I&EC research
Industrial & Engineering Chemistry Research

Article

Cite This: *Ind. Eng. Chem. Res.* 2019, 58, 6551−6561

pubs.acs.org/IECR

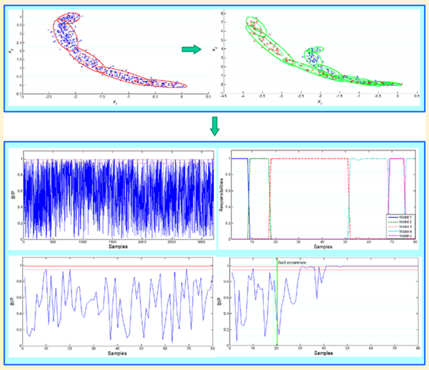
Recursive Gaussian Mixture Models for Adaptive Process Monitoring

Junhua Zheng, Qiaojun Wen, and Zhihuan Song*

State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, PR China

Supporting Information

ABSTRACT: Gaussian mixture models (GMM) have recently been introduced and widely used for process monitoring. This paper intends to develop a new recursive GMM model for adaptive monitoring of processes under time-varying conditions. Two model updating schemes with/without forgetting factors are both proposed. Bayesian inference probability index is used as the monitoring statistic in both of the continuous and batch process monitoring models. In order to reduce the online computational complexity, an updating strategy for both determinant and inverse of the covariance matrix during the monitoring process is particularly formulated. According to the simulation results of two case studies, efficiencies of both recursive modeling and adaptive monitoring performances are evaluated.



1. INTRODUCTION

In modern chemical engineering applications, it is of great interest to pursue high-quality products and to maintain safe operation. Model based fault detection and diagnosis methods have been well established for the purpose of process monitoring. However, due to the process complexity, accurate analytical models obtained from first-principles are often intractable. On the other hand, computer aided process control and instrumentation techniques lead to huge amounts of process data to be recorded and analyzed. As a result, data-based modeling and control methods have been widely applied in modern industrial processes.[1−5] For process monitoring, particularly, data-based methods have become popular in both academic research and real industrial applications.[6−12]

Nevertheless, many industrial processes may operate in different operation regimes, in those cases, the traditional MSPM methods will probably show poor monitoring performance. Recently, the Gaussian mixture model has been introduced for multimode process modeling. In the past years, various Gaussian mixture models (GMM) based methods have been constructed for process monitoring and soft sensor applications.[13−19] With the incorporation of the latent variable models, the GMM model has been further extended to low dimensional counterparts, such as mixtures of factor analyzers, mixtures of probabilistic PCA, etc.[20−25] More recently, the Gaussian mixture model has also been extended to handle semisupervised process data and big process data with a scalable form.[26]

Besides the multimode property, the time-varying behavior of the process data is also quite common in industrial processes. With the change of process specifications, the aging of main process equipment, seasoning effects, and so on, data-based modeling methods can only capture the data information recorded in the early stage of the process. Therefore, even for the most well trained MSPM method, it is difficult to describe all possible future states and condition of the process.[27] Thus, a strategy for online adaptation of the process data information is quite important. So far, different kinds of adaptive process monitoring approaches have been developed, including recursive PLS algorithms, fast moving window PCA (FMWPCA), moving window kernel PCA (MWKPCA), Just-In-Time Learning (JITL) strategy, and so on.[28−33]

In this paper, a new recursive scheme based on GMM is proposed to monitor time-varying processes. The initial Gaussian mixture model is estimated by the modified expectation-maximization (EM) algorithm. Then the recursive algorithm is adopted to update the mixture model parameters as new observations are obtained. Two model updating schemes with/without forgetting factors are both proposed. For both continuous and batch process monitoring purposes, Bayesian inference probability index is used as the monitoring statistic. In order to reduce the online computational complexity, an updating strategy for both determinant and inverse of the covariance matrix during the monitoring process is particularly formulated.

The rest of this article is organized as follows. In the next section, the preliminary knowledge of GMM and its training approach is briefly reviewed, followed by the proposed recursive form of the GMM algorithm. After that, model updating and process monitoring strategies for batch processes will be developed. In section 5, two case studies are provided to demonstrate the efficiency of the proposed method. Finally, conclusions are made.

Received: December 9, 2018
Revised: February 19, 2019
Accepted: April 1, 2019
Published: April 1, 2019

DOI: 10.1021/acs.iecr.8b06101

2. GAUSSIAN MIXTURE MODELS

The distribution of random variables $\mathbf{x} \in \Re^D$ from Gaussian mixture models can be written as a linear superposition of Gaussian components in the form[34]

$$p(\mathbf{x}) = \sum_{g=1}^{G} \pi_g \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{1}$$

where G is the number of Gaussian components, each of which has mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$, $g = 1, 2, ..., G$; and the nonnegative weights π_g, which are called mixing probabilities, satisfy the constraint $\sum_{g=1}^{G} \pi_g = 1$. Let $\boldsymbol{\theta}_g = \{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}$; for the g-th component $\mathbf{C}_g$, the Gaussian density function can be expressed as

$$\begin{aligned} p(\mathbf{x}|\mathbf{x} \in \mathbf{C}_g) &= N(\mathbf{x}|\boldsymbol{\theta}_g) \\ &= \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g^{-1} \right. \\ &\quad \left. \times (\mathbf{x} - \boldsymbol{\mu}_g) \right\} \end{aligned} \tag{2}$$

The model parameters can be written as $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_G, \pi_1, ..., \pi_G\}$. Given a set of N independent and identical distributed samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, the log-likelihood of the model is

$$\log p(\mathbf{X}|\boldsymbol{\Theta}) = \log \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\Theta}) = \sum_{i=1}^{N} \log \sum_{g=1}^{G} \pi_g p(\mathbf{x}_i|\boldsymbol{\theta}_g) \tag{3}$$

To estimate the model parameters, the maximum likelihood estimate (MLE) or maximum a posteriori (MAP) can be used. The expectation-maximization (EM) algorithm can be employed to solve the problem effectively. EM algorithm is an iterative algorithm that finds the local maxima of log-likelihood function. However, the objective function of EM algorithm is different from eq 3. According to the EM algorithm for GMM,[34] $\mathbf{X}$ is treated as an incomplete data set in which the missing part of GMM can be interpreted as N tags $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$; the objective function is the complete log-likelihood

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\Theta}) \tag{4}$$

With a previously known G, the parameters $\boldsymbol{\Theta}$ can be learned from eq 4 via the EM algorithm. In practice, however, it is not easy to set an appropriate G for GMM. With an unknown G, the process parameters $\{G, \boldsymbol{\Theta}\}$ can be estimated using several methods, among which a modified EM algorithm called F-J algorithm (Figueiredo Jain Tune Algorithm) is an efficient approach.[35] The F-J algorithm is based on the "minimum message length" (MML) criterion. For more details about the F-J algorithm, please see ref 35.

3. RECURSIVE GAUSSIAN MIXTURE MODELS (RGMM)

On the basis of the historical database, it is assumed that the Gaussian mixture models have been already trained. With the ongoing of the process, which may change from one condition to another, the static GMM model is updated with the incorporation of new informed data samples. In this section, the recursive form of GMM is proposed for adaptive monitoring of nonlinear processes.

3.1. Recursive GMM. Assume that there are m samples in the database that have been previously used for development of the GMM model. A new measurement $\mathbf{x}_{m+1}$ is then collected, and the model will be updated from $\boldsymbol{\Theta}^{(m)}$ to $\boldsymbol{\Theta}^{(m+1)}$.

In the previous model with parameters $\boldsymbol{\Theta}^{(m)}$, the responsibility (posterior probability) of sample $\mathbf{x}_i$ ($i = 1, ..., m, m + 1$) in the g-th Gaussian component $\mathbf{C}_g$ is given as

$$p^{(m)}(\mathbf{C}_g|\mathbf{x}_i) = \frac{\pi_g^{(m)} p(\mathbf{x}_i|\boldsymbol{\theta}_g^{(m)})}{\sum_{j=1}^{G} \pi_j^{(m)} p(\mathbf{x}_i|\boldsymbol{\theta}_j^{(m)})} \tag{5}$$

Because only one new sample is added to the model, we can assume that the model parameters do not change much from $\boldsymbol{\Theta}^{(m)}$ to $\boldsymbol{\Theta}^{(m+1)}$. Thus, the responsibility of sample $\mathbf{x}_i$ with model $\boldsymbol{\Theta}^{(m+1)}$, $p^{(m+1)}(\mathbf{C}_g \mid \mathbf{x}_i)$, can be approximated as[36]

$$p^{(m+1)}(\mathbf{C}_g|\mathbf{x}_i) \approx p^{(m)}(\mathbf{C}_g|\mathbf{x}_i) \tag{6}$$

after $\mathbf{x}_{m+1}$ is added to the model, since

$$\pi_g^{(m+1)} = \frac{1}{m+1} \sum_{i=1}^{m+1} p^{(m+1)}(\mathbf{C}_g|\mathbf{x}_i) \tag{7}$$

and

$$\pi_g^{(m)} = \frac{1}{m} \sum_{i=1}^{m} p^{(m)}(\mathbf{C}_g|\mathbf{x}_i) \tag{8}$$

therefore, the mixing probabilities $\pi_g^{(m+1)}$ are updated as

$$\pi_g^{(m+1)} = \pi_g^{(m)} + \frac{1}{m+1}[p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1}) - \pi_g^{(m)}] \tag{9}$$

and then the mixing probabilities are normalized to make sure

$$\sum_{g=1}^{G} \pi_g^{(m+1)} = 1 \tag{10}$$

The mean and covariance of the g-th Gaussian component are updated according to[36,37]

$$\boldsymbol{\mu}_g^{(m+1)} = \boldsymbol{\mu}_g^{(m)} + \frac{1}{m+1} \frac{p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1})}{\pi_g^{(m)}} (\mathbf{x}_{m+1} - \boldsymbol{\mu}_g^{(m)}) \tag{11}$$

$$\begin{aligned} \boldsymbol{\Sigma}_g^{(m+1)} &= \boldsymbol{\Sigma}_g^{(m)} + \frac{1}{m+1} \frac{p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1})}{\pi_g^{(m)}} [(\mathbf{x}_{m+1} - \boldsymbol{\mu}_g^{(m)}) \\ &\quad \times (\mathbf{x}_{m+1} - \boldsymbol{\mu}_g^{(m)})^T - \boldsymbol{\Sigma}_g^{(m)}] \end{aligned} \tag{12}$$

It is shown that in the updated eqs 9, 11, 12, the difference between model parameters in $\boldsymbol{\Theta}^{(m)}$ and $\boldsymbol{\Theta}^{(m+1)}$ is weighted with a term $1/(m + 1)$. Since m is the total number of training samples as well as testing samples that have been previously used to develop the Gaussian mixture model, m is sufficiently large and the model parameters between $\boldsymbol{\Theta}^{(m)}$ and $\boldsymbol{\Theta}^{(m+1)}$ have little difference. Thus, the approximation in eq 6 is valid.

3.2. Recursive GMM with Forgetting Factors. As the number of measurements in the process database grows, the impacts of current added measurements on the GMM model may become insignificant. However, to accommodate the process change, the recently recorded data samples that may contain new process information should be emphasized in the model updating procedure. To address this issue, another recursive GMM model is proposed here, with the employment of the forgetting factor used among data samples.

The reason that the current samples have less influence to the updated model is because the $1/(m + 1)$ terms in eqs 9, 11, 12 will become very small as m increases. In an extreme case, when m is sufficiently large, the current samples will have no influence

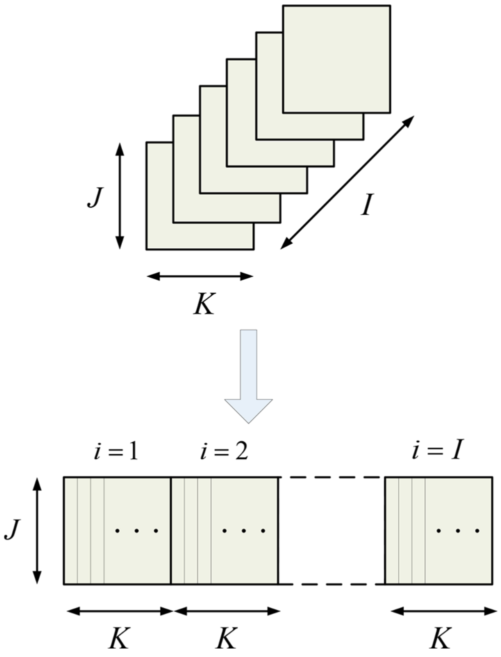


Figure 1. Unfolding of three-way data matrix $\mathbf{X}(I \times J \times K)$ into two-dimensional matrices.

to the model. Therefore, it is more appropriate to introduce a forgetting factor, in order to accommodate the change of the process condition. To maintain the influence of new added measurements for online model update, the term $1/(m + 1)$ in the RGMM model is replaced with a fixed small constant λ. Thus, the updating equations turn to be

$$\pi_g^{(m+1)} = \pi_g^{(m)} + \lambda[p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1}) - \pi_g^{(m)}] \tag{13}$$

$$\boldsymbol{\mu}_g^{(m+1)} = \boldsymbol{\mu}_g^{(m)} + \lambda\frac{p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1})}{\pi_g^{(m)}}(\mathbf{x}_{m+1} - \boldsymbol{\mu}_g^{(m)}) \tag{14}$$

$$\begin{aligned}\boldsymbol{\Sigma}_g^{(m+1)} = \boldsymbol{\Sigma}_g^{(m)} + \lambda\frac{p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1})}{\pi_g^{(m)}}[(\mathbf{x}_{m+1} - \boldsymbol{\mu}_g^{(m)}) \\ \times (\mathbf{x}_{m+1} - \boldsymbol{\mu}_g^{(m)})^T - \boldsymbol{\Sigma}_g^{(m)}]\end{aligned} \tag{15}$$

The constant λ is similar to the coefficient of exponentially weighted moving average (EWMA) models.[38] A higher value of λ represents more emphases on the recent samples, which on the other hand discounts older observations faster, while a lower λ leads to a lower forgetting rate on the historical information. Thus, a proper λ has to be determined on the basis of the trade-off between a finer model for current data and a model that keeps important process varying information.

In the rest of present work, the recursive Gaussian mixture model is adopted as the updating scheme for process monitoring, which is simply denoted as recursive GMM (RGMM).

4. PROCESS MONITORING SCHEME

4.1. Bayesian Inference Probability. After the GMM model and its updating strategy have been determined, the subsequent procedure is to derive the confidence boundary around the normal operating region for process monitoring. Previous researches on the GMM based monitoring scheme include the likelihood threshold and Bayesian inference probability (BIP).[17] Due to the high computational complexity, the likelihood threshold method is not a favorite one for recursive GMM. Therefore, BIP is chosen as the process monitoring scheme in the present work.

According to BIP, the posterior probability of the monitored sample $\mathbf{x}_m$ belonging to each Gaussian component is calculated at the first step

$$p^{(m)}(\mathbf{C}_g|\mathbf{x}_m) = \frac{\pi_g^{(m)}p(\mathbf{x}_m|\boldsymbol{\theta}_g^{(m)})}{\sum_{j=1}^{G}\pi_j^{(m)}p(\mathbf{x}_m|\boldsymbol{\theta}_j^{(m)})} \tag{16}$$

For each component $\mathbf{C}_g$ that follows a unimodal Gaussian distribution, the local Hoteling's T^2 statistic, which is the squared Mahalanobis distance of $\mathbf{x}_m$ from the center $\boldsymbol{\mu}_g^{(m)}$, can be calculated as

$$T^2(\mathbf{x}_m, \mathbf{C}_g) = (\mathbf{x}_m - \boldsymbol{\mu}_g^{(m)})^T(\boldsymbol{\Sigma}_g^{(m)})^{-1}(\mathbf{x}_m - \boldsymbol{\mu}_g^{(m)}) \tag{17}$$

By applying similar results from ref 17, the T^2 based probability index from each local Gaussian component

$$P_L^g(\mathbf{x}_m) = p\{T^2(\mathbf{x}, \mathbf{C}_g) \le T^2(\mathbf{x}_m, \mathbf{C}_g)\} \tag{18}$$

can be calculated by integrating the χ^2 probability density function with appropriate degree of freedom. Then, the global

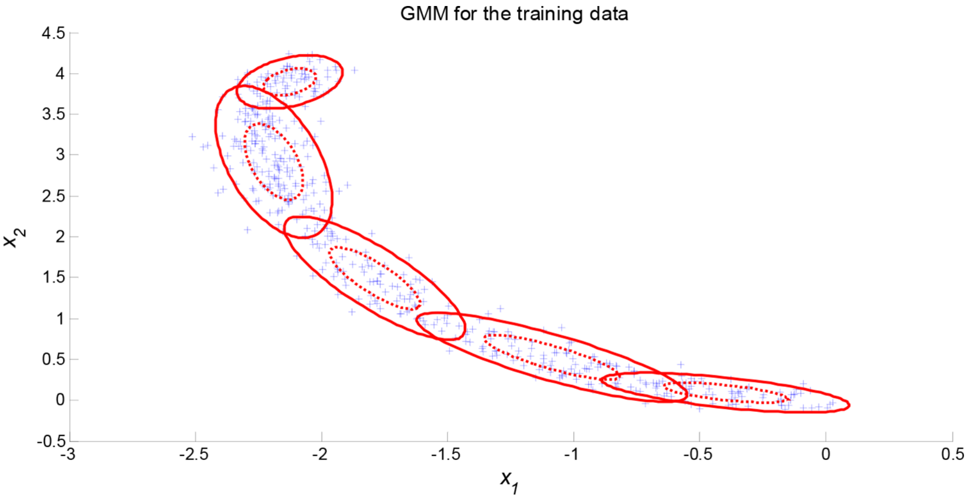


Figure 2. Initial modeled Gaussian mixtures.

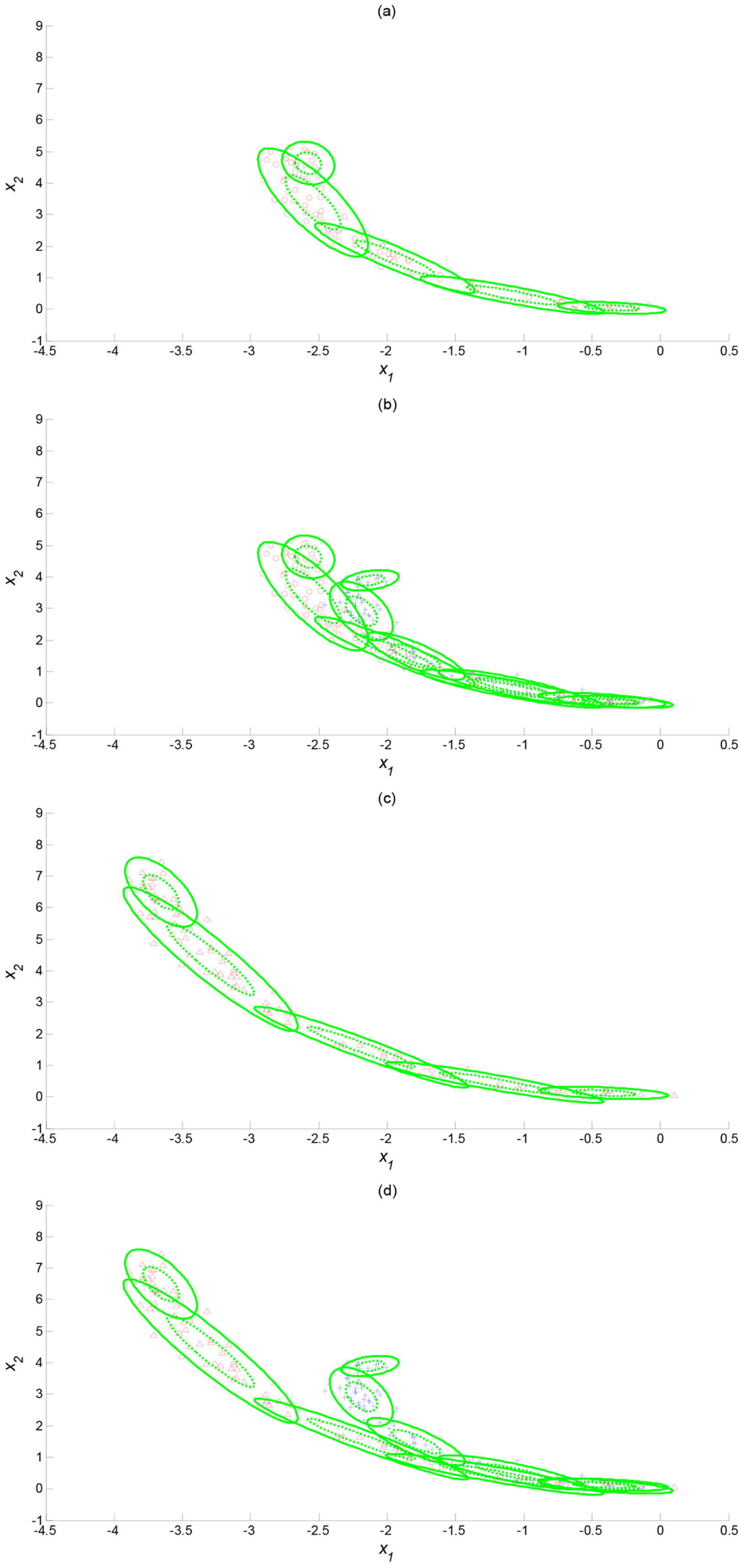
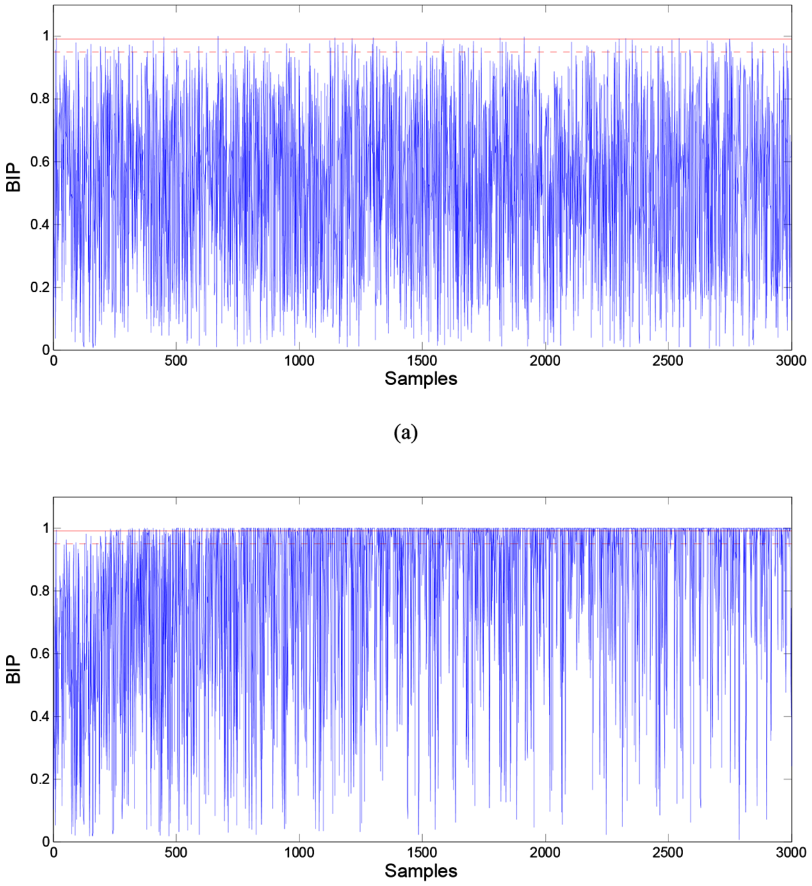


Figure 3. Updated GMM at different sampling points: (a) GMM at 1000th sampling point; (b) GMM at 1000th sampling point (in red ○) compared with the initial GMM (in blue +); (c) GMM at 3000th sampling point; (d) GMM at 3000th sampling point (in red △) compared with the initial GMM (in blue +).



(a)

(b)

Figure 4. BIP monitoring results via recursive GMM and ordinary GMM, respectively: (a) recursive GMM; (b) GMM. The dashed red line and solid red line represent the control limits with 95% and 99% confidence, separately.

BIP index is further defined to combine the local probability metrics across all the Gaussian clusters as follows

$$\mathrm{BIP} = \sum_{g=1}^{G} p(\mathbf{C}_g|\mathbf{x}_m)P_L^g(\mathbf{x}_m) \tag{19}$$

The BIP index satisfies

$$0 \le \mathrm{BIP} \le 1 \tag{20}$$

Under confidence level $1 - \alpha$, the process is determined within normal operation if

$$\mathrm{BIP} \le 1 - \alpha \tag{21}$$

Otherwise, the violation of BIP shows that there is probably an abnormal event happens in the process.

4.2. Computational Complexity for Model Updating and Process Monitoring. To update the model, the responsibility of each new sample to the local Gaussian components has to be calculated. According to eq 5, to obtain the responsibility, the Gaussian density function for each Gaussian component C_g

$$p(\mathbf{x}_m|\boldsymbol{\theta}_g^{(m)}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_g^{(m)}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}_m - \boldsymbol{\mu}_g^{(m)})^T \times (\boldsymbol{\Sigma}_g^{(m)})^{-1}(\mathbf{x}_m - \boldsymbol{\mu}_g^{(m)})\right\} \tag{22}$$

should be calculated. One can see that the determinant and inversion of local Gaussian components covariance $\boldsymbol{\Sigma}_g^{(m)}$ are involved. Meanwhile, it is noted in eq 17 that the inverse matrix of local Gaussian components covariance $\boldsymbol{\Sigma}_g^{(m)}$ is involved in the calculation of local T^2 statistics. As is well-known, the operation of matrix determinant and inversion is computationally demanding. To obtain the determinant of a $D \times D$ matrix, the most efficient algorithm has a running time of $O(D^{2.376})$.[39] With the inverse matrix calculation, the computational complexity of naïve operation is $O(D^3)$. Up to date, the most asymptotically efficient algorithm for matrix inversion is the Coppersmith–Winograd algorithm, which also has a running time of $O(D^{2.376})$.[39]

Table 1. Variation Range of Initial Value for Process Parameters

variables	variation range of initial values
culture volume (L)	100–104
substrate concentration (g/L)	12–16
dissolved oxygen concentration (g/L)	1–1.2
biomass concentration (g/L)	0.08–0.12
penicillin concentration (g/L)	0
carbon dioxide concentration (g/L)	0.8–1.0
pH	4.5–5
temperature (K)	298

For online utilization, the matrix determinant and inversion calculation will introduce a tedious computation step, making both of the GMM model updating and process monitoring inefficient. In this paper, in order to improve the efficiency of

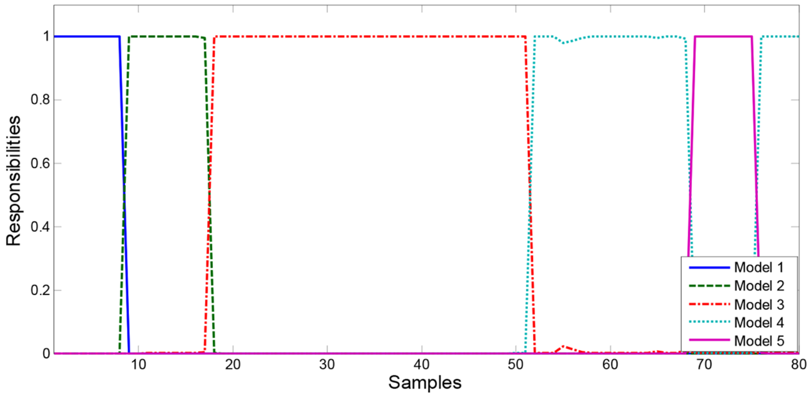


Figure 5. GMM responsibilities (posterior probabilities) of each sample in a batch.

Table 2. Summary of Fault Types Introduced at Different Batches of Fermentation

fault no.	fault type	occurrence batch no.	occurrence time (h)
1	aeration rate increases linearly with a slope 0.05	5	101
2	10% step decrease in agitator power	15	101
3	20% step increase in substrate feed rate	25	101

online computation, the inverse covariance matrix, as well as the determinant, is intended to be updated along with the GMM updating strategy.

Denote $\mathbf{S}_g^{(m)} = (\mathbf{\Sigma}_g^{(m)})^{-1}$; the update of $\mathbf{S}_g^{(m)}$ is given as follows

$$\mathbf{S}_g^{(m+1)} = (1-\beta)^{-1}\mathbf{S}_g^{(m)} - (1-\beta)^{-1}\mathbf{S}_g^{(m)}\boldsymbol{\delta} \times \{(1-\beta)\beta^{-1} + \boldsymbol{\delta}^T\mathbf{S}_g^{(m)}\boldsymbol{\delta}\}^{-1}\boldsymbol{\delta}^T\mathbf{S}_g^{(m)} \quad (23)$$

where $\beta = \lambda p^{(m)}(\mathbf{C}_g|\mathbf{x}_{m+1})/\pi_g^{(m)}$, and $\boldsymbol{\delta} = \mathbf{x}_m - \boldsymbol{\mu}_g^{(m)}$. Notice that $\boldsymbol{\delta}$ is a column matrix; there is no matrix inversion in eq 23. Detailed proof of eq 23 is provided in Supporting Information S1 of this paper.

The update of the covariance matrix determinant $|\mathbf{\Sigma}_g^{(m)}|$ is given as follows

$$|\mathbf{\Sigma}_g^{(m+1)}| = (1-\beta)^D\{1 + \beta\boldsymbol{\delta}^T\mathbf{S}_g^{(m)}\boldsymbol{\delta}/(1-\beta)\}|\mathbf{\Sigma}_g^{(m)}| \quad (24)$$

Similarly, there is no complex computation in eq 24. Detailed proof of eq 24 is provided in Supporting Information S2 of this paper.

4.3. Process Monitoring Scheme for Batch Processes.

Different from continuous processes, the data collected from batch processes are in the form of a three-way matrix that requires a preprocessing step of data unfolding prior to further analysis. Consider a three-way data matrix $\mathbf{X}(I \times J \times K)$ from batch process, where I represents the number of batches, J denotes the number of process variables, and K corresponds to the number of sampling points in each batch. In this paper, the data matrix is unfolded in such way that different batches are in a successive form, that is, the first sample of each batch will be added after the last sample in the previous batch. The data structure of the batch process and the unfolding strategy is illustrated in Figure 1.

The Gaussian mixture model can be updated in a sample-wise scheme, such as the case in the continuous process where the model will be updated as soon as new observations under normal operation are obtained. However, another method, which is called the batch-wise update scheme, may be more appropriate for modeling batch processes. According to this scheme, the model will be updated if and only if one normal batch has been added to the database and monitored. Compared with the sample-wise scheme, the batch-wise scheme has several advantages. First, since what we considered is slowly varying batch processes, it is not necessary to update the model frequently. Second, in the sample-wise scheme, it is hard to confirm whether the samples used for model updating are under normal operation condition. As a result, there is a potential risk that the model may be updated through a wrong direction. On the other hand, the batch-wise scheme will update the model only if a normal batch has been confirmed, which will greatly reduce the missing alarm rate. Moreover, it is possible to miss small disturbance faults in the sample-wise scheme,

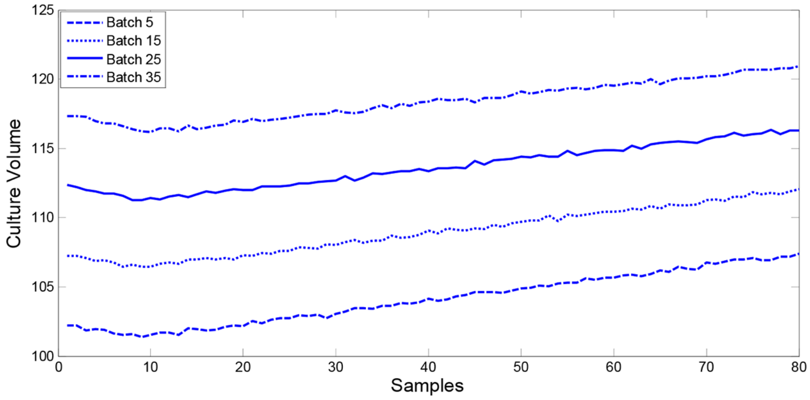


Figure 6. Culture volumes in different batches.

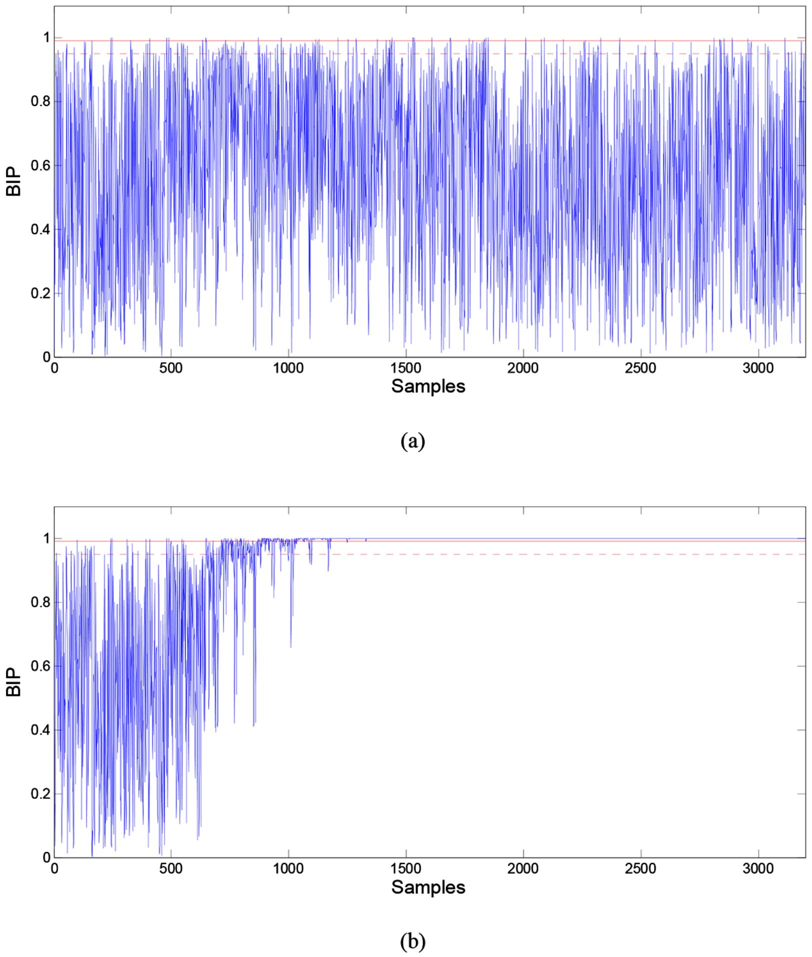


Figure 7. BIP monitoring results with normal penicillin process via recursive GMM and ordinary GMM, respectively: (a) recursive GMM; (b) GMM. The dashed red line and solid red line represent the control limits with 95% and 99% confidence, separately.

since the sample-wise scheme may "tolerate" inconspicuous faults during the model updating process. Therefore, the batch-wise update scheme is more reasonable for batch process monitoring.

Detailed procedures for monitoring batch processes are given as follows.

Step 1: Collect the data under normal operation condition and data preprocessing.

Step 2: Apply the F-J algorithm to estimate the parameters of GMM as well as the number of Gaussians.

Step 3: In online monitoring step, for each data sample in a new batch, the analysis procedures are given as follows:

- 3.1 For a new sample, compute the responsibility to each Gaussian component using eq 16.
- 3.2 Compute the corresponding Hoteling's T^2 statistic to each Gaussian component using eq 17.
- 3.3 Compute the BIP index for the sample as given in eqs 18, 19; determine if there is any abnormality in the process.
- 3.4 Go to 3.1, and keep monitoring the new batch until all the samples have been analyzed.

Step 4: If the batch is normal, update the model as eqs 13, 14, 15, and go to Step 3.

5. CASE STUDIES

In this section, two case studies are demonstrated to show the effectiveness of the proposed approach, one of which is a numerical case while the other is the well-known benchmark simulation of fed-batch penicillin production process.

5.1. Numerical Example. The numerical data used here for demonstrating our proposed method. For simplicity and clear figure illustration, only two variables are chosen for process modeling. The process is as follows:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} t^2 - 3at \\ -t^3 + 3at^2 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \tag{25}$$

where e_1 and e_2 are independent and identically distributed Gaussian sequences of zero mean and variance 0.01, which represent measurement uncertainty, t is a uniformly distributed stochastic sequence within the range of [0.01 2], and a is a time varying parameter. From the above process with $a = 1$, a data set $\mathbf{X}_1$ consisting of 500 samples is generated to model the Gaussian mixtures. For model update and monitoring, another data set $\mathbf{X}_2$ containing 3000 samples is also generated with a varying a that gradually increases by 0.0001 per sample, from an initial value of 1.

The scatter plot of $\mathbf{X}_1$ is shown in Figure 2. One can see that the training data show a strong nonlinear property. An initial GMM is trained on $\mathbf{X}_1$ via the F-J algorithm. $G = 5$ is automatically chosen as the number of Gaussian components. The dashed line and solid line represent the 1-sigma and 2-sigma contours of the corresponding local Gaussian components, respectively. It is shown that the data are well modeled with the mixtures of 5 Gaussian components.

DOI: 10.1021/acs.iecr.8b06101
Ind. Eng. Chem. Res. 2019, 58, 6551−6561

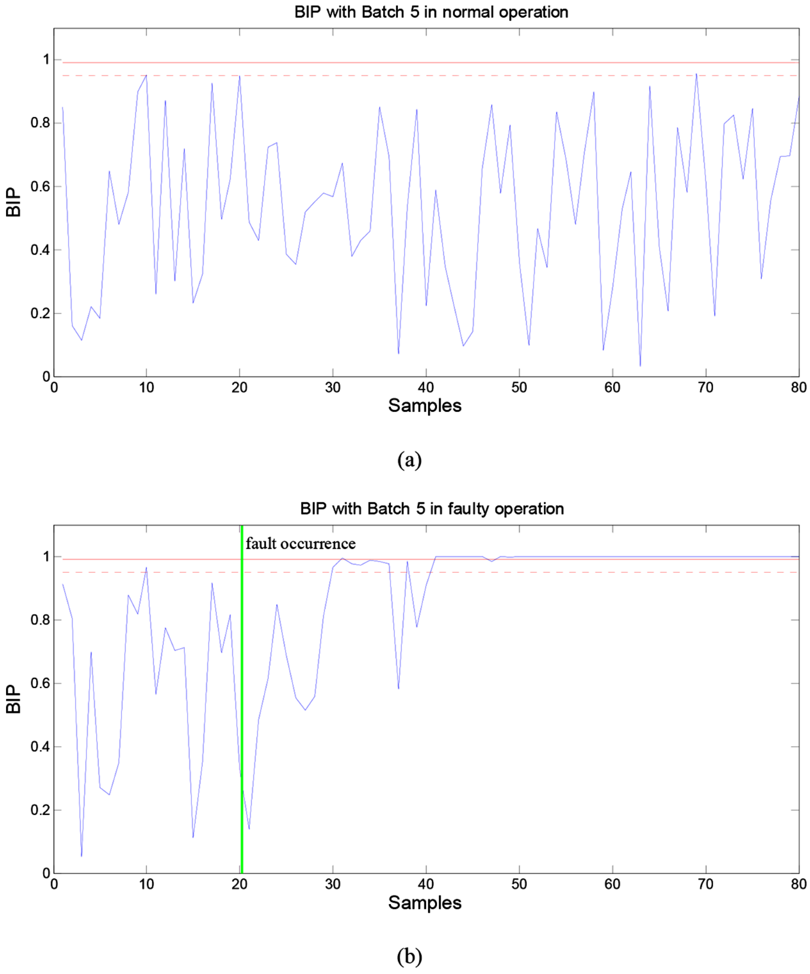


Figure 8. Monitoring results with Batch 5: (a) in normal operation; (b) in fault operation. The green line represents the occurrence point of the fault in the process.

Next, the data samples in $\mathbf{X}_2$ are successively added for model updating. Particularly, the mixture models at sampling points 1000 and 3000 are captured for comparing with the initial GMM model. The illustration is shown in Figure 3, where for each GMM, the most recent 100 samples are plotted. One can see that the distributions at $t = 1000$ and $t = 3000$ are quite different from the initial distribution. However, the model updating scheme makes the data well captured by the mixture models. In contrast, the static GMM does not model the process well; this is because it cannot capture the time-varying property of the process parameter a.

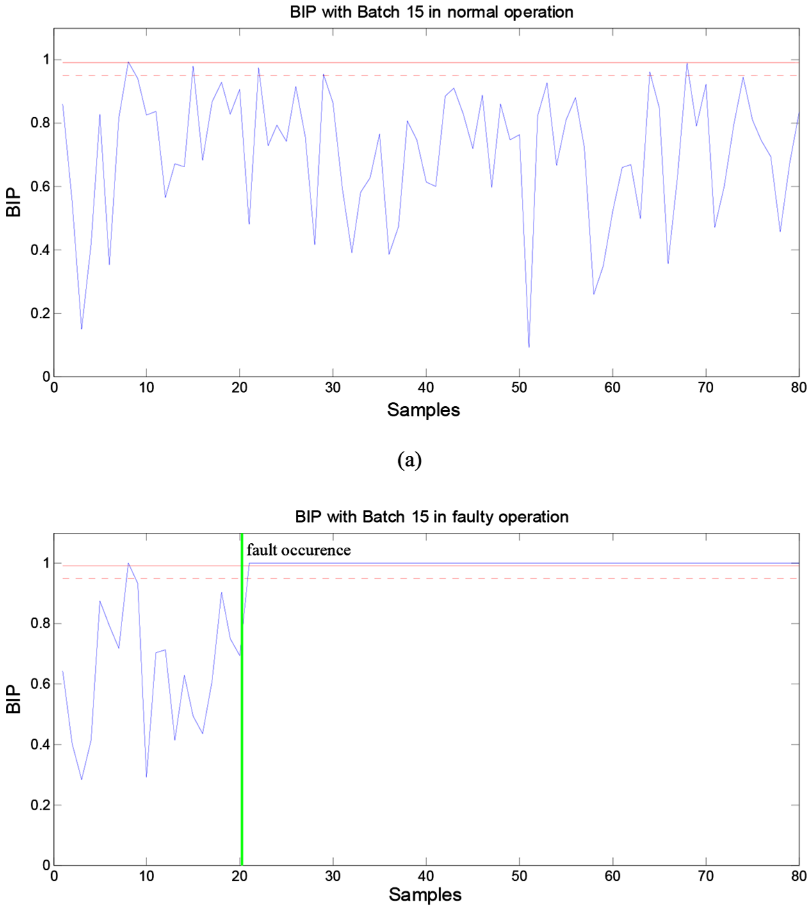
The monitoring results of both RGMM and static GMM models are demonstrated in Figure 4. It can be seen that the false alarm rate in Figure 4(b) becomes much more significant after about 230 samples. However, the false alarm rate provided by the recursive GMM model keeps as a low value. Among all of the 3000 data samples, only 9 data samples have exceeded the 99% control limit. In order to make the results justified, 100 Monte Carlo experiments have been carried out; the average result of the false alarm rate is less than 1% for this example.

5.2. Fermentation (Penicillin) Process.

The production of fed-batch penicillin fermentation is an important biochemical industrial cultivation process. Because of its nonlinearity, dynamics and multiphase characteristics during the production process, it has received wide attention as the subject of many studies. At the first stage of a common penicillin fermentation process, special microorganisms grow and multiply in a fermenter with certain conditions. Penicillin will be produced as the metabolite when the concentration of microorganisms in the fermenter reaches a high degree. During the penicillin production stage, substrate should be fed to the fermenter to maintain the growth of microorganisms.

The simulation experiment is carried out using the PenSim v2.0 simulator developed by the monitoring and control group of the Illinois Institute of Technology.[40] There are 11 process variables selected for modeling in this work. Each batch has duration of 400 h, including a preculture stage of about 50 h and a fed-batch stage of about 350 h. The sampling interval is chosen as 5 h per sample. To model the initial Gaussian mixtures, process data of 10 batches are collected, which are denoted as $\mathbf{X}_1(10 \times 11 \times 80)$. The variation ranges of initial process parameters are set as in Table 1, where the variables follow uniform distribution. To simulate the real process environment, Gaussian noises are added to each monitoring variables, with zero mean and 5% standard deviation of the corresponding variable.

Via F-J algorithm, an initial GMM is built on the basis of $\mathbf{X}_1$. $G = 5$ is automatically chosen as the number of Gaussian components. To see the modeling performance, the posterior probabilities of all samples in the first batch is shown in Figure 5. The clustering result is in accordance with the multiphase property of the process, although the process data from 52th to 68th



(a)

(b)

Figure 9. Monitoring results with Batch 15: (a) in normal operation; (b) in fault operation. The green line represents the occurrence point of the fault in the process.

sample and 76th to 80th sample have largest responsibility in the fourth Gaussian component. One can see that although the operation information on the process is unknown, the GMM still models the process well.

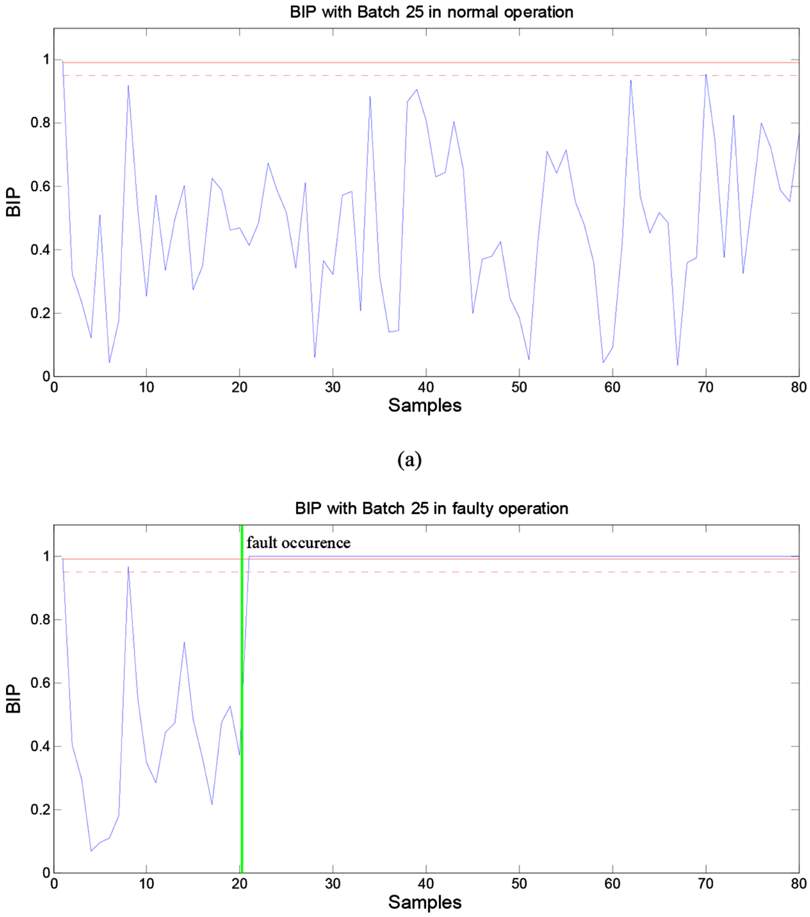
For model update, $\mathbf{X}_2(40 \times 11 \times 80)$ with 40 batches are generated, in which the initial culture volumes of these batches are artificially set to gradually increase from 100.5 to 120 L to mimic the normal batch-wise slow variations. The increasing rate of the initial culture volume is 0.5 L per batch. The initial values of other parameters are set as in Table 2, which are the same as in $\mathbf{X}_1$. Gaussian noises are also added to each monitoring variable. The culture volumes with Gaussian noises of the fifth, 15th, 25th, and 35th batch are plotted in Figure 6.

The model is updated using $\mathbf{X}_2$ via the recursive Algorithm given in section 3. The BIP monitoring result of $\mathbf{X}_2$ is given in Figure 7(a). For comparison, the monitoring result of static GMM is also shown in Figure 7(b). While the BIP violation rate keeps normal in Figure 7(a), for the latter approach, the BIP violation rate goes to a very high value after about 700th sample, which is actually starting from the ninth batch. With the increase of the variation of process parameters, the violation percent of static GMM based BIP becomes 100%. Therefore, it can be inferred that the static GMM cannot capture the process property any more. In contrast, the recursive GMM performs very well, which can be seen in Figure 7(b). Similarly, to make the results justified, 100 Monte Carlo experiments have also been carried out. It turns out that the average results of the false alarm rate for the recursive GMM method is around 1.5%, while the static GMM method shows a high false alarm rate.

Next, three different faults are introduced into the 5th, 15th, and 25th batch of the penicillin process. The detailed information on these faults is listed in Table 2. Other initial values of parameters are the same as used in the normal batches. Figure 8 illustrates the monitoring results for the fifth batch in normal operation and faulty operation, respectively, while the detailed monitoring results of the 15th batch and the 25th batch are given in Figure 9 and Figure 10. Particularly, for the faulty operation condition in the fifth batch, there are about 10 samples delayed before the fault is detected. This is probably because the fault in batch 5 is a ramp fault, which is drift slowly during the batch. Compared to this fault, the faults in other two batches have been detected as soon as they are introduced. On the basis of these monitoring results, it can be seen that the proposed method can detect different kinds of process faults accurately with low false alarm rate.

6. CONCLUSIONS

In this paper, the recursive form of Gaussian mixture model has been proposed to model time-varying processes. The static GMM algorithm is used to train initial models from historical data. As new process data have been observed, the mixture model is updated by the recursive algorithm with the forgetting



(a)

(b)

Figure 10. Monitoring results with Batch 25: (a) in normal operation; (b) in fault operation. The green line represents the occurrence point of the fault in the process.

factor strategy. The corresponding process monitoring scheme based on Bayesian inference probability has also been developed. The proposed RGMM based process monitoring algorithm is simple in form and efficient in online computation. The case study on a numerical process illustrates the adaptive modeling performance of RGMM graphically, while the application on the fed-batch penicillin fermentation process demonstrates the superiority of the proposed strategy in monitoring process faults.

For future research, the basic GMM model can be effectively extended to the nonlinear form, with the possible incorporation of some deep learning approaches, such as autoencoder, Restricted Boltzmann Machine (GRBM), and the recurrent neural network (RNN).[41,42] Second, it is noted that the dynamical data information has not been well considered by the GMM model, which should be important for online monitoring of the industrial process. Besides, with the ever increasing amounts of data in industrial processes, future works employing big data modeling tools are highly required in this area;[43−45] for example, the basic and the proposed recursive GMM model can both be extended to the distributed parallel form, in order to handle the big process data more efficiently.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.iecr.8b06101.

Proof of eq 23 (PDF)
Proof of eq 24 (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: songzhihuan@zju.edu.cn.

ORCID

Junhua Zheng: 0000-0002-8419-4572
Zhihuan Song: 0000-0002-6693-3247

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (61833014), the Natural Science Foundation of Zhejiang Province (LR18F030001), and the Fundamental Research Funds for the Central Universities under grant 2018XZZX002-09.

■ REFERENCES

(1) Ge, Z.; Song, Z.; Gao, Z. Review of Recent Research on Data-Based Process Monitoring. *Ind. Eng. Chem. Res.* **2013**, *52*, 3543−3562.

(2) Qin, S. Process data analytics in the era of big data. *AIChE J.* **2014**, *60*, 3092−3100.

(3) Ge, Z. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemom. Intell. Lab. Syst.* **2017**, *171*, 16−25.

(4) Ge, Z.; Song, Z.; Ding, S.; Huang, B. Data mining and analytics in the process industry: The role of machine learning. *IEEE Access* **2017**, *5*, 20590−20616.

(5) Zhu, J.; Ge, Z.; Song, Z.; Gao, F. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. *Annual Reviews in Control* **2018**, *46*, 107−133.

(6) Jiang, Q.; Yan, X. Plant-wide process monitoring based on mutual information multi-block principal component analysis. *ISA Trans.* **2014**, *53*, 1516−1527.

(7) Lee, J.; Yoo, C.; Choi, S.; Vanrolleghem, P.; Lee, I. Nonlinear process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* **2004**, *59*, 223−234.

(8) Ge, Z. Distributed predictive modeling framework for prediction and diagnosis of key performance index in plant-wide processes. *J. Process Control* **2018**, *65*, 107−117.

(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) Ge, Z.; Zhang, M.; Song, Z. Nonlinear process monitoring based on linear subspace and Bayesian inference. *J. Process Control* **2010**, *20*, 676−688.

(11) Yao, L.; Ge, Z. Locally weighted prediction methods for latent factor analysis with supervised and semi-supervised process data. *IEEE Transactions on Automation Science and Engineering* **2017**, *14*, 126−138.

(12) Ge, Z.; Liu, Y. Analytic Hierarchy Process Based Fuzzy Decision Fusion System for Model Prioritization and Process Monitoring Application. *IEEE Transactions on Industrial Informatics* **2019**, *15*, 357−365.

(13) Zhao, N.; Li, S. Nonlinear and Non-Gaussian Process Monitoring Based on Simplified R-Vine Copula. *Ind. Eng. Chem. Res.* **2018**, *57*, 7566−7582.

(14) Choi, S.; Park, J.; Lee, I. Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis. *Comput. Chem. Eng.* **2004**, *28*, 1377−1387.

(15) Choi, S.; Martin, E.; Morris, A. Fault detection based on a maximum-likelihood principal component analysis (PCA) mixture. *Ind. Eng. Chem. Res.* **2005**, *44*, 2316−2327.

(16) Jiang, Q.; Huang, B.; Yan, X. GMM and optimal principal components-based Bayesian method for multimode fault diagnosis. *Comput. Chem. Eng.* **2016**, *84*, 338−349.

(17) Yu, J.; Qin, S. Multimode process monitoring with Bayesian inference based finite Gaussian mixture models. *AIChE J.* **2008**, *54*, 1811−1829.

(18) Chen, T.; Zhang, J. On-line multivariate statistical monitoring of batch processes using Gaussian mixture model. *Comput. Chem. Eng.* **2010**, *34*, 500−507.

(19) Liu, J.; Liu, T.; Chen, J. Sequential local-based Gaussian mixture model for monitoring multiphase batch processes. *Chem. Eng. Sci.* **2018**, *181*, 101−113.

(20) Ge, Z. Process data analytics via probabilistic latent variable models: A tutorial review. *Ind. Eng. Chem. Res.* **2018**, *57*, 12646−12661.

(21) Raveendran, R.; Huang, B. Two layered mixture Bayesian probabilistic PCA for dynamic process monitoring. *J. Process Control* **2017**, *57*, 148−163.

(22) Liu, Y.; Liu, B.; Zhao, X.; Xie, M. A Mixture of Variational Canonical Correlation Analysis for Nonlinear and Quality-Relevant Process Monitoring. *IEEE Transactions on Industrial Electronics* **2018**, *65*, 6478−6486.

(23) Ge, Z.; Chen, X. Dynamic probabilistic latent variable model for process data modeling and regression application. *IEEE Transactions on Control Systems Technology* **2019**, *27*, 323−331.

(24) Wang, L.; Deng, X.; Cao, Y. Multimode complex process monitoring using double-level local information based local outlier factor method. *J. Chemom.* **2018**, *32*, e3048.

(25) Zhou, L.; Zheng, J.; Ge, Z.; Song, Z.; Shan, S. Multimode Process Monitoring Based on Switching Autoregressive Dynamic Latent Variable Model. *IEEE Transactions on Industrial Electronics* **2018**, *65*, 8184−8194.

(26) Yao, L.; Ge, Z. Scalable Semi-supervised GMM for Big Data Quality Prediction in Multimode Processes. *IEEE Transactions on Industrial Electronics* **2019**, *66*, 3681−3692.

(27) Kadlec, P.; Grbic, R.; Gabrys, B. Review of adaptation mechanisms for data driven soft sensors. *Comput. Chem. Eng.* **2011**, *35*, 1−24.

(28) Qin, S. Recursive PLS algorithms for adaptive data modeling. *Comput. Chem. Eng.* **1998**, *22*, 503−514.

(29) Chan, L.; Wu, X.; Chen, J.; Xie, L.; Chen, C. Just-In-Time Modeling with Variable Shrinkage Based on Gaussian Processes for Semiconductor Manufacturing. *IEEE Trans. Semiconductor Manuf.* **2018**, *31*, 335.

(30) Liu, Y.; Yang, C.; Gao, Z.; Yao, Y. Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes. *Chemom. Intell. Lab. Syst.* **2018**, *174*, 15−21.

(31) Liu, Y.; Liang, Y.; Gao, Z.; Yao, Y. Online Flooding Supervision in Packed Towers: An Integrated Data-Driven Statistical Monitoring Method. *Chem. Eng. Technol.* **2018**, *41*, 436−446.

(32) Jiang, Q.; Yan, X.; Huang, B. Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference,. *IEEE Transactions on Industrial Electronics* **2016**, *63* (1), 377−386.

(33) Wang, H.; Li, P.; Gao, F.; Song, Z.; Ding, S. Kernel classifier with adaptive structure and fixed memory for process diagnosis. *AIChE J.* **2006**, *52*, 3515−3531.

(34) Bishop, C. *Pattern Recognition and Machine Learning*; Springer, 2006.

(35) Figueiredo, M.; Jain, A. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2002**, *24*, 381−396.

(36) Zivkovic, Z.; Heijden, F. Revursive unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Machine Intell.* **2004**, *26*, 651−656.

(37) Titterington, D. Recursive parameter estimation using incomplete data. *Journal of Royal Statistical Society, Series B (Methodological)* **1984**, *46* (2), 257−267.

(38) Choi, S.; Martin, E.; Morris, A.; Lee, I. Adaptive multivariate statistical process control for monitoring time-varying processes. *Ind. Eng. Chem. Res.* **2006**, *45*, 3108−3118.

(39) Cormen, T.; Leiserson, C.; Rivest, R.; Stein, C. *Introduction to Algorithms*, 3rd ed.; MIT Press, 2009.

(40) Birol, G.; Undey, C.; Cinar, A. A modular simulation package for fed-batch fermentation: Penicillin production. *Comput. Chem. Eng.* **2002**, *26*, 1553−1565.

(41) Yao, L.; Ge, Z. Deep Learning of Semi-supervised Process Data with Hierarchical Extreme Learning Machine and Soft Sensor Application. *IEEE Transactions on Industrial Electronics* **2018**, *65*, 1490−1498.

(42) Yuan, X.; Huang, B.; Wang, Y.; Yang, C.; Gui, W. Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Transactions on Industrial Informatics* **2018**, *14*, 3235−3243.

(43) Zhu, J.; Ge, Z.; Song, Z. Distributed Parallel PCA for Modeling and Monitoring of Large-scale Plant-wide Processes with Big Data. *IEEE Transactions on Industrial Informatics* **2017**, *13*, 1877−1885.

(44) Yao, L.; Ge, Z. Big data quality prediction in the process industry: a distributed parallel modeling framework. *J. Process Control* **2018**, *68*, 1−13.

(45) Zhang, X.; Ge, Z. Local Parameter Optimization of LSSVM for Industrial Soft Sensing with Big Data and Cloud Implementation. *IEEE Transactions on Industrial Informatics* **2019**, DOI: 10.1109/TII.2019.2900479.

DOI: 10.1021/acs.iecr.8b06101