IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 7, 2019, accepted January 19, 2019, date of publication January 25, 2019, date of current version September 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2895120

Phase Partition and Fault Diagnosis of Batch Process Based on KECA Angular Similarity

CHANG PENG, QIAO JUNFEI, (Member, IEEE), ZHANG XIANGYU, AND LU RUIWEI
Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China

Corresponding author: Chang Peng (changpeng@ bjut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61174109.

ABSTRACT Fault monitoring of multiphase batch process is a difficult problem in multivariate statistical process monitoring. It needs to consider not only the process monitoring under stable mode, but also the transition mode with strong dynamic nonlinearity. Since the data has different correlations under different operating modes, it is necessary to establish different monitoring models for each process mode, especially the transition process between stable modes. The biggest feature is the dynamic characteristics of the variables. This feature can be better reflected in this transition using a time-varying covariance instead of a fixed covariance during the transition phase. In this paper, a new strategy for batch process sub-phase partition and process monitoring is proposed. Firstly, the three-dimensional data matrix is expanded into a new two-dimensional data according to the time slice expansion strategy. Secondly, the data of each time slice is transformed by Kernel Entropy Component Analysis (KECA), and then the production process is divided into phases according to the spatial angle of the kernel entropy. The production operation process is divided into a stable phase and a transition phase, and monitoring models are respectively established to monitor the production process; Finally, the application of the penicillin fermentation simulation platform shows that the Sub-MKECA phase partition results can reflect the mechanism of the batch process well, and the fault monitoring of the process shows that it can detect faults in time and accurately, and has high practicality value.

INDEX TERMS Batch process, fault monitoring, fault diagnosis, phase partition, kernel entropy component analysis.

I. INTRODUCTION

Multiphase is the inherent characteristic of batch process with not only their specific, unique operating modes and potential process characteristics [1]–[5], but also different key process variables and specific control objectives [6]–[10]. Traditional modeling methods such as MPCA [11], MPLS [12], MICA [13], and their extension algorithms model all data from a complete batch as a whole [14]–[19]. This idea of whole modeling uses the mean and variance information of all process data [20]–[24], describes the historical average trajectory of the process operation, and is susceptible to such effects as noise and outliers. When applied to process monitoring, it tends to produces a high miss alarm rate and false alarm rate. The main reason is that the whole model neglects the multiphase characteristics of batch process. Because the mean and variance of process variables in batch process are significantly different in each phase, it is difficult to reveal the change of process variables correlation by using the mean and variance of the whole modeling idea, and cannot be directly applied to batch process monitoring or fault diagnosis with multiphase characteristics. In other words, using the whole model to describe the whole production process, will make its overall control limit in the monitoring of each stage, either too loose or too strict, leading to a large number of miss alarms. In either case, it can lead to misjudgment of process monitoring and even complete failure of process monitoring. At present, a lot of work has been done by domestic and foreign scholars on the monitoring of multiphase batch processes [6], [25]. For example, Lu *et al.* [26] used the load matrix decomposed by PCA to divide the batch process into multiple operation phase and establish sub-models of each phase for process monitoring and achieved good results. However, the above methods are hard partitioning

The associate editor coordinating the review of this article and approving it for publication was Gongbo Zhou.

methods, which cannot reflect the characteristics of the transition phase, so that the characteristic changes of the transition process in the adjacent phase have a great influence on the detection result. This is because the transition process between phase and phase is not a mainstream mechanism process, but it is a ubiquitous phenomenon and an important process behavior compared to the stable phase as the main operating mode. This transitional phase appears as a dynamic gradual trend, not only in the variation of the batch process variables, but also in the changes in the correlation of the batch process variables. Because of the instability of batch production process in the transition stage, the production process in the transition stage is vulnerable to external interference and deviates from the normal process trajectory, thus affecting the final product quality and production process safety. Therefore, in order to ensure the whole production process, safe and stable, fault monitoring of the transition process is of great significance. In view of the unique process mechanism characteristics of the transition phase compared with the stable mode, it is necessary to identify the transition region for modeling and analysis independent of the stable stage. To this end, Zhao *et al.* [27] proposed a K-means-based sub-phase partitioning and process monitoring method for batch process. This method introduces fuzzy membership degree as the weight coefficient of two sub-phase models adjacent to the transition mode based on the stable sub-phase partitioning. The characteristics of the adjacent two stable sub-phases are integrated to approximate the characteristics of the transition sub-phase, and the monitoring accuracy of the model is improved. However, the above method only uses PCA to analyze the data and describe the production process with the load matrix without discarding any information. The load matrix is used to divide the phase, but if the load matrix contains outliers and noise, it will inevitably affect the correctness of the phase partition. For this reason, Wang *et al.* [28] proposed a two-step phase partitioning method. First, PCA was used to roughly divide the batch process according to the number of principal components after time slice data matrix analysis. Then, in each rough stage, the batch process was subdivided according to the change of process variables correlation. After two stages, the number of principal components in the same stage is the same and the variation direction of variables is similar. But in the transitional stage, the whole monitoring model is used, that is, a whole PCA model is used to monitor the process in the transitional stage, without considering the dynamic and non-linear characteristics of the transitional process, so its monitoring effect in the transitional stage is not good. Qi [29] introduced the concept of 0∼1 fuzzy membership degree to monitor the non-linear data in the transition stage. After dividing the stages with FCM, it proposed to establish PCA monitoring model in the stable stage and KPCA monitoring in the transition stage. The monitoring model in the transition stage fully considered the non-linear characteristics of the batch process and achieved successful application. However, FCM needs to input the number of clusters in advance when dividing the stages, which requires knowing the number of stages in advance. However, the actual batch process may not always know the number of stages in advance, which affects its application to unknown processes. Through the above analysis, the above methods have the following two shortcomingsčž1) The input of clustering algorithm is the load matrix decomposed by MPCA, and MPCA is a linearization method which cannot deal with the non-linear information of batch process. The decomposed load matrix will inevitably lose the non-linear characteristics, and the non-linearity is the inherent characteristics of batch process, resulting in the loss of non-linear information; 2) K-means or FCM clustering algorithm needs to specify the number of phases in advance. Once the number of phases is not selected properly, the result of partitioning will not conform to the real structure of the data set, that is, the actual operation mechanism of the process. Its monitoring of the process will cause a lot of false alarms and missed alarms. In order to solve the above problems, an integrated method based on MKECA similarity phase partition and monitoring is proposed for fault monitoring in batch process. The method maps three-dimensional historical data into high-dimensional kernel entropy space according to time slice expansion to calculate their similarity. Finally, the production process is divided into stable phase and transition phase. In each sub-phase, a monitoring model is constructed to monitor the batch process. When abnormal conditions are released, the time contribution graph method is used to diagnose the fault.

II. PHASE PARTITION OF SIMILARITY SPECTRUM CLUSTERING BASED ON KECA

A. SPECTRAL CLUSTERING OF KECA

KECA [30]–[34] clustering belongs to spectral clustering. The process of clustering depends entirely on the difference of features between samples. After the data is mapped to the high-dimensional kernel entropy space by KECA, the origins of different clusters and nuclear entropy feature space form different angles, and maintain a certain angle distance between different clusters. It is found that KECA clustering is based on the divergence of kernel entropy. The divergence of KECA clustering is studied and two corresponding divergences are summarized.

1) INTEGRATED SQUARED ERROR DIVERGENCE, ISE

The Integrated Squared Error Divergence can be estimated by Parzen window. The definition of ISE is given here.

$$D_{ISE}(p_i, p_j) = \int \left[p_i(\boldsymbol{x}) - p_j(\boldsymbol{x})\right]^2 \mathrm{d}\boldsymbol{x} = V(p_i) - 2\int p_i(\boldsymbol{x}) p_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}' + V(p_j) \quad (1)$$

Here when $p_i(x) = p_j(x)$, $D_{ISE}(p_i, p_j) \in [0, \infty]$, Replace $p_i(\boldsymbol{x})$, and $p_j(\boldsymbol{x})$, with $\widehat{p}_i(\boldsymbol{x})$ and $\widehat{p}_j(\boldsymbol{x})$ respectively. Then the

estimated value of ISE is obtained:

$$\begin{aligned} D_{ISE}(\widehat{p}_i, \widehat{p}_j) &= \frac{1}{N_i^2} \sum_{\boldsymbol{x}_i \in D} \sum_{\boldsymbol{x}_{i'} \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) \\ &= -\frac{1}{N_i N_j} \sum_{\boldsymbol{x}_i \in D} \sum_{\boldsymbol{x}_j \in D} k_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\quad + \frac{1}{N_i^2} \sum_{\boldsymbol{x}_j \in D} \sum_{\boldsymbol{x}_{j'} \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) \\ &= \left\| \boldsymbol{m}_i - \boldsymbol{m}_j \right\|^2 \end{aligned} \tag{2}$$

Here, $\boldsymbol{m}_i = \frac{1}{N_i} \sum_{\boldsymbol{x}_i \in D} \boldsymbol{\phi}(\boldsymbol{x}_i)$ and $\boldsymbol{m}_j = \frac{1}{N_i} \sum_{\boldsymbol{x}_j \in D} \boldsymbol{\phi}(\boldsymbol{x}_j)$ are the average vectors of the clusters corresponding to the data of the density $\widehat{p}_i(\boldsymbol{x})$ and $\widehat{p}_j(\boldsymbol{x})$ in the kernel feature space, respectively.

The ISE can be extended to:

$$\begin{aligned} D_{ISE}\left(\widehat{p}_1, \ldots, \widehat{p}_C\right) &= \sum_{i=1}^{C-1} \sum_{j>i} \int \left[p_i(\boldsymbol{x}) - p_j(\boldsymbol{x})\right]^2 \mathrm{d}\boldsymbol{x} \\ &= \sum_{i=1}^{C-1} \sum_{j>i} \left\| \boldsymbol{m}_i - \boldsymbol{m}_j \right\|^2 \end{aligned} \tag{3}$$

2) CAUCHY-SCHWARZ DIVERGENCE AND MEAN VECTOR

The probability density function of the *i-th* cluster $p_i(x)$ and the probability density function of the population $p(x)$ are known. Then the CS divergence can be defined as:

$$D_{CS}(p_i, p_j) = -\log \frac{\int p_i(\boldsymbol{x}) p_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}{\sqrt{\int p_i^2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \int p_j^2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}} \tag{4}$$

$$\begin{aligned} D_{CS}(p_i, p_j) &= -\log \int p_i(\boldsymbol{x}) p_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &\quad - \frac{1}{2} \widehat{H}(p_i) - \frac{1}{2} \widehat{H}\left(p_j\right) \end{aligned} \tag{5}$$

Here $\hat{p}(x) = \frac{1}{N} \sum_{x_i \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}, \boldsymbol{x}_i)$, then $\widehat{p}_i(\boldsymbol{x}) = \frac{1}{N_i} \sum_{x_i \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}, \boldsymbol{x}_i)$ and $\widehat{p}_j(\boldsymbol{x}) = \frac{1}{N_j} \sum_{x_i \in D} \boldsymbol{k}_\sigma\left(\boldsymbol{x}, \boldsymbol{x}_j\right)$. The specific form is given as (6) shown at the bottom of this page.

The divergence estimate of V_{CS} has the following correspondence with D_{CS}. The specific form is as follows:

$$\begin{aligned} D_{CS}(p_i, p_j) &= -\log V_{CS}\left(\widehat{p}_i, \widehat{p}_j\right) \\ &= -\log \cos \angle \left(\boldsymbol{m}_i, \boldsymbol{m}_j\right) \end{aligned} \tag{7}$$

Here, $\boldsymbol{m}_i = \frac{1}{N_i} \sum_{\boldsymbol{x}_i \in D} \boldsymbol{\phi}(\boldsymbol{x}_i)$ and $\boldsymbol{m}_j = \frac{1}{N_i} \sum_{\boldsymbol{x}_j \in D} \boldsymbol{\phi}\left(\boldsymbol{x}_j\right)$ are the average vectors of the clusters corresponding to the data of the density $\widehat{p}_i(\boldsymbol{x})$ and $\widehat{p}_j(\boldsymbol{x})$ in the kernel feature space, respectively. The CS divergence can be extended to C clusters by Parzen window density estimation. The specific form is as follows:

$$D_{CS}\left(p_1, \ldots, p_C\right) = -\log \frac{1}{\kappa} \sum_{i=1}^{C-1} \sum_{j>i} \frac{\int p_i(\boldsymbol{x}) p_j(\boldsymbol{x}) d\boldsymbol{x}}{\sqrt{\int p_i^2(\boldsymbol{x}) d\boldsymbol{x} \int p_j^2(\boldsymbol{x}) d\boldsymbol{x}}} \tag{8}$$

From the above derivation, it can be concluded that the CS divergence is mainly measured by the cosine of the angle formed by these average vectors, and the specific form is as follows:

$$\begin{aligned} &\max_{D_1, \ldots, D_C} D_{CS}\left(p_1, \ldots, p_C\right) \\ &= \max_{D_1, \ldots, D_C} -\log \frac{1}{\kappa} \sum_{i=1}^{C-1} \sum_{j>i} \frac{\int p_i(\boldsymbol{x}) p_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}{\sqrt{\int p_i^2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \int p_j^2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}} \end{aligned} \tag{9}$$

Figure 1 shows the CS divergence and ISE divergence based on the Parzen window Renyi entropy estimation in the kernel feature space. m_1, m_2 represent two mean vectors, CS divergence represents the angle between the two mean vectors, ISE divergence represents the Euclidean distance between the two mean vectors. Clustering can be represented

$$\begin{aligned} V_{CS}\left(\widehat{\boldsymbol{p}}_i, \widehat{\boldsymbol{p}}_j\right) &= \frac{\frac{1}{N_i N_j} \sum_{\boldsymbol{x}_i \in D} \sum_{\boldsymbol{x}_i \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{\frac{1}{N_i^2} \sum_{\boldsymbol{x}_i \in D} \sum_{\boldsymbol{x}_{i'} \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) \frac{1}{N_j^2} \sum_{\boldsymbol{x}_j \in D} \sum_{\boldsymbol{x}_{j'} \in D} \boldsymbol{k}_\sigma(\boldsymbol{x}_j, \boldsymbol{x}_{j'})}} \\ &= \frac{\left\langle \frac{1}{N_i} \sum_{\boldsymbol{x}_i \in D} \boldsymbol{\phi}(\boldsymbol{x}_i) \right\rangle \left\langle \frac{1}{N_i} \sum_{\boldsymbol{x}_j \in D} \boldsymbol{\phi}\left(\boldsymbol{x}_j\right) \right\rangle}{\sqrt{\left\langle \frac{1}{N_i} \sum_{\boldsymbol{x}_i \in D} \boldsymbol{\phi}(\boldsymbol{x}_i), \frac{1}{N_i} \sum_{\boldsymbol{x}_{i'} \in D} \boldsymbol{\phi}(\boldsymbol{x}_{i'}) \right\rangle \left\langle \frac{1}{N_i} \sum_{\boldsymbol{x}_j \in D} \boldsymbol{\phi}\left(\boldsymbol{x}_j\right), \frac{1}{N_i} \sum_{\boldsymbol{x}_{j'} \in D} \boldsymbol{\phi}\left(\boldsymbol{x}_{j'}\right) \right\rangle}} \\ &= \frac{\langle \boldsymbol{m}_i, \boldsymbol{m}_j \rangle}{\sqrt{\langle \boldsymbol{m}_i, \boldsymbol{m}_i \rangle \langle \boldsymbol{m}_j, \boldsymbol{m}_j \rangle}} = \cos \angle \left(\boldsymbol{m}_i, \boldsymbol{m}_j\right) \end{aligned} \tag{6}$$

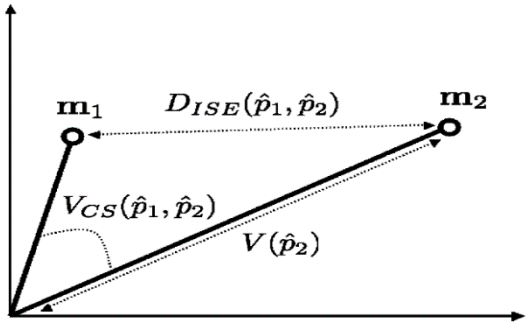


FIGURE 1. CS divergence, ISE divergence and mean vectors in Mercer kernel feature space.

by an angle-based value function as follows:

$$\boldsymbol{J}\left(C_1, \ldots, C_C\right) = \sum_{i=1}^{C} N_i \cos\angle\left(\boldsymbol{m}_i, \boldsymbol{m}_j\right) \tag{10}$$

B. KECA-BASED SIMILARITY INDEX

Since the nonlinear mapping function ϕ is unknown at the time of kernel mapping, the load matrix m of nonlinear data in the kernel space cannot be explicitly expressed, which means that the clustering algorithm such as k-means cannot be directly applied for time division. Although the load matrix m cannot be explicitly expressed, the angle and distance of the load matrix can be explicitly expressed by introducing the kernel matrix. Therefore, this section derives the kernel KECA similarity based on the angle definition. The calculation formula of the similarity of the load matrix in the kernel entropy space is derived below. According to the cosine formula and the inner product definition, the cosine between the *j-th* column vectors of the load matrices M^1 and $\boldsymbol{M}^2$ is:

$$\cos(\boldsymbol{m}_j^1, \boldsymbol{m}_j^2) = \frac{\left\langle \boldsymbol{m}_j^1, \boldsymbol{m}_j^2 \right\rangle}{\sqrt{\sum_{n=1}^{N} \left(\boldsymbol{m}_{nj}^1\right)^2} \sqrt{\sum_{n=1}^{N} \left(\boldsymbol{m}_{nj}^2\right)^2}} \tag{11}$$

The angle between $\boldsymbol{M}^1$ and $\boldsymbol{M}^2$ can be defined as:

$$\theta_{1,2} = \sum\nolimits_{j=1}^{A} \omega_j \theta_j \quad (j = 1, 2, \ldots, A) \tag{12}$$

where ω_j is the weighting factor and θ_j is the angle between the *j-th* column vector of $\boldsymbol{M}^1$ and $\boldsymbol{M}^2$, defined as follows:

$$\theta_j = \begin{cases} \dfrac{180 arc\cos(\boldsymbol{m}_j^1, \boldsymbol{m}_j^2)}{\pi} & \cos(\boldsymbol{m}_j^1, \boldsymbol{m}_j^2) \geqslant 0 \\ 180 - \dfrac{180 arc\cos(\boldsymbol{m}_j^1, \boldsymbol{m}_j^2)}{\pi} & \cos(\boldsymbol{m}_j^1, \boldsymbol{m}_j^2) \geqslant 0 \end{cases} \tag{13}$$

The similarity indicator according to the angle can be defined as:

$$S_{KECA-Angle} = (cos(\theta_{1,2}))^2 \tag{14}$$

It is known from the above definition, $S_{KECA-Angle} \in [0, 1]$.

C. SUB-PHASE PARTITION BASED ON KECA SIMILARITY

Suppose the modeling data of a batch process is a three-dimensional array $\boldsymbol{X}$ $(I \times J \times K)$, and the three dimensions represent the number of batch operations (i $=1, \ldots, I$), the number of process variables (j $= 1, \ldots,$ J), and sampling time (k $=1, \ldots,$K). By cutting the data $\boldsymbol{X}$ $(I \times J \times K)$ vertically along the time direction, K two-dimensional data slices X_k $(I \times J)$ can be obtained, which consist of all batch process measurements at the *k-th* sampling instant, called the "time slice matrix". Using the KECA method to extract the nonlinear characteristics of each time slice matrix, the load matrix $\boldsymbol{V}k$ characterizes the correlation information between the process variables of K sampling moments, and becomes the basic data unit of the time division. The detailed steps of the time division algorithm are as follows:

a) Calculate the density indicator for each data point according to the following formula:

$$D_1^k = \sum_{h=1}^{k} \exp\left\{-\frac{(1 - s_{k,h})}{(\gamma_a/2)^2}\right\} \quad (k = 1, 2, \ldots, K) \tag{15}$$

where $s_{k,h} = \lambda s_{KECA-angle}$ represents the similarity between the k-th time slice and the h-th time slice, and the weight coefficient $0 \leqslant \lambda \leqslant 1$. The positive number γ_a is the radius of the area of the data point $\boldsymbol{M}^k$. When the other data points are not within the radius γ_a, these points contribute little to the density index of $\boldsymbol{M}^k$. Therefore, if the density index of data $\boldsymbol{M}^k$ is high, it means that there must be multiple adjacent data points similar to each other around the point.

b) After calculating each data point density indicator, select the data point with the highest density indicator as the first cluster center. Let the number of cluster centers c=1, the selected points be recorded as $\boldsymbol{M}_c^*$ and item D_c^* as their density indicators, then the density index of each data point $\boldsymbol{M}^k$ can be corrected by the density index of the *c-th* cluster center:

$$D_{c+1}^k = D_c^k - D_c^* \sum_{h=1}^{k} \exp\left\{-\frac{(1 - s_{c,k}^*)}{(\gamma_b/2)^2}\right\} \quad (k = 1, 2, \ldots, K) \tag{16}$$

where $s_{c,k}^*$ represents the similarity between the *k-th* time slice and the *c-th* cluster center. The constant γ_b defines a field in which the density index is significantly reduced, and $\gamma_b > \gamma_a$ is satisfied to avoid the two cluster centers being closer together. In this way, the density index of the data points close to M_c^* will be significantly reduced, so these points are unlikely to be the next cluster center.

c) After correcting the density indicator for each data point, if the maximum value $D_{(c+1)}^{\max}$ in $D_{(c+1)}^k (k = 1, 2, \ldots, K)$ satisfies the following cluster termination criteria:

$$D_{c+1}^{\max} \Big/ D_1^* < \upsilon \tag{17}$$

where υ is a sufficiently small number. The physical meaning of the termination criterion is that the current highest density value is very small compared to the initial highest density value, that is, the current cluster center contains very few data points, so the cluster center can be ignored and the clustering can be ended. Otherwise, accept the data point with the highest density indicator as the c+1-*th* cluster center.

d) Let c=c+1, cycle through the above process until the final $\boldsymbol{C_0}$ initial cluster centers are finalized.

e) After determining the cluster center, calculate the similarity between the K time slice load matrices and the cluster center, and the similarity magnitude represents the degree of similarity. According to the principle of large similarity, $\boldsymbol{K}$ time slices are divided into $\boldsymbol{C_0}$ categories, which characterize different process characteristics of $\boldsymbol{C_0}$.

f) The time slices are arranged in order, and the time slices whose category attributes are the same and which are consecutive in time are divided into the same time period.

g) Due to problems such as noise or measurement error, it may happen that the data of a certain time or several consecutive moments and the data of the time period before and after it do not belong to the same period in the classification process form a ''jump'' phenomenon. This situation is treated as a singularity problem in the clustering process and still falls into temporally adjacent sub-phases. After processing, the batch process is divided into $\boldsymbol{C}$ sub-phases according to the change of process characteristics.

III. OFF-LINE MODELING AND ONLINE MONITORING BASED ON SUB-PHASE PARTITION

For each sub-phase, a uniform time-representation model can be used to characterize the stable correlation characteristics between process variables from the perspective of the entire time period. Based on the data point index contained in each class, the data of the same sub-period in the original data space are put together to establish a KECA model.

A. FAULT DETECTION

1) OFFLINE MODELING

a) $\boldsymbol{K}$ time slice matrices $\boldsymbol{X}_k$ $(I \times J)$ are obtained by cutting the three-dimensional data set $\boldsymbol{X}$ $(I \times J \times K)$ of batch operations collected under normal working conditions along the time axis.

b) Normalize the *i-th* (i = 1, 2,..., I) samples in the *k-th* time slice matrix:

$$\bar{x}_{ij}^k = \frac{x_{ij}^k - v_j^k}{s_j^k} \quad (j = 1, 2, \ldots, J) \tag{18}$$

where $v_j^k = \frac{1}{I}\sum_{i=1}^{I} x_{ij}^k$ and $\left(s_j^k\right)^2 = \frac{1}{I}\sum_{i=1}^{I}\left(x_{ij}^k - v_j\right)^2$ represent the mean and variance of the *j-th* variable, respectively.

c) The $\boldsymbol{K}$ load matrix is divided into $\boldsymbol{C}$ sub-phases using the time division algorithm introduced in the previous section, and the length of each sub-phases is $\boldsymbol{L_c}$.

d) The standardized time slice matrices belonging to a sub-period are put together to form a large data matrix $\tilde{X}(L_c I \times J)(c = 1, 2, \ldots, C)$

e) Using the data matrix $\tilde{X}_c$ $(L_c I \times J)$ to directly establish a unified KPCA model representing $\boldsymbol{C}$ sub-phases.

f) Calculate the $\boldsymbol{T^2}$ and $\boldsymbol{SPE}$ statistic for all data in each phase and determine the control limits for the statistic for each sub-phase.

2) ONLINE MONITORING

a) Obtain the current sample $\boldsymbol{x}_k$ and perform data normalization preprocessing according to the historical mean and variance of the corresponding time:

$$\bar{x}_{k,j} = \frac{x_{k,j} - v_j^k}{s_j^k} \quad (j = 1, 2, \ldots, J) \tag{19}$$

b) According to the process time indication, it can be determined which sub-phase the new data belongs to, and the corresponding monitoring model is directly called.

c) Calculate $\boldsymbol{T^2}$ and $\boldsymbol{SPE}$ statistics for current sample.

d) Determine if the $\boldsymbol{T^2}$ and $\boldsymbol{SPE}$statistics are outside the normal control limits. If neither of them exceeds the limit, the current time process is considered to be normal operation, and the process returns to step 1; otherwise, the process is abnormal, and the cause of the failure is analyzed and diagnosed.

B. FAULT DIAGNOSIS

In this section, the contribution of process variables to the two monitoring statistics T^2 and SPE is deduced by using the gradient descent algorithm of the kernel function. The maximum contribution of each variable to the two monitoring statistics is used to judge the fault. The Gaussian kernel function is used to calculate the kernel matrix. Assuming the existence of vector $v = [v_1, v_2, \ldots v_m]^T, (i = 1, 2, \ldots m)$, the kernel function can be written as:

$$\boldsymbol{k}(x_j, x_k) = \exp(-\left\|\boldsymbol{v}\cdot\boldsymbol{x}_j - \boldsymbol{v}\cdot\boldsymbol{x}_k\right\|^2/\sigma) \tag{20}$$

The partial derivative of the *i-th* variable v_i can be calculated by the following formula:

$$\begin{aligned}\frac{\partial \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_k)}{\partial v_i} &= \frac{\partial \boldsymbol{k}(\boldsymbol{v}\cdot\boldsymbol{x}_j, \boldsymbol{v}\cdot\boldsymbol{x}_k)}{\partial v_i} \\ &= -\frac{1}{\sigma}(\boldsymbol{v}_i\boldsymbol{x}_{j,i} - \boldsymbol{v}_i\boldsymbol{x}_{k,i})^2\boldsymbol{k}(\boldsymbol{v}\cdot\boldsymbol{x}_j, \boldsymbol{v}\cdot\boldsymbol{x}_k) \\ &= -\frac{1}{\sigma}(\boldsymbol{v}_i\boldsymbol{x}_{j,i} - \boldsymbol{v}_i\boldsymbol{x}_{k,i})^2\,\boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_k)\big|_{v_i=1}\end{aligned} \tag{21}$$

where $x_{j,i}$ is the *i-th* variable of the *j-th* sample, and the partial derivatives of the products of the two kernel functions are:

$$\begin{aligned}&\frac{\partial \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_{new})\boldsymbol{k}(\boldsymbol{x}_k, \boldsymbol{x}_{new})}{\partial \boldsymbol{v}_i} \\ &= -\frac{1}{\sigma}\left[\left(\boldsymbol{x}_{j,i} - \boldsymbol{x}_{new,j}\right)^2 + \left(\boldsymbol{x}_{k,i} - \boldsymbol{x}_{new,i}\right)\right] \\ &\quad\times \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_{new})\boldsymbol{k}(\boldsymbol{x}_k, \boldsymbol{x}_{new})\end{aligned} \tag{22}$$

Define two new statistics to calculate the contribution of each variable to the monitoring statistic:

$$C_{T^2,new,i} = \left|\frac{\partial T^2}{\partial \boldsymbol{v}_i}\right|, \quad C_{SPE,new,i} = \left|\frac{\partial SPE_{new}}{\partial \boldsymbol{v}_i}\right| \tag{23}$$

$C_{T^2,new,i}$ and $C_{SPE,new,i}$ respectively represent the contribution rate of the *i-th* process variable of the new time test data to the two process monitoring statistics $\boldsymbol{T^2}$ and ***SPE***. The monitoring statistic $\boldsymbol{T^2}$ of the KECA model can be calculated by the following formula:

$$\begin{aligned} T^2_{new} &= \boldsymbol{t}^T_{new}\boldsymbol{\Lambda}^{-1}\boldsymbol{t}_{new} = \bar{\boldsymbol{K}}^T_{new}\boldsymbol{a}\Lambda^{-1}\boldsymbol{a}^T\bar{\boldsymbol{K}}_{new} \\ &= trace(\boldsymbol{a}^{\mathrm{T}}\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}^T_{new}\boldsymbol{a}\boldsymbol{\Lambda}^{-1}) \end{aligned} \tag{24}$$

This is represented by using a mean Gram matrix as (25) shown at the top of the next page, and the contribution rate of the *i-th* variable to the statistic $\boldsymbol{T^2}$ can be expressed as:

$$\begin{aligned} C_{T^2,new,i} &= \left|\frac{\partial T^2}{\partial v_i}\right| = \left|\frac{\partial}{\partial v_i}(trace(\boldsymbol{a}^T\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}^T_{new}\boldsymbol{a}\boldsymbol{\Lambda}^{-1})\right| \\ &= \left|trace(\boldsymbol{a}^T(\frac{\partial}{\partial v_i}\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}^T_{new})\boldsymbol{a}\boldsymbol{\Lambda}^{-1})\right| \end{aligned} \tag{26}$$

The definition of ***SPE*** monitoring statistics in kernel space is as follows:

$$\begin{aligned} SPE_{new} &= \boldsymbol{k}(\boldsymbol{x}_{new},\boldsymbol{x}_{new}) - \frac{2}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j,\boldsymbol{x}_{new}) \\ &\quad + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j,\boldsymbol{x}_i) - \boldsymbol{t}^T_{new}\boldsymbol{t}_{new} \end{aligned} \tag{27}$$

The contribution rate of the *i-th* variable to the statistic SPE can be expressed as (28) shown at the top of the next page. The key question in the above solution process is how to solve the key question in the above solution process is how to solve the $\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}^T_{new}$ matrix. The solution process is given below. Each row vector in the Gram matrix consists of 4 items, and the 2nd and 4th items are composed of training samples. Constant $S_p = \frac{1}{N}\sum_{j=1}^{N}k(x_p,x_j)$ and $S = \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}k\left(x_j,x_i\right)$ during the test, therefore, the P row and Q column elements of matrix $\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}^T_{new}$ are represented as (29) shown at the top of the next page. The partial differentiation of element $[\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}^T_{new}]_{pq}$ to variable $\boldsymbol{v}_i$ can be expressed as (30) shown at the top of the next page. Finally, the amount by which $C_{T^2,new,i}$ and $C_{SPE,NEW,i}$ vary greatly is extracted as a fault characteristic variable. Figure 2 illustrates the offline modeling and online monitoring procedures.

IV. ALGORITHM VERIFICATION

A. NUMERICAL EXAMPLE SIMULATION

This section uses a simple numerical example to show that due to the unsynchronized trajectories between production batches and the transition between phases, there is often a stronger dynamic nonlinearity between the variables in the transition process. The main purpose of the experiment here

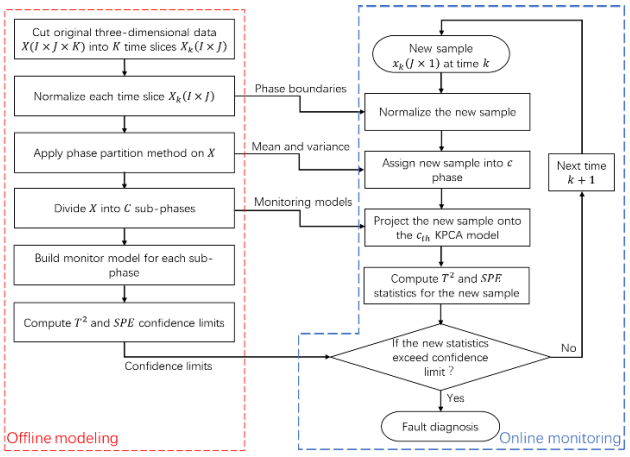


FIGURE 2. Illustration of offline modeling and online monitoring procedures 3.

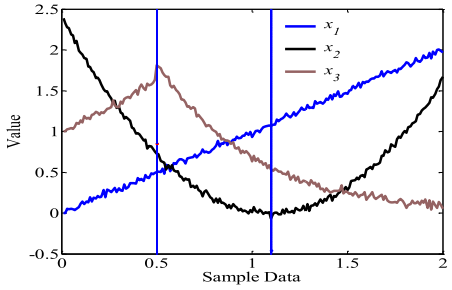


FIGURE 3. Trajectories of three process variables from a batch run.

is to prove the following points: (1) The process data based on the phase is nonlinear; (2) The data characteristics of the transition phase have stronger nonlinear characteristics than the stable phase;

Assume that the following numerical process has three process variables x_1, x_2, x_3, which are obtained as follows:

$$\begin{aligned} x_1 &= t + e_1 \\ x_2 &= 2(t-1.1)^2 + e_2 \\ x_3 &= \begin{cases} \exp(t) + e_3 & t < 0.5 \\ 5\times\exp(-2t) + e_3 & t \geqslant 0.5 \end{cases} \end{aligned} \tag{31}$$

where t∈[0.01,2], e1, e2, e3 are white noises that follow the normal distribution of N(0, 0.012). Figure 3 shows the variation of the three process variables over the domain of the sample sequence t. It can be seen from the figure that the nonlinear process in this example can be more clearly divided into three piecewise approximation linear processes. Furthermore, it can be seen from the figure that the variable x_2 has a distinct smooth transition characteristic near the turning point (t=1.1). If the sampling interval is 0.01, a batch of 200×3 data matrix can be generated for each batch. In order to simulate the characteristics of the actual batch process and realize the unequal length of each batch stage, the methods of translation or scaling are adopted for x_2 and x_3, so that the turning point of x_2 (t=1.1) falls randomly

$$\bar{\boldsymbol{K}}_{new} = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{x}_1, \boldsymbol{x}_{new}) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_1, \boldsymbol{x}_j) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_{new}, \boldsymbol{x}_j) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_i) \\ \boldsymbol{k}(\boldsymbol{x}_2, \boldsymbol{x}_{new}) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_2, \boldsymbol{x}_j) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_{new}, \boldsymbol{x}_j) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_i) \\ \vdots \\ \boldsymbol{k}(\boldsymbol{x}_N, \boldsymbol{x}_{new}) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_N, \boldsymbol{x}_j) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_{new}, \boldsymbol{x}_j) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_i) \end{bmatrix} \tag{25}$$

$$\begin{aligned} C_{SPE,new,i} &= \left|\frac{\partial \mathrm{SPE}_{new}}{\partial \boldsymbol{v}_i}\right| = \left|-\frac{1}{\boldsymbol{\sigma}}(-\frac{2}{N}\frac{\partial}{\partial \boldsymbol{v}_i}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_{new}) - \frac{\partial}{\partial \boldsymbol{v}_i}\boldsymbol{t}_{new}^T\boldsymbol{t}_{new})\right| \\ &= \left|\frac{1}{\sigma}(\frac{2}{N}\frac{\partial}{\partial \boldsymbol{v}_i}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_{new}) + trace(\boldsymbol{a}^T(\frac{\partial}{\partial \boldsymbol{v}_i}\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}_{new}^T\boldsymbol{a})))\right| \end{aligned} \tag{28}$$

$$\begin{aligned} [\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}_{new}^T]_{pq} &= \boldsymbol{k}(\boldsymbol{x}_p, \boldsymbol{x}_t)\boldsymbol{k}(\boldsymbol{x}_q, \boldsymbol{x}_t) + (S - S_p)\boldsymbol{k}(\boldsymbol{x}_q, \boldsymbol{x}_t) + (S - S_q)\boldsymbol{k}(\boldsymbol{x}_p, \boldsymbol{x}_t) - \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_t, \boldsymbol{x}_j)(\boldsymbol{k}(\boldsymbol{x}_p, \boldsymbol{x}_t) \\ &\quad + \boldsymbol{k}(\boldsymbol{x}_q, \boldsymbol{x}_t)) + \frac{1}{N}(S_p + S_q - 2S)\sum_{j=1}^{N}\boldsymbol{k}(\boldsymbol{x}_t, \boldsymbol{x}_j) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\boldsymbol{k}(\boldsymbol{x}_t, \boldsymbol{x}_j)\boldsymbol{k}(\boldsymbol{x}_t, \boldsymbol{x}_i) \end{aligned} \tag{29}$$

$$\begin{aligned} \frac{\partial(\bar{\boldsymbol{K}}_{new}\bar{\boldsymbol{K}}_{new}^T)_{pq}}{\partial \boldsymbol{v}_i} &= -\frac{1}{\sigma}\Big\{\Big[(\boldsymbol{x}_{p,i} - \boldsymbol{x}_{t,i})^2 + (\boldsymbol{x}_{q,i} - \boldsymbol{x}_{t,i})^2\Big] \times \boldsymbol{k}(\boldsymbol{x}_p, \boldsymbol{x}_t)\boldsymbol{k}(\boldsymbol{x}_q, \boldsymbol{x}_t) \\ &\quad + (S - S_q)(\boldsymbol{x}_{p,i} - \boldsymbol{x}_{t,i})^2 \times \boldsymbol{k}(\boldsymbol{x}_p, \boldsymbol{x}_t) + (S - S_p)(\boldsymbol{x}_{q,i} - \boldsymbol{x}_{t,i})^2 \times \boldsymbol{k}(\boldsymbol{x}_q, \boldsymbol{x}_t) \\ &\quad - \frac{1}{N}\sum_{j=1}^{N}\Big[(\boldsymbol{x}_{j,i} - \boldsymbol{x}_{t,i})^2 + (\boldsymbol{x}_{p,i} - \boldsymbol{x}_{t,i})^2\Big] \times \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_t)\boldsymbol{k}(\boldsymbol{x}_p, \boldsymbol{x}_t) + \frac{1}{N}(S_p + S_q + S)\sum_{j=1}^{N}(\boldsymbol{x}_{j,i} - \boldsymbol{x}_{t,i})^2 \times \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_t) \\ &\quad + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{k=1}^{N}\Big[(\boldsymbol{x}_{j,i} - \boldsymbol{x}_{t,i})^2 + (\boldsymbol{x}_{k,i} - \boldsymbol{x}_{t,i})\Big] \times \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}_t)\boldsymbol{k}(\boldsymbol{x}_k, \boldsymbol{x}_t)\Big\} \end{aligned} \tag{30}$$

between 1.0 and 1.2, and the turning point of x_3 (t=0.5) falls randomly between 0.4 and 0.6, resulting in 20 batch data. The average trajectory is extracted from 20 batches of data, and the numerical process is divided into stable phase (0∼0.4), (0.6∼1), (1.2∼2) and transitional phase (0.4∼0.6), (1.0∼1.2). Then analyze the correlation between the variables of each phase, Fig 4 (a) shows the relationship of variables in stable phase 1. When the two variables satisfy the linear relationship, they correspond to straight lines. Red lines represent the relationship between variables x_2 and x_1, and blue lines represent the relationship between variables x2 and x3. It can be seen from the graph that in the stable phase 1, x_2 and x_1 do not satisfy the linear relationship, and neither do x_2 and x_3. Figure 4(b) represents the relationship of the variables in the stable phase 2. As can be seen from the relationship curve in the figure, in the stable phase 1, x_2, x_1 and x_2, x_3 do not satisfy the linear relationship. Figure 4(c) and Figure 4(d) represent the relationship between the variables in transition phase 1 and transition phase 2, respectively. From the blue and red lines, we can see that there is a strong non-linear relationship between x_3 and x_2, x_3 and x_1. In summary, it can be analyzed as follows: in the stable phase, the variables show a strong non-linear relationship, while in the transition phase, so the linear PCA method is obviously inappropriate to establish a statistical model in the transition phase.

B. PENICILLIN SIMULATION PLATFORM VERIFICATION

In this paper, penicillin fermentation simulation platform PenSim2.0 is used as the simulation test platform of the algorithm. The main purpose of the experiment here is to prove the following points: (1) The process data based on the phase is nonlinear and multi-phase coexistence, not single existence; (2) The monitoring model based on the phase has effective fault monitoring capability; (3) The monitoring model based on the phase is conducive to the diagnosis of intermittent process faults. The monitoring strategy proposed in this chapter was comprehensively tested. The reaction time of each batch of penicillin fermentation was 400h, the sampling interval was 1 hour, and 10 process

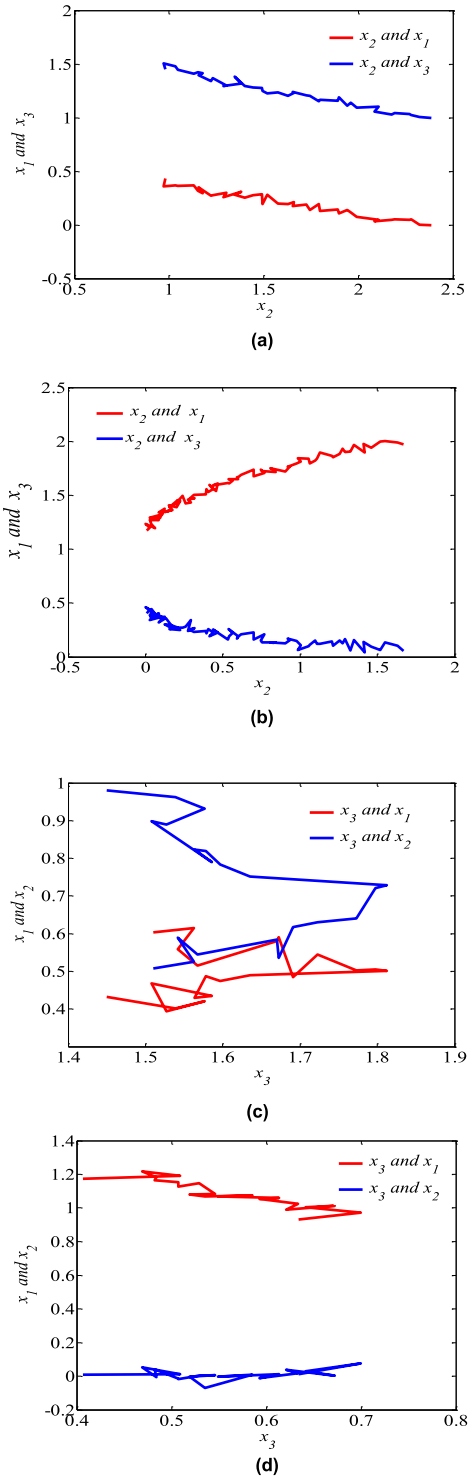


FIGURE 4. Correlation of process variables in different phases and transitions.

variables (Agitator power, Aeration rate, Substrate feed flow rate, Substrate feed temperature, DO conc, CO_2, PH, Temperature, Generated heat, Cold water flow rate) were selected. In order to make the training sample data reliable, assuming that there are enough training sample data, 100 batches of normal batch data are produced as the reference database

TABLE 1. Summary of fault types introduced in process.

Fault Number	Fault Variable	Fault Description	Fault Duration
1	base flow rate	Ramp changes from +0% to +5%	200h-400h
2	aeration rate	step changes by +5%	100h-400h
3	air head pressure	step changes by +5%	150h-400h
4	base flow rate	step changes by +5%	200h-400h
5	aeration rate	ramp changes from −0% to −5%	50h-400h
6	air head pressure	ramp changes from +0% to +3%	150h-400h

TABLE 2. Summary of monitoring results for MKPCA, sub-MKPCA and sub-MKECA.

Fault Number	Type I Error Rate(%)			Type II Error Rate(%)		
	MKPCA	Sub-MKPCA	Sub-MKECA	MKPCA	Sub-MKPCA	Sub-MKECA
1	22	9	0	82	48	0.6
2	5.7	12	2.3	7	5	0
3	7.8	14	0.2	12.5	3	0.8
4	3.8	5	1.2	42.5	20.7	4
5	3.2	6	1.3	38	12	2
6	3.2	7	0.8	52.4	44.2	4.5

of the model. The three-dimensional data of the monitoring model is X(100 × 10 × 400) after KECA data conversion. In the high-dimensional space, the batch process was divided into five phases according to the phase partition method in this paper. The fermentation process of penicillin was eventually divided into five sub-phases, of which 1-44h, 92-152h, 298-400h were stable phases, 45-91h, 152-297h were transitional phases. In order to verify the effectiveness of the proposed monitoring algorithm, we compare it with MKPCA and sub-MKPCA. The experimental kernel parameters are selected as 200. The fault type is shown in Table 1. The monitoring effect is shown in Table 2. Due to the limitation of the space, only the monitoring effect diagram and fault diagnosis diagram of fault 1 and fault 2 are given here. The method is compared with the MKPCA method and the sub-MKPCA method. Among them, MKPCA uses the cumulative variance contribution rate method and determined 17 principal components; in sub-MKPCA, the three stable phases of PCA modeling use cross-validation method to determine the principal components. In the sub-MKPCA method, PCA modeling is used in three stable phases, and the number of principal elements is determined by

cross-validation method is 14, 13, 16 respectively. The two transition phases are modeled by KPCA, and the principal component is determined by the variance contribution rate method, which number is 25 and 27 respectively; the three stable phases in sub-MKECA are modeled by PCA, and the number of principals is determined by cross-validation method to be 6, 8, 6 and 2 transition phases are modeled by sub-MKECA. The entropy contribution rate method determines the number of principals to be 10 and 12, respectively.

C. MONITORING RESULTS AND DISCUSSION

Table 2 gives a comparison of the monitoring performance of the three methods. It can be seen that the method proposed in this paper is effective for the detection of various types of faults, and the false alarm rate (type I error rate) is the lowest among the three methods, indicating the method of this paper can improve the reliability of the monitoring process to a certain extent. For faulty batches, the proposed method can achieve fast and accurate detection of faults with a small miss alarm rate (type II error rate). In addition, in some fault detection, the MKPCA and sub-MKPCA methods have higher false alarm rates. The reason for the analysis shows that for some of the faults, no abnormality is detected in the T^2 chart of the MKPCA and sub-MKPCA methods, and the miss alarm rate in the ***SPE*** chart is also significantly higher than the method in this paper. Figure 5 shows the batch monitoring effect for fault type 1. The fault is a slope disturbance with a slope of +5%, and 200h is introduced until the end of the reaction. Under normal circumstances, the stirring power is the main factor affecting the dissolved oxygen concentration, and the decrease of the stirring power will cause the concentration of dissolved oxygen in the medium to decrease, thereby causing the growth rate of the bacteria to slow down and finally reducing the yield of penicillin. It can be seen from Figure 5 that the method detects an abnormal situation at 200h, that is, almost at the same time as the fault occurs, which is about 20h and 14h earlier than the conventional MKPCA and sub-MKPCA methods, respectively. In the T^2 monitoring chart, the sub-MKPCA method lags the detection method by about 62 h, while the T^2 chart of the MKPCA method lags the method by about 136 h. The analysis shows that the fault in the experiment happens in the transition phase 2. Because sub-MKPCA divides the model into different sub-phases, the connection between adjacent process stages is separated, and the characteristics of the transition stage cannot be reflected. Therefore, failures occurring during the transition process can not be detected in time and effectively, and there is a large lag. Some are even obscured by the changes in the correlation of the variables in the transition phase, and the change in the correlation of the process variables caused by the fault is considered to be caused by the phase transition; The MKPCA method treats the complete batch data as a whole, and cannot accurately describe the characteristics of all phases of the process; Or it uses a monitoring model to characterize the entire operating range, resulting in too loose monitoring limits. It can be

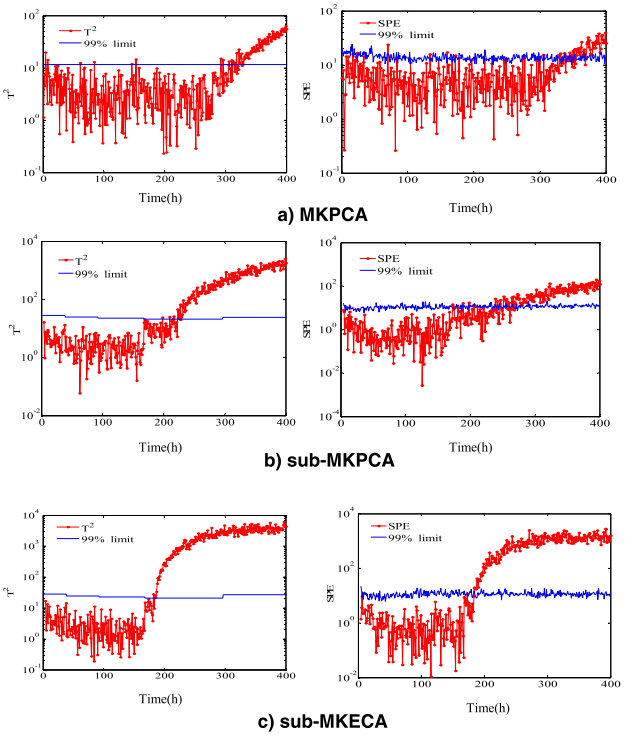


FIGURE 5. Monitoring results using MKPCA, sub-MKPCA and the proposed method for test batch 1.

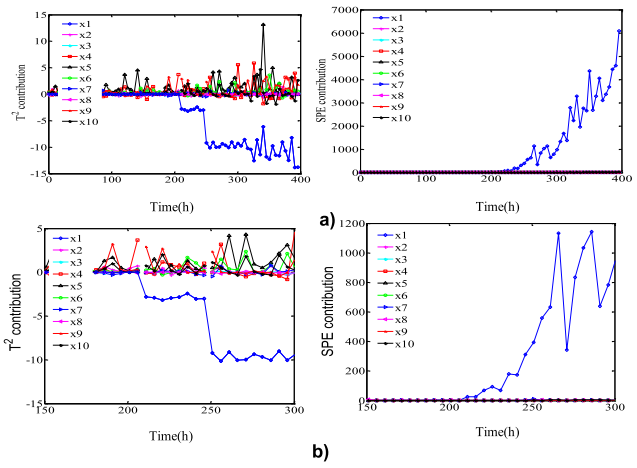


FIGURE 6. Fault diagnosis a) the contribution of the overall time diagram method b) the contribution of the phase time diagram method.

seen that the MKPCA method based on the overall modeling idea is no longer applicable when monitoring the multi-phase production process.

Figure 7 is the result of monitoring batches of fault type 2. The failure batch was that the feeding rate reduced by 15%, by adding step disturbance at Aeration rate of 100 h until the end of the reaction. It can be seen from the monitor chart that this method detects faults in 100 hours, 27 hours ahead of the traditional MKPCA method and 12 hours ahead of the sub-MKPCA method. In the T^2 monitoring chart of sub-MKPCA method, it not exceeded until 123 hours, and

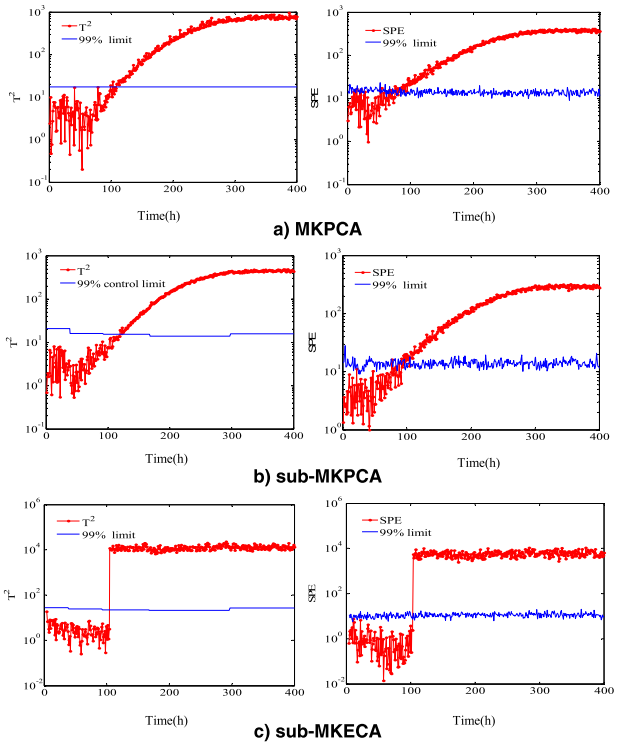


FIGURE 7. Monitoring results using MKPCA, sub-MKPCA and the proposed method for test batch 2V. SOME COMMON MISTAKES.

delay fault occurs for 23 hours. In addition, there are some false alarms in ***SPE*** monitoring chart of MKPCA and sub-MKPCA methods at the beginning of fermentation. All these fully show that the multi-phase monitoring model based on sub-MKECA is superior to the traditional MKPCA and sub-MKPCA methods in both accuracy and robustness.

After the fault is monitored, the time contribution diagram is used to diagnose the cause of the fault. Take fault 1 as an example, where the fault is caused by variable 1, and the contribution of process variables to the two statistics is shown in Figure 8. For the ***SPE*** statistics of MKECA, it accurately identifies the contribution of variable 1 to the abnormal change of the statistical index. But for $\boldsymbol{T^2}$ statistics, from Figure 6, besides variable 1, variables 5, 6 and 7 show a certain contribution to $\boldsymbol{T^2}$ statistics at different times, indicating that they may be fault variables, but from the overall trend, the contribution of variable 1 in the whole cycle is the most significant, which is also where the time contribution graph is superior to the traditional contribution graph. It is based on the overall periodic trend of fault diagnosis, so as to avoid a single variable at a particular time contributing significantly to the statistics, mistaken as a fault. In addition, Figure 6(b) graph shows the contribution of the monitored variable to the fault at each sampling instant, effectively reduces the amount of calculation, and can quickly locate the source of the fault. This is very necessary in the production process. The earlier the fault source is located, the sooner the fault can be handled correctly and the impact of the fault on the quality of the production. After detecting the fault, it is necessary to trace the

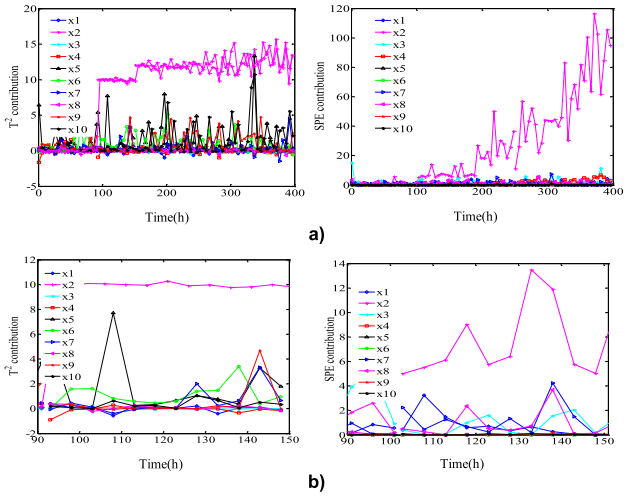


FIGURE 8. Fault diagnosis a) the contribution of the overall time diagram method b) the contribution of the phase time diagram method.

TABLE 3. Time contribution of pattern recognition fault variable results.

Fault Number	Whether the time contribution graph identifies the fault source
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	Yes

fault variable. Figure 6 shows the statistical time contribution graph of the stable phase 3 obtained by the time contribution graph method. As can be seen from the figure, the $\boldsymbol{T^2}$ and ***SPE*** monitoring statistics are given. The same diagnostic result, that is, the fault is caused by the abnormality of the variable 2. The test found that the phased contribution graph based on the kernel function takes about 1 minute, which can fully meet the real-time requirements of fault diagnosis of general industrial processes, while the overall contribution graph calculation time takes about 15 minutes. The results of other fault identification are shown in Table 3. The fault source can be identified accurately based on the phase-time contribution diagram method.

D. SUMMARY

In summary, the method proposed in this paper can better reveal the change of process variable's correlation, objectively reflect the diversity and uniqueness of the characteristics of each stage and transition process. Because there are obvious differences between different phases, the performance of process variables is that there are obvious differences in the mean and variance of process variables

between different stages. The monitoring model must accurately describe the characteristics of each phase. The idea of phase based modeling used in this paper satisfies this condition. It can effectively reduce the false alarm rate and miss alarm rate of the system. Especially when the fault occurs in the transitional stage, it reflects a higher fault recognition rate.

V. CONCLUSION

Fault monitoring of multi-phase batch process is a difficult problem in multivariate statistical process monitoring. It needs to consider not only the process monitoring under stable mode, but also the transition mode with strong dynamic nonlinearity. Since the data has different correlations under different operating modes, it is necessary to establish different monitoring models for each process mode, especially the transition process between stable modes. The most obvious feature is the dynamic characteristics of the variables. This feature can be better reflected in this transition using a time-varying covariance instead of a fixed covariance during the transition phase. In this paper, a new strategy for batch process sub-phase partition and process monitoring is proposed. Firstly, the three-dimensional data matrix is expanded into a new two-dimensional data according to the time slice expansion strategy. Secondly, the data of each time slice is transformed by KECA, and then the production process is divided into phases according to the spatial angle of the kernel entropy. The production operation process is divided into a stable phase and a transition phase, and monitoring models are respectively established to monitor the production process; Finally, the application of the penicillin fermentation simulation platform shows that the Sub-MKECA phase partition results can reflect the mechanism of the batch process well, and the fault monitoring of the multi-modal process shows that it can detect faults in time and accurately, and has high practicality value.

REFERENCES

[1] Z. Ge and Z. Song, *An Overview of Conventional MSPC Methods*. London, U.K.: Springer, 2013.
[2] Z. Ge, L. Xie, U. Kruger, and Z. Song, "Local ICA for multivariate statistical fault diagnosis in systems with unknown signal and error distributions," *AIChE J.*, vol. 58, pp. 2357–2372, Aug. 2012.
[3] X.-T. Doan and R. Srinivasan, "Online monitoring of multi-phase batch processes using phase-based multivariate statistical process control," *Comput. Chem. Eng.*, vol. 32, nos. 1–2, pp. 230–243, 2008.
[4] Y. Yao and F. Gao, "Phase and transition based batch process modeling and online monitoring," *J. Process Control*, vol. 19, no. 5, pp. 816–826, 2009.
[5] Y. Yao and F. Gao, "A survey on multistage/multiphase statistical modeling methods for batch processes," *Annu. Rev. Control*, vol. 33, no. 2, pp. 172–183, 2009.
[6] J.-M. Lee, C. K. Yoo, and I.-B. Lee, "Statistical process monitoring with independent component analysis," *J. Process Control*, vol. 14, no. 5, pp. 467–485, Aug. 2004.
[7] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.
[8] S. J. Qin, "Process data analytics in the era of big data," *AIChE J.*, vol. 60, no. 9, pp. 3092–3100, 2014.
[9] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Ind. Eng. Chem. Res.*, vol. 52, no. 10, pp. 3543–3562, 2013.
[10] P. Nomikos and J. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE J.*, vol. 40, pp. 1361–1375, Aug. 1994.
[11] T. Kourti and J. F. MacGregor, "Process analysis, monitoring and diagnosis, using multivariate projection methods," *Chemometrics Intell. Lab. Syst*, vol. 28, p. 3–21, Apr. 1995.
[12] S. Rännar, J. F. Macgregor, and S. Wold, "Adaptive batch monitoring using hierarchical PCA," *Ind. Eng. Chem. Res.*, vol. 43, no. 13, pp. 3343–3352, 2004.
[13] Z. Lv, X. Yan, and Q. Jiang, "Batch process monitoring based on just-in-time learning and multiple-subspace principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 137, pp. 128–139, Oct. 2014.
[14] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.
[15] P. Nomikos and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics Intell. Lab. Syst.*, vol. 30, no. 1, pp. 97–108, 1995.
[16] P. Kesavan, J. H. Lee, G. A. Krishnagopalan, and V. Saucedo, "Partial least squares (PLS) based monitoring and control of batch digesters," *J. Process Control*, vol. 10, nos. 2–3, pp. 229–236, 2000.
[17] S. Stubbs, J. Zhang, and J. Morris, "Multiway interval partial least squares for batch process performance monitoring," *Ind. Eng. Chem. Res.*, vol. 52, no. 35, pp. 12399–12407, 2013.
[18] J. Vanlaer, G. Gins, and J. F. M. Van Impe, "Quality assessment of a variance estimator for Partial Least Squares prediction of batch-end quality," *Comput. Chem. Eng.*, vol. 52, pp. 230–239, May 2013.
[19] Z. Ge, Z. Song, L. Zhao, and F. Gao, "Two-level PLS model for quality prediction of multiphase batch processes," *Chemometrics Intell. Lab. Syst.*, vol. 130, pp. 29–36, Jan. 2014.
[20] T. Kourti, "Application of latent variable methods to process control and multivariate statistical process control in industry," *Int. J. Adapt. Control Signal Process.*, vol. 19, no. 4, pp. 213–246,2005.
[21] C. K. Yoo, J.-M. Lee, P. A. Vanrolleghem, and I.-B. Lee, "On-line monitoring of batch processes using multiway independent component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 71, no. 2, pp. 151–163, 2004.
[22] L. Cai, X. Tian, and S. Chen, "A process monitoring method based on noisy independent component analysis," *Neurocomputing*, vol. 127, pp. 231–246, Mar. 2014.
[23] H. Guo and H. Li, "On-line batch process monitoring with improved multiway independent component analysis," *Chin. J. Chem. Eng.*, vol. 21, no. 3, pp. 263–270,2013.
[24] L. Wang and H. Shi, "Multivariate statistical process monitoring using an improved independent component analysis," *Chem. Eng. Res. Des.*, vol. 88, no. 4, pp. 403–414, 2010.
[25] G. Stefatos and A. B. Hamza, "Dynamic independent component analysis approach for fault detection and diagnosis," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8606–8617, 2010.
[26] N. Lu, F. Gao, and F. Wang, "Sub-PCA modeling and on-line monitoring strategy for batch processes," *AIChE J.*, vol. 50, no. 1, pp. 255–259, 2004.
[27] C. Zhao, F. Wang, M. Jia, and N. Lu, "Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes," *J. Process Control*, vol. 17, no. 9, pp. 728–741, 2007.
[28] F. Wang, S. Tan, Y. Chang, and J. Peng, "Process monitoring based on mode identification for multi-mode process with transitions," *Chemometrics Intell. Lab. Syst.*, vol. 110, no. 1, pp. 144–155, 2012.
[29] Y. Qi, P. Wang, and X. Gao, "Fault detection and diagnosis of multiphase batch process based on kernel principal component analysis-principal component analysis," *Control Therory Appl.*, vol. 29, no. 6, pp. 754–764, 2012.
[30] R. Jenssen, "Kernel entropy component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 847–860, May 2010.
[31] R. Jenssen, "Mean vector component analysis for visualization and clustering of nonnegative data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1553–1564, Oct. 2013.
[32] L. Gómez-Chova, R. Jenssen, and G. Camps-Valls, "Kernel entropy component analysis for remote sensing image clustering," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 2, pp. 312–316, Mar. 2012.
[33] R. Jenssen, "Entropy-relevant dimensions in the kernel feature space: Cluster-capturing dimensionality reduction," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 30–39, Jul. 2013.

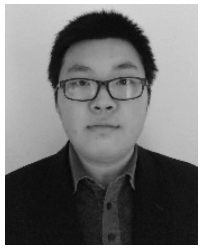
[34] E. Gokcay and J. C. Principe, "Information theoretic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 158–171, Feb. 2002.
[35] G. Birol, C. Ündey, and A. Çinar, "A modular simulation package for fed-batch fermentation: Penicillin production," *Comput. Chem. Eng.*, vol. 26, no. 11, pp. 1553–1565, 2002.

CHANG PENG received the B.S. and Ph.D. degrees from the College of Electronic Information and Control Engineering, Beijing University of Technology, China, in 2006 and 2015, respectively. In 2015, he joined the Beijing University of Technology, where he has been an Associate Professor, since 2016. His current research interest includes batch process monitoring and diagnosis.

QIAO JUNFEI (M'11) received the B.E. and M.E. degrees in control engineering from Liaoning Technical University, Fuxin, China, in 1992 and 1995, respectively, and the Ph.D. degree from Northeast University, Shenyang, China, in 1998.

From 1998 to 2000, he was a Postdoctoral Fellow with the School of Automatics, Tianjin University, Tianjin, China. He joined the Beijing University of Technology, Beijing, China, where he is currently a Professor. He is also the Director of the Intelligence Systems Laboratory, Beijing. His current research interests include neural networks, intelligent systems, self-adaptive learning systems, and process control systems. Prof. Junfei is also a member of the IEEE Computational Intelligence Society. He is also a Reviewer for more than 20 international journals, such as the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

ZHANG XIANGYU received the B.E. degree in measurement, control technology, and instrumentation from the Shandong University of Technology, Zibo, China, in 2016. He is currently pursuing the M.E. degree with the Beijing Key Laboratory of Computational Intelligence and Intelligent System, Faculty of Information Technology, Beijing University of Technology, Beijing, China. His current research interests include data-driven complex process monitoring, soft sensor modeling, data mining and analytics, and machine learning.



LU RUIWEI received the B.E. degree in electronic communication engineering and instrumentation from the Beijing University of Technology, Beijing, China, in 2017, where he is currently pursuing the M.E. degree with the Beijing Key Laboratory of Computational Intelligence and Intelligent System, Faculty of Information Technology. His current research interests include data-driven complex process monitoring, data mining and analytics, neural network, and system optimization.