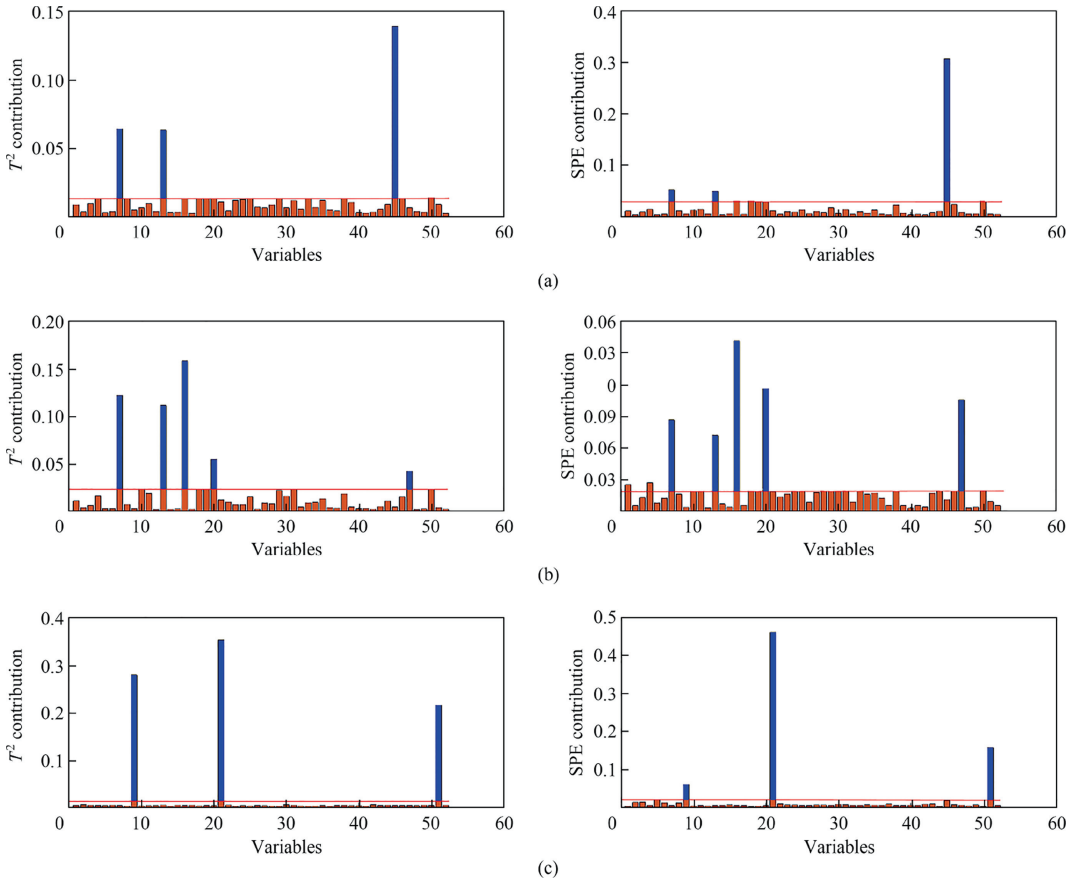

Fig. 8. Three types of fault diagnosis results (a – fault 7, b – fault 8, c – fault 14).

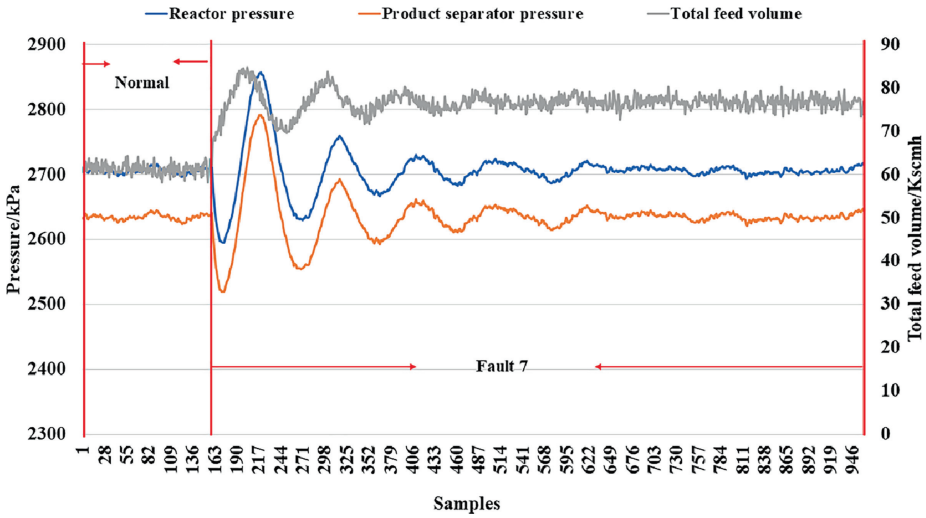

Fig. 9. Plot of the root cause variables (fault 7).

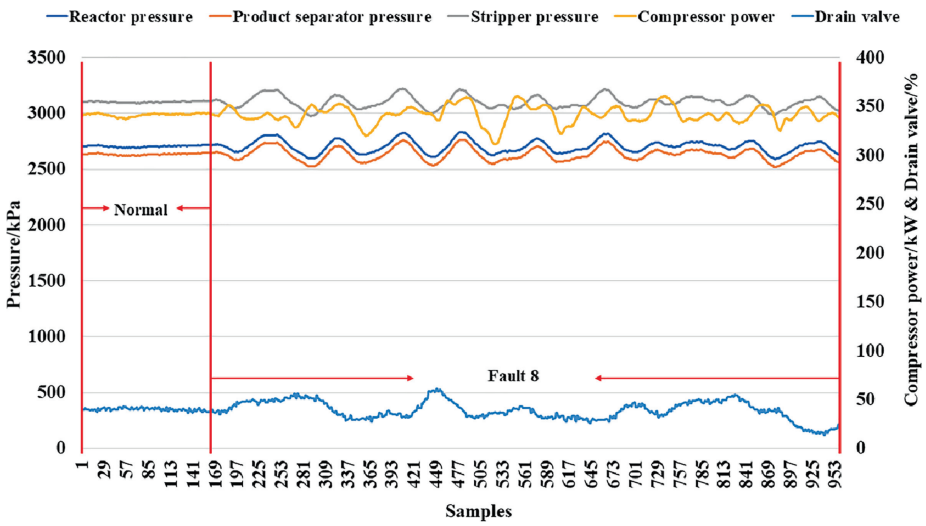

Fig. 10. Plot of the root cause variables (fault 8).

cooling water outlet temperature, which is directly affected by the viscous reactor cooling water valve. Therefore, the difference between the three types of faults lies in the propagation time of the fault caused by the fault type. The diagnosis result of the fault variable can intuitively judge the cause of the fault. The influence of the fault variables of faults 4, 11, and 14 is shown in Fig. 11 (a)–(c), respectively.

From the above diagnosis results, it can be seen that for the 21 types of faults in the TE process, the RBC method can more accurately locate the process variables (control) that cause the fault and its related measurement variables to determine the cause of the fault and clarify the path of the fault through variable reconstruction. The method reduces the autocorrelation between variables, improves the cross-correlation of related variables, and achieves the purpose of locating the fault location in time.

3.2. Batch process fault diagnosis

Different fault conditions of the penicillin fermentation process are used to judge the effectiveness of the proposed CTA-MDKPCA method. In the process simulation, the microorganisms were grown in a batch operation within the first 40 h to achieve high cell density. Then, the fermenter was switched to the fed-batch mode, with a continuous feed of glucose as the substrate to maintain a high level. The growth rate of biomass promotes the synthesis of penicillin. The target product, penicillin, is also a secondary metabolite in the process, which is mainly secreted in the fed-batch stage. This process is characterized by nonlinear dynamics and multiple operating stages. The process flow diagram of the fed-batch penicillin fermentation process is shown in Fig. 12 [35].

In this study, the observation data of the 16 measurement variables given in Table 5 were collected for batch process fault diagnosis.

The sampling time used is 0.1 h, and the total duration of each batch is 400 h, including the batch and fed-batch stages. The normal distribution of the measured variables during the fermentation process is shown in Fig. 13.

In our proposed method, only a complete batch of benchmark data is required, and the monitored data are checked for online fault detection. When the penicillin fermentation process is carried out for 80 h, the actual fault amplitude is 30.

The batch process is divided into cycles first, and the relationship between the number of cycles and the error is shown in the Fig. 14.

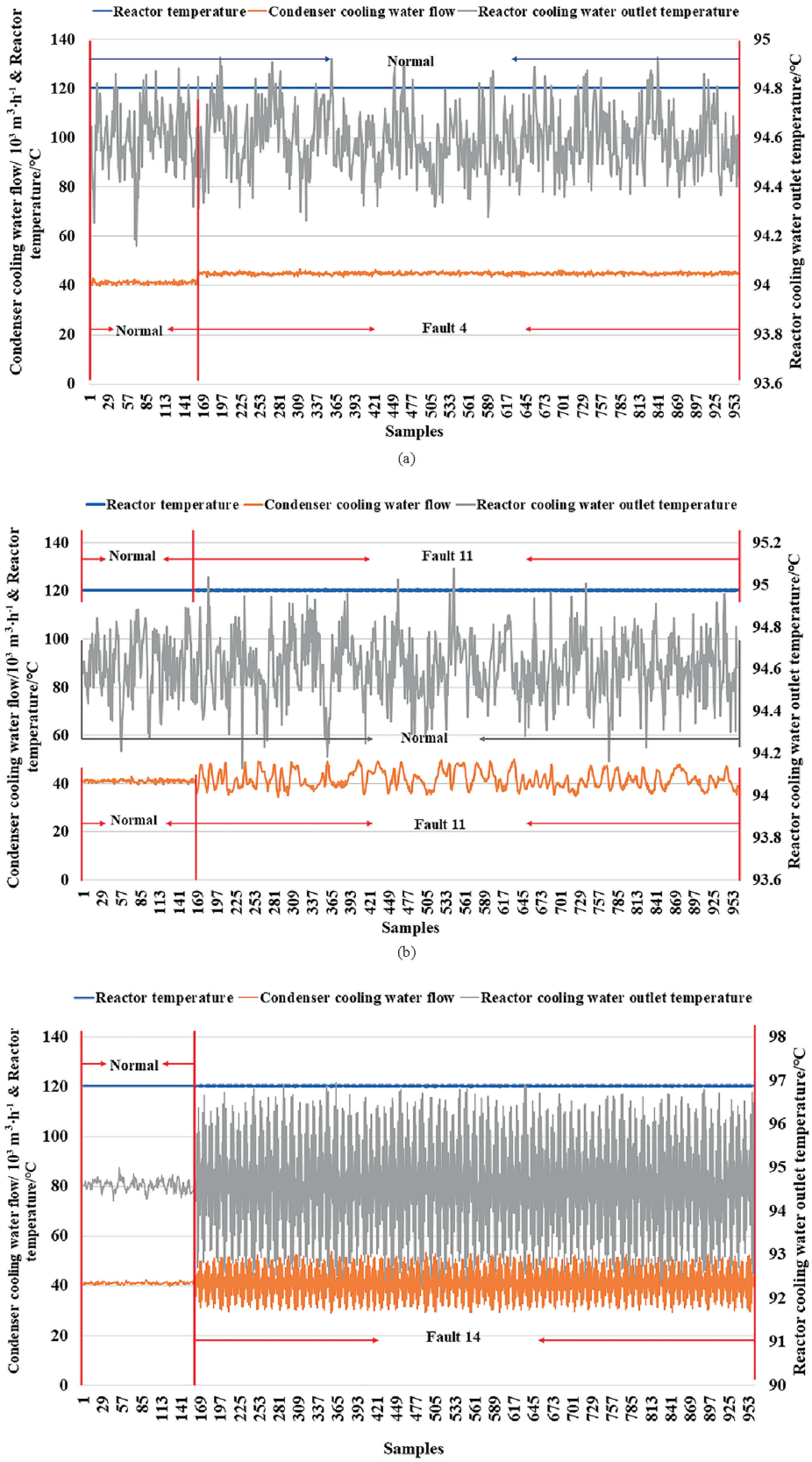

Fig. 11. The root cause of fault 4, fault 11 and fault 14 (a – fault 4, b – fault 11, c – fault 14).

This batch process's cycle segmentation part is divided 7 times, divided into 128 data segments, the error reached below the error threshold in the 7th cycle. Input the segmented data into the MDKPCA model. The number of selected principal components is 6, which is less than the 8 principal components of MDKPCA. More fault information can be

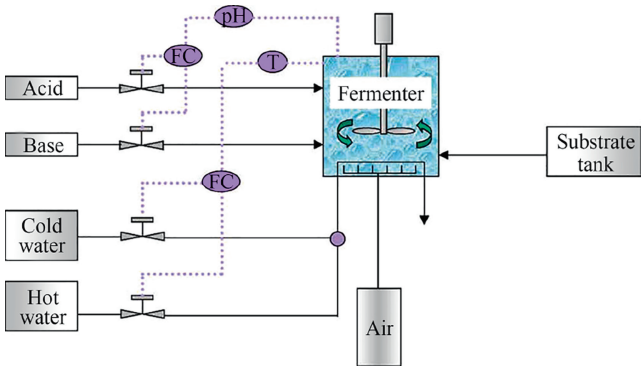

Fig. 12. The process flow diagram of fed-batch penicillin fermentation.

Table 5
Monitoring variables of the fed-batch penicillin fermentation process

Variable number	Description
1	Aeration rate/L·h^{-1}
2	Agitator power/W
3	Substrate feed rate/L·h^{-1}
4	Substrate feed temperature/K
5	Substrate concentration/g·L^{-1}
6	PH
7	Dissolved oxygen concentration/g·L^{-1}
8	Carbon dioxide concentration/g·L^{-1}
9	Biomass concentration/g·L^{-1}
10	Penicillin concentration/g·L^{-1}
11	Fermenter temperature/K
12	Cooling water flow rate/L·h^{-1}
13	Generated heat/kcal
14	Acid flow rate/L·h^{-1}
15	Base flow rate/L·h^{-1}
16	Culture volume/L

obtained, and subsequent fault detection results will be more accurate.

Then, we make a comparison with the MKPCA and MDKPCA methods for FDR, FAR and TD. The comparison results are shown in Table 6.

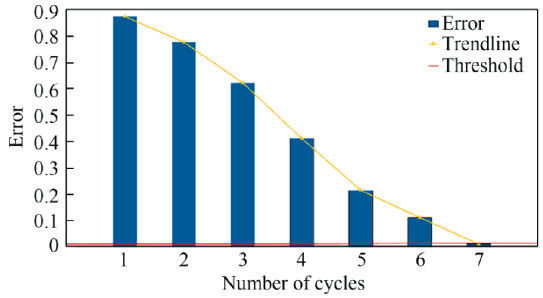

Fig. 14. The relationship between the number of cycles and the error.

Table 6
The comparison results of CTA-MDKPCA, MDKPCA and MKPCA

Fault	MKPCA			MDKPCA			CTA-MDKPCA		
	FDR	FAR	TD	FDR	FAR	TD	FDR	FAR	TD
1	71.0%	16.5%	12 s	91.5%	10.4%	8 s	99.2%	4.2%	2 s

The comparison with MDKPCA and MKPCA method shows that the batch process fault diagnosis model proposed in this paper outperforms the MDKPCA and MKPCA methods in FDR, FAR and TD statistics. Among them, the CTA-MDKPCA method's FDR is 99.2%, and TD is 2 s, which proves that the CTA-MDKPCA method has greatly improved the accuracy and speed of fault detection. In addition, in terms of FAR, it is also superior to MDKPC and MKPCA algorithms, which proves that the algorithm proposed in this paper is a comprehensive optimization.

The detection results are shown in Fig. 15.

The fault detection results show that the MKPCA method is not sensitive to faults. The fault situation introduced in this article is 80 h, but the MKPCA method has 60 h and 14 h response times in the principal component space and the residual subspace after the fault occurs, which is extremely large delayed the implementation of subsequent fault remedial measures. The fault detection results of the MDKPCA method show that although the MDKPCA method

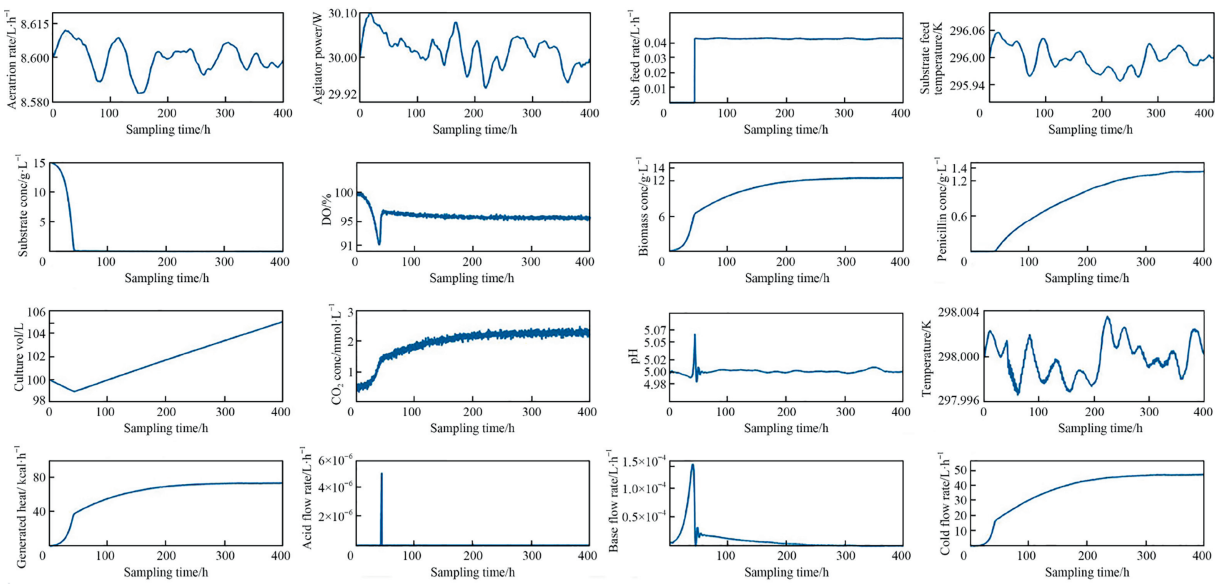

Fig. 13. Normal profiles of the monitored variables in the fed-batch penicillin fermentation process.

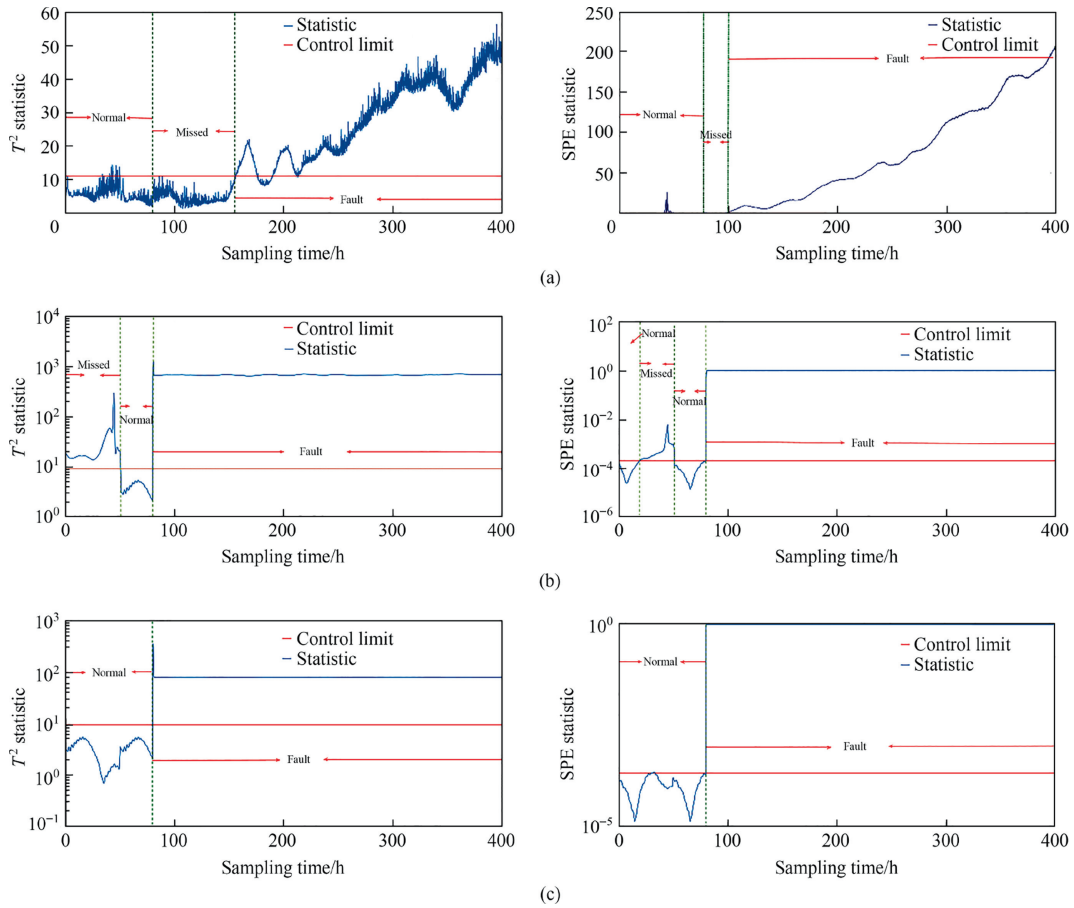

Fig. 15. The CTA-MDKPCA and MKPCA test comparison results (a – MKPCA, b – MDKPCA, c – CTA-MDKPCA).

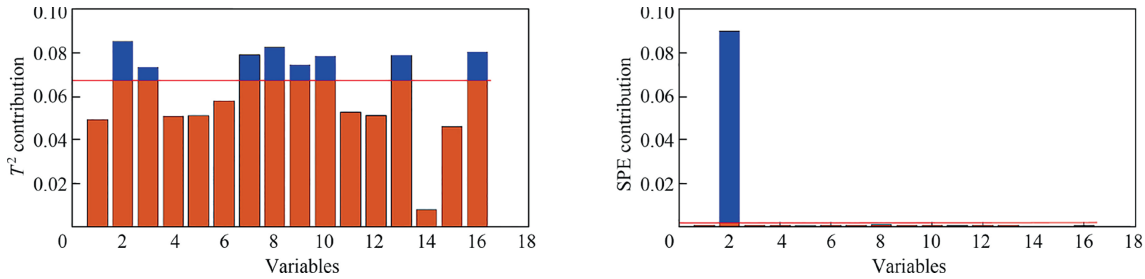

Fig. 16. The percentage of fault contribution of each variable in the batch process.

can respond to the fault at the point of fault, it may be affected by early data fluctuations, so it may cause misjudgments when detecting early faults. However, in the CTA-DKCPCA method, when a fault occurs, the principal component space and the residual subspace respond at the same time, and the occurrence of the fault can be detected quickly and accurately.

After detecting the fault of the batch process, the contribution of each variable is calculated. The diagnosis result is shown in Fig. 16.

The result of the contribution rate of the current fault shows that the stirring power of Variable 2 is increased by 80% of the fault range. From the SPE variable identification, the main error occurs in Variable 2; but for the T^2 identification result, the average fault contribution rate should be 6.25%. The substrate feed flow rate increases, thereby increasing the cell concentration, product concentration and reactor volume, which increases the fermentation reaction rate; and as the carbon dioxide concentration increases, the accumulated heat in the fermenter increases due to an overreaction, and finally, the cooling water flow rate increases to keep the system balanced.

For the penicillin fermentation process, CTA-MDKPCA can accurately detect faults in the batch process. The identification of two statistics through the RBC model shows the root cause of the fault, and the path of occurrence and propagation of the fault.

4. Conclusions

This paper proposes a CTA to improve the performance of chemical process fault diagnosis. Through the cycle segmentation and cycle merge process, the effective information is retained to the greatest extent and the calculation dimension of the matrix is reduced. The DKPCA and MDKPCA algorithms are embedded in the CTA and applied to the chemical continuous and batch process fault detection model. Improve the fault detection rate while reducing the fault time delay. When a fault is detected, it is input into the RBC model, which will determine the percentage of each variable's contribution to the fault based on the principle of reconstruction contribution, comprehensively judge the cause of the fault, locate the fault variable and determine the fault path.

For continuous processes, to determine the advantages of the CTA-DKPCA model in the detection effect of the continuous process, the FDR of 18 faults in the TE process was compared with PCA, KPCA and DKPCA. The results show that in the T^2 and SPE statistics, the average FDR in the TE process of the CTA-DKPCA method is 98.63% and 98.74%, respectively. Comparing the CTA-DKPCA algorithm with the 2-class SVM algorithm, it can be seen that the CTA-DKPCA algorithm is better than the 2-class SVM algorithm in FDR, FAR and TD, which shows that CTA-DKPCA solves the computational redundancy of the large-scale data problem and greatly improves the calculation speed and accuracy. Finally, three different types of faults 7, 8 and 14 were selected for comparison. CTA-DKPCA has better curve smoothness and detection effects than PCA and its derived algorithms. Then, the fault of the TE process are brought into the RBC model for diagnosis, and the root cause fault variables of each fault are obtained. Three types of faults consistent with the detection results are selected for diagnosis. In addition, the cause of the fault is also distinguished for the faults that contain the relationship of the fault variables. The results show that the CTA-DKPCA model and the RBC model for actual continuous chemical process faults have a high detection and identification effect. The calculation of the model can detect and determine the fault and its cause in time.

The CTA-MDKPCA model is applied to the fed-batch penicillin fermentation process; 16 variables are selected as the main variables, and a fault condition is set for diagnosis. The results show that the CTA-MDKPCA model can accurately detect fault and that the RBC model determines the cause and propagation path of the fault. Under fault conditions, the CTA-MDKPCA model is compared with the MKPCA and MDKPCA methods in terms of FDR, FAR and TD. The FDR, FAR and TD of the CTA-MDKPCA model reach 99.2%, 4.2% and 2 s, respectively. The advantages of the proposed model in fault detection are shown. In a variation from the continuous process and due to the multistage operation in the batch process, the recognition of the principal component space will not have a single variable exceeding 50%; this outcome will be supplemented in the next stage to achieve the existing balance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors gratefully acknowledge financial support from the National Natural Science Foundation of China (21706220).

References

[1] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, *Chemom. Intell. Lab. Syst.* 171 (2017) 16–25.
[2] S. Joe Qin, Statistical process monitoring: Basics and beyond, *J. Chemometrics* 17 (8–9) (2003) 480–502.
[3] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, *Ind. Eng. Chem. Res.* 52 (10) (2013) 3543–3562.
[4] Z.Q. Wang, L. Wang, Z.H. Yuan, B.Z. Chen, Data-driven optimal operation of the industrial methanol to olefin process based on relevance vector machine, *Chin. J. Chem. Eng.* 34 (2020) 106–115.
[5] H. Han, S. Zhu, J. Qiao, M. Guo, Data-driven intelligent monitoring system for key variables in wastewater treatment process, *Chin. J. Chem. Eng.* 26 (10) (2018) 2093–2101.
[6] M. Nawaz, A.S. Maulud, H. Zabiri, S.A.A. Taqvi, A. Idris, Improved process monitoring using the CUSUM and EWMA-based multiscale PCA fault detection framework, *Chin. J. Chem. Eng.* 29 (2021) 253–265.
[7] G. Wang, J. Liu, Y. Li, C. Zhang, Fault diagnosis of chemical processes based on partitioning PCA and variable reasoning strategy, *Chin. J. Chem. Eng.* 24 (7) (2016) 869–880.
[8] Y. Li, D.S. Yang, Local component based PCA model for multimode process monitoring, *Chin. J. Chem. Eng.* 34 (2020) 116–124.
[9] K. Liu, X. Jin, Z. Fei, J. Liang, Adaptive partitioning PCA model for improving fault detection and isolation, *Chin. J. Chem. Eng.* 23 (6) (2015) 981–991.
[10] W. Ku, R.H. Storer, C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis, *Chemom. Intell. Lab. Syst.* 30 (1) (1995) 179–196.
[11] J.-M. Lee, C. Yoo, S.W. Choi, P.A. Vanrolleghem, I.-B. Lee, Nonlinear process monitoring using kernel principal component analysis, *Chem. Eng. Sci.* 59 (1) (2004) 223–234.
[12] Q. Yang, F. Tian, D.Z. Wang, Nonlinear dynamic process monitoring based on lifting wavelets and dynamic kernel PCA, IEEE, Jinan, China, 2010, pp. 5712–5716.
[13] C. Zhao, F. Wang, F. Gao, N. Lu, M. Jia, Adaptive monitoring method for batch processes based on phase dissimilarity updating with limited modeling data, *Ind. Eng. Chem. Res.* 46 (14) (2007) 4943–4953.
[14] J.-M. Lee, C. Yoo, I.-B. Lee, On-line batch process monitoring using a consecutively updated multiway principal component analysis model, *Comput. Chem. Eng.* 27 (12) (2003) 1903–1912.
[15] J.-M. Lee, C. Yoo, I.-B. Lee, Fault detection of batch processes using multiway kernel principal component analysis, *Comput. Chem. Eng.* 28 (9) (2004) 1837–1847.
[16] Y.S. Ng, R. Srinivasan, An adjoined multi-model approach for monitoring batch and transient operations, *Comput. Chem. Eng.* 33 (4) (2009) 887–902.
[17] Q. Chen, U. Kruger, M. Meronk, A.Y.T. Leung, Synthesis of T^2 and Q statistics for process monitoring, *Control. Eng. Pract.* 12 (6) (2004) 745–755.
[18] Q.-X. Zhu, Y. Luo, Y.-L. He, Novel distributed alarm visual analysis using multicorrelation block-based PLS and its application to online root cause analysis, *Ind. Eng. Chem. Res.* 58 (45) (2019) 20655–20666.
[19] C.F. Alcala, S.J. Qin, Reconstruction-based contribution for process monitoring with kernel principal component analysis, *Ind. Eng. Chem. Res.* 49 (17) (2010) 7849–7857.
[20] H. Sun, S. Zhang, C. Zhao, F. Gao, A sparse reconstruction strategy for online fault diagnosis in nonstationary processes with No a priori fault information, *Ind. Eng. Chem. Res.* 56 (24) (2017) 6993–7008.
[21] M. Onel, C.A. Kieslich, Y.A. Guzman, C.A. Floudas, E.N. Pistikopoulos, Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection, *Comput. Chem. Eng.* 115 (2018) 46–63.
[22] H. Wang, Y. Chen, A robust fault detection and diagnosis strategy for multiple faults of VAV air handling units, *Energy Build.* 127 (2016) 442–451.
[23] C. Xu, S. Zhao, F. Liu, Sensor fault detection and diagnosis in the presence of outliers, *Neurocomputing* 349 (2019) 156–163.
[24] R. Srinivasan, M. Sheng Qian, Online fault diagnosis and state identification during process transitions using dynamic locus analysis, *Chem. Eng. Sci.* 61 (18) (2006) 6109–6132.
[25] Y. Dai, J. Zhao, Fault diagnosis of batch chemical processes using a dynamic time warping (DTW)-based artificial immune system, *Ind. Eng. Chem. Res.* 50 (8) (2011) 4534–4544.
[26] Y. Dai, Y. Qiu, Z. Feng, Research on faulty antibody library of dynamic artificial immune system for fault diagnosis of chemical process (Book Chapter), *Comput. Aided Chem. Eng.* 44 (2018) 493–498.
[27] G.E. Fainekos, G.J. Pappas, Robustness of temporal logic specifications for continuous-time signals, *Theor. Comput. Sci.* 410 (42) (2009) 4262–4291.
[28] M. Reynolds, Metric temporal logic revisited, *Acta Inf.* 53 (3) (2016) 301–324.
[29] V. Raman, A. Donzé, D. Sadigh, R.M. Murray, S.A. Seshia, Reactive synthesis from signal temporal logic specifications, In: International Conference on Hybrid Systems: Computation and Control, 2015.
[30] M. Onel, C.A. Kieslich, E.N. Pistikopoulos, A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the Tennessee Eastman process, *AIChE J.* 65 (3) (2019) 992–1005.

[31] A. Kumar, A. Bhattacharya, J. Flores-Cerrillo, Data-driven process monitoring and fault analysis of reformer units in hydrogen plants: Industrial application and perspectives, *Comput. Chem. Eng.* 136 (2020) 106756.
[32] M.R. Maurya, R. Rengaswamy, V. Venkatasubramanian, Fault diagnosis using dynamic trend analysis: A review and recent developments, *Eng. Appl. Artificial Intell.: Int. J. Intell. Real-Time Autom.* 20 (2007) 133–146.
[33] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, *Comput. Chem. Eng.* 17 (3) (1993) 245–255.
[34] H. Wu, J.S. Zhao, Fault detection and diagnosis based on transfer learning for multimode chemical processes, *Comput. Chem. Eng.* 135 (2020) 106731.
[35] M.M. Rashid, J. Yu, Nonlinear and non-Gaussian dynamic batch process monitoring using a new multiway kernel independent component analysis and multidimensional mutual information based dissimilarity approach, *Ind. Eng. Chem. Res.* 51 (33) (2012) 10910–10920.