I&EC research
Industrial & Engineering Chemistry Research

Article

Cite This: Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

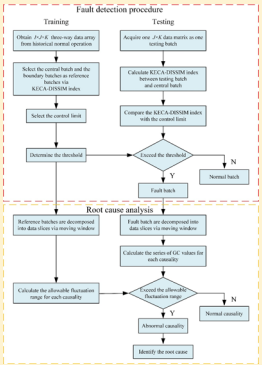
pubs.acs.org/IECR

Fault Detection and Root Cause Analysis of a Batch Process via Novel Nonlinear Dissimilarity and Comparative Granger Causality Analysis

He Fei,*,† Wang Chaojun,† and Fan Shu-Kai S*,‡

†Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, Beijing 100083, China
‡Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan

ABSTRACT: Data-driven fault detection and root cause analysis methods become attractive in modern industrial production that can guarantee the safety and stability of process operation. If process monitoring technology is implemented for fault detection, and the root cause of faults is analyzed timely, it is beneficial to maintain and improve the quality of coming batches. In this paper, a framework of fault detection and root cause analysis is proposed to address the aforementioned issue, particularly for a batch process. First, a new algorithm, termed kernel entropy component analysis (KECA)−DISSIM that combines KECA and dissimilarity analysis (DISSIM), is proposed for the batch process monitoring purpose. The KECA can extract nonlinear characteristics of the batch process effectively based on nonlinear mapping with the Renyi quadratic entropy. Then, dissimilarity indices between normal reference datasets and testing datasets can be calculated. If the testing dataset is detected as the non-normal batch by KECA−DISSIM, a novel root cause analysis named comparative Granger causality analysis is introduced for root cause analysis. The testing dataset is decomposed into a series of data slices via the moving window along the time domain. A series of causality values for each pair of variables are obtained by performing Granger causality analysis on these time slices. Lastly, the case studies based on a typical seven-variable nonlinear numerical process and a benchmark fed-batch penicillin fermentation process are studied to illustrate the practicality and effectiveness of the proposed framework.



1. INTRODUCTION

With a huge amount of data from industrial production processes, data-driven techniques become widely accepted in the field of process monitoring and root cause analysis.[1,2] A batch process, a typical production process in the chemical, food, and medicine industries, among others, brings about challenges in process monitoring. Distinct from two-way datasets, traditional statistical process monitoring methods are not suited to the batch process datasets with a three-way structure. Nomikos and MacGregor[3] introduced their developed monitoring scheme of the multiway principal component analysis (MPCA) with the idea of unfolding a three-way data structure into a two-way data structure, in which case the time dimension is combined with the variable dimension. So far, various algorithms have been proposed based on the idea of unfolding, such as multiway partial least squares,[4] multiway independent component analysis (MICA),[5] and multiway locality preserving projections.[6] The related nonlinear methods include multiway kernel principal component analysis[7] and multiway kernel independent component analysis.[8,9] However, as the number of variables increases dramatically, the calculation cost incurred increases sharply for the foregoing methods.[10] To resolve this computational difficulty, a dissimilarity index[11,12] named DISSIM can be introduced, which is designed to take advantage of phase dissimilarity between batches. However, it belongs to a typical linear statistical analysis method that reveals the linear data structure. To that end, kernel entropy component analysis (KECA) can be employed, which extracts nonlinear characteristics of the batch process effectively via nonlinear mapping with Renyi quadratic entropy[13] and reduces dimensionality according to different contributions of each eigenvalue to the Renyi quadratic entropy. Then, the dataset is intercepted into data slices by a moving window, which is helpful in collecting useful information. Finally, the dissimilarity indices based on DISSIM are used for fault detection.

Once the fault detection is accomplished, root cause analysis becomes an imperative topic for the identified fault batch.[14,15] The modern industrial process configuration has become extremely complex because it contains many control loops. The fault signal propagates along the direction of material and information flows once the fault occurred. Subsequently, other locations will also generate fault signals, which dictates the necessity of root cause analysis. Root cause analysis is a systematic procedure that investigates process variables to pinpoint the root cause that may cause fault.[16,17] In fault diagnosis methods, the contributions of different variables are determined by their own contribution rates. Then, the variables that make more contributions to statistical monitor-

Received: August 13, 2019
Revised: October 10, 2019
Accepted: November 2, 2019
Published: November 3, 2019

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

ing are selected as the major reasons for the process fault. However, root cause analysis is different from fault diagnosis, the former of which pays more attention to fault propagation. It is reasonable to conjecture that the fault appears in the root cause variable and then propagates to other variables. That is, there exists a time lag between the root cause and the non-root cause. Apparently, fault diagnosis methods do not take this kind of time lag into consideration. Due to the discrepancy between fault diagnosis and root cause analysis, the variable with a large contribution may not be the root cause variable. This will be illustrated shortly in a later section.

Commonly used algorithms for industrial root cause analysis include the Bayesian network,[18−20] Granger causality (GC) analysis,[21] transfer entropy,[22,23] and so forth. Among these methods, Granger causality (GC) analysis is preferred for its low computational effort, less data requirements, and easy implementation in practice. The related improved methods have been proposed to meet different needs of real industrial applications. To satisfy the multivariate characteristics of the industrial production process, a conditional Granger causality analysis method is proposed and then the causality map is simplified by a maximum spanning tree.[24] A novel method based on data-driven time series is introduced for locating the sources and propagation paths of plant-wide oscillations.[25] Traditional GC analysis based on the linear autoregressive model (AR) can only extract linear characteristics of the process but cannot grasp the nonlinear correlation among variables. To this end, a novel method combining GC analysis with Gaussian process regression is used to find out the fault propagation path and identify the fault sources.[26] In fact, traditional GC analysis can only describe the current causal state of each batch. The difference between normal and fault batches is not considered. Meanwhile, they always generate indirect causality due to information propagation between variables, which may result in misleading judgments.[27−29] Thus, a two-step diagnosis approach named comparative Granger causality (CGC) analysis is introduced to identify the root cause correctly. In the first step, the causality is compared between normal and fault batches. The abnormal causality, defined as the causality with evident change of Granger causality values, is determined according to the comparative results. Then, the root cause variable is deemed to be the one with the largest number of abnormal causality.

The remainder of this paper is organized as follows. The KECA−DISSIM method is introduced in Section 2. In Section 3, the CGC analysis is proposed. Case studies and the corresponding experimental results are presented in Section 4, demonstrating the effectiveness of the proposed framework. Finally, Section 5 concludes the research findings.

2. KECA−DISSIM MODELING AND PROCESS MONITORING

2.1. DISSIM Statistical Analysis. To quantify the dissimilarity between two matrices, Kano et al.[30,31] proposed a statistical method, termed DISSIM. Consider two matrices with the same number of variables, $\mathbf{X}_1 \in \mathcal{R}^{N_1 \times M}$ and $\mathbf{X}_2 \in \mathcal{R}^{N_2 \times M}$, where N_i denotes the sample size, and M denotes the number of variables. The variance-covariance matrices of two datasets are given by

$$\mathbf{R}_i = \frac{1}{N_i}\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i, \qquad i = 1,\ 2 \tag{1}$$

Two covariance matrices are combined together according to

$$\mathbf{R} = \frac{N_1}{N_i + N_2}\mathbf{R}_1 + \frac{N_2}{N_i + N_2}\mathbf{R}_2 \tag{2}$$

The mixture covariance matrix $\mathbf{R}$ in (2) can be diagnosed by an orthogonal matrix $\mathbf{P}_0$, which satisfies

$$\mathbf{P} = \mathbf{P}_0\mathbf{\Lambda}^{-1/2} \tag{3}$$

where $\mathbf{\Lambda}$ is a diagonal matrix, the elements of which are the eigenvalue of $\mathbf{R}$. Then, the original matrices $\mathbf{X}_i$ are translated into $\mathbf{Y}_i$ in the following

$$\mathbf{Y}_i = \sqrt{\frac{N_i}{N_1 + N_2}}\mathbf{X}_i\mathbf{P}_0\mathbf{\Lambda}^{-1/2} \tag{4}$$

The covariance matrix of $\mathbf{Y}_i$ is given by

$$\mathbf{S}_i = \frac{1}{N_i}\mathbf{Y}_i^{\mathrm{T}}\mathbf{Y}_i = \frac{N_i}{N_1 + N_2}\mathbf{P}^{\mathrm{T}}\mathbf{R}_i\mathbf{P} \tag{5}$$

where $\mathbf{P} = \mathbf{P}_0\mathbf{\Lambda}^{-1/2}$ and $\mathbf{S}_i$ satisfies

$$\mathbf{S}_1 + \mathbf{S}_2 = \mathbf{I} \tag{6}$$

$$\mathbf{S}_i\boldsymbol{w}_j^i = \lambda_j^i\boldsymbol{w}_j^i \tag{7}$$

where $\boldsymbol{w}_j$ and λ_j are the eigenvectors and the corresponding eigenvalues, respectively, $\mathbf{I}$ is the identity matrix. From eqs 6 and 7, such relations can be overthrown by

$$\mathbf{S}_2\boldsymbol{w}_j^1 = (1 - \lambda_j^1)\boldsymbol{w}_j^1 \tag{8}$$

$$1 - \lambda_j^1 = \lambda_j^2 \tag{9}$$

If the two matrices are similar, their corresponding eigenvalues should be close to each other, indicating that the eigenvalue is close to 0.5 according to eq 9. In contrast, the value of eigenvalues is close to 0 or 1 if two matrices are different from each other. To evaluate the dissimilarity between two datasets, the dissimilarity index D is defined as follows

$$D = \frac{4}{m}\sum_{j=1}^{m}(\lambda_j - 0.5)^2 \tag{10}$$

where m is the number of eigenvalues. Obviously, it can be seen that the range of the difference index is between 0 and 1. The larger the dissimilarity is, the smaller the similarity is.

2.2. Kernel Entropy Component Analysis. Consider a dataset $\mathbf{X}$: $\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots \boldsymbol{x}_n$ with the probability density function $p(\boldsymbol{x})$. The Renyi quadratic entropy is defined according to ref 32 as follows

$$\hat{V}(p) = -\log(V(p)) = -\log\left(\int p^2(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right) \tag{11}$$

and the Parzen window density estimator is described as

$$\hat{p}(\boldsymbol{x}) = \frac{1}{N}\sum_{\boldsymbol{x}_i \in \mathbf{X}} k_\sigma(\boldsymbol{x}, \boldsymbol{x}_i) \tag{12}$$

where $k_\sigma(\boldsymbol{x},\boldsymbol{x}_i)$ is a Mercer kernel function, and σ is the parameter of the kernel function or the Parzen window. Using the sample mean approximation of the expectation operator, we then have

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

$$\hat{V}(p) = \frac{1}{N}\sum_{\boldsymbol{x}_i \in \mathbf{X}} \hat{p}(\boldsymbol{x}_i) = \frac{1}{N}\sum_{\boldsymbol{x}_i \in \mathbf{X}} \frac{1}{N}\sum_{\boldsymbol{x}_i \in \mathbf{X}} k_\sigma(\boldsymbol{x}, \boldsymbol{x}_i) \tag{13}$$

Equation 14 can be obtained in the following

$$\hat{V}(p) = \frac{1}{N^2}\mathbf{I}^{\mathrm{T}}\mathbf{K}\mathbf{I} \tag{14}$$

where $\mathbf{K}$ is the kernel matrix and $\mathbf{I}$ is an $(n \times 1)$ column vector where each element equals one. Then, the Renyi entropy estimator may be expressed in terms of the eigenvalues and the corresponding eigenvectors of the kernel matrix, decomposed as $\mathbf{K} = \mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi} = \mathbf{E}\mathbf{D}\mathbf{E}$ with $\mathbf{D}$ being a diagonal matrix storing the eigenvalues $\lambda_1,\lambda_2,...,\lambda_N$ and $\mathbf{E}$ being a matrix with the eigenvectors $\boldsymbol{e}_1,\boldsymbol{e}_2,...,\boldsymbol{e}_N$ as columns. Rewriting (3) produces

$$\hat{V}(p) = \frac{1}{N^2}\sum_{i=1}^{N} (\sqrt{\lambda_i}\boldsymbol{e}_i^{\mathrm{T}}\mathbf{I})^2 \tag{15}$$

Equation 15 demonstrates that both the eigenvalue and eigenvector make different contributions to the Renyi quadratic entropy. Therefore, the number t of eigenvalues which make the first t largest contributions to Renyi quadratic entropy is selected. The KECA matrices are formed and defined as $\boldsymbol{\varphi}_{\mathrm{eca}} = \mathbf{D}_t^{1/2}\mathbf{E}_t^{\mathrm{T}}$.

2.3. KECA−DISSIM Algorithm. As a typical linear algorithm, DISSIM has been proven to be unsuitable for nonlinear processes.[33] Thus, a novel algorithm named KECA−DISSIM that combines the KECA method with the DISSIM method is introduced, which can not only deal with nonlinear problems, but also reduce dimensionality. The nonlinear characteristics of batch processes can be extracted effectively via nonlinear mapping with Renyi quadratic entropy and then the feature selection can be done in a high dimensional feature space based on the different contributions with respect to the Renyi quadratic entropy. In fact, the datasets from the industrial process have the characteristic of multi-mode along with the time dimension. Thus, an appropriate moving window is useful for capturing more information from different time stages. In practice, the nonstationary characteristics are typical in batch processes. However, there is no absolutely effective way to eliminate the nonstationary characteristics. To mitigate the nonstationary characteristics in KECA−DISSIM, a series of data slices generated by using the moving window can be considered. A small moving window is chosen here. Then, the generated data slices can be assumed stationary for a short period of time. The batch process is the one that works repeatedly over a pre-defined production cycle. The KECA−DISSIM index between the normal and fault batches will become prominent once the fault occurs. For this, the KECA−DISSIM index can be considered an appropriate measure for batch process monitoring. To apply the KECA−DISSIM algorithm, two batch datasets $\mathbf{X}_1$ and $\mathbf{X}_2$ with the same size of $J \times K$ are needed and the size of the moving window, L, is selected. The detailed procedure of the KECA−DISSIM index can be summarized officially in the following.

Algorithm 1, the KECA−DISSIM index.

1 Two datasets, $\mathbf{X}_1$ and $\mathbf{X}_2$ with the size of $J \times K$, are acquired;

2 Set the size of the moving window, L. For each dataset, a series of data slices with the size of $J \times L$ can be generated via the moving window step by step along the time dimension, $\mathbf{X}_i^w(J \times L)$ $(w = 1,2, ..., K - L + 1)$. Thus, two data slice sets $\mathbf{X}_1^w$ and $\mathbf{X}_2^w$ are obtained;

3 Perform KECA on $\mathbf{X}_1^w$ and $\mathbf{X}_2^w$, and then two data slice sets in the low dimensional space, $\mathbf{Y}_1^w$ and $\mathbf{Y}_2^w$, are generated;

4 Calculate the dissimilarity index between $\mathbf{Y}_1^w$ and $\mathbf{Y}_2^w$ via DISSIM as w goes from 1 to $k - L + 1$. A sequence of dissimilarity indices $\mathbf{d}_{12}$ with $k - L + 1$ indices is obtained.

3. ROOT-CAUSE ANALYSIS

Once a quality-related fault is detected by KECA−DISSIM, the root cause analysis should be performed timely. In this section, a novel method named CGC analysis (CGC) is introduced for root cause analysis. It contains a training process with the purpose of learning the causality state of normal batches.

3.1. Granger Causality Analysis. Granger causality analysis is used for investigating whether a time series is helpful in predicting another one. Consider two time series $\boldsymbol{x}_1$: ${x_1}^1, {x_2}^1, ..., {x_N}^1$, $\boldsymbol{x}_2$: ${x_1}^2, {x_2}^2, ..., {x_N}^2$. To investigate the causality between them, two models are built, including a bivariate AR model as defined below

$${x_t}^1 = \sum_{i=1}^{\mathrm{lag}} \alpha_i {x_{t-i}}^1 + \sum_{i=1}^{\mathrm{lag}} \beta_i {x_{t-i}}^2 + \varepsilon_{1,1}(t) \tag{16}$$

$${x_t}^2 = \sum_{i=1}^{\mathrm{lag}} \alpha_i {x_{t-i}}^1 + \sum_{i=1}^{\mathrm{lag}} \beta_i {x_{t-i}}^2 + \varepsilon_{2,2}(t) \tag{17}$$

and its reduced model

$${x_t}^1 = \sum_{i=1}^{\mathrm{lag}} \beta_i {x_{t-i}}^1 + \varepsilon_1(t) \tag{18}$$

$${x_t}^2 = \sum_{i=1}^{\mathrm{lag}} \beta_i {x_{t-i}}^2 + \varepsilon_2(t) \tag{19}$$

where $\varepsilon_{i,i}$ is the predicted error from the AR model, it represents the error of predicting ${x_t}^i$ jointly by ${x_{t-i}}^1$ and ${x_{t-i}}^2$, ε_i refers to the predicted error from the reduced model, lag is the time lag (order) of the model which is determined by maximizing the akaike information criterion[34] or the Bayesian information criterion,[35] and α and β are the unknown parameters.

This causality between two time series can be quantified as in eq 20. For instance, if $\varepsilon_{2,2}$ is less than ε_2, $F_{x_1 \to x_2} > 0$. It indicates that the time series $\boldsymbol{x}_1$ is beneficial to predicting time series $\boldsymbol{x}_2$. That is, time series $\boldsymbol{x}_1$ is the Granger cause of time series $\boldsymbol{x}_2$.

$$F_{x_i \to x_j} = \ln\left(\frac{\mathrm{var}(\varepsilon_j)}{\mathrm{var}(\varepsilon_{j,j})}\right) \tag{20}$$

If $F_{x_i \to x_j} > 0$, there exists a Granger causality between them. If $F_{x_i \to x_j} = 0$, there is no Granger causality. Note that $F_{x_i \to x_j}$ cannot be negative. The statistical significance of this causal relationship can be defined via the F-statistic defined as follows

$$F_{\mathrm{statistic}} = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_1)/\mathrm{lag}}{\mathrm{RSS}_1/(N - 2 \times \mathrm{lag} - 1)} \approx F(\mathrm{lag}, N - 2 \times \mathrm{lag} - 1) \tag{21}$$

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

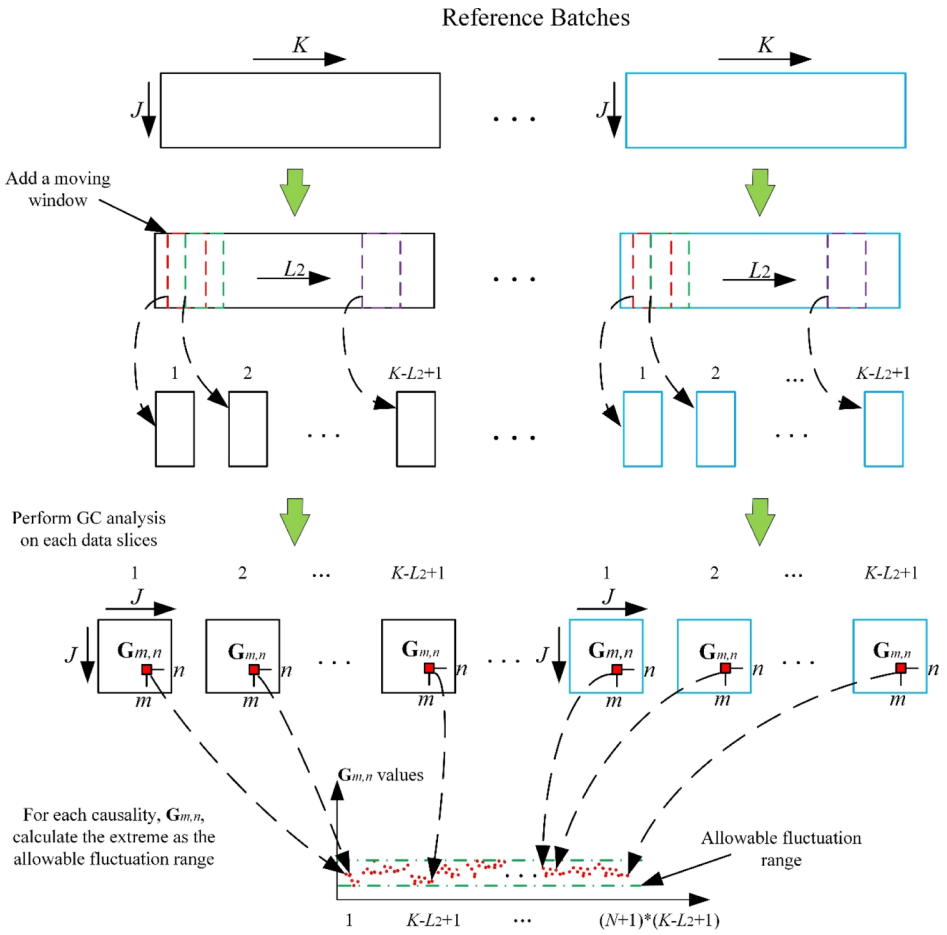


Figure 1. Allowable fluctuation range of each causality.

where RSS_0 is the residual sum of squares in the reduced model in eqs 18 and 19, RSS_1 is the residual sum of squares in the bivariate AR model in eqs 16 and 17, and n is the total number of observations used to build the model. If $F_{x_i \to x_j} = 0$, the null hypothesis is rejected with a significance level α, the time series $\boldsymbol{x}_1$ is said to have a causal influence on another time series $\boldsymbol{x}_2$.

Consider the dataset $\mathbf{X}$ with the size of $J \times K$. Then, GC analysis is performed and the causality matrix $\mathbf{G}$ with the size of $J \times J$ is obtained, which reflects the causality between variables. For each value in the causality matrix, if $\mathbf{G}_{m,n} > 0$, it is defined as the degree of causality from the n-th variable to the m-th variable. It is also worth noting that the obtained causality $\mathbf{G}_{m,n}$ is equal to 0 if $m = n$.

3.2. Comparative Granger Causality Analysis. In actuality, conventional GC analysis describes the current causality state of the dataset while the inherent causality between variables cannot be removed. In other words, it is quite difficult to judge whether the causality is generated due to fault information or not if only the causality of fault batch is calculated. This paper proposes a CGC analysis method for solving this issue, in which the training process is augmented to obtain the current causality state of normal batches. With the purpose of eliminating the inherent causality, the proposed method conducts a comparison of the causality between normal and fault batches, in which the abnormal causality due to fault information is determined based on the comparative results. Finally, all abnormal causal relationships are counted to seek their root cause variables.

3.2.1. Moving Window. As the result of the traditional GC analysis for each pair of variables only returns a single value, it is difficult to reflect the difference between the causality of the variables accurately. Thus, it is hoped that a sequence of causality values can be obtained, which is more capable of reflecting the change of causality objectively and improving the accuracy of the comparative result. Towards this end, a moving window is added along the time dimension to decompose the datasets into a series of time slices. For any pair of variables, a sequence of causality values is generated by performing GC analysis on each slice. No doubt, the comparative results between two sequences should provide much more trustworthy information than that between two values, so the accuracy and stability of the comparative results must be improved in this manner.

It is apparent that the size of the moving window should be dependent on the total sample size. If the moving window is too large, the number of data slices is insufficient and the accuracy of the comparative results cannot be guaranteed. It is a legitimate choice to choose between 0.8 and 0.9 times of the total sample size as the size of the moving window.

3.2.2. Allowable Fluctuation Range for Each Causality. It is worth noting that the calculation cost has increased due to the moving window. To save the calculation cost, the central batch and many boundary batches, which can represent the whole training batches, are selected as the reference batches during the training process. The KECA−DISSIM indices

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

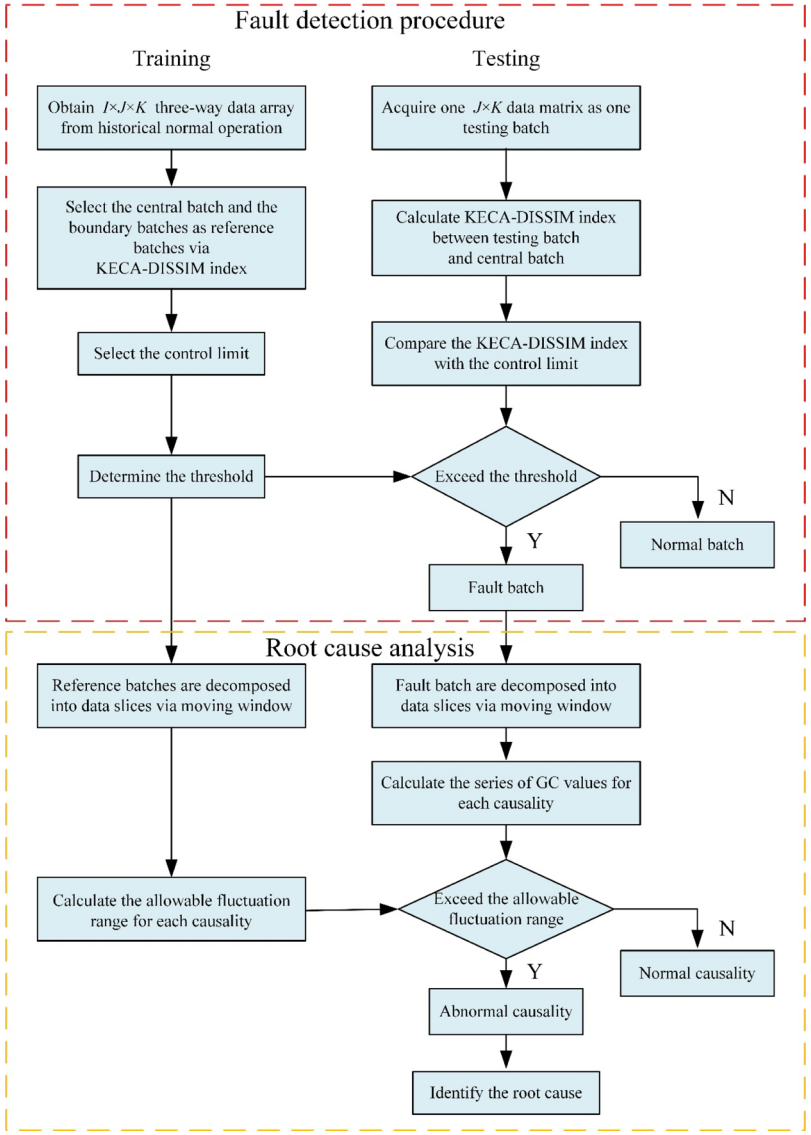


Figure 2. Flowchart of the proposed framework.

between all batches are calculated. The one with the minimum sum of the indices is defined as the central batch that represents the center of the whole training datasets. The boundary batches represent the boundary of the whole training datasets. Thus, the KECA−DISSIM indices between the boundary batches and the central batch are larger than those between other batches and the central batch. Thus, the central batch and boundary batches are selected as the reference batches to represent the whole training batches. For example, one central batch and N boundary batches are selected by this way. There are $N + 1$ sequences for each causality, showing the fluctuation range of the corresponding causality under normal operation. Thus, the allowable fluctuation range is defined as the extremum of the $N + 1$ sequences.

The process of calculating the allowable fluctuation range for any causality is shown in Figure 1. First, the reference batches are selected from the training datasets according to the KECA−DISSIM index. The reference batches consist of the central batch and N boundary batches with J variables and K time. Second, a moving window with the size of L_2 is selected. Then, a series of slices with the same size of $J \times L_2$ are generated via the moving window step by step along the time dimension for each batch, $\mathbf{X}_i^w(J \times L_2)$, where $i = 1, ..., N + 1$, $w = 1, ..., K - L_2 + 1$. Thirdly, a series of Granger causality matrices $\mathbf{G}$ with the size of $J \times J$ are obtained by performing GC analysis on each slices. Obviously, there are totally $((N + 1) \times (K - L_2 + 1))$ Granger causality matrices. For each causality value, $\mathbf{G}_{m,n}$, where $m = 1, ..., J$, $n = 1, ..., J$, a sequence of causality values is obtained. Lastly, the extremum for each sequence of causality can be located as its allowable fluctuation range. Any causality values that exceed their corresponding allowable fluctuation range are deemed as the abnormal causality. This process is repeated to determine the allowable fluctuation range of all causality.

In fact, both the normal batch and the faulty batch have their corresponding causality maps. Comparing the causality between normal batches and faulty batches, the abnormal causality that has obvious changes between normal batches and faulty batches is captured. Only these abnormal causality changes can uncover the fault conditions under the circumstances where the influence of the causality map from the normal batches has been removed.

3.2.3. Root Cause Diagnosis. Traditional GC analysis identifies the root cause according to the direction of the

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

causality. However, there are two shortcomings for traditional GC analysis. The first one is that GC analysis just describes the current causality state of the dataset. In this paper, the comparison of the causality between normal and fault batches is accomplished, and then the comparative result is used to determine which causality leads to abnormality. There are many factors that influence the causality results such as noise, fault propagation and control loops. The causality map becomes more complex and even goes perplexing against the fact, which may lead to wrong results. Thus, the comparative GC analysis no longer focuses on the direction of causality but the comparative results of causality. Once the fault occurs, fault signals appear in other variables due to the fault propagation effect. In other words, the causality connected with the root cause variable may become abnormal due to the fault propagated, and the root cause variable will generate excessive abnormal causality.

3.3. Procedures of Fault Detection and Root Cause Analysis. The entire procedure is presented in Figure 2. The proposed framework includes fault detection and root cause analysis. The detailed steps can be shown as follows:

3.3.1. Fault Detection Procedure. Off-line training:

1 The training datasets $\mathbf{X}$ from historical normal operating condition with three-way structure of batches I, variables J, and time K are acquired and normalized by their own means and variances.
2 The central batch and N boundary batches are selected from the training datasets $\mathbf{X}$ as the reference batches via the KECA−DISSIM index. All batches are re-normalized by the mean and variance obtained from the central batch.
3 N sequences of KECA−DISSIM indices between the central and boundary batches are computed. From these KECA−DISSIM indices, the control limit is determined based on the three-sigma principle.
4 Set a threshold α. According to the three-sigma principle, there are inevitably a few indices exceeding the control limit. The N sequences are compared to the control limits; the threshold α is defined as the maximum number exceeding the control limit among these sequences.

On-line detection:

1 One $J \times K$ data matrix is acquired as the testing batch and normalized by the mean and variance obtained from the central batch.
2 A sequence of the KECA−DISSIM indices between the central batch and the testing batch is calculated.
3 If the number of this sequence exceeding the control limit is larger than the threshold α, the testing batch is judged as a fault batch, judged as a normal batch if not.

3.3.2. Root Cause Analysis. Off-line training:

1 The reference batches have been selected during the off-line training process of the fault detection procedure.
2 Select the moving window with the size of L_2. All the reference batches are decomposed into a series of slices with the size of $J \times L_2$ via the moving window step by step.
3 Perform GC analysis on each of the data slices and then a series of Granger causality matrices are obtained. For each causality, a sequence of GC values is obtained and the extremum among this sequence is set as the corresponding allowable fluctuation range.

On-line root cause analysis:

1 The fault batch is decomposed into a series of data slices via the same moving window L_2.
2 Perform GC analysis on a series of slices and then a sequence of GC values for each causality is generated.
3 These sequences of causality values are compared with the corresponding allowable fluctuation range. If the causality values exceed the allowable fluctuation range, it is judged as abnormal causality.
4 The root cause variable is the one that generates the largest number of abnormal causality.

4. SIMULATIONS AND DISCUSSIONS

The processes discussed in this section are the typical seven-variable nonlinear numerical process and the well-known fed-batch fermentation process for penicillin production. The first case focuses on analyzing the efficiency and necessity of the proposed framework. The second case is intended for verifying its practicality in a complex industrial process. The detailed results of the proposed framework will be demonstrated.

Table 1. Fault Detection Rates of KECA−DISSIM and KDISSIM for the Numerical Process (%)

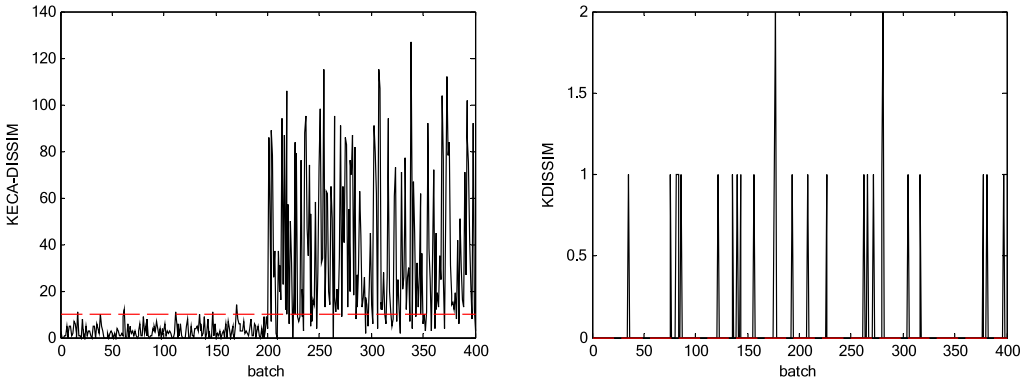
fault	KECA−DISSIM		KDISSIM	
kernel parameter	10^2		10^2	
L	100		85	
	FAR	FDR	FAR	FDR
	2.5	**78.5**	7	6
	2.5	**90.5**	7	4

4.1. Seven-Variable Linear Numerical Process. In this subsection, the KECA−DISSIM algorithm is compared with the KDISSIM algorithm[33] for the process monitoring purpose. In addition, the comparative GC analysis is compared with the traditional GC analysis. The discussions of traditional GC analysis have been already presented in Section 3.2. The functional relationship between variables is expressed in eq 22.

$$
\begin{aligned}
X_1(t) &= 0.5 \times k^2 - 2 \times k + 0.5 + e;\\
X_2(t) &= k^2 - k + \sin(\pi \times k) + e;\\
X_3(t) &= 0.5 \times X_2(t-3) - 1.01 \times X_1(t-4)\\
&\quad + 0.1 \times \cos(\pi \times k) + e;\\
X_4(t) &= -0.22 \times X_2(t-1) \times X_1(t) - k + 5 + e;\\
X_5(t) &= (X_3(t-2))^2 - 0.47 \times X_1(t) + 2 \times k + e;\\
X_6(t) &= X_5(t-1) + 6.6 + e;\\
X_7(t) &= X_6(t-4) - 0.47 \times X_4(t) - k + e;
\end{aligned}
\tag{22}
$$

where e denotes the independent random error term simulated by means of a uniform distribution between −0.1 and 0.1, which represents the normal stochastic variation, t ranging from 1 to 400 represents the time index, and k is a constant between −2 and 2. In addition, the disturbances are also used to generate fault datasets according to the following configuration:

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX



(a) KECA-DISSIM (b) KDISSIM

Figure 3. Control charts based on KECA−DISSIM and KDISSIM for fault 1.

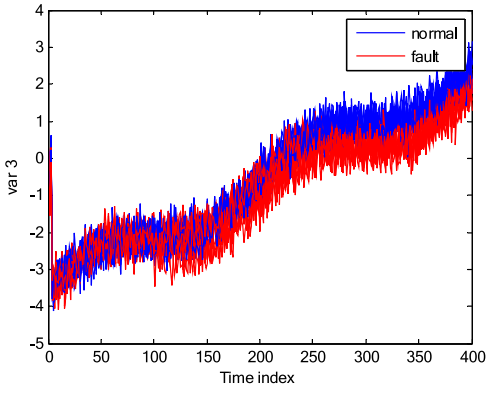
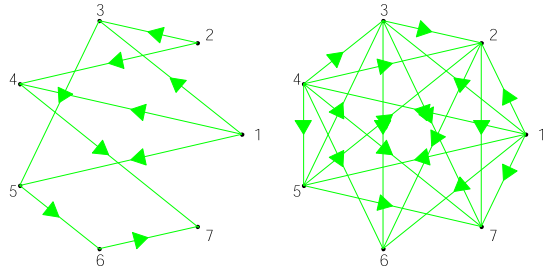


Figure 4. The variable trajectory of fault 1.

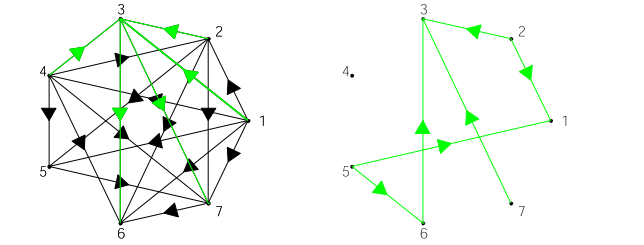
- A step change of variable 3 by −0.6 is introduced at a randomly time.
- A step change of variable 3 to −0.9 is introduced at a randomly time.

A total of 100 batches are generated as the training datasets, each of which consists of 7 variables and 400 samples. Then, a total of 400 batches are generated as testing datasets, in which the first 200 batches are normal and the last 200 batches are faulty with the disturbance aforementioned. The fault signal is added randomly during the time interval $t \in [200, 300]$. Therefore, there are two testing datasets corresponding to the two disturbances, generated with a size of 400 ize of400 each.



(a) causality map in theory (b) causality map for normal batches

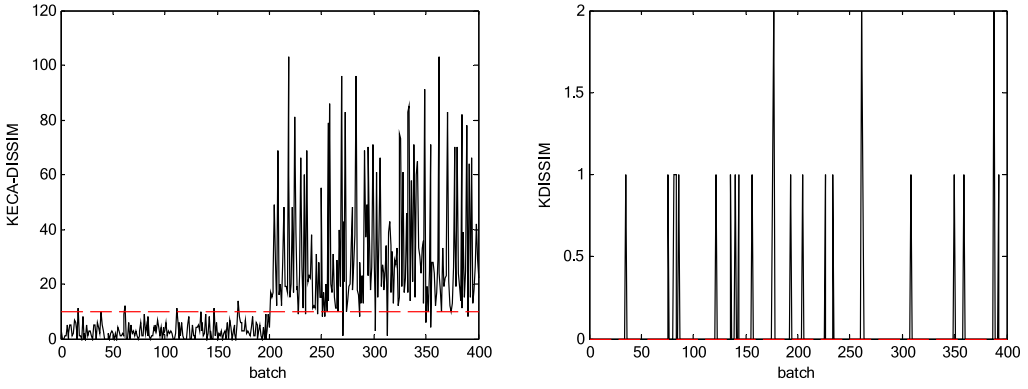
Figure 6. Causality maps of the seven-variable numerical process.



(a) causality map by traditional GC analysis (b) causality map by the comparative GC analysis

Figure 7. Causality maps for fault 1.

For a fair comparison, the false alarm rate (FAR) and the fault detection rate (FDR) are defined as follows



(a) KECA-DISSIM (b) KDISSIM

Figure 5. Control charts based on KECA−DISSIM and KDISSIM for fault 2.

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

Table 2. Overview of Available Measurements

number	name	unit
x_1	fermentation volume	m^3
x_2	dissolved oxygen	mg/L
x_3	dissolved CO_2	mg/L
x_4	reactor temperature	K
x_5	pH	-
x_6	reaction heat	cal
x_7	feed rate	L/h
x_8	feed temperature	K
x_9	aeration rate	L/h
x_{10}	agitator power	W
x_{11}	water flow rate	L/h
x_{12}	cold water temperature	K
x_{13}	hot water temperature	K
x_{14}	base flow rate	mL/h
x_{15}	acid flow rate	mL/h
x_{16}	cumulative base flow	mL
x_{17}	cumulative acid flow	mL

Table 3. Three Types of Faulty Batches

fault no.	variables that introduce disturbance	root-cause variable
1	change in coolant temperature −0.2 °C	X_{12}
2	agitator power drop −4%	X_{10}
3	aeration rate drop −5%	X_9

$$\mathrm{FAR} = \frac{\mathrm{NN}}{\mathrm{TNN}} \tag{23}$$

$$\mathrm{FDR} = \frac{\mathrm{AN}}{\mathrm{TAN}} \tag{24}$$

where NN represents the number of normal batches that are judged as fault batches, and TNN is the total number of the normal batches; AN represents the number of fault batches that are judged as fault batches, and TAN is the total number of the fault batches.

The input space is mapped into the high dimensional feature space via the Gaussian kernel $k(x,x') = \exp(-\|x - x'\|^2/c)$. Here, different parameters are employed to build different models for each method, such as different sizes of the moving window, the kernel parameter $c \in \{10^{-3}, 10^{-2}, ..., 10^{7}\}$. Then the best results of different methods are compared. The FDRs of KECA−DISSIM and KDISSIM are shown in Table 1.

On the whole, the results demonstrate that the performance of the KECA−DISSIM method is superior to the KDISSIM method. In every fault type, KECA−DISSIM can produce much better detection results in FAR and FDR than KDISSIM. Detection results of fault 1 are shown in Figure 3. Evidently, KECA−DISSIM outperforms KDISSIM overwhelmingly. Specifically, the KECA−DISSIM method returns better monitoring results with FDR of 78.5% and FAR of 2.5%, while the KDISSIM method yields the FDR of merely 6% and FAR of 7%. In order to show the fault pattern pictorially, the change in fault 1 is exhibited in Figure 4 in comparison to the normal batches.

The results of fault 2 are displayed in Figure 5. Looking at the monitoring results returned by KDISSIM, there seems no notable difference between normal and fault batches. The normal and fault batches are indistinguishable with respect to KDISSIM but KECA−DISSIM returns FDR of 90.5% for fault 2 instead.

The original intension of GC analysis is to investigate if a time series is helpful in predicting another time series. That is, the relationship shown in eq 22 can be considered as the causality between variables. Hence, the causality map in theory according to the relationship in eq 22 is plotted in Figure 6a, where black points represent seven variables and the green lines are the Granger causality. Here, variable 1 and variable 2 can be seen as the root cause of the system that have an effect on other variables. However, noises exist during data collection and indirect causality is involved with different variables so that the causality complexity becomes aggregated. The causality map arising from the normal batches is shown in Figure 6b. There could be several new causalities that are not conducive to locate the root cause. Note that variable 1 is selected as the root-cause variable while variable 2 is not. Comparing the two causality maps, both causalities from variables 2 to 3 and from variables 2 to 4 give the opposite direction. This phenomenon indicates that the direction of causality is susceptible to other factors. Sometimes, it does not even indicate the true direction of fault propagations.

For fault batches, the root-cause is variable 3 in which the fault signal is added. In fault 1, the causality map of fault batches calculated by traditional GC analysis is shown in Figure 7a, where green lines represent the causality connected with variable 3. However, the root cause is still variable 1 according to Figure 7a. The results for both normal and fault batches indicate that variable 1 is the root cause variable.

Table 4. Fault Detection Rates of KECA−DISSIM, KDISSIM, MICA, and MPCA for the Penicillin Cultivation (%)

fault	KECA−DISSIM		KDISSIM	
kernel parameter	10^2		10^2	
L/latent number	200		200	
	FAR	FDR	FAR	FDR
1.	**8**	**78.5**	49.5	52
2.	**8**	**87.5**	49.5	48.5
3.	**8**	**91**	49.5	50

fault	MICA				MPCA			
L/latent number	82		82		30		30	
	I^2		SPE		T^2		SPE	
	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR
1.	1	65.5	0.5	1	0	0	0	0
2.	1	63	0.5	1.5	0	4	0	1.5
3.	1	69	0.5	2	0	10	0	0.5

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

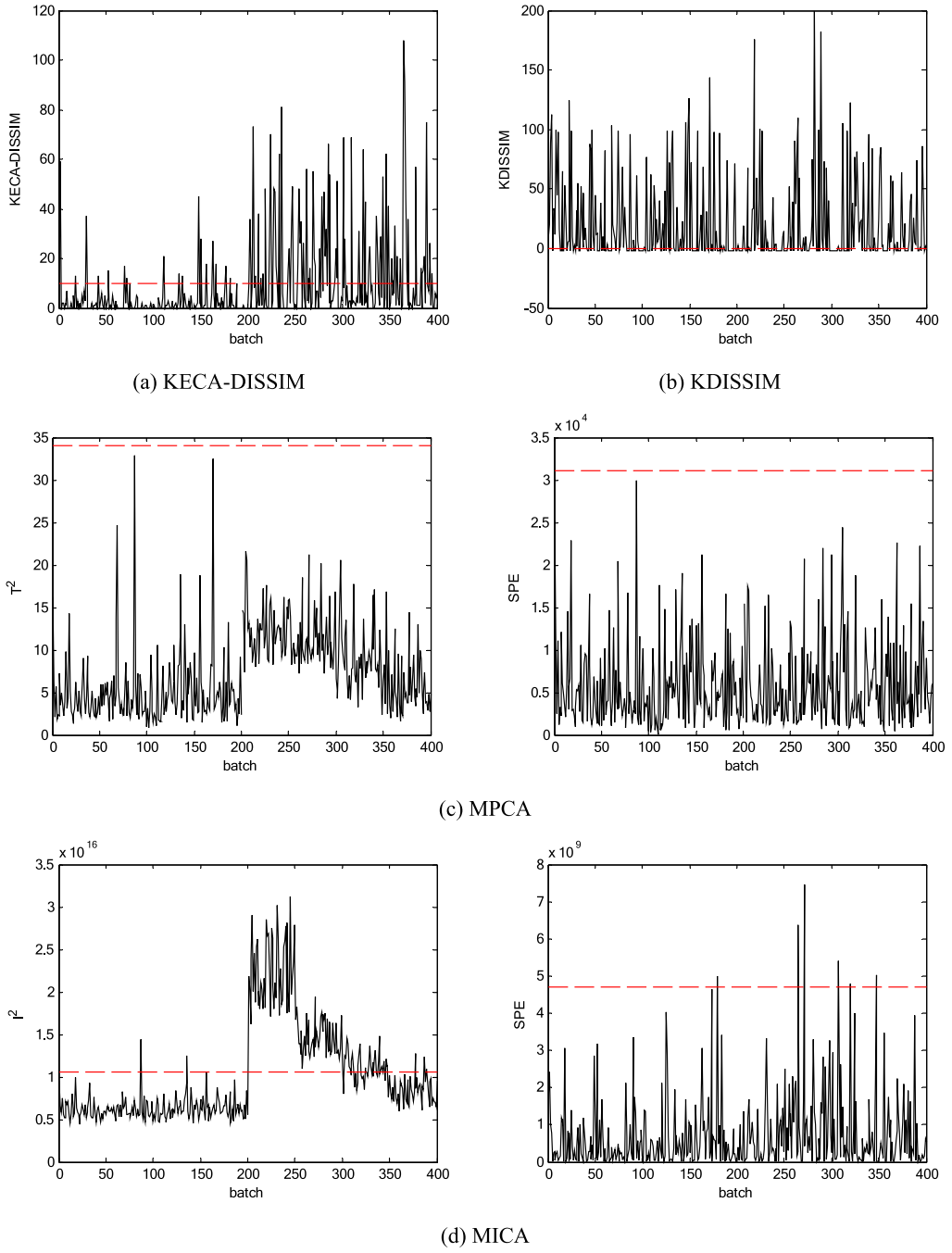


Figure 8. Control charts based on KECA−DISSIM, KDISSIM, MPCA, and MICA for fault 1.

However, the variable 3 is the real root cause. Obviously, the root cause variable derived from traditional GC analysis is incompletely credible. For example, the causalities from variables 3 to 4 and from variables 1 to 3 appear not only in the normal batch but also in the fault batch. It is difficult to judge whether these causalities turn abnormal due to the fault propagations or not. Meanwhile, GC analysis only describes the current causality state of the dataset without considering the difference between normal and fault batches. How to make use of the information is the key to finding out the right root cause.

This paper introduces a novel method termed the comparative GC analysis, in which the issues mentioned previously will be fully addressed. In the proposed method, the causality is compared with the corresponding allowable fluctuation range calculated during the training process, and then those causality values that exceed the allowable fluctuation range are judged as the abnormal variables. If the fault takes place, the root cause generates a fault signal. Other variables become abnormal due to the fault signal propagated from the root cause. The causality associated with the root cause is more likely to be realistically abnormal. Thus, the root cause can be considered the variable that generates more abnormal causality than other variables.

The causality map of abnormal causality is shown in Figure 7b, where green lines represent the abnormal causality. According to this map, variable 3 should be the root cause in that it owns the largest number of abnormal causality than

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

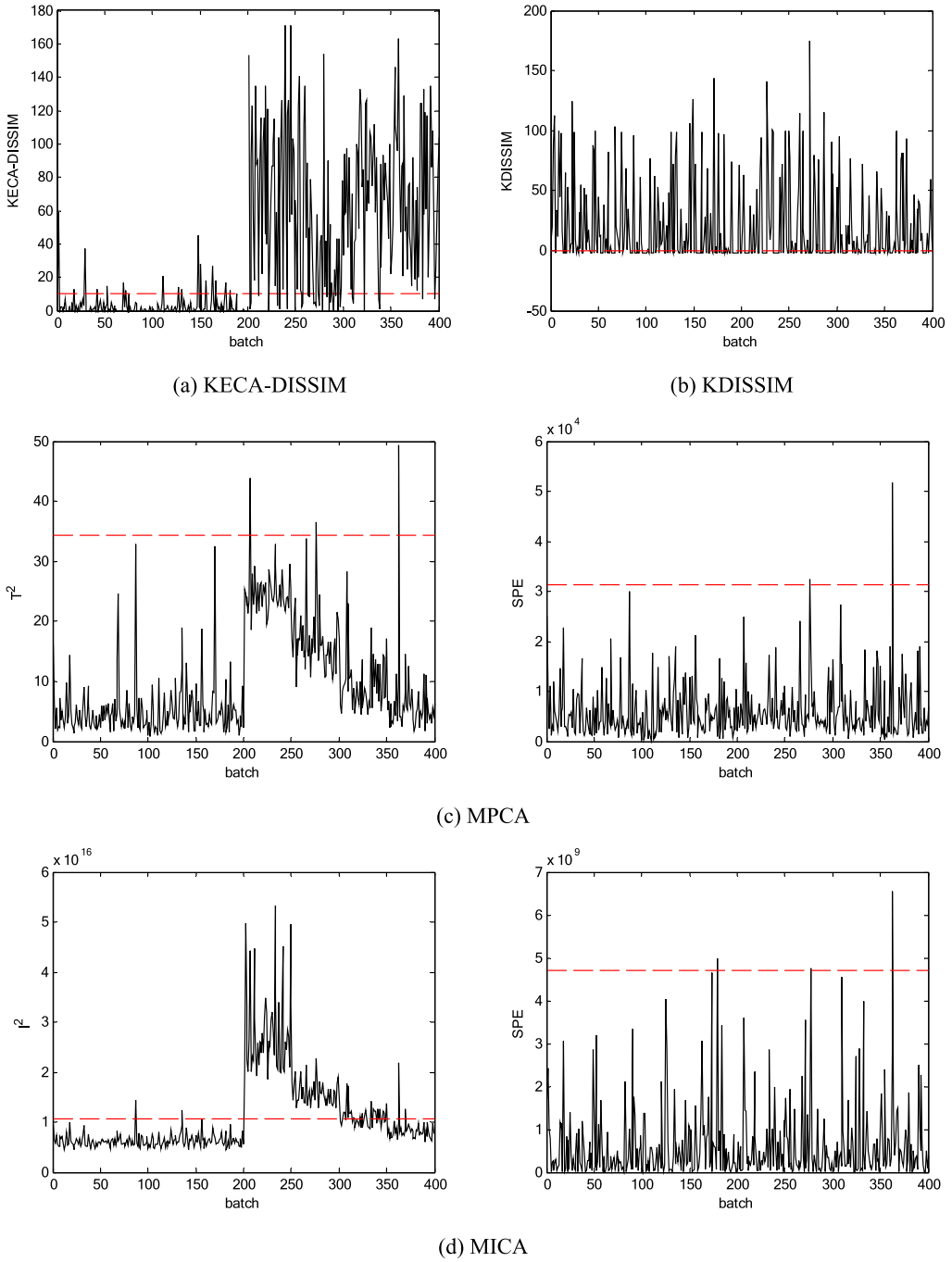


Figure 9. Control charts based on KECA−DISSIM, KDISSIM, MPCA, and MICA for fault 2.

others. Obviously, the root cause is correctly located as the fault datasets with a step change in variable 3. This method compares the causality states between normal and fault batches, which is of great importance to determine if the causality becomes abnormal. Then, a causality map is plotted and the root cause is located. In a word, the results of the linear numerical process discussed in this section demonstrate that the framework proposed in this paper is practically viable.

4.2. Fed-Batch Penicillin Fermentation Process. To further evaluate the proposed framework, a simulated fed-batch fermentation process for penicillin production is used. Such a well-known benchmark process with the characteristics of nonlinear dynamics and multiphase is widely used in data-driven fault monitoring and root cause analysis.[36−39] The simulation package is available at http://www.chee.iit.edu/~control/simulation.html.

The dataset consists of 400 normal batches and 15 different fault types. Each fault type consists of several fault magnitudes, with 200 batches per fault magnitude. It is worth noting that the 200 fault batches can be divided into four stages according to the initial location where the fault really happens, each with 50 batches. The initial fault location for the first stage ranges from 0 to 300, the second stage from 300 to 600, the third stage from 600 to 800, and the last stage from 800 to 1000. There are totally 17 variables, listed in Table 2, which are measured online throughout every batch. All the variables are mostly related to the product quality, which are used for process monitoring and root cause analysis.

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

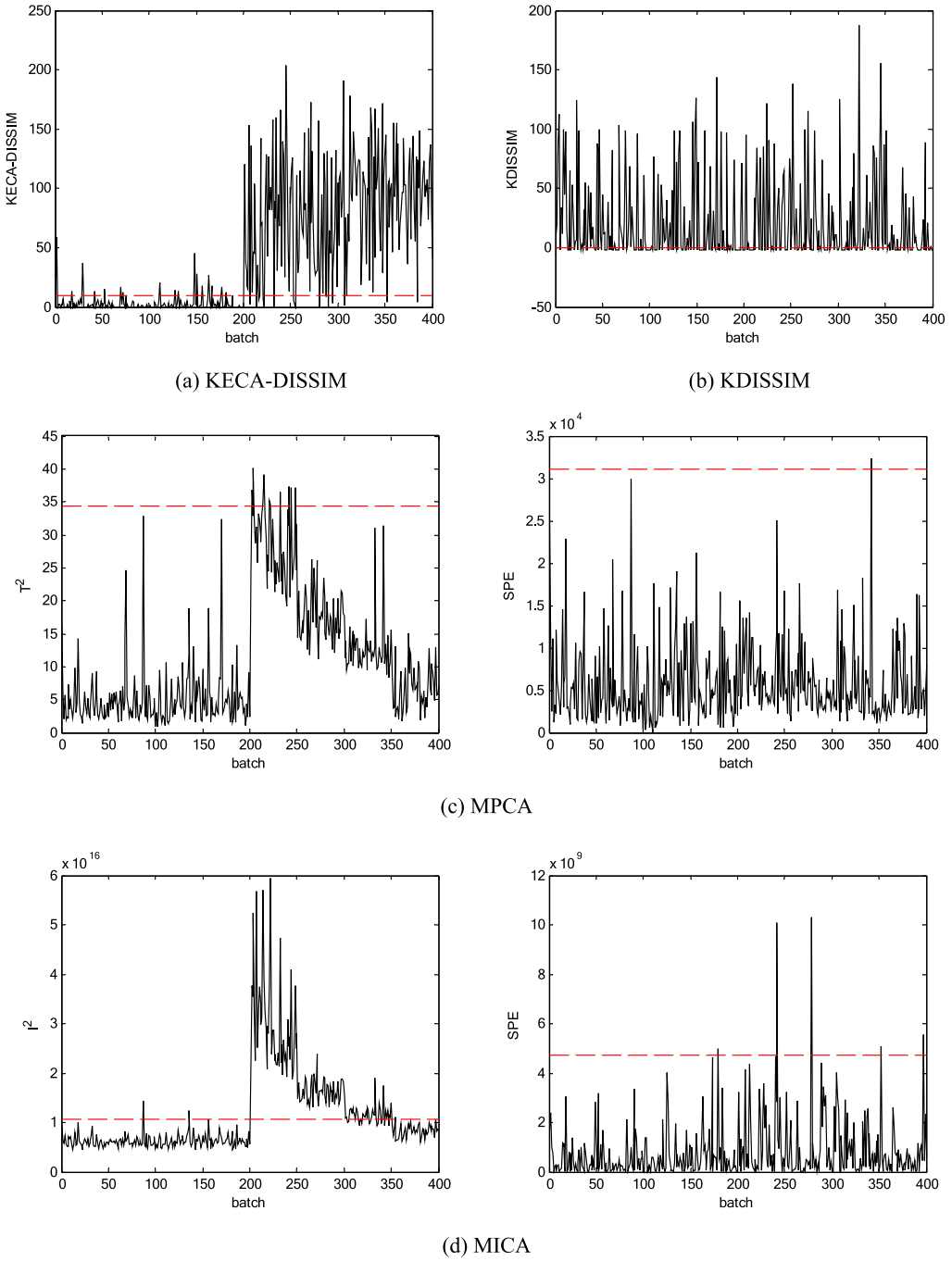


Figure 10. Control charts based on KECA−DISSIM, KDISSIM, MPCA, and MICA for fault 3.

Three fault types listed in Table 3 are selected to evaluate the proposed framework. The first 200 batches are chosen as the training datasets, the size of which is 200 are sel1201. The other 200 normal batches are used as the testing datasets with the same size of 200 201. Th1201. For each fault type shown in Table 3, 200 batches per fault type are also used as the testing datasets, amounting to a total of 400 batches for each fault type. Therefore, there are three testing datasets created corresponding to the three fault types, the size of which is 400 200 ba1201.

In this subsection, the monitoring results based on KECA−DISSIM are compared with the results based on KDISSIM, MICA, and MPCA. Herein, different parameters are employed to build different models, such as different sizes of the moving window, the kernel parameters $c \in \{10^{-3}, 10^{-2}, ..., 10^{7}\}$. The control limits based on KDISSIM, MICA, and MPCA are determined by using the kernel density estimation.[40] Then, the best results of different methods are compared and presented. The FDRs of the four different methods are tabulated in Table 4.

Overall, the computational results demonstrate that the performance of KECA−DISSIM is superior to other three compared methods. In every fault type, the KECA−DISSIM method returns the best FDR. In the meantime, KECA−DISSIM can distinguish normal and fault batches with a highest FDR and a comparable FAR. The corresponding detection results for fault 1 are shown in Figure 8. Among the compared methods, the best FDR of 78.5% was obtained using

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

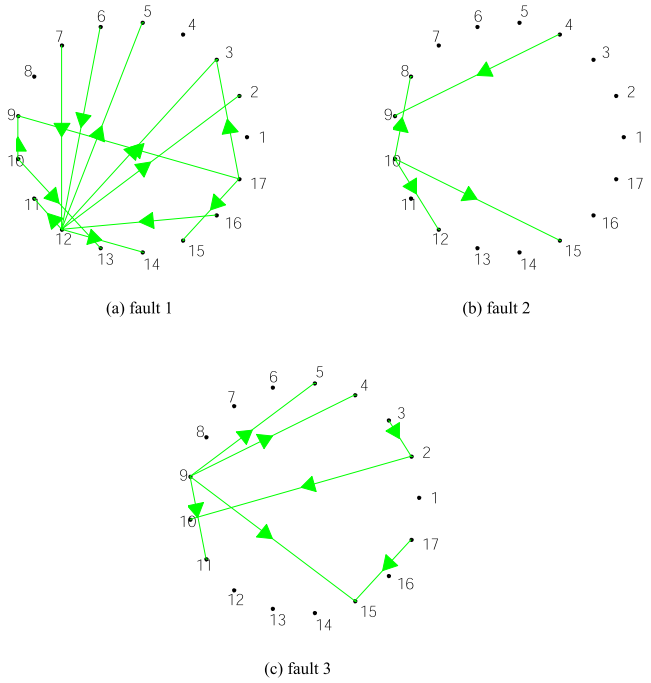


Figure 11. Causality maps based on the comparative GC analysis for three fault types.

KECA−DISSIM. For the remaining three methods, only a FDR of 65.5% was obtained using the control chart of I^2 based on MICA. Notice that all the fault batches are detected mistakenly as normal via MPCA.

In Figure 9, the detection results illustrate that KECA−DISSIM still works quite effectively for screening fault 2 among all the compared methods. By contrast, KDISSIM totally cannot distinguish normal and fault batches. FDRs based on MPCA are all less than 5%. The control chart of I^2 based on MICA returns a FDR of 63%, while KECA-DISSIM yields a better FDR of 87.5%.

The detection results of fault 3 are shown in Figure 10. Among all the compared methods, KECA−DISSIM returns the best FDR of 91%, which is significantly better than the FDR of 69% generated by the I^2 control chart of MICA. Recall that the fault batches for each fault type are divided into four stages according to the initial location where the fault occurs. As can be seen from (c) and (d) in Figure 10, both the results of T^2 statistics based on MPCA and I^2 statistics based on MICA appear self-evident that the four-stage variation is present. Clearly, these two methods are sensitive to the initial location where the fault occurs, inasmuch as the I^2 and T^2 statistics vary drastically with the initial fault location. The computational results shown in this section demonstrate that KECA−DISSIM exhibits much more stability than MPCA and MICA.

Once the fault batch is detected, it is necessary to determine the root cause. The causality maps of three fault types via the comparative GC analysis are presented in Figure 11. In fault 1, most of the abnormal causalities are connected with the variable 12, which is selected as the root cause variable. In fault 2 and fault 3, the root cause variable is repeatedly sought by the same way, concluding variables 10 and 9, respectively. The comparative GC analysis can generate a more succinct causality map where the normal causality is ignored, which greatly helps locate potential root cause variables. To sum, the comparative GC analysis is a meaningful and efficient method to determine the root-cause via the number of abnormal causality.

5. CONCLUSION

In this paper, a novel framework of fault detection and root cause analysis is proposed for batch processes. Both the seven-variable numerical process and the benchmark studies of the fed-batch penicillin fermentation process have been used to illustrate the monitoring performance of the proposed framework. In comparison to KDISSIM, MPCA, and MICA, the computational results show that KECA−DISSIM gives the best monitoring performance. Meanwhile, the proposed novel method, by comparing the causality between normal and fault batches, provides a new direction of root-cause analysis. This proposed method proves to be efficient to overcome the disadvantages of traditional GC analysis. The results show that the comparative GC analysis works efficiently in root cause analysis.

On the whole, the proposed framework provides data scientists with a viable proposal for both fault detection and root cause analysis. Even so, there are still several issues worth further scrutiny. How to improve the efficiency of the entire algorithm execution, especially in root cause analysis, points out an important direction of future research in the era of big data.

AUTHOR INFORMATION

Corresponding Authors

*E-mail: hefei@ustb.edu.cn (H.F.).
*E-mail: morrisfan@ntut.edu.tw (F.S.-K.S.).

ORCID

He Fei: 0000-0002-1739-5649

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This research is supported by the National Key Technology R&D Program of the 12th Five-year Plan of China (Grant no. 2015BAF30B01), the Open Foundation of the State Key Laboratory of rolling and automation, Northeastern University (grant no. 2018RALKFKT003), and USTB-NTUT Joint Research Program (grant no. TW2019013).

REFERENCES

(1) Zhou, P.; Lu, S.-W.; Chai, T. Data-driven soft-sensor modeling for product quality estimation using case-based reasoning and fuzzy-similarity rough sets. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 992−1003.

(2) Sheng, N.; Liu, Q.; Qin, S. J.; Chai, T. Comprehensive monitoring of nonlinear processes based on concurrent kernel projection to latent structures. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 1129−1137.

(3) Nomikos, P.; MacGregor, J. F. Monitoring batch processes using multiway principal component analysis. *AIChE J.* **1994**, *40*, 1361−1375.

(4) Nomikos, P.; MacGregor, J. F. Multi-way partial least squares in monitoring batch processes. *Chemom. Intell. Lab. Syst.* **1995**, *30*, 97−108.

(5) Yoo, C. K.; Lee, J.-M.; Vanrolleghem, P. A.; Lee, I.-B. On-line monitoring of batch processes using multiway independent component analysis. *Chemom. Intell. Lab. Syst.* **2004**, *71*, 151−163.

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

(6) Hu, K.; Yuan, J. Multivariate statistical process control based on multiway locality preserving projections. *J. Process Control* **2008**, *18*, 797−807.
(7) Lee, J.-M.; Yoo, C. K.; Lee, I. B. Fault detection of batch processes using multiway kernel principal component analysis. *Comput. Chem. Eng.* **2004**, *28*, 1837−1847.
(8) Tian, X.; Zhang, X.; Deng, X.; Chen, S. Multiway kernel independent component analysis based on feature samples for batch process monitoring. *Neurocomputing* **2009**, *72*, 1584−1596.
(9) 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. *Chemom. Intell. Lab. Syst.* **2011**, *107*, 106−115.
(10) He, Q. P.; Wang, J. Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes. *IEEE Trans. Semicond. Manuf.* **2007**, *20*, 345−354.
(11) Zhao, C.; Wang, F.; Gao, F.; Lu, N.; Jia, M. Adaptive monitoring method for batch processes based on phase dissimilarity updating with limited modeling data. *Ind. Eng. Chem. Res.* **2007**, *46*, 4943−4953.
(12) Zhao, C.; Wang, F.; Jia, M. Dissimilarity analysis based batch process monitoring using moving windows. *AIChE J.* **2007**, *53*, 1267−1277.
(13) Qi, Y.; Wang, Y.; Lu, C.; Wang, L. Improved batch process monitoring and diagnosis based on multiphase KECA. *IFAC-PapersOnLine* **2018**, *51*, 827−832.
(14) Garg, A.; Tangirala, A. K. Metrics for Interaction Assessment in Multivariable Control Systems Using Directional Analysis. *Ind. Eng. Chem. Res.* **2018**, *57*, 967−979.
(15) Xu, S.; Baldea, M.; Edgar, T. F.; Wojsznis, W.; Blevins, T.; Nixon, M. Root Cause Diagnosis of Plant-Wide Oscillations Based on Information Transfer in the Frequency Domain. *Ind. Eng. Chem. Res.* **2016**, *55*, 1623−1629.
(16) Gajjar, S.; Palazoglu, A. A data-driven multidimensional visualization technique for process fault detection and diagnosis. *Chemom. Intell. Lab. Syst.* **2016**, *154*, 122−136.
(17) Bin Shams, M. A.; Budman, H. M.; Duever, T. A. Fault detection, identification and diagnosis using CUSUM based PCA. *Chem. Eng. Sci.* **2011**, *66*, 4488−4498.
(18) Gharahbagheri, H.; Imtiaz, S. A.; Khan, F. Root Cause Diagnosis of Process Fault Using KPCA and Bayesian Network. *Ind. Eng. Chem. Res.* **2017**, *56*, 2054−2070.
(19) Liu, Z.; Ge, Z.; Chen, G.; Song, Z. Adaptive soft sensors for quality prediction under the framework of bayesian network. *Contr. Eng. Pract.* **2018**, *72*, 19−28.
(20) Ma, L.; Dong, J.; Peng, K. A practical propagation path identification scheme for quality-related faults based on nonlinear dynamic latent variable model and partitioned Bayesian network. *J. Frankl. Inst.* **2018**, *355*, 7570−7594.
(21) Rashidi, B.; Singh, D. S.; Zhao, Q. Data-driven root-cause fault diagnosis for multivariate non-linear processes. *Contr. Eng. Pract.* **2018**, *70*, 134−147.
(22) Xu, S.; Baldea, M.; Edgar, T. F.; Wojsznis, W.; Blevins, T. Root Cause Diagnosis of Plant-Wide Oscillations Based on Information Transfer in the Frequency Domain. *Ind. Eng. Chem. Res.* **2016**, *55*, 1623−1629.
(23) Zhu, Q.-X.; Luo, Y.; He, Y.-L. Novel Multiblock Transfer Entropy Based Bayesian Network and Its Application to Root Cause Analysis. *Ind. Eng. Chem. Res.* **2019**, *58*, 4936−4945.
(24) Chen, H.-S.; Yan, Z.; Zhang, X.; Lin, Y.; Yao, Y. Root cause diagnosis of process faults using conditional Granger causality analysis and Maximum Spanning Tree. *IFAC-PapersOnLine* **2018**, *51*, 381.
(25) Yuan, T.; Qin, S. J. Root cause diagnosis of plant-wide oscillations using granger causality. *J. Process Control* **2014**, *24*, 450−459.
(26) Chen, H.-S.; Yan, Z.; Yao, Y.; Huang, T.-B.; Wong, Y.-S. Systematic Procedure for Granger-Causality-Based Root Cause Diagnosis of Chemical Process Faults. *Ind. Eng. Chem. Res.* **2018**, *57*, 9500−9512.
(27) Duan, P.; Yang, F.; Chen, T.; Shah, S. L. Direct causality detection via the transfer entropy approach. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 2052−2066.
(28) Gao, W.; Cui, W.; Ye, W. Directed information graphs for the Granger causality of multivariate time series. *Phys. A* **2017**, *486*, 701−710.
(29) Ma, L.; Dong, J.; Peng, K. Root cause diagnosis of quality-related faults in industrial multimode processes using robust Gaussian mixture model and transfer entropy. *Neurocomputing* **2018**, *285*, 60−73.
(30) Kano, M.; Ohno, H.; Hasebe, S.; Hashimoto, I. Process monitoring based on dissimilarity of time series data. *Kagaku Kogaku Ronbunshu* **1999**, *25*, 1004−1009.
(31) Kano, M.; Nagao, K.; Ohno, H.; Hasebe, S.; Hashimoto, I. Dissimilarity of process data for statistical process monitoring. *IFAC Proc. Vol.* **2000**, *33*, 231−236.
(32) Zhang, H.; Qi, Y.; Wang, L.; Gao, X.; Wang, X. Fault detection and diagnosis of chemical process using enhanced KECA. *Chemom. Intell. Lab. Syst.* **2017**, *161*, 61−69.
(33) Zhao, C.; Wang, F.; Zhang, Y. Nonlinear process monitoring based on kernel dissimilarity analysis. *Contr. Eng. Pract.* **2009**, *17*, 221−230.
(34) Akaike, H. A new look at the statistical model identification. *IEEE Trans. Autom. Control* **1974**, *19*, 716−723.
(35) Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461−464.
(36) Birol, G.; Üdeyl, C.; Çinar, A. A modular simulation package for fed-batch fermentation: penicillin production. *Comput. Chem. Eng.* **2002**, *26*, 1553−1565.
(37) Aimin, M.; Peng, L.; Lingjian, Y. Neighborhood preserving regression embedding based data regression and its applications on soft sensor modeling. *Chemom. Intell. Lab. Syst.* **2015**, *147*, 86−94.
(38) Van Impe, J.; Gins, G. An extensive reference dataset for fault detection and identification in batch processes. *Chemom. Intell. Lab. Syst.* **2015**, *148*, 20−31.
(39) Yuan, X.; Ge, Z.; Song, Z. Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. *Ind. Eng. Chem. Res.* **2014**, *53*, 13736−13749.
(40) Chen, Q.; Wynne, R. J.; Goulding, P.; Sandoz, D. The application of principal component analysis and kernel density estimation to enhance process monitoring. *Contr. Eng. Pract.* **2000**, *8*, 531−543.

DOI: 10.1021/acs.iecr.9b04471
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX