Data-Driven Batch-End Quality Modeling and Monitoring Based on Optimized Sparse Partial Least Squares

Qingchao Jiang, *Member, IEEE*, Xuefeng Yan, Hui Yi, and Furong Gao

***Abstract*—Batch-end quality modeling is used to predict the quality by using batch measurements and generally involves a large number of predictor variables. However, not all of the variables are beneficial for the prediction. Conventional multiway partial least squares (PLS) may not function properly for batch-end quality modeling because of many irrelevant predictor variables. This paper proposes an optimized sparse PLS (OSPLS) modeling approach for simultaneous batch-end quality prediction and relevant-variable selection. The effect of irrelevant variables on the quality-prediction performance is analyzed, and the importance of the relevant-variable selection is emphasized. Then, an OSPLS batch-end quality modeling approach is developed by incorporating the variable resolution optimization and sparse PLS modeling. The quality-prediction accuracy and modeling interpretability are improved because only quality-relevant variables are selected, and quality-irrelevant variables are eliminated. Based on the selected quality-relevant variables, a statistic is established for monitoring the quality status. The proposed OSPLS-based modeling and monitoring approach is applied on a fed-batch penicillin fermentation process and an industrial injection molding process. The results are compared with the state-of-the-art methods to verify the effectiveness of the OSPLS approach.**

***Index Terms*—Batch-end quality prediction, batch processes, optimized sparse partial least square (OSPLS), soft sensing, sparse modeling.**

Manuscript received November 4, 2018; revised February 11, 2019, March 15, 2019, and April 26, 2019; accepted June 5, 2019. Date of publication June 19, 2019; date of current version January 3, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61603138 and Grant 21878081, in part by the Shanghai Pujiang Program under Grant 17PJD009, in part by the Hong Kong Research Grant Council Project under Grant 16207717, and in part by the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017. *(Corresponding author: Xuefeng Yan.)*

Q. Jiang and X. Yan are with the Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: qchjiang@ecust.edu.cn; xfyan@ecust.edu.cn).

H. Yi is with the College of Electronic Engineering and Control Science, Nanjing University of Technology, Nanjing 211816, China (e-mail: jsyihui@126.com).

F. Gao is with the Department of Chemical and Biomolecular Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: kefgao@ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2019.2922941

I. Introduction

LARGE portions of value-added products are produced in chemical and pharmaceutical industries by batch processes. Generally, a batch process consists of several phases, and the variables in a batch run are expected to follow a predefined recipe. Due to the variations in environmental conditions, reaction depths, or raw materials, the variable evolution recipe may be deviated, and the final product quality may be unsatisfactory. Thus, timely assessment of the process state and estimation of the final product quality is important [1], [2]. However, the quality variable is generally obtained with some delay because of the technique used or economic limitation. Establishing a soft-sensor model for quality prediction is important. Quality modeling and monitoring techniques are typically classified into two types, namely, mechanism (white-box) models and data-driven (black-box) models [3]–[7]. On the one hand, establishing a mathematic model is difficult because the reaction during a process is generally complex. On the other hand, abundant of history data are stored with the rapid advancement of sensing techniques. Data-driven modeling and monitoring techniques are gaining increasing attention [8]–[13].

Least square (LS) is the basic linear regression method for quality or key-performance-indicator modeling [14]. However, the LS generally fails in dealing with high-dimensional and highly correlated data, because of the regression coefficient stability and computational efficiency problems. To handle high-dimensional and highly correlated data, partial least squares (PLS) is proposed and among the most popular data-driven soft-sensor development methods [15]. For batch processes, the multiway PLS (MPLS) that unfolds the three-way data as two-way data is generally used [16]. However, the following defects of classical MPLS method exist, which may degrade the prediction performance. After data unfolding, the number of predictor variables can be remarkably large, whereas the number of predictor measurements is generally small. For example, in a batch process that has ten variables and 200 measurements in each batch and a set of data with 100 batches, the number of predictor variables is $10 \times 200 = 2000$ while the number of predictor measurements is 100. The number of samples n is much smaller than the number of variables m, this refers to the large m small n problem. Not all predictor variables are beneficial for predicting the final quality; the existence of irrelevant variables may damage useful information and degrade prediction

performance. Therefore, it is important to select the relevant variables and eliminate the irrelevant variables in PLS-based modeling.

In PLS-based soft-sensor development, several variable-selection methods are developed, which are categorized into three categories, namely, filter methods, wrapper methods, and embedded methods [17]–[19]. In filter methods, a PLS regression (PLSR) model is first established on the data, then variable selection is performed according to certain rules in evaluating the variable importance. Wrapper methods are computationally expensive because they generally involve double-iterative procedures. The embedded methods perform the selection of variables using a one-iterative procedure, therefore are generally less time consuming than wrapper methods. However, these methods may not function effectively in batch-end quality prediction because of the process complexity and the large *m* small *n* problem.

In dealing with the large *m* small *n* problem in batch-end quality prediction, representative techniques include the sparse PLS (SPLS) [20] and the multiresolution PLS (MRPLS) [21]. SPLS performs simultaneous regression and selects variables by adding the L_1 penalty term to the optimization objective function of PLS. SPLS produces sparse linear combinations of predictor variables and makes a subtle variable selection. However, the prediction performance will also degrade as the number of quality-irrelevant variable increases. MRPLS performs variable selection by considering the variable time resolution or the time periods. Facco *et al*. highlighted that the quality prediction of PLS can be improved by including time series dynamic information into the modeling. A moving average three-phase PLS estimator was developed for a real-world industrial batch polymerization process [22]. Gins *et al*. discussed the effect of time resolution selection on the prediction performance and proposed a multiresolution quality prediction (MRQP) method [23]. The improved performance is obtained when the structured correlation in time and variable dimensions is considered. More recently, Rato and Reis further extended the MRQP method and proposed an MRPLS modeling method for batch data analysis [21]. The optimal-variable selection is conducted in three dimensions, namely, the variable dimension, the resolution dimension, and the stage dimension. The quality-prediction performance is enhanced because the operational stage dimension is introduced, and more information is available.

The existing MRQP method adjusts the predictor variable dimension by controlling the variable resolution, which effectively avoids high-dimensional optimization. However, the MRQP can only make rough variable selections. For example, optimal resolution of a variable may vary at different production periods. Even in the same production period, the optimal variable resolution can be different. Moreover, the MRQP method selects only one resolution during the entire batch running or an entire operation phase. If too many production periods are separated, then the variable selection becomes much more complex.

Given the aforementioned observations, we study the data-driven batch-end quality modeling and monitoring for batch processes. The novelty and contributions of the current work are as follows.

1) The impact of variable selection on the batch-end quality prediction is analyzed to enhance the basic data-driven batch-end quality modeling and monitoring theory.
2) An optimized SPLS (OSPLS) modeling approach is proposed for efficient batch-end quality prediction. The OSPLS aims to achieve simultaneous quality prediction and relevant-variable selection by optimizing the variable resolution before SPLS modeling through a stochastic optimization approach. The resolution optimization conducts a rough variable selection, whereas the SPLS performs a further subtle variable selection.
3) A monitoring statistic is then established based on the selected quality-relevant variables to identify the difficult-to-measure batch-end quality status.
4) The advantages of the proposed OSPLS modeling scheme are theoretically analyzed. The OSPLS superiority is verified through experimental studies on the simulated fed-batch penicillin fermentation (FBPF) and an industrial injection molding (IIM).

The rest of this paper is organized as follows. Section II reviews the standard MPLS modeling approach and provides a motivational analysis on sparse modeling. Section III details the scheme and discusses the properties of OSPLS-based modeling. The experimental studies on the FBPF process and the IIM process are carried out in Section IV. Finally, Section V concludes this paper.

Notation: The notations used here are standard except where otherwise specified. The superscript "$\boldsymbol{T}$" represents the transport of vectors or matrices. $|.|$ refers to the absolute value of a scalar. $|.|_1$ and $|.|_2$ represent the L_1 and L_2 norm of vectors, respectively. $\mathrm{corr}(\boldsymbol{X}, \boldsymbol{Y})$ represents the correlation between $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\boldsymbol{\Sigma}_X = \mathrm{var}(\boldsymbol{X})$ denotes the variance (covariance) of $\boldsymbol{X}$. $\lfloor a \rfloor$ represents the largest integer less than or equal to a. $\boldsymbol{S}_{XY}$ denotes the estimated covariance (from data) of $\boldsymbol{X}$ and $\boldsymbol{Y}$. $F_\alpha(a, b)$ denotes the F distribution with degrees of freedom a and b, and the level of significance α.

II. Preliminaries and Motivations

The standard MPLS for quality prediction is presented and the effect of irrelevant variables on prediction performance is analyzed. Then, the state-of-the-art SPLS and MRPLS approaches are reviewed, and the areas for improvements are discussed.

A. MPLS Basics

Assume a batch process that has J process sensors to measure process variants, a batch running consists of K sample points, and after a batch running M product-quality variables. After $\boldsymbol{I}$ batches, three-way tensor data consisting of the measured variables $\underline{\boldsymbol{X}}(I \times J \times K)$ and a quality matrix $\boldsymbol{Y}$ $(I \times M)$ are obtained. Obtaining quality variables are generally costly or time consuming. Therefore, the establishment of an estimation model to predict the batch-end product quality is important.

The MPLS was proposed to deal with the three-way data, wherein three-way data are unfolded into two-way data, and then PLS modeling is performed on the two-way data [16]. Several data unfolding methods have been developed. We used the

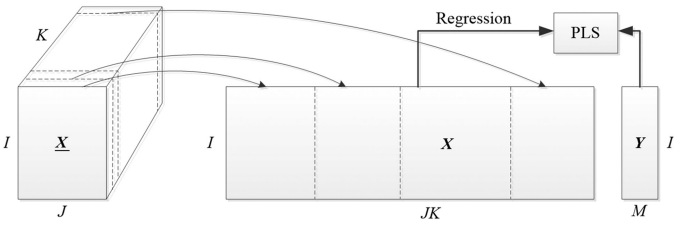

Fig. 1. MPLS with batch-wise unfolding.

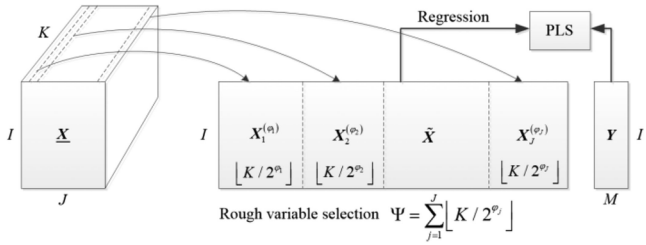

Fig. 2. MRPLS for batch-end quality prediction.

batch-wise unfolding method in this paper. In batch-wise unfolding, different time periods are placed side by side, as shown in Fig. 1. The batch measurements are placed in a row and the columns correspond to the measurements at different time instants. Then, the three-way data $\underline{\boldsymbol{X}}(I \times J \times K)$ are unfolded into the two-way predictor data matrix $\boldsymbol{X}(I \times JK)$. Generally, I is much smaller than JK. Then, PLS is used to establish relationship between the $\boldsymbol{X}$ and the quality matrix $\boldsymbol{Y}$.

The PLS model can be established in terms of $\boldsymbol{X}$ and $\boldsymbol{Y}$ as

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E}_X \\ \boldsymbol{Y} &= \boldsymbol{T}\boldsymbol{B}\boldsymbol{Q}^T + \boldsymbol{E}_Y \end{aligned} \tag{1}$$

where $\boldsymbol{T}(I \times p)$ contains latent variables relating $\boldsymbol{X}$ and $\boldsymbol{Y}$; $\boldsymbol{P}(JK \times p)$ denotes the $\boldsymbol{X}$-loading matrix; $\boldsymbol{Q}(M \times p)$ denotes the $\boldsymbol{Y}$-loading matrix; $\boldsymbol{B}(p \times p)$ denotes the regression matrix; $\boldsymbol{E}_X(I \times JK)$ and $\boldsymbol{E}_Y(I \times M)$ denote residual; and p denotes the number of latent variables, which is usually determined through the cross-validation method [24].

After the data unfolding, the dimension of the predictor matrix is JK, which can be remarkably large. However, obtaining the quality variables is generally difficult, and the number of samples in $\boldsymbol{Y}$ is generally small. A considerable rank-deficient problem occurs on the one hand. On the other hand, not all predictor variables in $\boldsymbol{X}$ are beneficial for predicting quality variables, and a large number of irrelevant variables generally exist. The existence of irrelevant variables may affect the prediction performance. We present the effect of irrelevant variables on the prediction performance using the univariate Y as an example.

Let the predictor data $\boldsymbol{X}$ be partitioned into $(\boldsymbol{X}_1 \ \boldsymbol{X}_2)$, where $\boldsymbol{X}_1$ denotes the m_1 relevant variables and $\boldsymbol{X}_2$ contains the $m - m_1$ irrelevant variables. With normalization, each variable in $\boldsymbol{X}_2$ follows the Gaussian distribution as $N(0, I)$. Helland derived the closed form solution for univariate PLSR as [20], [25]

$$\hat{\beta}^{\mathrm{PLS}} = \hat{\boldsymbol{R}}\left(\hat{\boldsymbol{R}}^T \boldsymbol{S}_{XX} \hat{\boldsymbol{R}}\right)^{-1} \hat{\boldsymbol{R}} \boldsymbol{S}_{XY} \tag{2}$$

where $\hat{\boldsymbol{R}} = (\boldsymbol{S}_{XY}, \ldots, \boldsymbol{S}_{XX}^{p-1}\boldsymbol{S}_{XY})$ and $\boldsymbol{S}_{XY}$ denotes the covariance matrix. Consider $p = 1$, then $\hat{\boldsymbol{R}} = \boldsymbol{S}_{XY}$. Thus, (2) becomes

$$\hat{\beta}^{\mathrm{PLS}} = \boldsymbol{S}_{XY}\left(\boldsymbol{S}_{XY}^T \boldsymbol{S}_{XX} \boldsymbol{S}_{XY}\right)^{-1} \boldsymbol{S}_{XY} \boldsymbol{S}_{XY}. \tag{3}$$

Assume m grows at the rate of $O(kn)$ and the constant k is substantially large to have

$$\max\left(\sigma_{X_1Y}^T \sigma_{X_1Y}, \sigma_{X_1Y}^T \Sigma_{X_1X_1} \sigma_{X_1Y}\right) \ll k\sigma_1^2\sigma_2^2 \tag{4}$$

where σ is the covariance. Then, for a fixed m_1, the PLS estimator is approximated by [20], [25]

$$\begin{aligned} &\hat{\beta}^{\mathrm{PLS}} \\ &= \frac{\boldsymbol{S}_{X_1Y}^T \boldsymbol{S}_{X_1Y} + \boldsymbol{S}_{X_2Y}^T \boldsymbol{S}_{X_2Y}}{\boldsymbol{S}_{X_1Y}^T \boldsymbol{S}_{X_1X_1} \boldsymbol{S}_{X_1Y} + 2\boldsymbol{S}_{X_1Y}^T \boldsymbol{S}_{X_1X_2} \boldsymbol{S}_{X_2Y} + \boldsymbol{S}_{X_2Y}^T \boldsymbol{S}_{X_2X_2} \boldsymbol{S}_{X_2Y}} \boldsymbol{S}_{XY} \\ &\approx \frac{\boldsymbol{S}_{X_2Y}^T \boldsymbol{S}_{X_2Y}}{\boldsymbol{S}_{X_2Y}^T \boldsymbol{S}_{X_2X_2} \boldsymbol{S}_{X_2Y}} \boldsymbol{S}_{XY} \\ &= \mathrm{O}\left(k^{-1}\right) \boldsymbol{S}_{XY}. \end{aligned} \tag{5}$$

The large number of irrelevant variables distorts the directions of the covariance $\boldsymbol{S}_{XY}$, therefore degrades the prediction performance. The coefficients of irrelevant variables should be zero in theory; however, the interpretation becomes difficult because of the noise influence or the estimation bias from the limited process data. The following reasons motivate us to establish a reasonable sparse prediction model.

1) First, the number of predictor variables is generally much larger than the number of measurements.
2) Second, the existence of irrelevant variables can distort the loadings in the regression coefficients.
3) Third, the influence of the noise makes the coefficients of the irrelevant variables not zero that leads to the difficulty in model interpretation.

Given the reasons above, a prediction method that can simultaneously perform regression and select relevant variables is required. The description of the two state-of-the-art sparse modeling methods, MRPLS and SPLS follows.

B. Multiresolution PLS

In MRPLS modeling, the predictor variables are first set with the same high resolution. A resolution index φ is introduced to adjust the variable resolution. When $\varphi = 0$, the resolution is highest. If $\varphi > 0$, then the variable signal is operated by averaging some observations in a nonoverlapping window. For example, a signal with K observations, the lower resolution signal at the φ-level is composed of $\lfloor K/2^{\varphi} \rfloor$ observations corresponding to the average value in a nonoverlapping window. An illustration of the MRPLS method is provided in Fig. 2.

The three-way data are unfolded by taking slices in the variable dimension. A slice is represented by $\boldsymbol{X}_j^{(\varphi_j)}(I \times \lfloor K/2^{\varphi_j} \rfloor)$ with j being the variable index and φ_j being the respective resolution index. Then, the unfolded multiresolution data matrix is formulated by stacking $\boldsymbol{X}_j^{(\varphi_j)}$ as $\tilde{\mathbf{X}}^{(\varphi)} = [\mathbf{X}_1^{(\varphi_1)} \mathbf{X}_2^{(\varphi_2)} \ldots \mathbf{X}_J^{(\varphi_J)}]$ $(I \times \Psi)$, where $\Psi = \sum_{j=1}^{J} \lfloor K/2^{\varphi_j} \rfloor$. A forward

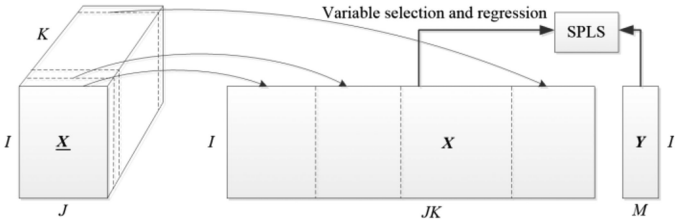

Fig. 3. SPLS for batch-end quality prediction.

stepwise algorithm was developed to select the best resolution for each variable [21], [23].

C. Sparse PLS

The PLS finds the projecting matrix $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_q)$ from successive optimization problems to obtain $\boldsymbol{T}$ from $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}$. $\boldsymbol{w}_i$ for univariate $\boldsymbol{Y}$ is obtained by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{w}_i &= \underset{\boldsymbol{w}}{\operatorname{argmax}} \left\{ \operatorname{corr}^2(\boldsymbol{Y}, \boldsymbol{X}\boldsymbol{w}) \operatorname{var}(\boldsymbol{X}\boldsymbol{w}) \right\} \\ &\text{s.t.}\, \boldsymbol{w}^T\boldsymbol{w} = 1,\ \boldsymbol{w}^T\boldsymbol{\Sigma}_X\boldsymbol{w}_j = 0 \end{aligned} \tag{6}$$

where $j = 1, \ldots, i-1$. To impose the coefficients of irrelevant variables equal to zero, an approximation direction vector $\boldsymbol{c}$ of $\boldsymbol{w}$ is introduced, and L_1 penalty is introduced in the direction vector $\boldsymbol{c}$. Then the optimization problem becomes [20]

$$\begin{aligned} \min_{\boldsymbol{w}\boldsymbol{c}} \Big\{ &-\kappa \boldsymbol{w}^T\boldsymbol{M}\boldsymbol{w} + (1-\kappa)(\boldsymbol{c}-\boldsymbol{w})^T\boldsymbol{M}(\boldsymbol{c}-\boldsymbol{w}) \\ &+ \lambda_1 |\boldsymbol{c}|_1 + \lambda_2 |\boldsymbol{c}|_2^2 \Big\} \\ &\text{s.t.}\, \boldsymbol{w}^T\boldsymbol{w} = 1 \end{aligned} \tag{7}$$

where λ_1 and λ_2 are sparsity parameters, κ is a parameter that controls the effect of the concave portion. The optimization problem can be solved by iterating alternately between solving for $\boldsymbol{w}$ with a fixed $\boldsymbol{c}$, and solving for $\boldsymbol{c}$ with a fix $\boldsymbol{w}$. We did not detail the solution, and more details can be found in [20]. An illustration of the multiway SPLS modeling is presented in Fig. 3.

In the MRPLS method, variables are selected by averaging the variables with a certain time interval. A regular variable selection area is constructed in this method. This selection method is regarded as a rough variable because the important variables for quality prediction may be time varying. The SPLS performs simultaneous variable selection and quality prediction, and is a subtle variable selection method. Moreover, the SPLS selects related variables directly from a large number of process variables. Although is the SPLS demonstrated to be more effective in dealing with the large m small n problem than that of the conventional PLS method, the prediction performance may be degraded with a considerable number of predictor variables. Therefore, a comprehensive variable selection method that performs rough and subtle variable selection is developed to increase the quality-prediction accuracy and model interpretability.

Variable number	1			2			…	J		
Chromosome (Genes)	1	0	1	0	1	1	…	0	0	0
Resolution index φ	$\varphi_1 = 5$			$\varphi_2 = 3$			⋯	$\varphi_J = 0$		
Resolution	$\lfloor K/2^5 \rfloor$			$\lfloor K/2^3 \rfloor$			⋯	$\lfloor K/2^0 \rfloor$		

Fig. 4. Chromosome design for variable resolution optimization.

III. OSPLS-Based Batch-End Quality Modeling and Monitoring

A. OSPLS Model

The OSPLS performs comprehensive rough and subtle variable selection through a stochastic optimization method. The OSPLS makes a rough variable selection by optimizing the variable resolution during a batch running. Then, the SPLS model is established to make simultaneous subtle variable selection and quality-prediction performance improvement based on the selected variables with optimal resolutions.

Genetic Algorithm (GA) is an efficient tool in solving nonconvex optimizing problems when the decision variables are discrete [26]. Although the convergence of GA is still under investigation, GA has been widely used to find a better or a more satisfactory solution in engineering application. Here, we used the GA as the stochastic optimization algorithm for optimizing the variable resolution. The chromosome in GA is utilized to encode the OSPLS variable resolution (see Fig. 4). Assume that J variables are measured, and each variable is encoded in D bits of genes. The length of a chromosome is $D \times J$. Then, the resolution levels of each variable can be selected from 0 to $(2^D - 1)$. For example, each variable resolution is encoded in three bits of genes $(D = 3)$, and the optimization process searches for the optimal resolution from $\lfloor K/2^0 \rfloor$ to $\lfloor K/2^7 \rfloor$. Three bits of genes can provide sufficiently high-resolution level, i.e., 2^7 for selection. The bits of genes D can be adjusted to meet the practical requirement. A larger number of bits provide more resolution choices for the variables, but it will also increase the optimization computational burden. Generally, we use two or three bits of genes for each variable resolution selection.

Once a chromosome is designed, the next step is to construct the fitness function. The object of the resolution optimization is to minimize the prediction error of the validation data, which is evaluated by the root mean squared error (RMSE). Then, the fitness function is expressed as follows:

$$\varphi = \arg\min_{\varphi} \mathrm{RMSE} = \sqrt{\sum_{n=1}^{N_V} (y_n - \hat{y}_n)^2 / (N_V - 1)} \tag{8}$$

where N_V denotes the number of validation data; y_n is the output value of the nth sample; and $\hat{y}_n$ is the predicted value of the nth sample. The resolution optimization works as follows.

First, an initial population of chromosomes is generated. The initial population of all chromosomes is set to be with zero elements. The resolutions of all variables are at the same high level. A temporary SPLS model is established, and the fitness value is calculated. Second, the chromosomes evolve to the next generation with gene operations, including selection, mutation, and crossover. The fitness value corresponding to individual

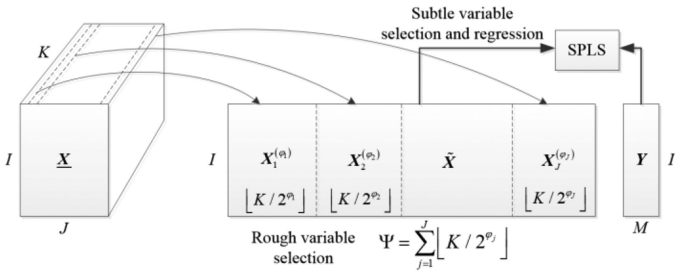

Fig. 5. OSPLS with batch-wise data unfolding.

chromosomes is calculated, and the chromosome with the lowest fitness value is noted. The evolutions are continued until the optimal solution is reached or the predefined maximal number of generations is reached.

Although there are four tuning parameters $(\kappa, \lambda_1, \lambda_2, p)$ in the SPLS modeling, only the sparsity parameter λ_1 and the number of retained variables p are considered to be the key tuning parameters [20]. The two key parameters are determined according to the false discovery rate and cross-validation method [20]. The false discovery rate approach is used to determine the thresholding parameter λ_1 first. Then, p is determined through the cross-validation method. The algorithm for the parameter determination is computational efficiency [20]. An illustration of the OSPLS model for batch-end quality prediction is presented in Fig. 5. As illustrated in Fig. 5, the resolution optimization makes a rough variable selection. The SPLS on the reorganized predictor data (with optimal resolution) then makes a subtle variable selection and regression simultaneously.

B. OSPLS-Based Quality Modeling and Monitoring

The proposed OSPLS-based quality prediction and monitoring scheme is detailed in this section.

1) Offline Modeling:

Step 1. *Variable resolution optimization:* Following the details in the Section III-A, the GA-based variable resolution optimizing is performed, which selects the optimal resolution in each measured variable. After the resolution optimization, the dimension of the predictor matrix is remarkably reduced. All batches are assumed to be synchronized because various synchronization methods can be applied in the OSPLS model [27].

Step 2. *OSPLS model establishment:* Once the optimal variable resolution is determined, the OSPLS model is established on the basis of the new organized predictor matrix. The regression matrix β^{OSPLS} can be obtained, in which the coefficients of irrelevant variables are set to zero.

Step 3. *Quality monitoring statistic construction:* Once the quality variable is estimated accurately, a quality monitoring method should be developed, which is important for finding and eliminating a fault timely. We assume that the regression matrix β^{OSPLS} has been obtained. Then, the relationship between a predictor sample x and the corresponding quality sample $\boldsymbol{y}$ can be established as

$$\boldsymbol{y} = \boldsymbol{x}^T \beta^{\mathrm{OSPLS}}. \tag{9}$$

The quality in normal status (not the available measurements) needs to be justified because the quality variable is generally obtained with a considerable delay. The absolute value $|\beta_{ij}|$ in the regression coefficient β^{OSPLS} reflects the importance of the ith predictor variable to the jth predicted-quality variable. The larger the $|\beta_{ij}|$ is, the more important the ith predictor variable is. Therefore, the most important predictor variables for examining the status of the quality variable are reasonably selected. The quality-relevant important variables are selected in accordance with the criteria $|\beta_{ij}| \geq cl$, where cl is a threshold parameter that controls the number of selected variables. Generally, the number of selected variables is less than the number of the training samples. Assume that M most important variables are selected, and the selected variables make up a new sample, $\boldsymbol{x}_s = [\, x_s^1 \; x_s^2 \; \ldots \; x_s^M \,]^T$. T^2 test on the selected variables is expressed as

$$T^2 = \boldsymbol{x}_s^T \boldsymbol{\Sigma}_X^{-1} \boldsymbol{x}_s = \boldsymbol{x}_s^T \left(\frac{\boldsymbol{X}_s^T \boldsymbol{X}_s}{N-1} \right)^{-1} \boldsymbol{x}_s \tag{10}$$

where $\boldsymbol{X}_s$ denotes the training data matrix with selected relevant variables.

Step 4. *Monitoring statistic threshold determination:* Based on the assumption that the selected predictor variables are multivariate Gaussian distributed, and T^2 statistic follows a χ^2 distribution. Then, the threshold of T^2 statistic is determined as $\chi_\alpha^2(r)$, where $r = \mathrm{rank}(\boldsymbol{\Sigma}_X)$ is the degree of freedom and α is the level of significance. The threshold T_{th}^2 is generally determined according to the F distribution because the covariance is estimated from a limited number of samples. T_{th}^2 is defined as

$$T_{\mathrm{th}}^2 = \frac{r(N^2-1)}{N(N-r)} F_\alpha \left(r, N-r\right). \tag{11}$$

If ever the Gaussian distribution assumption is not met, then kernel density estimation is suitable for determining the threshold [28].

2) Online Quality Prediction and Monitoring:

Step 1. *Data unfolding and preprocessing:* Once an online batch data $\underline{\boldsymbol{X}}^{\mathrm{new}}$ is available, the dataset is first unfolded, and the variable resolution is set as the offline determined. The reorganized predictor variables $\boldsymbol{x}^{\mathrm{new}}$ are obtained and normalized using the stored mean and the variance of training data.

Step 2. *Batch-end quality prediction:* The quality variable $\hat{\boldsymbol{y}}^{\mathrm{new}}$ is estimated according to the established OSPLS model as

$$\hat{\boldsymbol{y}}^{\mathrm{new}} = (\boldsymbol{x}^{\mathrm{new}})^T \beta^{\mathrm{OSPLS}}. \tag{12}$$

Step 3. *Quality status identification:* T^2 *statistic* for the predicted quality variables is established. Then, the status of the quality is identified in accordance with the

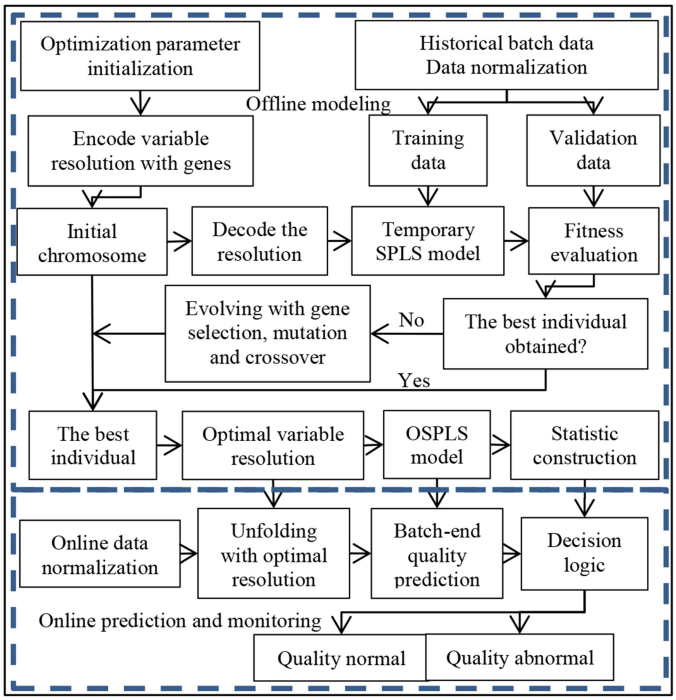

Fig. 6. OSPLS-based batch-end quality modeling and monitoring scheme.

following decision logic:

$$\begin{cases} T^2 > T_{\text{th}}^2 \Rightarrow \text{quality faulty} \\ T^2 \le T_{\text{th}}^2 \Rightarrow \text{quality fault} - \text{free} \end{cases}. \tag{13}$$

This paper focuses only on the quality-related-process faults and selects the most important variables for monitoring. The monitoring statistics is also established for the irrelevant variables, and the status of the entire process is examined. The proposed OSPLS-based batch-end quality modeling and monitoring scheme is illustrated in Fig. 6.

C. Characteristics and Remarks

Characteristic 1. The Computational Complexity Analysis: The computational complexity of OSPLS modeling can be analyzed on two aspects, namely, the design complexity and process complexity. The design complexity involves many parameters that can affect the computational burden. The major computational burden comes from the GA optimization, which involves the resolution optimization and temporary SPLS modeling. First, when the number of the measured variables J is generally not large, then only a few bits of genes are needed to encode the resolution. For example, three bits of genes $(D = 3)$ can set the minimal resolution to $\lfloor K/2^7 \rfloor$, which can generally satisfy the practical requirement. Second, the number of cross-validation randomizations in determining the parameters in temporary SPLS modeling can affect the computational burden. Third, the GA optimization can generally determine the optimal resolution in an appropriate number of generations because the optimization problem is not complex. The process complexity affects the computational burden in terms of the number of measured variables and the number of observations. The computational burden can be addressed using a general computer and meets the practical application requirement.

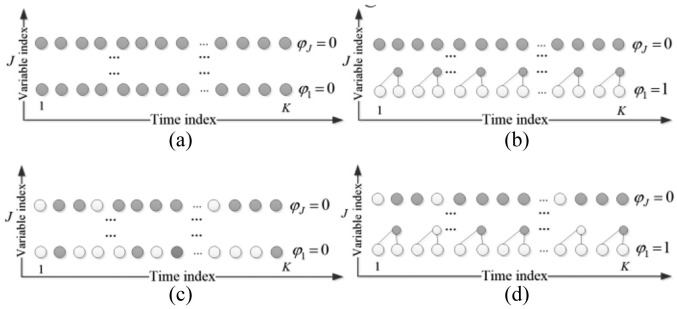

Fig. 7. Graphical representation of variable selection of different modeling methods. (a) PLS. (b) MRPLS. (c) SPLS. (d) OSPLS.

Characteristic 2. Relationship With MRPLS and SPLS: The OSPLS is highly related to the state-of-the-art MRPLS and MSPLS methods. The OSPLS makes a comprehensive rough variable selection and a subtle variable selection, which combines the advantages of the MRPLS and MSPLS. The highest variable resolution is used, i.e., $\varphi = 0$ for all variables, the OSPLS becomes conventional multiway SPLS modeling. This property makes sure that the SPLS is the worst case of the OSPLS, and the performance of the OSPLS will be better than the SPLS when the highest resolution is not optimal. Once the SPLS is replaced by the PLS, and the subtle variable selection is ignored, the OSPLS is degraded to the MRPLS with the difference exhibited on the optimization. Since the SPLS modeling approach makes simultaneous variable selection and quality prediction, the model robustness and interpretability are generally higher compared with that of the PLS model, especially when the large m small n problem exists. A graphical representation of the variable selection of different modeling methods is presented in Fig. 7. In Fig. 7, the gray circles denote the selected variables for modeling.

Characteristic 3. Comparison With GA-PLS: GA was utilized for variable selection in several multivariate analysis methods [29], [30]. In GA–PLS, each predictor variable is encoded by one bit of a gene. The gene value indicates that whether corresponding predictor variable should be selected or not. However, conventional GA–PLS is inappropriate for modeling the batch-end quality because the number of predictor variables is generally considerably large. For example, in the batch process mentioned above, there are 2000 predictor variables, and thus 2000 bits of genes are needed to encode the variables. This task is computationally expensive, and the optimal solution is generally difficult to obtain. In OSPLS, it is the variable resolution that is encoded by the gene. Then, the number of genes is $D \times J$. Moreover, the computational burden is substantially reduced. The optimal resolution can generally be obtained within an appropriate number of generations.

Remark. On the multiple phases and transient processes: A batch process generally consists of multiple operation phases and the transient processes exist between different phases. The OSPLS uses the batch-wise data unfolding and normalizes the

TABLE I
MAIN PARAMETERS IN THE GA OPTIMIZATION

Population	Size: 30; Creation function: Uniform
Fitness scaling	Function: Rank
Selection	Function: Stochastic uniform
Mutation	Function: Uniform
Crossover	Function: Scattered
Stopping criteria	Stall generations: 30 (or 20) Fitness limit: Minus infinity

data along the batch direction. Therefore, once the phases are synchronized, the process data can generally be well normalized, and the OSPLS model can be well established. Numerous process synchronization methods have been proposed to deal with data preprocessing problems [31], [32]. The OSPLS can cooperate with these existing methods to deal with some specific situations with data preprocessing difficulties. In addition, various phase partition and identification methods have been developed to distinguish the operation phases [31], [32]. Once process operation phases can be well identified, the chromosome in GA can be designed to represent resolutions in different phases. This paper focuses on the OSPLS basics, and various extensions and improvements can be performed to address more specific practice requirements.

IV. EXPERIMENTAL STUDIES AND COMPARISONS

Two experimental studies are carried out, namely, a simulated FBPF process and an IIM process. The RMSE and the coefficient of determination R^2 are two widely used indices for modeling performance evaluation. These two indices are defined as follows:

$$\text{RMSE} = \sqrt{\sum_{n=1}^{N_T} (y_n - \hat{y}_n)^2 / (N_T - 1)} \tag{14}$$

$$R^2 = 1 - \sum_{n=1}^{N_T} (y_n - \hat{y}_n)^2 \Big/ \sum_{n=1}^{N_T} (y_n - \bar{y})^2 \tag{15}$$

where N_T denotes the number of all testing samples, and $\bar{y}$ is the mean of the output values.

The computations are carried out in MATLAB R2017b. The hardware is Intel (R) Core (TM) I7-5500U @ 2.4 GHz 8.00G RAM. The MATLAB GA Toolbox is used here because the current optimization problem is not complex. The specified parameters are presented in Table I.

A. Experimental Study 1: The FBPF Process

The FBPF process is a widely used benchmark for testing batch process modeling performance [8], [21]. A simulator of the FBPF is available at http://simulator.iit.edu/web/pensim/index.html. A flow diagram of the process is presented in Fig. 8. The batch-end penicillin concentration (g/L) is treated as the quality variable. According to [8], [10], and [11], the following eleven variables are considered as predictor variables, which include aeration rate (L/h), agitator power

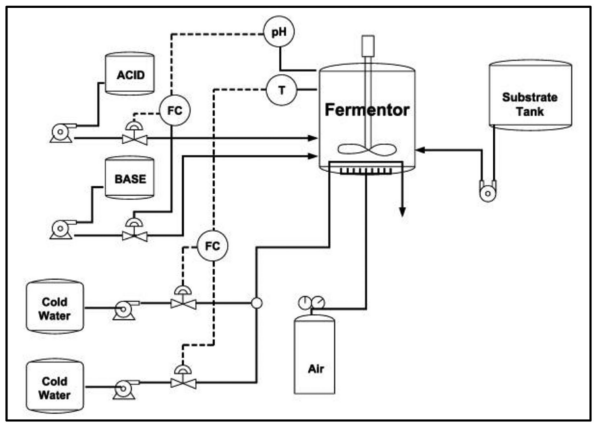

Fig. 8. Flowchart of the FBPF process.

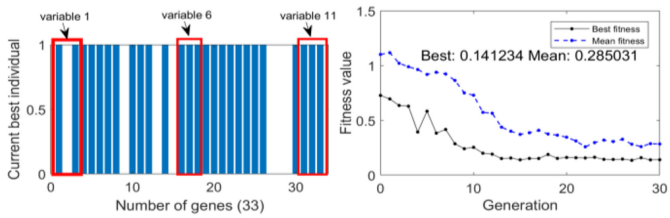

Fig. 9. Resolution optimization results for the FBPF process. (a) The best individual. (b) Fitness values during the optimization.

(W), substrate feed rate (L/h), substrate concentration (g/L), dissolved oxygen concentration (g/L), biomass concentration (g/L), culture volume (L), carbon dioxide concentration (g/L), pH, bioreactor temperature (K), and cooling water flow rate (L/h). The simulation conditions are set the same as in [8]. The starting point of each batch is randomly generated within the in-control interval. The sampling interval is 1 h, and the entire duration of a batch running is 200 h. The normal process data of 150 batches are collected for establishing the OSPLS model.

Based on the normal operating data, the variable resolutions are optimized through the GA. The bits of genes for each variable is set as $D = 3$, and the bits of genes in a chromosome is 27. The entire optimization process is completed within 130 min, which satisfies the practical application requirement. The optimization results are presented in Fig. 9. We can see that the highest resolution does not provide the best prediction results, and the resolution optimization does have influence on the modeling performance. We obtain the optimal variable resolution by decoding the chromosome.

Based on the predictor data matrix with optimized resolution, a SPLS model is established. To evaluate the quality-prediction performance, 50 other normal operating batches are simulated. The prediction results using multiway OSPLS, MRPLS, SPLS [20], PLS, PCR, and the Lasso regression [33] are presented in Fig. 10. From the results, we can identify that the OSPLS provides the lowest RMSE and the largest R^2. The OSPLS outperforms the PCR and Lasso because the SPLS focuses more

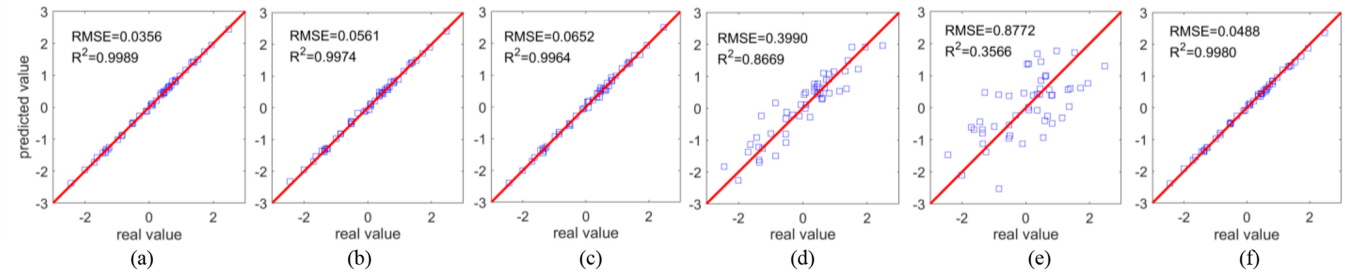

Fig. 10. Comparison results on the FBPF. (a) OSPLS. (b) MRPLS. (c) SPLS. (d) PLS. (e) PCR. (f) Lasso.

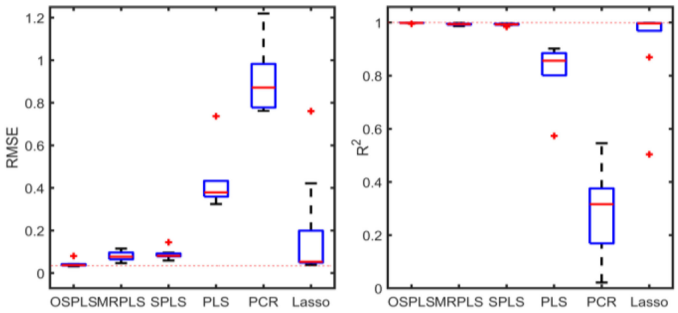

Fig. 11. Random test prediction performance on the FBPF process.

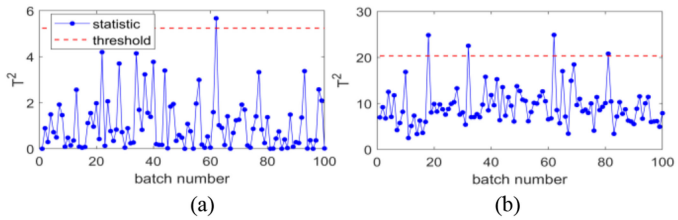

Fig. 12. Quality monitoring results of the FBPF fault 1. (a) Unavailable real quality monitoring results. (b) Predictive quality monitoring results.

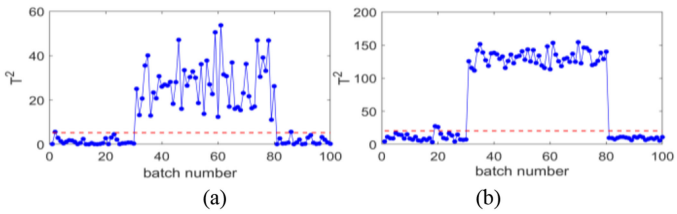

Fig. 13. Quality monitoring results of the FBPF fault 2. (a) Unavailable real quality monitoring results. (b) Predictive quality monitoring results.

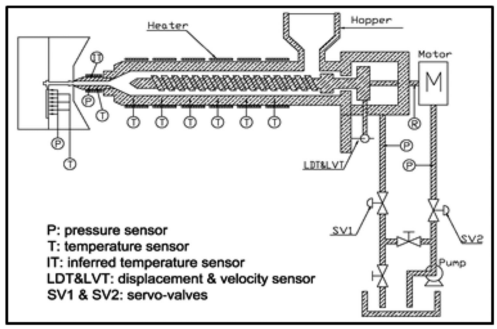

Fig. 14. Schematic of the IIM machine.

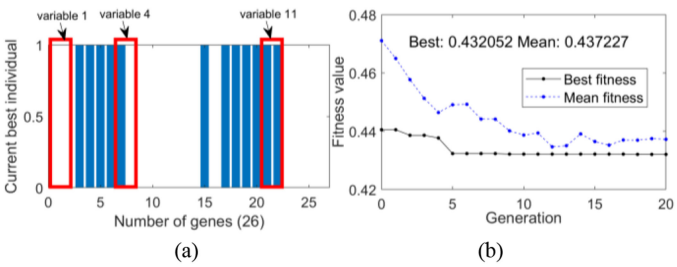

Fig. 15. Resolution optimization results for the IIM process. (a) The best individual. (b) Fitness values during the optimization.

on the prediction performance than the PCR and Lasso. The OSPLS outperforms PLS because the SPLS is more efficient than the PLS in the existence of large m small n problem. The OSPLS outperforms the SPLS because SPLS is a special case of the OSPLS and the variable resolution is optimized to increase the modeling performance in OSPLS. We repeat the entire experiment 20 times, and the random test results are presented in Fig. 11, from which we can see that the OSPLS generally provides the best prediction performance.

To evaluate the quality monitoring performance, two other sets of batches with faults occurring between batches 31–80 are generated. Fault 1 introduces a change into the agitator power from the 31st time instance, and the fault does not influence the quality variable. Fault 2 introduces a change into the substrate feed rate, and the fault has considerable effect on the quality variable. Then, quality monitoring results using the OSPLS are presented in Figs. 12 and 13. Fig. 12(a) plots the statistic of the (practical) unavailable real quality variable, and Fig. 12(b) plots the statistic of the OSPLS model for fault 1. Although there is a fault, the quality is not influenced, and the quality status is successfully identified. Fig. 13(a) presents the statistic of real quality variable, and Fig. 13(b) presents the statistic from the OSPLS model. The out-of-control batches are successfully detected.

B. Experimental Study 2: The IIM Process

The IIM process is suitable for testing the performance of a process modeling scheme [1]. An illustration of an IIM machine is presented in Fig. 14. The product weight is regarded as the quality variable. There are 13 measured variables considered, which are mold position, ejector pin position, injection position, system pressure, nozzle pressure, injection velocity,

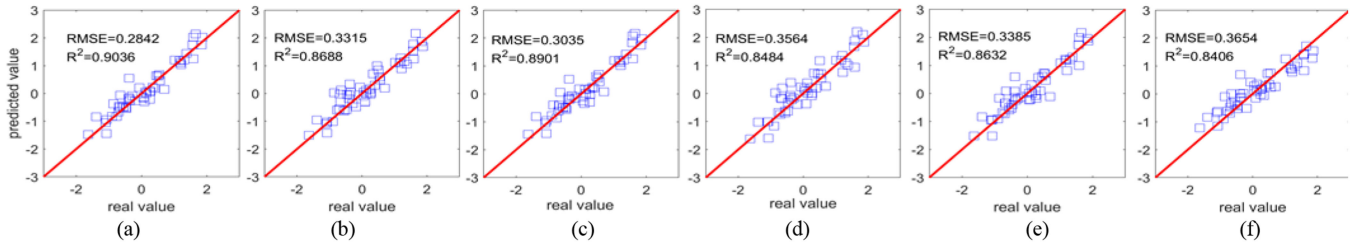

Fig. 16. Comparison results on the IIM. (a) OSPLS. (b) MRPLS. (c) SPLS. (d) PLS. (e) PCR. (f) Lasso.

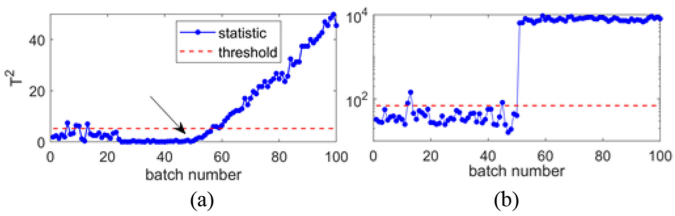

Fig. 17. Quality monitoring results of the IIM. (a) Unavailable real quality monitoring results. (b) Predictive quality monitoring results.

backpressure, mold velocity, ejector pin velocity, nozzle temperature, and zone 1–3 temperatures. Normal operating process data of 150 batches are collected for establishing the prediction model. Based on the training data, the resolution optimization is performed. The bits of genes for encoding a variable resolution is set as $D = 2$. The entire optimization process takes about 120 min and the results are presented in Fig. 15. The resolution selection has influence on the prediction performance, and the prediction error has been reduced after the resolution optimization. We obtain the optimal resolution for the process by decoding the chromosome. Using the predictor data with optimal resolution, we established the SPLS model.

To test the prediction performance, 50 other batches under normal conditions are collected. The prediction results using multiway OSPLS, MRPLS, SPLS, PLS, PCR, and the Lasso regression are presented in Fig. 16. We can see that the proposed OSPLS method provides the lowest RMSE and the largest R^2 among the considered state-of-the-art methods. One hundred other batches of testing data with faults are introduced from the 51st batch. The quality monitoring results are presented in Fig. 17. Fig. 17(a) plots the statistic of the (practical) unavailable real quality variable, and Fig. 17(b) plots the statistic of the OSPLS model. We can see that the quality variable starts to change on the 51st sample, and the fault is successfully detected by the OSPLS monitoring statistic. The efficiency of the OSPLS-based modeling and monitoring scheme is verified.

V. Conclusion

In this paper, we first analyzed the importance of the relevant-variable selection for batch-end quality prediction, and proposed an OSPLS-based batch-end quality modeling and monitoring scheme. The OSPLS performed dimension reduction and relevant-variable selection from resolution optimization and SPLS modeling. First, the resolution optimization performed a rough variable selection using the stochastic optimization algorithm, which reduced the predictor matrix dimension by averaging variables over a certain period. Second, the SPLS modeling performed a subtle variable selection by introducing L_1 penalty term, which achieved simultaneous quality prediction and variable selection. The proposed OSPLS combined the advantages of resolution optimization and sparse modeling, which increased the model prediction accuracy and model interpretability remarkably. After the quality-relevant variables were selected through the OSPLS model, a monitoring statistic was established to identify the quality status. The OSPLS-based modeling and monitoring was applied on a simulated FBPF benchmark and an IIM process. The results when compared with the state-of-the-art methods confirmed the effectiveness of the OSPLS-based modeling and monitoring scheme.

Notably, this paper established a data-driven model by using only process data. Thus, the model reliability relied largely on the reliability of process data. Although exact process mechanism was difficult to obtain, some process knowledge was generally available. Therefore, process knowledge into the data-driven model may be incorporated to increase the model reliability. Moreover, the current data-driven quality modeling and monitoring does not consider the influences from the disturbances. A method to increase the robustness of the data-driven model and remove the effects from the disturbances is desirable.

References

[1] Q. Jiang, F. Gao, X. Yan, and H. Yi, "Multiobjective two-dimensional CCA-Based monitoring for successive batch processes with industrial injection molding application," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3825–3834, May 2019.

[2] C. Ündey, S. Ertunç, and A. Çınar, "Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis," *Ind. Eng. Chem. Res.*, vol. 42, pp. 4645–4658, 2003.

[3] X. Liu, Z. Gao, and M. Chen, "Takagi–Sugeno fuzzy model based fault estimation and signal compensation with application to wind turbines," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5678–5689, Jul. 2017.

[4] Z. Gao, X. Liu, and M. Z. Q. Chen, "Unknown input observer based robust fault estimation for systems corrupted by partially-decoupled disturbances," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2537–2547, Apr. 2016.

[5] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2226–2238, Nov. 2013.

[6] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.

[7] C. Tong and X. Shi, "Decentralized monitoring of dynamic processes based on dynamic feature selection and informative fault pattern dissimilarity," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3804–3814, Jun. 2016.

[8] G. Birol, C. Ündey, and A. Çinar, "A modular simulation package for fed-batch fermentation: Penicillin production," *Comput. Chem. Eng.*, vol. 26, pp. 1553–1565, 2002.
[9] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Comput. Chem. Eng.*, vol. 33, pp. 795–814, 2009.
[10] C. Ündey, E. Tatara, and A. Çınar, "Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations," *J. Biotechnology*, vol. 108, pp. 61–77, 2004.
[11] C. Zhao, F. Wang, Z. Mao, N. Lu, and M. Jia, "Improved batch process monitoring and quality prediction based on multiphase statistical analysis," *Ind. Eng. Chem. Res.*, vol. 47, pp. 835–849, 2008.
[12] C. Shang, B. Huang, F. Yang, and D. Huang, "Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling," *AIChE J.*, vol. 61, pp. 4126–4139, 2015.
[13] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.
[14] S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems*. London, U.K.: Springer, 2014.
[15] H. Wold, "Estimation of principal components and related models by iterative least squares," in *Multivariate Analysis*. New York, NY, USA: Academic, 1966, pp. 391–420.
[16] P. Nomikos and J. F. Macgregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics Intell. Lab. Syst.*, vol. 30, pp. 97–108, 1995.
[17] T. Mehmood, K. H. Liland, L. Snipen, and S. Sæbø, "A review of variable selection methods in partial least squares regression," *Chemometrics Intell. Lab. Syst.*, vol. 118, pp. 62–69, 2012.
[18] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, pp. 2507–2517, 2007.
[19] K. Hasegawa, Y. Miyashita, and K. Funatsu, "GA strategy for variable selection in QSAR studies: GA-based PLS analysis of calcium channel antagonists," *J. Chem. Inf. Comput. Sci.*, vol. 37, pp. 306–310, 1997.
[20] C. Hyonho and K. Sündüz, "Sparse partial least squares regression for simultaneous dimension reduction and variable selection," *J. Roy. Statistical Soc.*, vol. 72, pp. 3–25, 2010.
[21] T. J. Rato and M. S. Reis, "Optimal selection of time resolution for batch data analysis. Part I: Predictive modeling," *AIChE J.*, vol. 64, pp. 3923–3933, 2018.
[22] P. Facco, F. Doplicher, F. Bezzo, and M. Barolo, "Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process," *J. Process Control*, vol. 19, pp. 520–529, 2009.
[23] G. Gins, J. F. Van Impe, and M. S. Reis, "Finding the optimal time resolution for batch-end quality prediction: MRQP–A framework for multi-resolution quality prediction," *Chemometrics Intell. Lab. Syst.*, vol. 172, pp. 150–158, 2018.
[24] L. Ståhle and S. Wold, "Partial least squares analysis with cross-validation for the two-class problem: A Monte Carlo study," *J. Chemometrics*, vol. 1, pp. 185–196, 1987.
[25] I. S. Helland, "Partial least squares regression and statistical models," *Scand. J. Statist.*, vol. 17, pp. 97–114, 1990.
[26] D. E. Golberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[27] J. M. González-Martínez, O. E. D. Noord, and A. Ferrer, "Multisynchro: A novel approach for batch synchronization in scenarios of multiple asynchronisms," *J. Chemometrics*, vol. 28, pp. 462–475, 2014.
[28] J. M. Lee, C. K. Yoo, and I. B. Lee, "Statistical process monitoring with independent component analysis," *J. Process Control*, vol. 14, pp. 467–485, 2004.
[29] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 377–386, Jan. 2016.
[30] Q. Jiang, S. Ding, Y. Wang, and X. Yan, "Data-driven distributed local fault detection for large-scale processes based on GA-regularized canonical correlation analysis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8148–8157, Oct. 2017.
[31] R. Rendall, L. H. Chiang, and M. S. Reis, "Data-driven methods for batch data analysis–A critical overview and mapping on the complexity scale," *Comput. Chem. Eng.*, vol. 124, pp. 1–13, 2019.
[32] C. Zhao, S. Mo, F. Gao, N. Lu, and Y. Yao, "Statistical analysis and online monitoring for handling multiphase batch processes with varying durations," *J. Process Control*, vol. 21, pp. 817–829, 2011.
[33] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *J. Roy. Statistical Soc.*, vol. 73, pp. 273–282, 2011.

Qingchao Jiang (M'17) received the B.E. degree in measurement and control technology and instrument and the Ph.D. degree in control theory and engineering from the East China University of Science and Technology, Shanghai, China, in 2010 and 2015, respectively.

He had been a Postdoctoral Fellow with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada, a Humboldt Research Fellow with the Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, Duisburg, Germany, and a Visiting Research Fellow with the Department of Chemical and Biomolecular Engineering, The Hong Kong University of Science and Technology, Hong Kong. He is currently an Associate Professor in Automation with the East China University of Science and Technology. His research interests include data mining and analysis, data-driven soft sensing, multivariate statistical process monitoring, and Bayesian fault diagnosis.

Xuefeng Yan received the B.S. degree in biochemical engineering and the Ph.D. degree in control theory engineering from Zhejiang University, Hangzhou, China, in 1995 and 2002, respectively.

He is currently a Professor in Automation with the East China University of Science and Technology, Shanghai, China. His current research interests include complex chemical process modeling, optimizing and controlling, process monitoring, fault diagnosis, and intelligent information processing.

Hui Yi received the B.E. degree in biomedical engineering and the Ph.D. degree in control theory and engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005 and 2012, respectively.

In 2012, as a Lecturer in Automation, he joined the College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing, China, where he is currently an Associate Professor. His research interests include fault diagnosis, health monitoring, and artificial intelligence.

Furong Gao received the B.E. degree in automation from the China University of Petroleum, Beijing, China, in 1985, and the M.E. and Ph.D. degrees in chemical engineering from McGill University, Montreal, QC, Canada, in 1989 and 1993, respectively.

He was a Senior Research Engineer with Moldflow International Company, Ltd. Since 1995, he has been with the Hong Kong University of Science and Technology, Hong Kong, where he is currently a Chair Professor with the Department of Chemical and Biomolecular Engineering. His research interests include the areas of process monitoring and control and polymer processing.