http://pubs.acs.org/journal/acsodf

Article

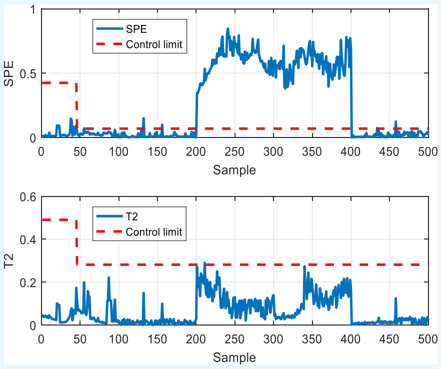
Efficient Iterative Dynamic Kernel Principal Component Analysis Monitoring Method for the Batch Process with Super-large-scale Data Sets

Yajun Wang,* Hongli Yu, and Xiaohui Li

Cite This: *ACS Omega* 2021, 6, 9989−9997

Read Online

ACCESS | Metrics & More | Article Recommendations



ABSTRACT: The Internet environment has provided massive data to the actual industrial production process. It not only has large amounts of data but also has a high data dimension, which brings challenges to the traditional statistical process monitoring. Aiming at the nonlinearity and dynamics of industrial large-scale high-dimensional data, an efficient iterative multiple dynamic kernel principal component analysis (IMDKPCA) method is proposed to monitor the complex industrial process with super-large-scale high-dimensional data. In KPCA, a new KK^T matrix is first created by using kernel matrix K. According to the properties of the symmetric matrix, the newly constructed matrix has the same eigenvector as the original matrix K; hence, each column of the matrix K can be used as the input sample of the iteration algorithm. After iterative operation, the kernel principal component can be deduced fleetly without the eigen decomposition. Because the kernel matrix is not stored in the algorithm beforehand, it can effectively reduce the computation complexity of the kernel. Especially for a tremendous data scale, the traditional eigen decomposition technology is no longer appropriate, yet the presented method can be solved quickly. The autoregressive moving average (ARMA) time series model and kernel principal component analysis (KPCA) are combined to build the IDKPCA model for dealing with the dynamics and nonlinearity in the industrial process. Eventually, it is applied to monitor faults in the penicillin fermentation process and compared with MKPCA to certify the accuracy and applicability of the proposed method.

■ INTRODUCTION

Batch processes have been exploited to manufacture high-value-added products in pharmaceutical, chemical, and semiconductor industries. To guarantee the high quality of production products and the safety and reliability of production process, multivariate statistical process monitoring has been broadly applied for monitoring batch processes[1−5] because complex process mechanisms do not need to be considered when establishing monitoring models, only process data are needed. For monitoring the nonlinear batch processes, representative methods include the multiway kernel principal component analysis (MKPCA)[6,7] and multiway partial kernel least square method (MKPLS).[8,9] However, a new problem arises in the kernel training process, which requires the calculation and storage of the kernel matrix. Because the square of the sampling point determines the dimension of the kernel matrix, the eigenvalue solution and the matrix inverse operation will lead to a huge amount of computation facing a large number of sample points, which will be very time consuming. For the dynamic characteristics of the process, Yu[10] proposed multiway discrete hidden Markov models to realize the classification and fault detection in the complicated batch process with intrinsic uncertainty and dynamics. Two-dimensional dynamic PCA[11,12] was presented to catch the dynamics within and between batches concurrently by depicting the batch process in a two-dimensional space. To further capture nonlinearity and dynamics in the process, Wang *et al.*[13] developed two-dimensional DKPCA to extract the foremost principal components. Nonetheless, the probability of the false positive error (type I error) enhances with the increase of the number of principal components used by the KPCA coding structure. In addition, the dimensionality of the kernel matrix is high. Jia *et al.*[14] developed a dynamic kernel principal component analysis (BDKPCA) monitoring method which combined the ARMAX time series models and kernel PCA. However, without adequately considering the differences between batches, only one model was built with all the batch data, as a result, the model is not very sensitive to fault monitoring. Wang *et al.*[15] proposed up-down double limit multimodel DPCA for the processes with

Received: December 11, 2020
Accepted: March 19, 2021
Published: April 6, 2021

https://doi.org/10.1021/acsomega.0c06039
ACS Omega 2021, 6, 9989−9997

finite modeling data sets. Nevertheless, when numerical changes of the normal new types of data are quite different from the original modeling data, the monitoring results are prone to false positives.

The emergence of various sensors can collect massive process data in industrial processes. New problems arise from establishing the monitoring model by using floods of process data collected online, such as redundant data, long time consumption, and large amount of calculation.[16,17] Big data analysis and processing brings new challenges to industrial process monitoring, as well as a key enabler.

For some complex industrial processes, different kinds of sensors embedded in the system collect a large amount of monitoring data in real time, thus large-scale data sets are obtained. A manifold-based machine learning approach was presented to excavate patterns in correlated, enormous, high-dimensional data.[18] Liu[19] developed a data-driven Takagi−Sugeno fuzzy model to model an actual plant situation with relevant inputs and nonlinear and time-varying input−output relationships. Principal component analysis eliminated the collinearity of inputs. Wang[20] proposed a multiscale neighborhood normalization-based multiple DPCA method to monitor the process running status and detect fault in complicated batch processes with relevant manipulations. This method can manage the strong non-Gaussian distribution problem. However, the computational process of scouting the nearest neighbors is intensive, especially for the large-scale data. Song *et al.*[21] presented the minimum-spanning tree (MST) method-based feature selection algorithm (FAST). The fast clustering-based technique of FAST can obtain useful and independent feature subset with high probability. Feature selection is beneficial to obtain an accurate data model and simplify calculation.[22] Kohlert and König[23] developed the one class classification (OCC) approach to monitor the industrial production process, which can obtain an accuracy as high as 99%. In recent years, the hybrid soft computing method has been applied and achieved quite a high degree of accuracy. However, the hybrid methods are fairly complicated because of the process optimization. Zaman and Hassan[24] developed an efficient hybrid recognition approach, which applies a fuzzy C-means method to the adaptive neuro-fuzzy reasoning system. The proposed approach can identify eight types of X-bar control chart patterns, which has been extensively surveyed and the accuracy rate can reach 99.82%. Chao[25] combined multivariate statistical analysis with a Bayesian inference method for large-scale high-dimensional process monitoring and proposed a stochastic optimization algorithm-based performance-driven process decomposition method. By decomposing the process, the best monitoring performance has been achieved. Unluckily, the methods mentioned above are not absolutely trustworthy and hard to guarantee the real-time monitoring performance.

Distinct from the methods mentioned above, an efficient iterative DKPCA modeling method is developed to monitor the batch processes with dynamics, nonlinearity, and large-scale high-dimensional data sets. The data analysis efficiency and modeling would be a challenge with the massive increase of a large amount of data. In the proposed method, the principal components can be obtained without decomposing the characteristic matrix, so as to improve the modeling efficiency and save the storage space.

The organization of the article is arranged as follows. The IPCA, KPCA, and IKPCA algorithms are first introduced sententiously. Next, the studied algorithm is developed according to the modeling and online monitoring. Ultimately, the proposed IMDKPCA monitoring method is applied in a representative chemical process, and comparison is demonstrated with the MKPCA monitoring method. The conclusions of this work are summarized in the last section.

Iterative Kernel Principal Component Analyses.

Iterative Principal Component Analysis. The first principal component is first considered. The sample vectors $y(1)$, $y(2)$,... are infinite, each of which is a d-dimensional vector. The sample is zero mean and A is the covariance matrix of $d \times d$, then $A = \{y(m)\ y^{\mathrm{T}}(m)\}$. Assuming that the eigenvector of matrix A is x, then $\lambda x = Ax$, where λ is the corresponding eigenvalue, the covariance matrix is brought into A. In the iterative process, $x(i)$ is substituted for x in each step, and the following expression can be obtained by $v = \lambda x$.

$$v(m) = \frac{1}{m}\sum_{i=1}^{m} y(i)y^{\mathrm{T}}(i)x(i) \tag{1}$$

where $v(m)$ is the estimation of v in step m, and the equation is mainly determined by the statistical efficiency. If the estimation of v is ideal, the eigenvalues $\lambda = \|v\|$ and eigenvectors $x = v/\|v\|$ can be calculated. In eq 1, $v(i-1)/\|v(i-1)\|$ can be selected instead of $x(i)$, and the following incremental expression can be derived

$$v(m) = \frac{1}{m}\sum_{i=1}^{m} y(i)y^{\mathrm{T}}(i)\frac{v(i-1)}{\|v(i-1)\|} \tag{2}$$

First, setting the first direction $v(0) = v(1)$ of data propagation, the incremental estimation eq 2 can be expressed in the recursive form

$$v(m) = \frac{m-1}{m}v(m-1) + \frac{1}{m}y(m)y^{\mathrm{T}}(m)\frac{v(m-1)}{\|v(m-1)\|} \tag{3}$$

where $(m-1)/m$ is the weight of the original estimate and $1/m$ is the weight of the newly acquired data. It can be derived that $v_1(m) \to \pm\lambda_1 e_1$ for $m \to \infty$, thus, the eigenvalue of the largest covariance matrix $y(m)$ is expressed as λ_1 and the corresponding eigenvector is expressed as e_1.

Equation 3 can estimate the first eigenvector. For other higher order eigenvectors, stock gradation ascent (SGA) is used. Starting from a set of normalized vectors, they are updated using iterative steps, and orthogonality is restored using the Gram−Schmidt orthogonalization method (GSO). To calculate the second-order eigenvector, the projection needs to be subtracted from the data, and then, the new data are used to estimate the second-order eigenvector, as shown below

$$y_2(m) = y_1(m) - y_1^{\mathrm{T}}(m)\frac{v_1(m)}{\|v_1(m)\|}\frac{v_1(m)}{\|v_1(m)\|} \tag{4}$$

where $y_1(m) = y(m)$, the residual is $y_2(m)$, which is the complementary space of v_1, and can be used as the input iterative data.

The steps of applying an iterative method to solve the principal element are as follows.

Input: $y(1)$, $y(2)$,...

Output: the first r principal components.

For m = 1,2,...

(1) Make $y_1(m) = y(m)$;

(2) Loop iteration for i = 1 to min(r,m)

When $i = m$, initialize the ith principal element, and $v_1(m) = y_i(m)$

When $i \neq m$

$$v_i(m) = \frac{m - l - 1}{m} v_i(m - 1) + \frac{1 + l}{m} y_i(m) y_i^{\mathrm{T}}(m) \frac{v_i(m - 1)}{\|v_i(m - 1)\|} \tag{5}$$

$$y_{i+1}(m) = y_i(m) - y_i^{\mathrm{T}}(m) \frac{v_i(m)}{\|v_i(m)\|} \frac{v_i(m)}{\|v_i(m)\|} \tag{6}$$

where l is the forgetting factor. If the forgetting factor is not added, then

$$v_i(m) = \frac{m - 1}{m} v_i(m - 1) + \frac{1}{m} y_i(m) y_i^{\mathrm{T}}(m) \frac{v_i(m - 1)}{\|v_i(m - 1)\|} \tag{7}$$

where $(m - 1)/m$ is the weight of the original estimate and $1/m$ is the weight of the newly acquired data. v_i is the estimate of $\lambda_i e_i$ at a certain time, then, the eigenvector $e_i = v_i/\|v_i\|$ and the eigenvalue $\lambda_i = \|v_i\|$. In this method, for each new sample data, the first principal component estimation is updated and the next principal component is updated in turn.

Iterative Kernel Principal Component Analysis. *KPCA.* KPCA is a nonlinear extension of the PCA algorithm. First, the input space is projected to a feature space by nonlinear projecting. PCs are followed by extraction from the projected feature space. Assuming that the input sample data $x_i \in R^m$, $i = 1,...,m$, are projected to a feature space S. The dimension of feature space S can be arbitrarily large or endless. Principal components of KPCA are calculated by solving the eigenvalue problem

$$\lambda v = C^{\mathrm{F}} v \tag{8}$$

$$C^{\mathrm{F}} = \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j) \Phi(x_j)^{\mathrm{T}} \tag{9}$$

where C^{F} is the covariance matrix, eigenvalue $\lambda \geq 0$. eq 8 can be written with the kernel matrix

$$m\lambda\alpha = K\alpha \qquad \alpha = [\alpha_1, ..., \alpha_m]^{\mathrm{T}} \tag{10}$$

where K is the Gram matrix, which is defined as $[K_{ij}] = K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle$. Furthermore, by computing the projection on the feature vector, we can obtain the score t of the nth sampling point x, as shown below

$$t_n = \langle v_n, \Phi(x_n) \rangle = \sum_{i=1}^{m} \alpha_i^{\ n} \langle \Phi(x_i), \Phi(x) \rangle = \sum_{i=1}^{m} \alpha_i^{\ n} k(x_i, x)$$
$$n = 1, ..., m \tag{11}$$

Iterative KPCA. When using the iterative method to solve the kernel principal component, the following properties of linear algebra are used.

$$K\omega = \lambda\omega$$

$$K^2\omega = KK\omega = \lambda K\omega = \lambda^2\omega$$

where ω and λ are the eigenvectors and eigenvalues corresponding to the matrix K, respectively. It can be seen that matrix K and K^2 have the same eigenvector, while the eigenvalues $\lambda_{K^2} = (\lambda_{K^2})/m$ are different, and m is the dimension of matrix K. Due to

$$KK^{\mathrm{T}} = K^2 = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}$$
$$= (K(x_1), K(x_2), ..., K(x_m))(K(x_1), K(x_2), ..., K(x_m))^{\mathrm{T}}$$
$$= \sum_{i=1}^{m} K(x_i) K(x_i)^{\mathrm{T}} \tag{12}$$

where $K(x_i) = (k_{i1}, k_{i2}, ..., k_{im})^{\mathrm{T}}$. If each column $K(x_i)$ of matrix K is regarded as the input sample of the iterative algorithm, the eigenvector of matrix KK^{T} can be obtained quickly after a series of iterations, and the eigenvector and eigenvalue need not be solved by eigen decomposition matrix K. According to the algebraic knowledge, assuming that the eigenvector corresponding to matrix K^2 is U_{K^2} and the eigenvalue is λ_{K^2}, then

$$\omega(n) = \lambda_{K^2} U_{K^2} = K^2 U_{K^2} \tag{13}$$

where $\omega(n)$ is the estimate of the eigenvector at time n. After updating the estimation step by step, the eigenvector $U_{K^2} = \omega/\|\omega\|$ and eigenvalue $\lambda_{K^2} = \|\omega\|$ can be obtained. In the kernel space, because the input samples are $K(x_1)$, $K(x_2)$,...,$K(x_m)$, the new sample $K(x_i)$ can be substituted into the iterative algorithm as input data in turn, in this way, the estimated value of the ith order principal component at time n can be expressed as

$$\omega_i(n) = K^2 U_{K^2} = \frac{1}{n} \sum_{t=1}^{n} K_i(x_t) K_i^{\mathrm{T}}(x_t) \frac{\omega_i(t - 1)}{\|\omega_i(t - 1)\|}$$
$$= \frac{1}{n} \sum_{t=1}^{n-1} K_i(x_t) K_i^{\mathrm{T}}(x_t) \frac{\omega_i(t - 1)}{\|\omega_i(t - 1)\|} + \frac{1}{n} K_i(x_n) K_i^{\mathrm{T}}(x_n) \frac{\omega_i(n - 1)}{\|\omega_i(n - 1)\|}$$
$$= \frac{n - 1}{n} \omega_i(n - 1) + \frac{1}{n} K_i(x_n) K_i^{\mathrm{T}}(x_n) \frac{\omega_i(n - 1)}{\|\omega_i(n - 1)\|} \tag{14}$$

where $K_i(x_t)$ is the input sample, and its main function is to update the first-order principal component at time t. When calculating high-order principal components, the residual data should be used as the input data for the iterative calculation. The projection of the original data on the low-order eigenvector should be first removed from the residual input data, where $K_1(x_n) = K(x_n)$, the calculation formula is

$$K_{i+1}(x_n) = K_i(x_n) - K_i(x_n)^{\mathrm{T}} \frac{\omega_i(n)}{\|\omega_i(n)\|} \frac{\omega_i(n)}{\|\omega_i(n)\|} \tag{15}$$

According to the above method, the required principal element and eigenvalue can be obtained after a series of iterative calculations.

Modeling and Monitoring Based on Iterative MDKPCA. *Establishment of the Iterative MDKPCA Model.* Considering the dynamics of the process, a ARMA time series model is established. The ARMAX regression model for batch i is listed as follows at sampling time a.

$$X_i(a) = [X_i(a)^{\mathrm{T}} X_i(a - 1)^{\mathrm{T}} X_i(a - d)^{\mathrm{T}}] \tag{16}$$

where d describes the delay duration. Similarly, the time-lagged augmented matrix of the whole batch i can be built

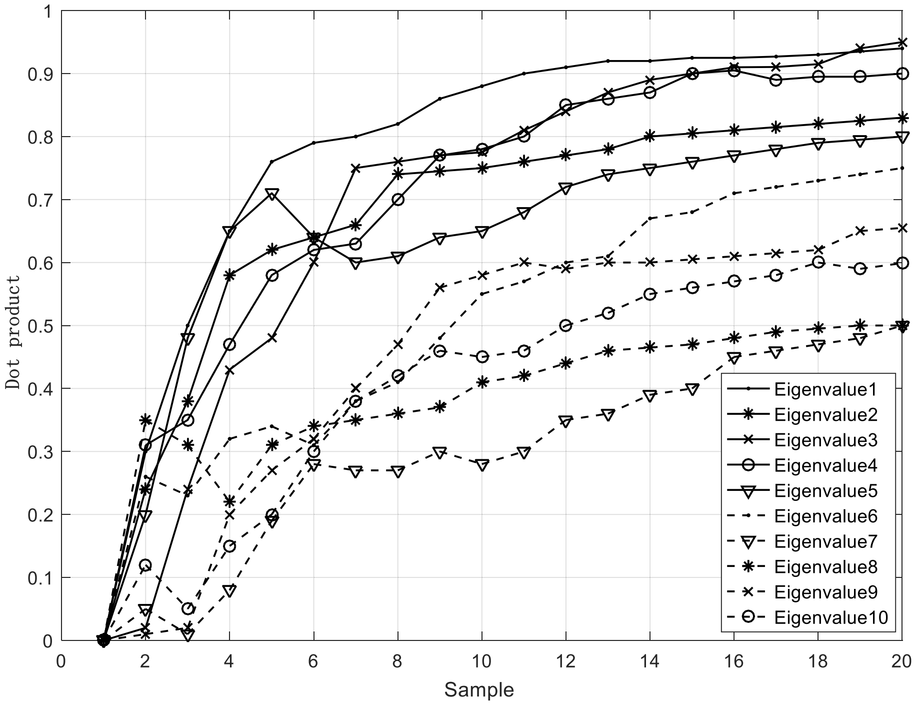


Figure 1. Influence of the forgetting factor on the first 10 eigenvectors when $l = 0$.

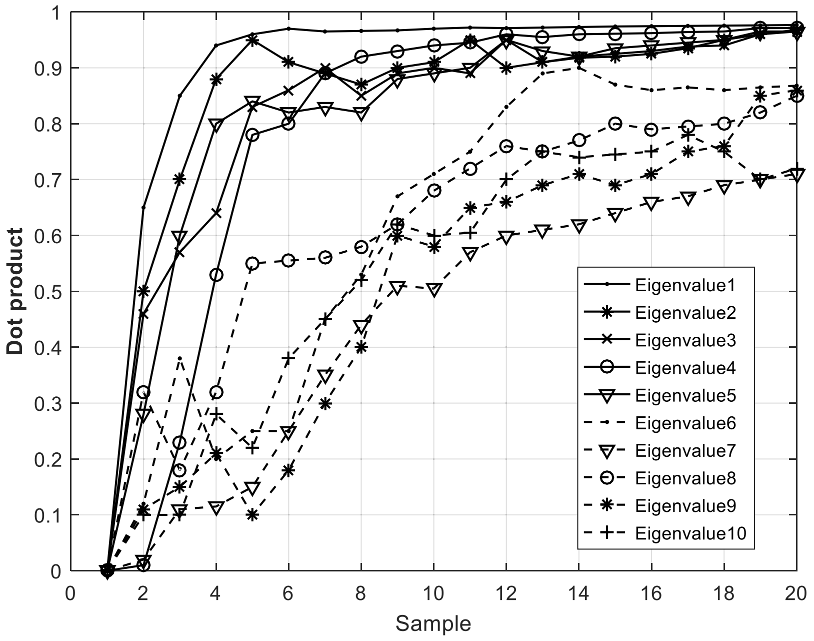


Figure 2. Influence of the forgetting factor on the first 10 eigenvectors when $l = 2$.

$$X_i^d = \begin{bmatrix} x_i^{\mathrm{T}}(a) & x_i^{\mathrm{T}}(a-1) & \cdots & x_i^{\mathrm{T}}(a-d) \\ x_i^{\mathrm{T}}(a-1) & x_i^{\mathrm{T}}(a-2) & \cdots & x_i^{\mathrm{T}}(a-d-1) \\ \vdots & \vdots & \cdots & \vdots \\ x_i^{\mathrm{T}}(a-(K_i-d)) & x_i^{\mathrm{T}}(a-(K_i-d)-1) & \cdots & x_i^{\mathrm{T}}(a-K_i) \end{bmatrix} \tag{17}$$

where selection method of d can be searched in the literature.[26] Next iterative KPCA is applied to the model. Because the model is established at each stage, the established model is called the iterative MDKPCA model.

Because the influence factor l has an influence on convergence, its value is generally taken as 0 or 2. Use 10 eigenvectors as an example to analyze l. When $l = 0$, as shown in Figure 1, the convergence speed of the feature vector is relatively slow and when $l = 2$, as shown in Figure 2, the convergence speed is relatively fast. This is due to the larger forgetting factor selected, the contribution of historical samples to the eigenvector is smaller, and the contribution of new samples to the eigenvector is larger; hence, the convergence is faster. If the forgetting factor is small, the contribution of the new sample to the principal component is small, in this way, the final principal component estimation can well contain the main information.

However, l cannot be infinite. If the features of the entire sample are mainly concentrated in the historical sample, the forgetting factor will lead to an overmuch weakening of the feature vector in the early stage, which will eventually make the obtained principal component estimation inaccurate. In order to make the principal component obtained by the iterative method contain the original data information as much as possible, the forgetting factor is chosen as 0.

Figure 3 shows the first 10 eigenvalues of a part of sample data obtained by the iterative method. The eigenvalue corresponding to the first eigenvector is the largest, while the others gradually decrease. The cumulative contribution rate of the first 9 eigenvalues is over 85% by the cumulative contribution rate method; therefore, the first nine principal components need to be calculated iteratively, while the latter need not. At this time, the first 9 principal components can reflect the main information of the tested sample. According to this idea, only the first r principal components need to be calculated iteratively during the iterative calculation.

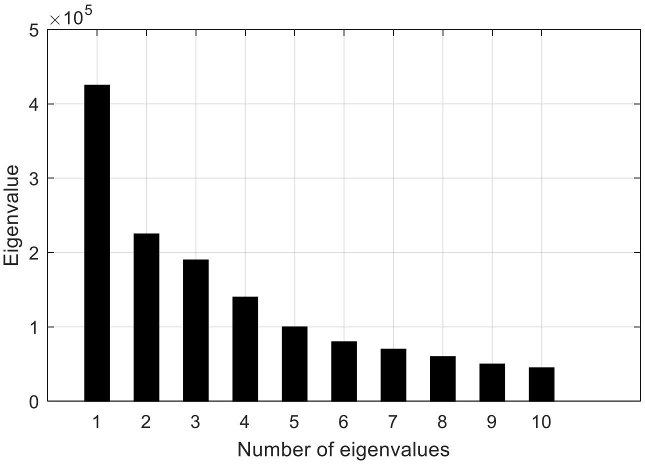


Figure 3. Size of the first 10 eigenvalues.

The convergence of eigenvalues is verified by experiments with $\|v_i\|/\lambda_i$, where λ_i, $i = 1,2,...,10$, are the eigenvalues. Figure 4 is the convergence diagram of eigenvalues. The convergence of eigenvalues approaches to 1 with the increase of the sample data by applying the iterative method to solve the principal component, as shown in Figure 4, which shows that the kernel principal component and eigenvalues can be completely solved by the iterative method when the samples are large enough.

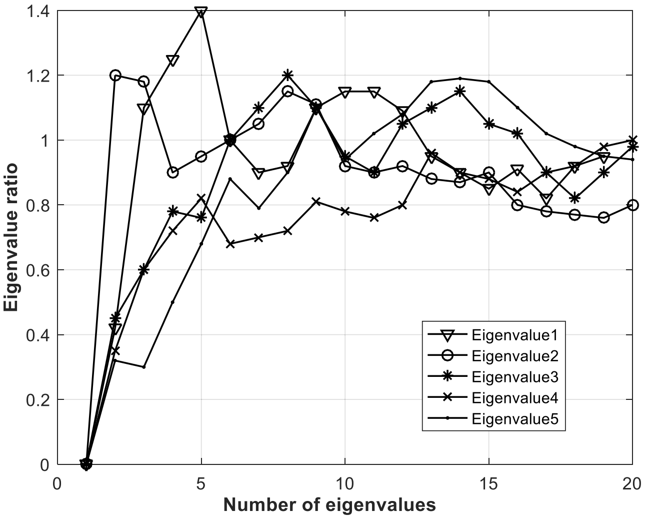


Figure 4. Convergence accuracy of eigenvalues.

In online monitoring, two typical statistics, Hotelling T^2 and SPE, are commonly applied to detect the operating status of the industrial production process, as listed below.

$$T^2 = [t_1, ..., t_r]\Lambda^{-1}[t_1, ..., t_r]^{\mathrm{T}} \tag{18}$$

where t is the score, r is the number of PCs, and $\Lambda^{-1}$is the inverse of the diagonal matrix, which is obtained by solving eigenvalues corresponding to the first r eigenvectors.

The control limit of Hotelling T^2 can be achieved using the F-distribution.

$$T^2 \sim \frac{r(n-1)}{n-r} F_{r,n-r,\alpha} \tag{19}$$

where n is the number of sample points. $F_{r,n-r,\alpha}$ is the upper limit value of an F-distribution with a degree of freedom r and $n - r$ with level of significance α.

SPE is obtained by

$$\mathrm{SPE} = \left\|\Phi(x) - \hat{\Phi}_p(x)\right\|^2 = \sum_{j=1}^{n} t_j^2 - \sum_{j=1}^{p} t_j^2 \tag{20}$$

The control limit of SPE can be achieved using χ^2 distribution.

$$\mathrm{SPE}_\alpha \sim g\chi_h^2 \qquad g = \frac{u}{2v} \qquad h = \frac{2v^2}{u} \tag{21}$$

where u and v are the statistical mean and variance of SPE statistical values.[27]

Modeling and Monitoring Procedures. Modeling program of iterative MDKPCA is as follows.

(1) Under normal operation, I batches' data $X_i = R^{J\times M}$, $i = 1,...I$, are collected as training samples.

(2) Stage division. The data of each stage is expressed as X_is, $s = 1,2,..., q$, q is the stage number.

(3) Established time-lagged augmented matrices with the whole batch data in each stage.

(4) Unfolded time-lagged augmented matrices by the variable-wise method successively.

(5) Utilized the iterative MDKPCA method to establish the model.

(6) Calculated the two statistics Hotelling T^2 and SPE (eqs 18 and 20) and solved the control limits of Hotelling T^2 and SPE (eqs 19 and 21).

The online application process is summarized as follows.

(1) For a new test sample $x_{\text{test}} \in R^{1\times J}$, determined the belonged stage of the new test sample.

(2) Established the time series augmented matrix x_{test}d and standardized it by employing the mean and standard deviation of the corresponding model.

(3) Substituted the standardized x_{test} into the stage model and then computed T^2 and SPE.

(4) Used the two calculated statistical values of T^2 and SPE and the corresponding control limits to declare a fault if any three continuous values of either T^2 or SPE go beyond the control limits.

RESULTS AND DISCUSSION

The penicillin fermentation process is a representative multistage and nonlinear dynamic production process. In this work, the developed iterative MDKPCA approach is applied to detect the various faults in fed-batch penicillin production. The massive modeling and monitoring data are generated by a penicillin fed-batch simulator PenSim v2.0. The PenSim v2.0 can simulate the velocity of the input flow, aeration rate, pH, temperature, heat generated, CO_2, substrate utilization, and concentrations of penicillin under different conditions.

Experimental Design and Modeling Data. In this work, a total of 40 batches were simulated to generate the normal modeling batches. The ten variables monitored are listed in Table 1. The duration of each batch is 500 h. Every batch consists of two stages: the pre-culture stage and the fed-batch stage. In the data generated by the simulator, the sampling interval of the modeling data is 0.05 h, and the sampling interval of the online monitoring data is set to be1.0 h. Slight changes are added to mimic the actual process perturbations and random features under the normal running status. The ranges of initial conditions and set points used to simulate the penicillin fermentation process are given in Table 2.

Table 1. Monitored Variables in the Penicillin Fermentation Process

no.	monitored variables
1	cooling water flow rate/h^{-1}
2	pH
3	dissolved oxygen concentration/% (saturation)
4	substrate feed temperature/K
5	culture volume/L
6	bioreactor temperature/K
7	substrate concentration/gL^{-1}
8	agitator power/W
9	carbon dioxide concentration/mmol L^{-1}
10	aeration rate/h^{-1}

On-Line Monitoring MKPCA and Iterative MDKPCA. As a comparison, the measurement data sets can be used to construct the MKPCA and iterative MDKPCA models. In this work, the high-dimensional feature space projection uses the second-order polynomial kernel function because it is more conducive to capturing the nonlinearity of the considered system. On-line monitoring is carried out with a control limit of 95% confidence level. The delay durations are 1 and 2 in the first and second stages for iterative MDKPCA, respectively. For MKPCA, thirty-five and sixty-four PCs are solved by the average eigenvalue method for the two stages, which can express 99.78 and 99.97% variations, respectively. For iterative MDKPCA, thirty-two and sixty principal components are selected for the two stages, which can express 99.6 and 99.94% variations, respectively.

To illustrate the monitoring performance of MKPCA and iterative MDKPCA, four scenarios including three types of process faults and one normal operation are designed and examined with MKPCA and iterative MDKPCA, as shown in Table 3. (1) Normal batch, (2) linearly decreased aeration rate, (3) step-decreased agitation power, and (4) linearly decreased glucose feed rate.

Table 3. Four Test Cases in Penicillin Fermentation

no.	scenarios	fault description
1	normal	normal batch
2	ramp fault 1	linearly decreased aeration rate at a slope of 0.05 from 250 h to the end of batch operation.
3	step fault 2	step decreased agitation power by 5% from 200 to 400 h of batch operation.
4	ramp fault 3	linearly decreased substrate feed rate at a slope of 0.002 from 200 h to the end of batch operation.

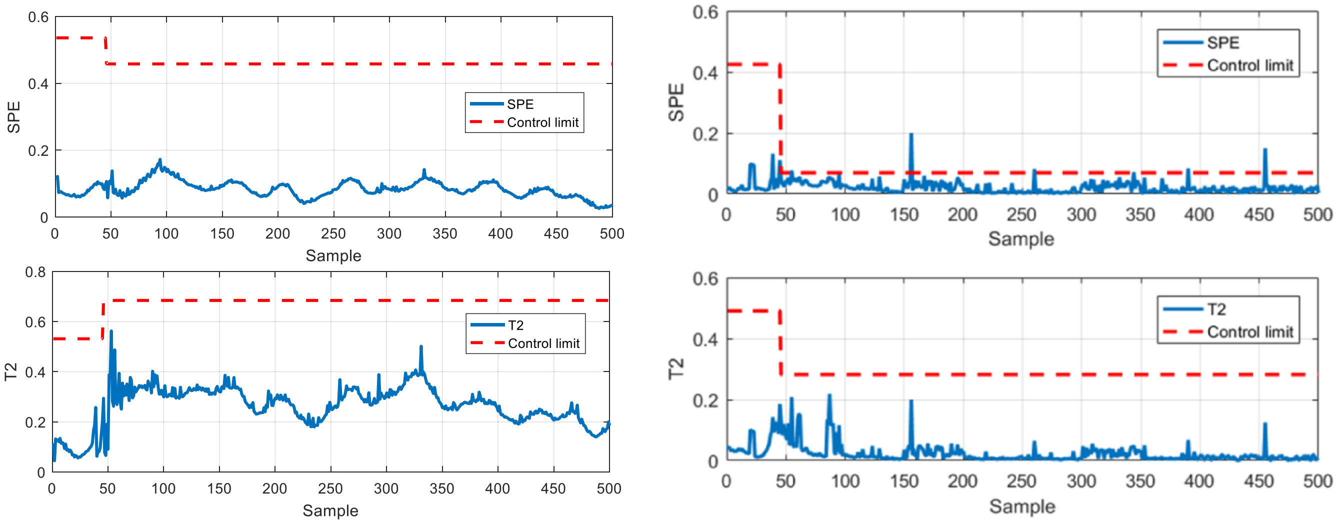
In the first test scenario, it is a normal batch. The test results of MKPCA and iterative MDKPCA methods are shown in Figure 5, respectively. It shows that the entire batch is not beyond the control limits for the two methods from beginning to end, which indicates that the first test process is normal. Two empirical models, MKPCA and iterative MDKPCA, can be used to describe the fermentation trajectory.

In the second test case, the ramp error in the aeration rate occurs from the 250th h and lasts to the 500th h of the batch operation. The test results of the two models are shown in Figure 6. In the case of MKPCA, the corresponding SPE and T^2 surpass the control limits at 266 and 255 h, respectively. Comparing the actual fault generation time points, the delays are 16 and 5 h, respectively, as shown in Figure 6a.Compared with MKPCA, the Hotelling's T^2 control chart and SPE control chart of iterative MDKPCA exceed control limits at the 252nd h.

For the second fault batch, the step error occurs in the agitation power from the 200th to the 400th h of batch operation. The agitation power determines the quality level of oxygen dissolved in the fermentation broth, and the decrease of agitation power reduces the oxygen content in the fermentation broth, which reduces the product concentration. The test results are shown in Figure 7. For MKPCA, SPE fails to identify this step fault, while T^2 only identifies the fault in a small part of the time during the fault, as shown in Figure 7a. In contrast, the SPE chart of the iterative MDKPCA method can accurately identify the beginning and end time of the fault. Unfortunately, the test

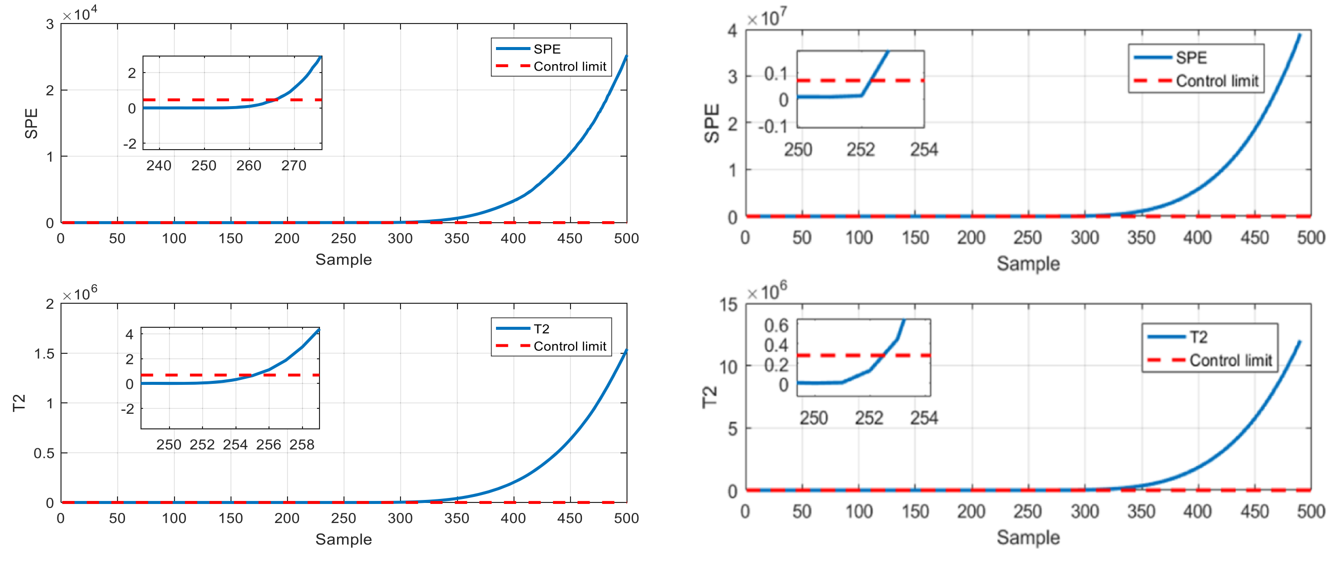
Table 2. Ranges of Initial Conditions and Set Points of Operation Parameters

initial conditions	range	set points	range
dissolved oxygen concentration (mmol/L)	1.05−1.25 (g/L)	bioreactor temperature	295−299 (K)
penicillin concentration	0 (g/L)	agitator power	28.5−31.5 (W)
biomass concentration	0.05−0.1 (g/L)	substrate feed flow rate	0.036−0.042 (L/h)
substrate concentration	13−18 (g/L)	substrate feed temperature	296−299 (K)
culture volume	100−104 (L)	aeration rate	8−9 (g/h)
bioreactor temperature	298−299 (K)	pH	5.1−5.2
carbon dioxide concentration (mmol/L)	0.5−0.6 (g/L)		
PH	4.5−5.5		
generated heat	0 (kcal)		



(a) (b)

Figure 5. Monitoring results of the normal batch. (a) SPE and T^2 control charts of MKPCA. (b) SPE and T^2 control charts of iterative MDKPCA.



(a) (b)

Figure 6. Monitoring results of fault 1. (a) SPE and T^2 control charts of MKPCA. (b) SPE and T^2 control charts of iterative MDKPCA.

results of T^2 also fail to identify the fault. From the overall monitoring results, the iterative MDKPCA method can still identify this fast changing fault.

For the third fault batch, the ramp error in the substrate feed rate is added to the fed-batch operation in the 200th h and remains for 300 h. The substrate feed rate is the main source of carbon in the reaction, which can increase the biomass, penicillin synthesis efficiency, and maintain metabolism. If the acceleration rate of the bottom stream decreases, the yield and efficiency of the penicillin fermentation synthesis will also decrease. The test results are shown in Figure 8. The SPE chart and T^2 chart of MKPCA overstep the control limits at 209 and 210 h, which are lagged behind 9 and 10 h, respectively. Compared with MKPCA, the two statistics SPE and T^2 of iterative MDKPCA are beyond the control limits at time 201 h, which is 1 h later than the time when the fault occurs.

Computational Complexity Analysis. The computational complexity of the two algorithms is analyzed by the time consumed in the process. There are 40 batches of modeling data, and the running time is 500 h. The matrix dimension is 1800 × 10 after expansion by the variables for the first stage; in the second stage, the dimension of the matrix is 18,200 × 10 after expansion by variables. For the MKPCA method, the dimension of the kernel matrix in the first stage can reach 1800 × 1800, and in the second stage can reach 18,200 × 18,200. When using the MKPCA method to solve principal components, it is indispensable to decompose the covariance matrix, so the amount of computation becomes very large. When using the iterative MDKPCA method to solve the principal components of the kernel space matrix, we only need to take the kernel matrix as a sample to input data in turn, and then iteratively solve the corresponding principal components in the two stages.

The running time is selected as the comparison standard, and the simulation environment is MATLAB 16; CPU: Intel(R) Core(TM) i7-8565U; memory: 16 G; basic frequency: 2.19 G; operating system: Windows 10. The running time is mainly

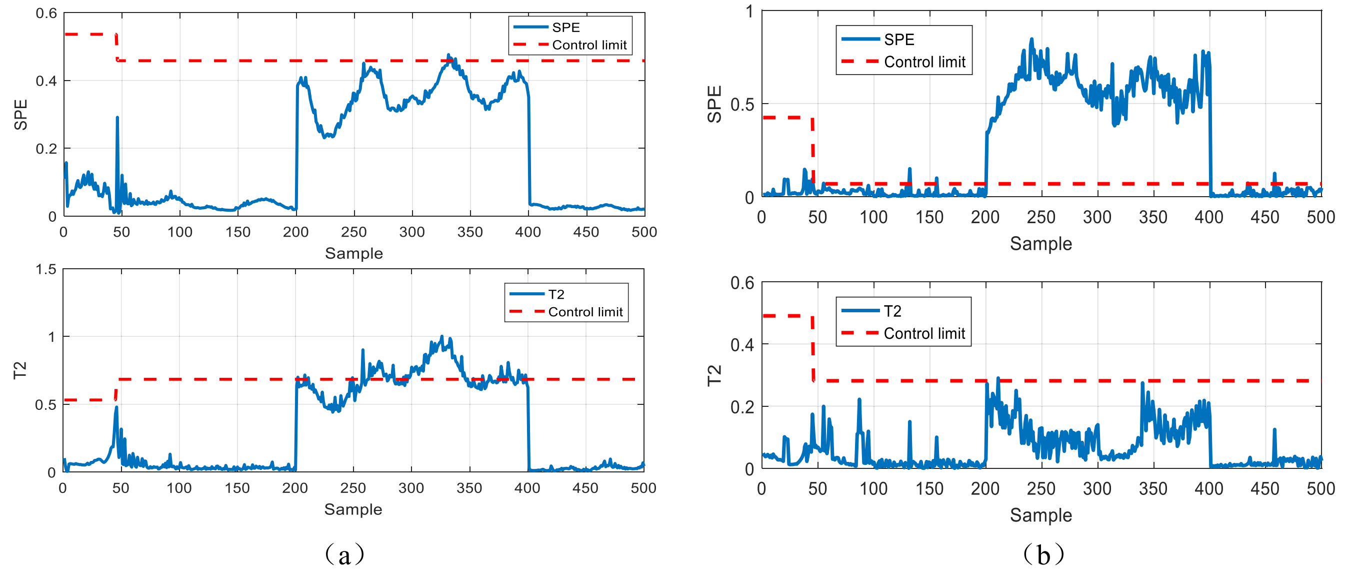


Figure 7. Monitoring results of fault 2. (a) SPE and T^2 control charts of MKPCA. (b) SPE and T^2 control charts of iterative MDKPCA.

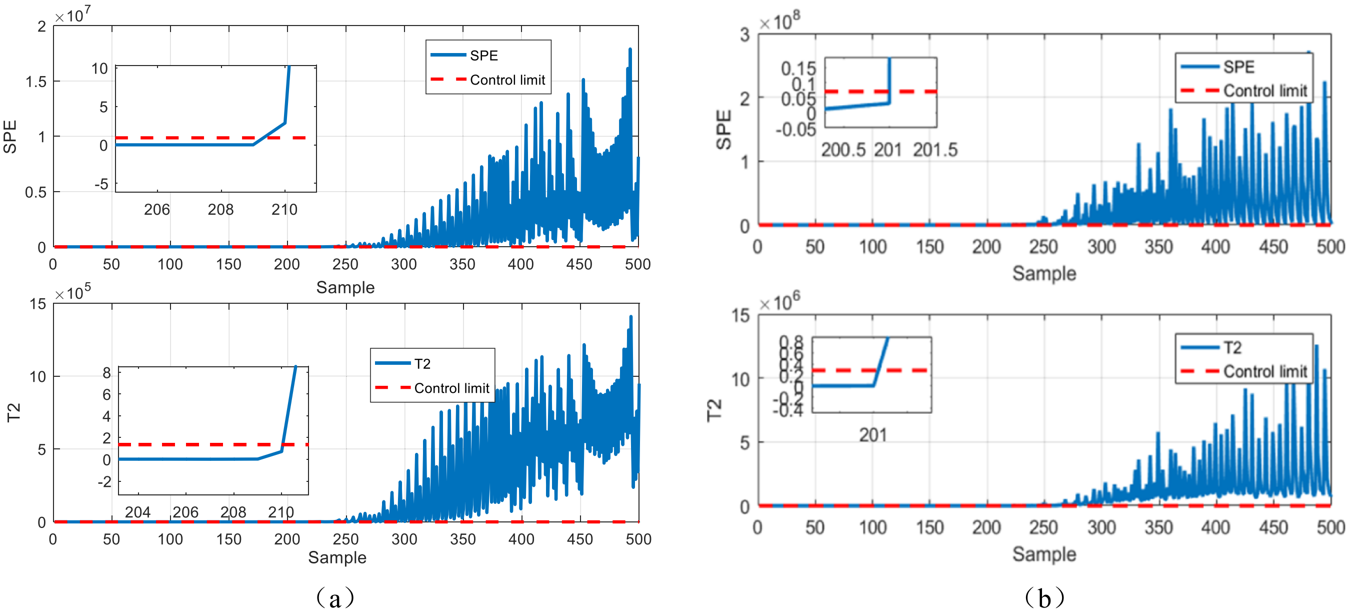


Figure 8. Monitoring results of fault 3. (a) SPE and T^2 control charts of MKPCA. (b) SPE and T^2 control charts of iterative MDKPCA.

obtained by internal instructions. Table 4 shows the running results.

As seen in Table 4, MKPCA takes more time in each stage than the iterative MDKPCA method, especially in the modeling stage. Moreover, the modeling time of the second stage is as high as about 255 h, which is much higher than that in the iterative MDKPCA method.

Table 4. Time Comparison Between Two Modeling Monitoring Methods

	modeling				
method	stage 1 (s)	stage 2	fault 1 (s)	fault 2 (s)	fault 3 (s)
MKPCA	101.8	254.7 h	17.1	16.9	17.2
Iterative MDKPCA	12.1	312.7 s	4.3	4.9	5.0

■ CONCLUSIONS

Aiming at the high computational complexity problem in the complicated industrial processes with large-scale high-dimensional data sets, an efficient iterative multiple DKPCA monitoring method is presented successfully. In this method, the principal component is calculated by using the iterative technique of statistical efficiency estimation. Using sample data to input and iterate one by one, the principal component information is obtained without using the feature decomposition of the sample covariance matrix to solve the feature vector, which greatly reduces the calculation time in the process of modeling and monitoring, reduces the storage space, and improves the work efficiency.

The proposed iterative MDKPCA approach is used to detect the different faults of the penicillin fermentation process. The experimental results show that the developed iterative approach has a better monitoring effect than the conventional MKPCA approach in the short delay, high accuracy of various types of faults. Besides, this method also greatly reduces the computation, storage space, and running time of the algorithm, particularly these advantages are more obvious when the amount of data is enormous. On this basis, our future work will study the determination of the process state evaluation grade and the identification of nonoptimal reasons.

AUTHOR INFORMATION

Corresponding Author

Yajun Wang – *School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China*; orcid.org/0000-0002-1164-2768; Email: wyj_lg@163.com

Authors

Hongli Yu – *School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China*

Xiaohui Li – *School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.0c06039

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (grant: 61503169, 61802161) and Natural Science Foundation of Liaoning province (grant: 2020-MS-291).

REFERENCES

(1) Gao, Z.; Jia, M.; Mao, Z.; Zhao, L. A new monitoring method for the between-mode transition of multimode processes. *Can. J. Chem. Eng.* **2020**, *98*, 254–262.

(2) Rato, T.; Reis, M.; Schmitt, E.; Hubert, M.; De Ketelaere, B. A systematic comparison of PCA-based Statistical Process Monitoring methods for high-dimensional, time-dependent Processes. *AIChE J.* **2016**, *62*, 1478–1493.

(3) Luo, L. Monitoring uneven multistage/multiphase batch processes using trajectory-based fuzzy phase partition and hybrid MPCA models. *Can. J. Chem. Eng.* **2019**, *97*, 178–187.

(4) Wang, Y.; Sun, F.; Li, X. Compound dimensionality reduction based multi-dynamic kernel principal component analysis monitoring method for batch process with large-scale data sets. *J. Intell. Fuzzy Syst.* **2020**, *38*, 471–480.

(5) Zhang, S.; Zhao, C.; Wang, S.; Wang, F. Pseudo time-slice construction using variable moving window-k nearest neighbor (VMW-kNN) rule for sequential uneven phase division and batch process monitoring. *Ind. Eng. Chem. Res.* **2017**, *56*, 728–740.

(6) Lee, J.-M.; Yoo, C.; Lee, I.-B. Fault detection of batch processes using multi-way kernel principal component analysis. *Comput. Chem. Eng.* **2004**, *28*, 1837–1847.

(7) Cao, D.-S.; Liang, Y.-Z.; Xu, Q.-S.; Hu, Q.-N.; Zhang, L.-X.; Fu, G.-H. Exploring nonlinear relationships in chemical data using kernel-based methods. *Chemometr. Intell. Lab. Syst.* **2011**, *107*, 106–115.

(8) Luo, L.; Bao, S.; Mao, J.; Tang, D. Nonlinear process monitoring based on kernel global-local preserving projections. *J. Process Control* **2016**, *38*, 11–21.

(9) Zhang, Y.; Teng, Y.; Zhang, Y. Complex process quality prediction using modified kernel partial least squares. *Chem. Eng. Sci.* **2010**, *65*, 2153–2158.

(10) Yu, J. Multiway discrete hidden Markov model-based approach for dynamic batch process monitoring and fault classification. *AIChE J.* **2012**, *58*, 2714–2725.

(11) Lu, N.; Yao, Y.; Gao, F.; Wang, F. Two-dimensional dynamic PCA for batch process monitoring. *AIChE J.* **2005**, *51*, 3300–3304.

(12) Yao, Y.; Chen, T.; Gao, F. Multivariate statistical monitoring of two-dimensional dynamic batch processes utilizing non-Gaussian information. *J. Process Control* **2010**, *20*, 1188–1197.

(13) Wang, T.; Wang, X.; Zhang, Y.; Zhou, H. Fault Detection of Nonlinear Dynamic Processes Using Dynamic Kernel Principal Component Analysis. *Proceedings of the 7th World Congress on Intell. Control Automation*; 2008; pp 3009–3014.

(14) Jia, M.; Chu, F.; Wang, F.; Wang, W. On-line batch process monitoring using batch dynamic kernel principal component analysis. *Chemometr. Intell. Lab. Syst.* **2010**, *101*, 110–122.

(15) Wang, Y.-J.; Jia, M.-X.; Mao, Z.-Z. A fast monitoring method for multiple operating batch processes with incomplete modeling data types. *J. Ind. Eng. Chem.* **2015**, *21*, 328–337.

(16) Hans, N.; Mahajan, S.; Omkar, S. N. Big Data Clustering Using Genetic Algorithm On Hadoop Mapreduce. *Int. J. Sci. Technol. Educ. Res.* **2015**, *4*, 58–62.

(17) Chen, Y. E.; Ren, B. L. Research on Large Scale Data Set Processing Based on SVM. *Adv. Mater. Res.* **2011**, *216*, 738–741.

(18) Guo, J. K.; Hofmann, M. O. Interactive Pattern Discovery in High-Dimensional, Multimodal Data Using Manifolds. *Procedia Comput. Sci.* **2017**, *114*, 258–265.

(19) Liu, J. Modeling a Large-Scale Nonlinear System Using Adaptive Takagi–Sugeno Fuzzy Model on PCA Subspace. *Ind. Eng. Chem. Res.* **2007**, *46*, 788–800.

(20) Wang, Y.; Sun, F.; Li, B. Multiscale Neighborhood Normalization-Based Multiple Dynamic PCA Monitoring Method for Batch Processes With Frequent Operations. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 1053–1064.

(21) Song, Q.; Ni, J.; Wang, G. Fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 1–14.

(22) Xu, Y.-M.; Wang, C.-D.; Lai, J.-H. Weighted multi-view clustering with feature selection. *Pattern Recogn.* **2016**, *53*, 25–35.

(23) Kohlert, M.; König, A. Large, high-dimensional, heterogeneous multi-sensor data analysis approach for process yield optimization in polymer film industry. *Neural Comput. Appl.* **2015**, *26*, 581–588.

(24) Lavanya, B.; Inbarani, H. H. A novel hybrid approach based on principal component analysis and tolerance rough similarity for face identification. *Neural Comput. Appl.* **2018**, *29*, 289–299.

(25) Jiang, Q.; Huang, B. Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method. *J. Process Contr.* **2016**, *46*, 75–83.

(26) Choi, S. W.; Lee, I.-B. Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chem. Eng. Sci.* **2004**, *59*, 5897–5908.

(27) Lee, J.-M.; Yoo, C.; Choi, S. W.; Vanrolleghem, P. A.; Lee, I.-B. Nonlinear process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* **2004**, *59*, 223–234.