I&EC research
Industrial & Engineering Chemistry Research

Article

pubs.acs.org/IECR

Just-in-Time Kernel Learning with Adaptive Parameter Selection for Soft Sensor Modeling of Batch Processes

Yi Liu,*[†] Zengliang Gao,[†] Ping Li,[‡] and Haiqing Wang[§]

[†] Key Laboratory of Pharmaceutical Engineering of Ministry of Education, Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou, 310032, People's Republic of China

[‡] State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, Zhejiang University, Hangzhou, 310027, People's Republic of China

[§] College of Mechanical and Electronic Engineering, University of Petroleum (East China), West Changjiang Road, No. 66, Qingdao, 266555, People's Republic of China

Supporting Information

ABSTRACT: An efficient nonlinear just-in-time learning (JITL) soft sensor method for online modeling of batch processes with uneven operating durations is proposed. A recursive least-squares support vector regression (RLSSVR) approach is combined with the JITL manner to model the nonlinearity of batch processes. The similarity between the query sample and the most relevant samples, including the weight of similarity and the size of the relevant set, can be chosen using a presented cumulative similarity factor. Then, the kernel parameters of the developed JITL-RLSSVR model structure can be determined adaptively using an efficient cross-validation strategy with low computational load. The soft sensor implement algorithm for batch processes is also developed. Both the batch-to-batch similarity and variation characteristics are taken into consideration to make the modeling procedure more practical. The superiority of the proposed soft sensor approach is demonstrated by predicting the concentrations of the active biomass and recombinant protein in the streptokinase fed-batch fermentation process, compared with other existing JITL-based and global soft sensors.

1. INTRODUCTION

Batch processes are of considerable importance in the biochemical pharmaceutical and fine chemical industries for producing high-value products. There has been growing interest in batch process modeling,[1−7] monitoring,[8,9] control,[10−14] and optimization[15] in recent years. To achieve good control and optimization performance, it is very important to obtain a suitable model capable of providing accurate predictions for product qualities and other key variables.[1−7] Comprehensive mechanistic models for batch processes are useful because they are valid over a wide range of process operations. However, the development of such first-principles models often requires detailed physical insight into batch processes and a large amount of time and resources. The long time scale in detailed mechanistic model development may make them infeasible for agile responsive manufacturing, where the products of batch processes are typically short-lived and of small volume.[1−15]

Although hardware analyzers have been widely used in the chemical industry, they are usually expensive and difficult to maintain. Furthermore, the large measurement delay of hardware analyzers significantly degrades the corresponding control and other automation performance.[2,10] Therefore, to address this issue, it becomes more acceptable to use empirical models or soft sensor models based on process operational data to estimate product qualities online instead of using hardware ones for direct measurement. Compared to comprehensive mechanistic models, one main advantage of empirical models is that they can generally be developed quickly without requiring substantial understanding of the phenomenology involved.[1−15] Additionally, process data have become widely available with digital instrumentation in many chemical plants nowadays. Therefore, many data-driven empirical modeling approaches,[1−8,16−35] such as multivariate statistical regression (MSR) (e.g., principal component regression (PCR) and partial least-squares (PLS)),[3−7,16,18,19] neural networks (NN),[17] support vector regression (SVR), and other kernel learning (KL) methods,[26−33] have been applied to chemical processes. Among these methods, due to the good modeling performance with limited training samples,[20−25] SVR and other KL methods have become more attractive for batch processes recently.[31−36]

Most chemical processes are actually time-variant due to the combined contributions of many random events such as catalyst deactivation, equipment aging, and sensor and process drifting. Therefore, how to keep an existing soft sensor model updated to trace the time-varying and nonlinear characteristics is still intractable.[18,19,37] Generally, the moving window technique and recursive strategy are employed to trace the time-varying dynamics by various process modeling and monitoring methods.[4,18,37−41] Different kinds of recursive soft sensor models have been proposed, such as recursive least-squares,[4] recursive PCR,[38] recursive PLS,[39] recursive kernel principal component analysis, and recursive SVR.[40,41] Although these methods can adapt a soft sensor model to new operational conditions recursively, they have difficulty coping

Received: July 28, 2011
Revised: February 11, 2012
Accepted: February 15, 2012
Published: February 15, 2012

dx.doi.org/10.1021/ie201650u | *Ind. Eng. Chem. Res.* 2012, 51, 4313−4327

with abrupt changes of a process. Additionally, most recursive soft sensors may not function well in a new operational region until a sufficient period of time due to the time delay when they adapt themselves to the new operational condition.[19]

As for batch and fed-batch processes, recursive soft sensors might be insufficient to adapt themselves quickly since batch processes have no steady-state operating point and often exhibit multiphase characteristics, complex dynamics, and nonlinearity.[2,3,5,6] Recursive adaptation approaches often assume that the batches are closest in time to the current batch. This assumption is likely to hold when batch-to-batch variations are slow or small. However, the conjecture of batch-to-batch similarity is often invalid whenever process changes are not gradual, e.g., because of changing operating policies.[7] Consequently, compared to continuous processes, it is more difficult to develop an approving soft sensor model and maintain the performance for many batch processes with uneven durations.

Recently, the just-in-time learning (JITL) method,[42,43] which is also named the "lazy learning" or "model-on-demand" method and is originally from the machine learning area, has been developed as an attractive alternative for chemical process modeling, monitoring, and control problems.[44−48] In the JITL model structure, a local model is built using the most relevant samples from the historical data set around a query sample when the estimated value of the sample is required. Different from traditional offline modeling and recursive modeling methods, which can both be considered as global models, the JITL-based method exhibits a local model structure and it is built online with a lazy learning manner. Thus, the current state of the process can be tracked by the JITL model and then it can cope with the process nonlinearity directly.[42−48]

However, most JITL methods are combined with multivariable linear regression to construct a local linear model.[42−48] Until now, few nonlinear modeling approaches have been integrated with JITL to build a local nonlinear model for better description of the process nonlinearity. To this end, in our previous study, a JITL-based least-squares SVR (JITL-LSSVR) is developed for modeling of fermentation processes.[49] Recently, Ge and Song extended the method using a real-time performance improvement strategy to enhance the online modeling efficiency and applied it to continuous processes.[50] However, as an important issue, how to select online the parameters of nonlinear JITL-based models, e.g., JITL-LSSVR, has been less investigated until now. As we know, the performance of soft sensor models can be degraded if the related parameters are not chosen properly.[18,19,37] Additionally, to the best of our knowledge, the application of JITL modeling methods to the batch processes has rarely been reported, especially for the purpose of soft sensor development.

Therefore, this paper intends to develop a nonlinear JITL-based soft sensor for online modeling of batch processes. The recursive LSSVR (RLSSVR) method is adopted to construct the JITL model because it can describe a nonlinear multi-input multi-output (MIMO) process using limited modeling samples.[41] Additionally, the nonlinear JITL-RLSSVR model structure, including the model parameters and the similarity between the query sample and the most relevant samples, can be determined adaptively using an efficient cross-validation (CV) strategy.[51,52] Therefore, both the good prediction performance and small computational load of the nonlinear JITL-RLSSVR-based soft sensor can be achieved.

The remainder of this paper is organized as follows. After a brief introduction of RLSSVR for nonlinear MIMO process modeling, detailed descriptions of the strategy for determination of the adaptive JITL-RLSSVR model structure are provided in section 2. The JITL-RLSSVR soft sensor implement algorithm for batch processes is also developed in this section. In section 3, the JITL-RLSSVR soft sensor is evaluated by predicting the key product concentrations in the streptokinase fed-batch fermentation process, compared with other existing JITL-based and global soft sensors. Finally, some conclusions are drawn in section 4.

2. NONLINEAR JITL-RLSSVR SOFT SENSOR MODEL DEVELOPMENT

2.1. RLSSVR Modeling Method. The process modeling issue and the soft sensor model development based on the KL framework can be both described as a problem where the aim is to learn a mapping f: $X \to Y$ using a modeling set $S = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_l, \mathbf{y}_l)\} \subset X \times Y$. As a temporal step of the KL method, one can consider that the input data $\mathbf{x}_i \in X$ is first mapped implicitly into the feature space H by ϕ: $X \to H$, where ϕ is a feature map associated with some positive definite kernel which satisfies the Mercer theorem, i.e., the so-called kernel trick: $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle_H$.[20−25] The function f is then determined in some optimal sense to yield the soft sensor model. A general form of the kernelized nonlinear MIMO model for process modeling can be formulated as[41]

$$\begin{aligned} y_{k,m} &= f(\mathbf{w}_{k,m}, b_{k,m}, \mathbf{x}_k) + e_{k,m} \\ &= \mathbf{w}_{k,m}^{\mathrm{T}}\phi(\mathbf{x}_k) + b_{k,m} + e_{k,m} \end{aligned} \tag{1}$$

where $f \in H$ is the wanted model; $y_{k,m}$ denotes the mth output measurement of the wanted model at k instance with $m = 1, ..., M$, and M is the number of outputs; $\mathbf{x}_k$ is a general input vector that is usually composed of several measured variables at time k, probably combined with their corresponding delayed forms and with the delayed outputs. The symbols $\mathbf{w}_{k,m}$, $b_{k,m}$, and $e_{k,m}$ are the model parameter vector, the bias term, and the process noise of the mth subsystem, respectively.[41]

When the philosophy of statistical learning theory[20] and the LSSVR framework[23−25] is applied to eq 1, the following optimization problem is formulated:[23−25]

$$\begin{aligned} &\min J(\mathbf{w}_{k,m}, b_{k,m}) = \frac{1}{2}\|\mathbf{w}_{k,m}\|^2 + \frac{\gamma}{2}\|\mathbf{e}_{k,m}\|^2 \\ &\text{s.t.} \quad y_{i,m} - \mathbf{w}_{k,m}^{\mathrm{T}}\phi(\mathbf{x}_i) - b_{k,m} - e_{i,m} = 0,\ i = 1, ..., k \end{aligned} \tag{2}$$

where $\mathbf{e}_{k,m} = [e_{1,m}, e_{2,m}, ..., e_{k,m}]^{\mathrm{T}}$ is the approximation error. The formulation consists of equality constraints instead of inequality constraints in the conventional SVR algorithm and takes into account a squared error with the regularization term. Therefore, this reformulation greatly simplifies a problem such that the LSSVR solution follows directly from solving a set of linear equations rather than from a convex quadratic program and, moreover, makes the recursive algorithm more straightforward and computationally efficient.[23−25] The user-defined regularization parameter γ ($\gamma > 0$) determines the trade-off between the model's complexity and approximation accuracy. Also, a suitable choice of γ can prevent from overfitting.[23−25]

To solve the optimization problem, the Lagrangian can be constructed below:[23−25]

$$L(\mathbf{w}_{k,m}, \mathbf{e}_{k,m}, \boldsymbol{\alpha}_{k,m}, b_{k,m}) = (\|\mathbf{w}_{k,m}\|^2 + \gamma \|\mathbf{e}_{k,m}\|^2)/2 + \sum_{i=1}^{k} \alpha_{k,m,i}[y_{i,m} - \mathbf{w}_{k,m}^{\mathrm{T}}\phi(\mathbf{x}_i) - b_{k,m} - e_{i,m}], \quad i = 1, ..., k \tag{3}$$

where $\boldsymbol{\alpha}_{k,m} = [\alpha_{k,m,1}, ..., \alpha_{k,m,k}]^{\mathrm{T}}$ are Lagrange multipliers. The optimality conditions are below:

$$\begin{aligned}
&\frac{\partial L}{\partial \mathbf{w}_{k,m}} = 0 \rightarrow \mathbf{w}_{k,m} = \sum_{i=1}^{k} \alpha_{k,m,i}\phi(\mathbf{x}_i)\\
&\frac{\partial L}{\partial e_{i,m}} = 0 \rightarrow \alpha_{k,m,i} = \gamma e_{i,m}, \qquad i = 1, ..., k\\
&\frac{\partial L}{\partial \alpha_{k,m,i}} = 0 \rightarrow y_{i,m} - \mathbf{w}_{k,m}^{\mathrm{T}}\phi(\mathbf{x}_i) - b_{k,m} - e_{i,m} = 0,\\
&\quad i = 1, ..., k\\
&\frac{\partial L}{\partial b_{k,m}} = 0 \rightarrow \sum_{i=1}^{k} \alpha_{k,m,i} = 0
\end{aligned} \tag{4}$$

After elimination of the variables $\mathbf{w}_{k,m}$ and $e_{k,m}$, the following solution can be obtained:

$$\begin{bmatrix} \mathbf{K}_k + \mathbf{I}_k/\gamma & \mathbf{1}_k \\ \mathbf{1}_k^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{k,m} \\ b_{k,m} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k,m} \\ 0 \end{bmatrix} \tag{5}$$

where $\mathbf{y}_{k,m} = [y_{1,m}, ..., y_{k,m}]^{\mathrm{T}}$, $\mathbf{I}_k \in \mathbf{R}^{k\times k}$ is a unit matrix, and $\mathbf{1}_k$ is a vector column vector of 1's; $\mathbf{K}_k$ is a kernel matrix, and the "kernel trick" applied here is[20−25]

$$\mathbf{K}_k(i, j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle, \qquad \forall\ i, j = 1, ..., k \tag{6}$$

For simplicity, the quantities are defined as $\mathbf{H}_k = \mathbf{K}_k + \mathbf{I}_k/\gamma$ and $\mathbf{P}_k = \mathbf{H}_k^{-1}$; then the solution can be expressed as

$$\boldsymbol{\alpha}_{k,m} = \mathbf{P}_k \left[\mathbf{y}_{k,m} - \frac{\mathbf{1}_k \mathbf{1}_k^{\mathrm{T}} \mathbf{P}_k \mathbf{y}_{k,m}}{\mathbf{1}_k^{\mathrm{T}} \mathbf{y}_{k,m} \mathbf{1}_k} \right] \tag{7}$$

$$b_{k,m} = \frac{\mathbf{1}_k^{\mathrm{T}} \mathbf{P}_k \mathbf{y}_{k,m}}{\mathbf{1}_k^{\mathrm{T}} \mathbf{y}_{k,m} \mathbf{1}_k} \tag{8}$$

The LSSVR model estimation of the mth subsystem at time $k + 1$, i.e., $\hat{y}_{k+1,m}$, can be obtained:

$$\begin{aligned}
\hat{y}_{k+1,m} &= f(\mathbf{w}_{k,m}, \mathbf{x}_{k+1})\\
&= \sum_{i=1}^{k} \alpha_{k,m,i}\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{k+1}) \rangle + b_{k,m}\\
&= \boldsymbol{\alpha}_{k,m}^{\mathrm{T}} \mathbf{k}_{k+1} + b_{k,m}
\end{aligned} \tag{9}$$

where $\mathbf{k}_{k+1}(i) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{k+1}) \rangle$, $\forall\ i = 1, ..., k$, is a kernel vector.

In summary, the development of a soft sensor model amounts to solving a set of linear equations in the high-dimensional feature space introduced by the kernel transform. Generally, the direct solution of a set of linear equations involves the inverse of matrix $\mathbf{H}_k$, which will make the computation inefficient if the data set becomes large. Additionally, offline modeling is not suitable for practical process automation applications where the data come in a sequential way and it is difficult for new information of the process to be directly absorbed into the established model. This is a very common situation in many online adaptive learning problems.[37] Although the model can be retrained from scratch when the modeling set is modified, it is cumbersome and computationally inefficient.

Therefore, the recursive learning algorithm is adopted to make the computation more efficient. The common framework of the RLSSVR algorithm consists of two recursive learning stages: the forward incremental stage and the backward decremental stage.[41] Here, only the incremental learning stage is adopted for JITL because the decremental learning stage is not necessary (which will be described later). In this paper, the samples adopted into the RLSSVR model are referred to as nodes.[41] Assume at time k the RLSSVR soft sensor model has N_k nodes, and then eq 5 can be rewritten as

$$\begin{bmatrix} \mathbf{K}_{N_k} + \mathbf{I}_{N_k}/\gamma & \mathbf{1}_{N_k} \\ \mathbf{1}_{N_k}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{N_k,m} \\ b_{N_k,m} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{N_k,m} \\ 0 \end{bmatrix} \tag{10}$$

When a new sample is introduced into the RLSSVR model as a node, the following inverse relation between matrix computation of $\mathbf{P}_{N_{k+1}}$ and $\mathbf{P}_{N_k}$ can be obtained using the Sherman−Morrison−Woodbury formula:[41,53]

$$\mathbf{P}_{N_{k+1}} = \begin{bmatrix} \mathbf{P}_{N_k} & 0 \\ 0 & 0 \end{bmatrix} + \mathbf{r}_{N_{k+1}} \mathbf{r}_{N_{k+1}}^{\mathrm{T}} z_{N_{k+1}} \tag{11}$$

where $\mathbf{r}_{N_{k+1}} = [\mathbf{V}_{N_{k+1}}^{\mathrm{T}} \mathbf{P}_{N_k}, -1]^{\mathrm{T}}$ is a column vector, $z_{N_{k+1}} = 1/(v_{N_{k+1}} - \mathbf{V}_{N_{k+1}}^{\mathrm{T}} \mathbf{P}_{N_k} \mathbf{V}_{N_{k+1}})$ and $v_{N_{k+1}} = \mathbf{K}(\mathbf{x}_{N_{k+1}}, \mathbf{x}_{N_{k+1}}) + 1/\gamma$ are scalars, and $\mathbf{V}_{N_{k+1}} = [\mathbf{K}(\mathbf{x}_1, \mathbf{x}_{N_{k+1}}), ..., \mathbf{K}(\mathbf{x}_{N_k}, \mathbf{x}_{N_{k+1}})]^{\mathrm{T}}$ is the corresponding kernel vector of the new node.

Generally, the direct computation of the matrix $\mathbf{P}_{N_k}$ requires about $O(N_k^3)$ operations. It might be less by application of a conjugate gradient algorithm.[23] However, the computational load is still large and cumbersome. Whenever a new node is available, the recursive algorithm of the forward incremental learning stage only needs $O(N_k^2)$ operations.[41] The improvement on the computing speed is noticeable when the number of nodes N_k becomes large or the LSSVR model has to be built many times combined with the JITL manner. Therefore, compared to conventional JITL-LSSVR,[49,50] a real-time performance improvement can be obtained by the proposed JITL-RLSSVR method due to its recursive update mechanism.

2.2. Global Learning and JITL.

As described above, many conventional data-based modeling methods, such as MSR,[3−7,16,18,19] NN,[17] SVR, and other KL methods,[26−36] can be considered as global learning approaches. However, the direct application of global modeling methods for complex chemical processes, especially multiphase batch/fed-batch processes, may become less attractive because it is often difficult to specify the complexity and the structure of a global model. Also, as aforementioned, another limitation of the global methods is that it is difficult for them to be updated quickly when the process dynamics are moved away from the nominal operating area. On the other hand, a nonlinear process can be approximated with a set of relatively simple local models valid in certain operating regimes. Instead of using a global model, some suitable local models can be constructed to describe a

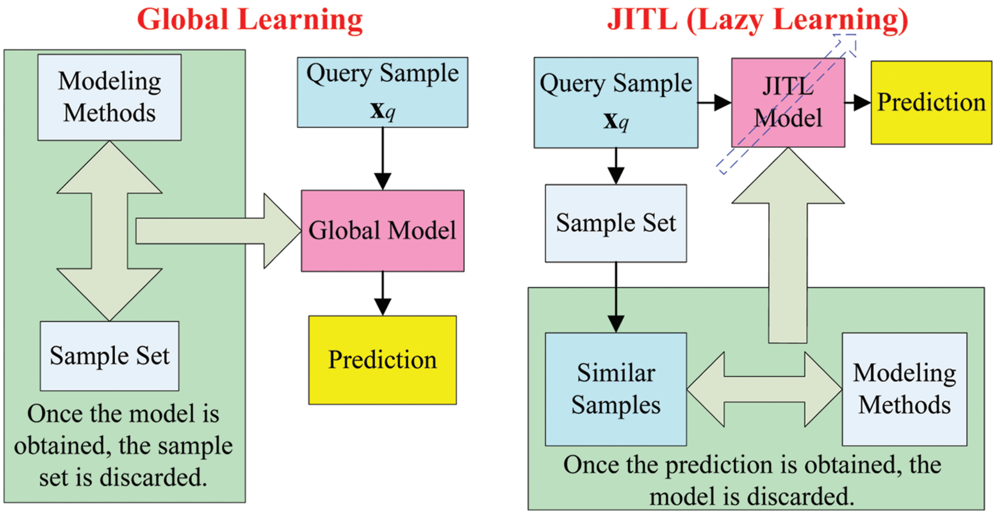
Global Learning
Modeling Methods
Sample Set
Once the model is obtained, the sample set is discarded.
Query Sample $\mathbf{x}_q$
Global Model
Prediction
JITL (Lazy Learning)
Query Sample $\mathbf{x}_q$
JITL Model
Prediction
Sample Set
Similar Samples
Modeling Methods
Once the prediction is obtained, the model is discarded.

Figure 1. Comparison between conventional global learning and JITL (or lazy learning) approaches.

nonlinear process.[44] Nevertheless, it is often complicated to determine how many local models for a nonlinear process are suitable when a priori knowledge is lacking.[44]

To alleviate these problems and construct the local models automatically, the JITL method, which is inspired by ideas from local modeling and database technology, has been developed as an attractive alternative for nonlinear processes.[42−50] Bontempi et al. first applied the JITL approach to modeling and control problems.[43] Cheng and Chiu introduced the JITL method into process modeling and monitoring areas.[44,45] As for process control, some JITL-based controllers have recently been designed for nonlinear chemical processes.[46]

The comparison between conventional global learning and JITL (or lazy learning) approaches is shown in Figure 1. Compared with traditional global learning, JITL exhibits three main characteristics. First, the modeling stage is postponed until an output for a query sample is requested. Then, the prediction for the query sample is computed by exploiting the stored samples in the database. Finally, once the prediction is obtained, the model and any intermediate results are discarded.[44,45] Therefore, the JITL approach shows a lazy manner (i.e., model on demand), which distinguishes itself from conventional global or local modeling methods with an active manner (i.e., model on hand).

Generally, for a query sample $\mathbf{x}_q$, there are three main steps to build a JITL model for online prediction:[44,45]

Step 1. Search the relevant samples to construct a similar set S_{sim} in the database S based on some defined similarity criterions.

Step 2. Build a linear or nonlinear JITL model $f_{\text{JITL}}(\mathbf{x}_q)$ using the relevant data set S_{sim}.

Step 3. Predict the output $\hat{\mathbf{y}}_q$ online for the current query sample $\mathbf{x}_q$ and then discard the JITL model $f_{\text{JITL}}(\mathbf{x}_q)$.

With the same three-step procedure, a new JITL model can be built for the next query sample. As a key step, how to construct the similarity criterion to match the query sample is important. Previously, a Euclidean distance-based similarity criterion was commonly utilized for JITL.[43] However, only utilization of the distance for description of the similarity is not comprehensive. Cheng and Chiu adopted the distance and angle to construct a more comprehensive similarity factor (SF) and showed better performance than with only utilization of the Euclidean distance.[44] The SF s_{qi} between the query sample $\mathbf{x}_q$ and the sample $\mathbf{x}_i$ in the data set is defined below:[44]

$$s_{qi} = \rho \exp(-d_{qi}) + (1 - \rho) \cos(\theta_{qi}), \quad \text{for } \cos(\theta_{qi}) \geq 0, \quad i = 1, ..., k$$
$$d_{qi} = \left\|\mathbf{x}_i - \mathbf{x}_q\right\|_2 \tag{12}$$
$$\cos(\theta_{qi}) = \langle \mathbf{x}_i, \mathbf{x}_q \rangle / (\|\mathbf{x}_i\|_2 \left\|\mathbf{x}_q\right\|_2)$$

where d_{qi} and $\cos(\theta_{qi})$ are the distance similarity and the angle similarity between $\mathbf{x}_q$ and $\mathbf{x}_i$ in S, respectively; $0 \leq \rho \leq 1$ is a weight parameter, and only the distance similarity (or the angle similarity) is adopted when $\rho = 1$ (or $\rho = 0$). The value of s_{qi} is bounded between 0 and 1, and when s_{qi} approaches 1, $\mathbf{x}_q$ closely resembles $\mathbf{x}_i$. Note that eq 12 will not be used to compute the similarity s_{qi} between $\mathbf{x}_q$ and $\mathbf{x}_i$ if $\cos(\theta_{qi})$ is negative. This can be illustrated in the three-dimensional space shown in Figure 2. For $\mathbf{x}_q$, $\mathbf{x}_i$ is more similar than $\mathbf{x}_j$, although the distances of d_{qi} and d_{qj} might be the same. Consequently, the corresponding $\mathbf{x}_j$ in the database will be not involved in the subsequent JITL modeling procedure.

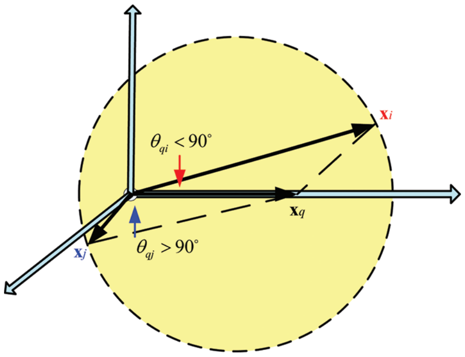

Figure 2. Illustration of the similarity with distance and angle in three-dimensional space.

As a critical issue, the prediction performance can be enhanced with a suitable choice of ρ because more similar samples are utilized to construct a local model. However, how to select the weight ρ between the distance and angle has still been researched little.

To this end, the distance and angle based similarity criterion is utilized and its related weight ρ is optimized in this paper, which is demonstrated in section 2.3. Although a new correlation-based similarity criterion has been proposed recently,[47,48] it is not the main scope of the proposed method. Thus, the correlation-based similarity criterion is not investigated in this paper.

Based on the similarity criterion, the l ($l_{\min} \le l \le l_{\max}$) related samples should be chosen before building a JITL model. Generally, two parameters $l_{\min}$ and $l_{\max}$ are chosen such that only the relevant data sets formed by the $l_{\min}$th relevant data to the $l_{\max}$th relevant data are used in the model.[44] However, how to determine the number of similar samples remains unsolved. Cheng and Chiu adopted a linear search method with a search step Δl to choose l.[44] Nevertheless, the computational burden may become large if the range of $[l_{\min}, l_{\max}]$ is wide. Also, how to determine both the two parameters of $l_{\min}$ and $l_{\max}$ is not provided, especially for batch processes.

To solve the problem, an improved similarity criterion is proposed. For a similarity weight ρ, the $l_{\max}$ most similar samples can be selected due to the SF criterion in eq 12. Then, rank the $l_{\max}$ similar samples according to the degree of similarity. Define a cumulative similarity factor (CSF) s_l below:

$$s_l = \frac{\sum_{i=1}^{l} s_{qi}}{\sum_{i=1}^{l_{\max}} s_{qi}}, \qquad l_{\min} \le l \le l_{\max} \tag{13}$$

which represents the cumulative similarity of l most similar samples compared to the relevant data set S_{sim}. The CSF index can be utilized to access the cumulative similarity and then determine the l most similar samples more reasonably. For example, s_l can be set as a value of 0.9 to choose l. Additionally, for batch processes, the modeling samples available are accumulated according to batches. Due to this reason, it is infeasible to set the range of $[l_{\min}, l_{\max}]$ as fixed. As an alternative method, the search range of $[l_{\min}, l_{\max}]$ can be substituted by the choice of s_l, e.g., $0.8 \le s_l \le 1$. Therefore, compared to the method of searching l in the range of $[l_{\min}, l_{\max}]$ directly,[44] the proposed CSF index is more meaningful and its computational burden can be reduced by narrowing the range of s_l suitably.

2.3. Adaptive Nonlinear JITL-RLSSVR Model Development.

The development of a good LSSVR/RLSSVR soft sensor model depends on suitable selection of related model parameters. The parameter selection is an important issue in the kernel learning area. There exist many theories which lead to different model selection criterions, e.g., cross-validation and Bayesian inference.[20−26] However, there is still no optimal parameter selection theory aiming for industrial applications, especially for the online modeling issue.

Two common kernel functions, i.e., the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp[-\|\mathbf{x}_i - \mathbf{x}_j\|/\sigma]$ (the width parameter $\sigma > 0$) and the polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$ (the degree parameter d is a natural number), are usually utilized in SVR and other KL methods.[20−25] Generally, there are two parameters: the regularization parameter γ and the related kernel parameter (σ or d) to be chosen. Based on the literature, these two parameters can work well in a relatively wide range for continuous processes due to the stable operating conditions.[54] Nevertheless, for multiphase nonlinear batch processes, the LSSVR-based soft sensor model might degrade with unsuitable parameters for some operating conditions. In a previous study, the JITL-LSSVR-based soft sensors were not optimized due to the large computational load, especially for the online modeling.[49,50] Therefore, as can be expected, the prediction performance can be enhanced if the parameters of JITL-LSSVR are optimized for every query sample.

To this end, an adaptive nonlinear JITL-RLSSVR model is developed with online optimization of parameters. The parameter selection procedure is mainly based on the fast leave-one-out cross-validation (FLOO-CV) criterion.[51,52] The traditional LOO-CV method has been shown to give an unbiased estimate of the generalization properties of the statistical models and thus provides a sensible criterion for model selection. Especially for the conventional SVR method, bounds on the LOO-CV estimator have been proved to be an effective criterion for model selection.[51,52] However, the conventional LOO-CV for SVR/LSSVR is computationally expensive and thus not suitable for online performing. Recently, a FLOO-CV criterion has been proposed by Cawley and Talbot.[51,52] Therefore, applying the FLOO-CV criterion as the foundation of parameter selection is feasible.

For the JITL-RLSSVR model, the parameters include the regularization parameter $\gamma \in \gamma_S$, the kernel parameter $\sigma \in \sigma_S$ for the Gaussian kernel (or the kernel parameter $d \in d_S$ for the polynomial kernel), and the similarity weight $\rho \in \rho_S$. Using the Gaussian kernel, the parameter set can be defined as $S_{\text{par}} = [\gamma_S, \sigma_S, \rho_S]$. If the similarity weight $\rho \in \rho_S$ and the number of similar samples l are determined, the relevant data set S_{sim} can be obtained. Then, with a pair of parameters $[\gamma, \sigma]$, the JITL-RLSSVR model can be built. Therefore, for a relevant data set S_{sim} with l samples, the adaptive JITL-RLSSVR model can be formulated as follows. With quantities defined as $\mathbf{H}_l = \mathbf{K}_l + \mathbf{I}_l/\gamma$ and $\mathbf{P}_l = \mathbf{H}_l^{-1}$, the matrix on the left-hand side of eq 5 can be decomposed into the block-matrix representation below:

$$\begin{bmatrix} \mathbf{K}_l + \mathbf{I}_l/\gamma & \mathbf{1}_l \\ \mathbf{1}_l^{\mathrm{T}} & 0 \end{bmatrix} = \begin{bmatrix} q_{11} & \mathbf{q}_1^{\mathrm{T}} \\ \mathbf{q}_1 & \mathbf{Q}_1 \end{bmatrix} = \mathbf{Q} \tag{14}$$

Then the following equations from eq 14 can be obtained.[51]

$$q_{11}\alpha_{1,m} + \mathbf{q}_1^{\mathrm{T}}[\bar{\boldsymbol{\alpha}}_{l,m}^{\mathrm{T}}, b_{l,m}]^{\mathrm{T}} = y_{1,m} \tag{15}$$

$$[\mathbf{q}_1 \quad \mathbf{Q}_1][\bar{\boldsymbol{\alpha}}_{l,m}^{\mathrm{T}}, b_{l,m}]^{\mathrm{T}} = [\bar{\mathbf{y}}_{l,m}^{\mathrm{T}}, 0]^{\mathrm{T}} \tag{16}$$

where $\bar{\boldsymbol{\alpha}}_{l,m}$ and $\bar{\mathbf{y}}_{l,m}$ denote the vectors $\boldsymbol{\alpha}_{l,m}$ and $\mathbf{y}_{l,m}$ excluding the first line, respectively; that is, $\boldsymbol{\alpha}_{l,m}^{\mathrm{T}} = [\alpha_{1,m}, \bar{\boldsymbol{\alpha}}_{l,m}^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{y}_{1,m}^{\mathrm{T}} = [y_{l,m}, \bar{\mathbf{y}}_{l,m}^{\mathrm{T}}]^{\mathrm{T}}$. Let $\boldsymbol{\alpha}_{l,m}^{(-i)}$ and $b_{l,m}^{(-i)}$ represent the model parameters of JITL-RLSSVR during the ith iteration of the FLOO-CV procedure; then in the first iteration, in which the first node is deleted, the following equation is obtained.[51]

$$\begin{bmatrix} \boldsymbol{\alpha}_{l,m}^{(-1)} \\ b_{l,m}^{(-1)} \end{bmatrix} = \mathbf{Q}_1^{-1} \begin{bmatrix} \bar{\mathbf{y}}_{l,m} \\ 0 \end{bmatrix} \tag{17}$$

The FLOO-CV prediction for the first node is given by[51]

$$\hat{y}_{1,m}^{(-1)} = \mathbf{q}_1^{\mathrm{T}} \begin{bmatrix} \boldsymbol{\alpha}_{l,m}^{(-1)} \\ b_{l,m}^{(-1)} \end{bmatrix} = \mathbf{q}_1^{\mathrm{T}} \mathbf{Q}_1^{-1} [\bar{\mathbf{y}}_{l,m}^{\mathrm{T}}, 0]^{\mathrm{T}} \tag{18}$$

After substituting eqs 16 and 15 into eq 18 subsequently, we can deduce

$$\begin{aligned}\hat{y}_{1,m}^{(-1)} &= \mathbf{q}_1^{\mathrm{T}}\mathbf{Q}_1^{-1}[\mathbf{q}_1 \;\; \mathbf{Q}_1][\bar{\boldsymbol{\alpha}}_{l,m}, b_{l,m}]^{\mathrm{T}} \\ &= \mathbf{q}_1^{\mathrm{T}}\mathbf{Q}_1^{-1}\mathbf{q}_1\alpha_{1,m} + \mathbf{q}_1^{\mathrm{T}}[\bar{\boldsymbol{\alpha}}_{l,m}, b_{l,m}]^{\mathrm{T}} \\ &= y_{1,m} - \alpha_{1,m}(q_{11} - \mathbf{q}_1^{\mathrm{T}}\mathbf{Q}_1^{-1}\mathbf{q}_1)\end{aligned} \tag{19}$$

Meanwhile, via the block matrix inversion lemma, the following equation can be deduced.[51,53]

$$\begin{aligned}\mathbf{Q}^{-1} &= \begin{bmatrix} q_{11} & \mathbf{q}_1^{\mathrm{T}} \\ \mathbf{q}_1 & \mathbf{Q}_1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} c^{-1} & -c^{-1}\mathbf{q}_1\mathbf{Q}_1^{-1} \\ \mathbf{Q}_1^{-1} + c^{-1}\mathbf{Q}_1^{-1}\mathbf{q}_1^{\mathrm{T}}\mathbf{q}_1\mathbf{Q}_1^{-1} & -c^{-1}\mathbf{Q}_1^{-1}\mathbf{q}_1^{\mathrm{T}} \end{bmatrix}\end{aligned} \tag{20}$$

where $c = q_{11} - \mathbf{q}_1^{\mathrm{T}}\mathbf{Q}_1^{-1}\mathbf{q}_1$. Consequently, the prediction error after deleting the first node by using the FLOO-CV procedure is formulated:[51]

$$e_{1,m}^{\mathrm{FLOO}} = y_{1,m} - \hat{y}_{1,m}^{(-1)} = \alpha_{1,m}/Q_{11}^{-1} \tag{21}$$

where Q_{11}^{-1} is the term at the first row and the first column of $\mathbf{Q}^{-1}$. Note that the system of linear equations is insensitive to permutations of the ordering of the equations and of the unknowns.[51] Therefore, the prediction error when deleting the ith node using FLOO-CV is obtained:

$$e_{i,m}^{\mathrm{FLOO}} = y_{i,m} - \hat{y}_{i,m}^{(-1)} = \alpha_{i,m}/Q_{ii}^{-1} \tag{22}$$

where Q_{11}^{-1} is the item at the ith row and ith column of $\mathbf{Q}^{-1}$. Furthermore, using the block matrix inversion formula on the matrix $\mathbf{Q}$, the following equation is formulated:[51,53]

$$\mathbf{Q}^{-1} = \begin{bmatrix} \mathbf{P}_l + \mathbf{P}_l\mathbf{1}_l\mathbf{1}_l^{\mathrm{T}}\mathbf{P}_l/o & -\mathbf{P}_l\mathbf{1}_l/o \\ -\mathbf{1}_l^{\mathrm{T}}\mathbf{P}_l/o & \mathbf{1}/o \end{bmatrix} \tag{23}$$

where $o = -\mathbf{1}_l^{\mathrm{T}}\,\mathbf{P}_l\mathbf{1}_l$ is a scalar, which is principally concerned only with the diagonal elements of $\mathbf{Q}^{-1}$ and $\mathbf{P}_l$. Thus, the equation below can be deduced:

$$Q_{ii}^{-1} = P_{l,ii} + s_i^2/o, \qquad i = 1, ..., l \tag{24}$$

where $\mathbf{s} = \mathbf{P}_l\mathbf{1}_l = [s_1, ..., s_l]^{\mathrm{T}}$ is the corresponding vector. After substituting eq 24 into eq 22, the final formula of prediction error when deleting the ith node by using the FLOO-CV procedure can be obtained.[51]

$$\begin{aligned}e_{i,m}^{\mathrm{FLOO}} &= y_{i,m} - \hat{y}_{i,m}^{(-1)} = \alpha_{i,m}/(P_{l,ii} + s_i^2/o), \\ i &= 1, ..., l\end{aligned} \tag{25}$$

Therefore, from eq 25, based on the FLOO-CV criterion, the total error of the JITL-RLSSVR model with l samples can be obtained:

$$E_l^{\mathrm{FLOO}} = \sum_{i=1}^{l} \left\| \mathbf{e}_i^{\mathrm{FLOO}} \right\|_2 \tag{26}$$

where $\mathbf{e}_i^{\mathrm{FLOO}} = [e_{i,1}^{\mathrm{FLOO}}, ..., e_{i,M}^{\mathrm{FLOO}}]^{\mathrm{T}}$. The related terms (i.e., $\mathbf{P}_l$ and $\boldsymbol{\alpha}_{i,m}$) are available. Additionally, the computational load of $\mathbf{s}$ and o is small. Consequently, compared to conventional LOO-CV method, the computation of E_i^{FLOO} is much more efficient.

In the naive implementation of LOO-CV, the LSSVR model will be trained l times, each time leaving out one node to perform the prediction. The implementation involves the solutions of l linear systems of dimension $(l - 1)$. Consequently, for each pair of potential model parameters $[\gamma, \sigma]$, the complexity of the naive LOO-CV is about $O(l^4)$ operations.[51,52] To reduce the computational load, the FLOO-CV algorithm is adopted to calculate the predictive error, which leads to only one linear system to be solved for each potential model parameter. Therefore, the complexity can be reduced to about $O(l^3)$ operations.[51,52]

For each pair of model parameters $[\gamma, \sigma]$, some potential JITL-RLSSVR models should be evaluated for $l_{\min} \le l \le l_{\max}$. Here, the recursive algorithm (i.e., eqs 10 and 11) can be implemented to improve the computational speed when a higher order LSSVR model is built. The computational complexity of about $\sum_{l=l_{\min}}^{l_{\max}} O(l^3)$ operations without recursive learning can be reduced to about $[O(l_{\min}{}^3) + \sum_{l=l_{\min}+\Delta l}^{l_{\max}} \quad O(l^2)]$ operations using recursive learning with a search step Δl. Consequently, if the potential parameter set $S_{\mathrm{par}} = [\gamma_S, \sigma_S, \rho_S]$ is large, the computational load for an optimized JITL-RLSSVR model can be saved a lot.

As aforementioned, the parameter set is defined as $S_{\mathrm{par}} = [\gamma_S, \sigma_S, \rho_S]$. Based on the FLOO-CV strategy and the recursive algorithm, an adaptive JITL-RLSSVR model can be determined by the following four-step procedure.

Step 1. For a similarity weight $\rho \in \rho_S$, the $l_{\max}$ most similar samples can be selected due to the similarity criterion in eq 12. Rank the $l_{\max}$ similar samples according to the similarity.

Step 2. Set a range of s_l and then the related range of $[l_{\min}, l_{\max}]$ can be obtained using the CSF criterion in eq 13. Choose the corresponding l samples to build a JITL-RLSSVR model with a pair of parameters $[\gamma, \sigma] \in [\gamma_S, \sigma_S]$. Then compute the total error of the JITL-RLSSVR model E_l^{FLOO}.

Step 3. Let l be increased by Δl, i.e., $l = l + \Delta l$. Then, perform step 2, that is, compute the total LOO-CV error of the JITL-RLSSVR model with l samples, i.e., E_l^{FLOO}, where $l_{\min} \le l \le l_{\max}$. The incremental learning stage of the RLSSVR algorithm, i.e., eqs 10 and 11, is applied here to improve the computational speed. For certain parameters $[\gamma, \sigma, \rho]$, the optimal samples can be achieved by $\min(E_l^{\mathrm{FLOO}})$, for $l_{\min} \le l \le l_{\max}$.

Step 4. Finally, for the whole parameter set $S_{\mathrm{par}} = [\gamma_S, \sigma_S, \rho_S]$, the optimal parameters can be obtained by $\arg_{S_{\mathrm{par}}} \min(E_l^{\mathrm{FLOO}})$, for $l_{\min} \le l \le l_{\max}$.

Consequently, the number of similar samples and model parameters can be selected adaptively using the FLOO-CV criterion and then an optimized JITL-RLSSVR soft sensor model is obtained for every query sample. Compared to previous JITL-LSSVR without optimization (using fixed model parameters),[49,50] the proposed nonlinear JITL model exhibits a more adaptive and flexible structure. Moreover, the computation is also efficient due to two improvements. First, the FLOO-CV strategy is applied to evaluate the model with a computational complexity of about $O(l^3)$ operations rather than about $O(l^4)$ operations. Second, during a batch run, for the prediction of every query $\mathbf{x}_{k,q}$, some potential JITL-RLSSVR models should be evaluated for $l_{\min} \le l \le l_{\max}$. The recursive

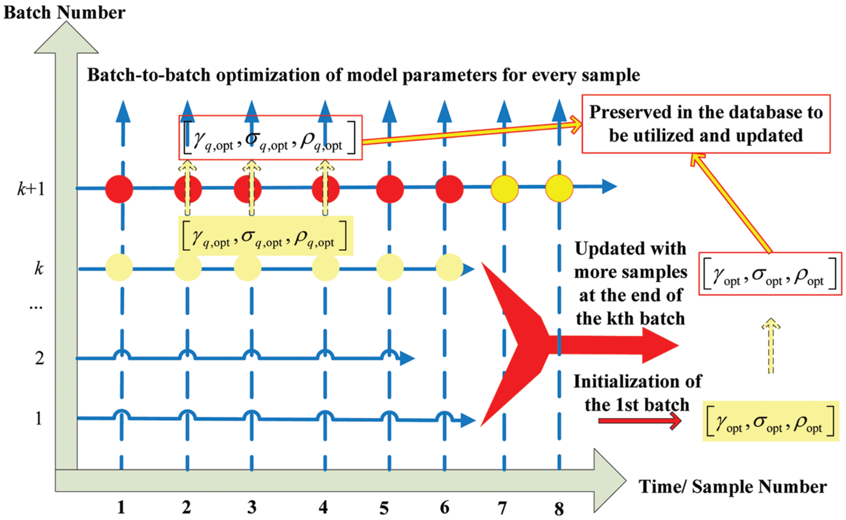
Batch Number
Batch-to-batch optimization of model parameters for every sample
$[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}]$
Preserved in the database to be utilized and updated
k+1
k
...
2
1
$[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}]$
Updated with more samples at the end of the kth batch
$[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$
Initialization of the 1st batch
$[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$
Time/ Sample Number
1
2
3
4
5
6
7
8

Figure 3. Batch-to-batch and query-to-query optimization of JITL-RLSSVR soft sensor model for online modeling of batch processes.

learning algorithm can be applied to improve the computational speed when a higher order LSSVR model is built. Therefore, the prediction performance can be improved with a more suitable model for every query sample.

2.4. Batch Process Modeling Strategy. In this section, the proposed JITL-RLSSVR soft sensor modeling method is applied to batch processes. Considering both the batch-to-batch similarity and variation characteristics, a modeling strategy is proposed as follows:

Step 1. Initialization: For the first batch, the initial optimal parameters can be determined using the obtained modeling samples at the end of the batch, noted as $[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$.

Step 2. Batch-to-batch online optimization: For the qth query of the kth batch ($k \geq 2$), i.e., $\mathbf{x}_{k,q}$, where $1 \leq q \leq J_k$ and J_k is the total number of samples of the kth batch, the parameter set S_{par} can be determined around $[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}]$, which can be obtained just from the last batch. Then, the optimal parameters, i.e., $[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}, l_{q,\mathrm{opt}}]$, for $\mathbf{x}_{k,q}$ are selected using the FLOO-CV criterion. Using the optimal pair of parameters $[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}, l_{q,\mathrm{opt}}]$, a JITL-RLSSVR model can be built and applied to make a prediction for $\mathbf{x}_{k,q}$. Generally, the duration of each batch is not equal. The parameter set S_{par} can be determined around $[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$ obtained from the last batch if $J_k > \max(J_1, ..., J_{k-1})$.

Step 3. Updating parameters after a batch: Denote $S_K = \{(\mathbf{x}_i,\mathbf{y}_i)_{i=1}^{J_K}\}$ as the total sample set obtained after the end of the kth batch, where $J_K = \sum_{i=1}^{k} J_i$ is the number of samples. For S_K, the new optimal parameters $[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$ can be obtained using the FLOO-CV criterion offline. Additionally, all of the optimal parameters $[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}]$ for the qth query of the current batch are preserved in the database for the next batch. Let $k \rightarrow k + 1$ and go to the step 2 for the (k + 1)th batch.

In summary, the batch-to-batch and query-to-query optimization flowchart of the JITL-RLSSVR soft sensor modeling method for batch processes is illustrated in Figure 3. At the end of the first batch, the initial optimal parameters $[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$ are obtained using the FLOO-CV criterion. After the end of the kth batch, the updated parameters $[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$ and $[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}]$ for the qth query of the current batch are all preserved in the database. These parameters can be utilized for the next batch. Suppose the (k + 1)th batch is the longest duration until now. For the first to sixth samples of the (k + 1)th batch, as shown in Figure 3, the parameter set S_{par} can be determined around $[\gamma_{q,\mathrm{opt}}, \sigma_{q,\mathrm{opt}}, \rho_{q,\mathrm{opt}}]$. For the seventh and eighth samples of the (k + 1)th batch, S_{par} can be approximately determined around $[\gamma_{\mathrm{opt}}, \sigma_{\mathrm{opt}}, \rho_{\mathrm{opt}}]$. Besides, S_{par} should be defined larger because these two samples may be different from previous samples. Then, using a similar strategy, the JITL-RLSSVR models for each query can be built, respectively.

Note that if the batch-to-batch variations caused by changing operating conditions are very large, for data-driven modeling methods, they can only accommodate the differences to some extent. A common situation is that batch processes have unequal lengths. In this situation, the SF between the query sample and samples in the data set may be small (e.g., the eighth sample of the (k + 1)th batch shown in Figure 3). For JITL modeling methods, very similar samples for modeling may not be enough because the query sample is much different from previous samples. Therefore, the previous JITL soft sensor models[49,50] with fixed parameters should be improved to adapt batch-to-batch variations. Otherwise, they may degrade or show unstable prediction performances.

Additionally, the modeling method can be extended for multistage/multiphase batch processes. Using the process knowledge, a batch operation can be divided into various stages approximately. Then, the JITL model can be established based on the similar samples and updated parameters obtained in the same stage of previous batches because the stage-to-stage similarity in different batches is utilized.

In traditional JITL modeling methods,[42−48] once the prediction is obtained, the JITL model and any intermediate results are always discarded. Different from that, in the proposed modeling mechanism, the model parameters can be updated based on the optimal parameters in the previous batches because the batch-to-batch similarity is utilized. This can be regarded as a rough adjusting step for initial optimization. On the other hand, the batch-to-batch variation is also taken into consideration to build a more suitable JITL-RLSSVR soft sensor model for each query sample. This can be regarded as a detailed adjusting step for further query-to-query optimization. Based on the modeling procedure, the JITL-RLSSVR model can be constructed just from the second batch. With batch-to-batch and query-to-query optimization of model parameters, the main characteristics of batch processes can be captured and batch-to-batch variations can be reduced. Therefore, the performance of the JITL-RLSSVR soft sensor model can be guaranteed.

3. JITL-RLSSVR SOFT SENSOR FOR BATCH PROCESSES: SIMULATION AND DISCUSSION

3.1. Streptokinase Fed-Batch Fermentation Process. Nowadays, genetically engineered microorganisms have become important vehicles for the production of valuable biomolecules. In recombinant technology, the host microorganism is forced to overproduce the desired metabolite by infecting it with multiple copies of the plasmids. However, cells always try to get rid of this burden and consequently the fermentation may run away with plasmid-free cells.[32,55] Therefore, it becomes necessary to strictly monitor and control these cells, which can be achieved by developing soft sensors for the online estimation of the concentrations and product quality variables.[1,2,9,32,55] Accordingly, this study aims to develop JITL-RLSSVR-based soft sensors for predicting the concentrations of the active biomass and recombinant protein in the streptokinase (SK) fed-batch fermentation process.

For many fermentation processes, an efficient simulator can serve as a useful tool for not only understanding the effects of a variety of operational variables on process dynamics but also for utilizing it for research and educational activities in process modeling, monitoring, control, and optimization.[56] Birol et al.[56] developed a useful simulator named “PenSim”, which has been utilized for process monitoring and modeling by many researchers.[3,5,7−9,18,49]

Based on the SK simulator (see Supporting Information), a total of 20 batches with 1-h sampling intervals of process data have been generated for study. Different from the previous study with equal batch lengths,[32,49] the batch durations of the SK process here are uneven. To show the batch-to-batch variation, the process initial conditions and kinetic parameter values of each batch are different. Additionally, the feeding strategy is changed batch-to-batch to simulate the real-life process scenario. Finally, all the process variables are added with an amplitude of 4−6% Gaussian noise. As an illustrative example, the active biomass and SK concentrations of the first 10 batches generated by the SK simulator are shown in Figure 4. The batch durations are uneven and batch-to-batch variations are exhibited, especially at the end of each batch.

In the following, two case studies are investigated. The simulation environment for both case studies is Matlab V2007b with CPU main frequency 2.4 GHz and 3.87 GB of memory. In section 3.2, the first 10 batches are modeled to test the effect of similarity for existing JITL methods. In section 3.3, all 20 batches are investigated to show the adaptive parameter selection characteristic of the JITL-RLSSVR model.

3.2. Effect of Similarity for JITL Methods. First, an illustrative example is investigated to demonstrate that a suitable choice of the similarity can achieve better performance than only utilization of the Euclidean distance. Also, the effect of CSF is investigated. Cheng and Chiu investigated the performance of similarity for a linear JITL model for a continuous process.[44] Different from that, the nonlinear JITL soft sensors for batch processes are investigated here. In the previous study, three existing JITL-based methods with the similarity of Euclidean distance, i.e., JITL-LSSVR, JITL-SVR, and JITL-PLS, were investigated.[49,50] Therefore, in this section, they are compared with improved methods with a more comprehensive SF combined with the distance and angle.

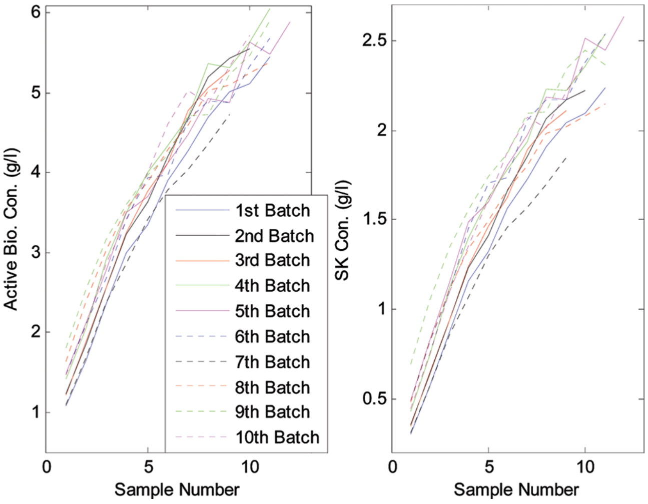

Figure 4. Active biomass and SK concentrations of the first 10 batches with 106 samples.

As aforementioned, the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp[-\|\mathbf{x}_i - \mathbf{x}_j\|/\sigma]$ $(\sigma > 0)$ and the polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$ $(d = 1, 2, 3, 4, ...)$ are two common kernel functions.[20−25] Both can show nonlinear modeling ability with suitable parameters.[20−36,54] However, the range of σ is much larger than that of d. Based on the literature,[20−25,54] the polynomial kernel can be utilized to model many processes with a certain degree in the range $1 \le d \le 5$. Therefore, the polynomial kernel is investigated in the following study to illustrate the implementation procedure and evaluate its characteristics of the JITL modeling methods.

There are seven secondary variables, mainly including three operating variables and their lagged values, to model soft sensors of the SK process, which can be referred to Desai et al.[32] The number of PLS components is selected by the common cumulative percentage variance criterion.[50] For the SK process, the component number of the PLS model is chosen as 4, which can explain most of the process data information. Here, the root-mean-square error (RMSE = $(\sum_{i=1}^{k}(\hat{y}_{i,m} - y_{i,m})^2/k)^{1/2}$) index is adopted to assess the prediction performance, where $\hat{y}_{i,m}$ denotes the prediction term of $y_{i,m}$.

Note that this case study is to show the effect of ρ on JITL-based soft sensors; thus all of the similar samples obtained using the SF criterion of eq 12 are adopted to establish local models for the query sample. Additionally, the parameters of $S_{\mathrm{par}} = [\gamma_S, d_S, \rho_S]$ are temporarily not optimized online for every query. Instead, only offline optimization is implemented at the end of each batch. That is, only the “step 3, updating parameters after a batch” in section 2.4 is implemented.

As an illustrative example, the comparison of JITL-LSSVR with different similarity weights ρ for online prediction of the active biomass and SK concentrations for the second to tenth batches is shown in Figure 5. The comparison results of the JITL-LSSVR, JITL-SVR, and JITL-PLS methods with different ρ values for the second to tenth batches are listed in Table 1. As can be shown in Table 1, for three JITL methods, the prediction performance can be generally improved by the combination of angle information into the similarity. For JITL-LSSVR, the prediction performance can be improved if ρ is

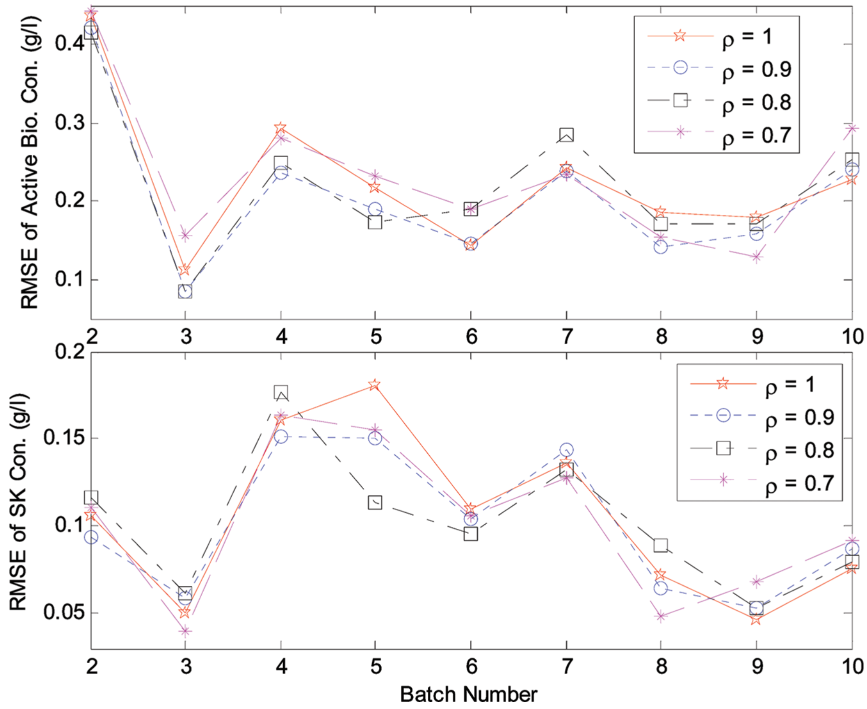

Figure 5. JITL-LSSVR with different similarity weights for online prediction of active biomass and SK concentrations of the second to tenth batches (RMSE comparison).

Table 1. Comparison Results of the Three JITL-Based Methods with Different ρ Values and Related Global Methods for the Second to Tenth Batches[a]

	JITL-Based Soft Sensors with Distance and Angle Based Similarity					
	JITL-LSSVR		JITL-SVR		JITL-PLS	
similarity weight ρ	X_a	S_t	X_a	S_t	X_a	S_t
ρ = 1[49,50]	0.242	0.115	0.250	0.111	0.307	0.145
ρ = 0.9	**0.223**	0.108	0.227	**0.104**	0.288	0.138
ρ = 0.8	0.236	0.108	0.229	0.108	0.295	0.142
ρ = 0.7	0.249	0.110	0.259	0.113	0.318	0.143
	Global Soft Sensors					
	LSSVR[23]		SVR[20]		PLS[18]	
similarity weight ρ	X_a	S_t	X_a	S_t	X_a	S_t
without ρ	0.304	0.145	0.312	0.138	0.353	0.166

[a] The best results with the smallest RMSE index are in boldface.

elaborately selected, e.g., ρ = 0.9 and ρ = 0.8. Besides, as shown in Figure 5, a suitable selection of ρ for every individual sample of each batch is different. For example, the fifth batch and eighth batch show batch-to-batch variations, which is shown in Figure 4, and consequently each batch has its suitable parameter.

As is also shown in Table 1, the prediction performances of JITL-LSSVR and JITL-SVR are comparative. Both JITL-LSSVR and JITL-SVR have shown better prediction performances than JITL-PLS has. The prediction results of three global modeling methods, i.e., LSSVR, SVR, and PLS, are also tabulated in Table 1. In contrast to the corresponding JITL methods, the three global modeling methods have less accurate prediction results. Among them, LSSVR and SVR have comparative prediction results. The conventional PLS-based soft sensor method, which has been widely utilized in chemical process modeling, is a little insufficient to reach good prediction performance for the SK process with so few samples. This is because the PLS model acquires less nonlinear information than the other two KL models. Generally, the modeling samples available for batch processes are fewer than those available for continuous processes. Therefore, for batch/fed-batch fermentation processes with limited modeling data, the nonlinear JITL-LSSVR and JITL-SVR models are more suitable than JITL-PLS.

As for the computational load, JITL-SVR is much larger than both JITL-LSSVR and JITL-PLS, which is also pointed out in the previous study.[49,50] This is because a quadratic programming optimization problem is required for the training process of traditional SVR.[20−25] Therefore, as can be drawn from Figure 5 and Table 1, the JITL-LSSVR method is the most suitable method for online modeling of batch processes of the three JITL methods, considering both the prediction performance and the computational load.

To construct the relevant data set S_{sim}, how many similar samples should be chosen is important. If fewer samples are

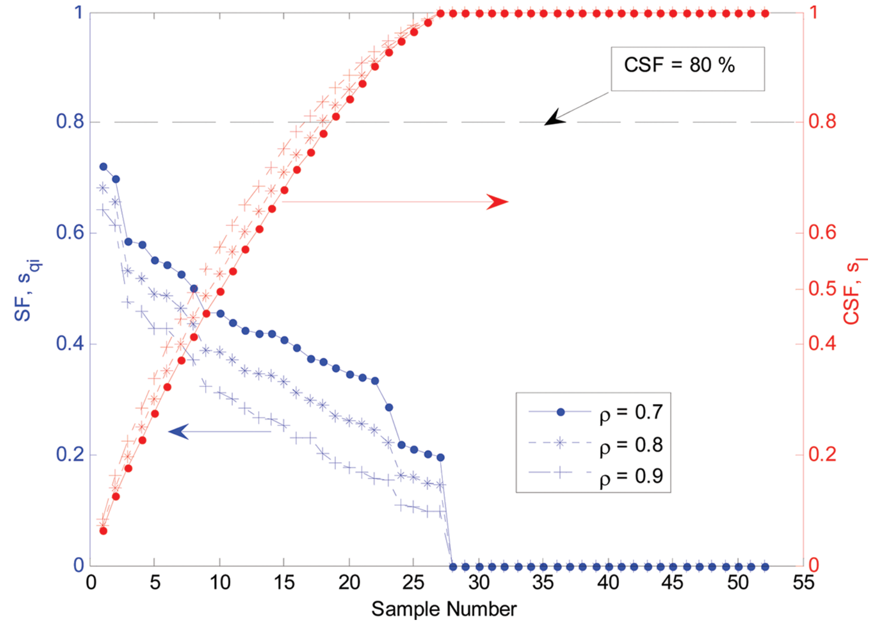

Figure 6. Effect of similarity weight on SF and CSF indices (for the last query sample of the fifth batch).

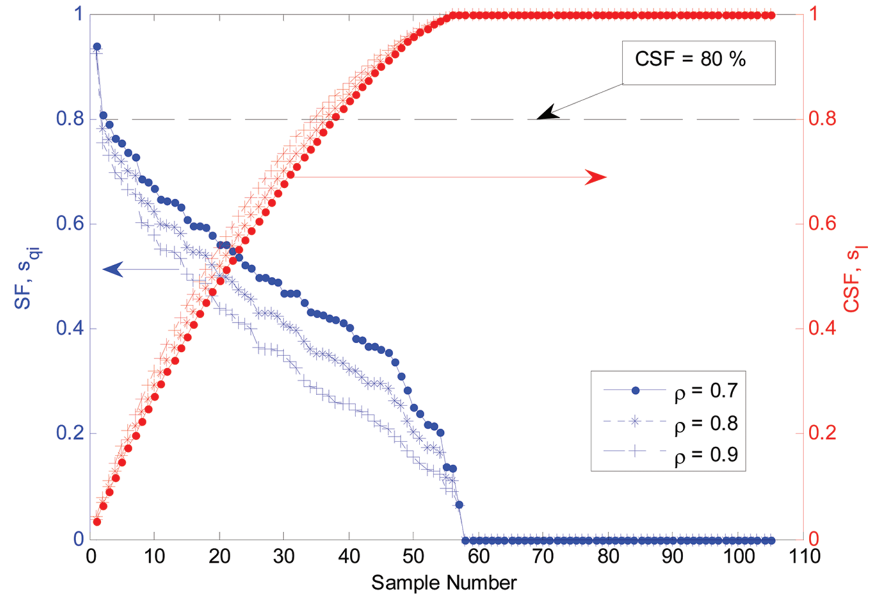

Figure 7. Effect of similarity weight on SF and CSF indices (for the last query sample of the tenth batch).

Table 2. Comparison of Characteristics of JITL-RLSSVR, JITL-LSSVR, and JITL-SVR Methods

	selection strategy of model parameters		
methods	$[\gamma, d]$ or $[\gamma, \sigma]$	ρ	l
JITL-RLSSVR	FLOO-CV-based online selection for every query	online selection for everyquery	online selection using the CSF criterion with $0.8 \le s_l \le 1$ for every query
JITL-LSSVR	offline optimization	ρ = 0.9	using the SF criterion with $l_{\max}$
JITL-SVR	offline optimization for every batch	ρ = 0.9	using the SF criterion with $l_{\max}$

Table 3. Comparison of JITL-RLSSVR, JITL-LSSVR, and JITL-SVR Methods for Online Prediction of Active and Product Concentrations of the SK Process from the Second to 20th Batches[a]

	RMSE		RE (%)		MAE	
methods	X_a	S_t	X_a	S_t	X_a	S_t
JITL-RLSSVR	**0.174**	**0.095**	**8.1**	**11.1**	**0.681**	**0.275**
JITL-LSSVR	0.208	0.111	9.3	11.8	0.856	0.344
JITL-SVR	0.209	0.107	9.1	11.4	0.822	0.290

[a] The best results with the smallest error index are in boldface.

selected, the modeling information is insufficient. Otherwise, if more samples are chosen, the excessive computational load may be burdened. The effects of different similarity weights on SF and CSF indices for the last query samples of the fifth batch and the tenth batch are shown in Figures 6 and 7, respectively.

As is shown in Figure 6, for building the JITL model of the last query sample of the fifth batch, at most 28 similar samples

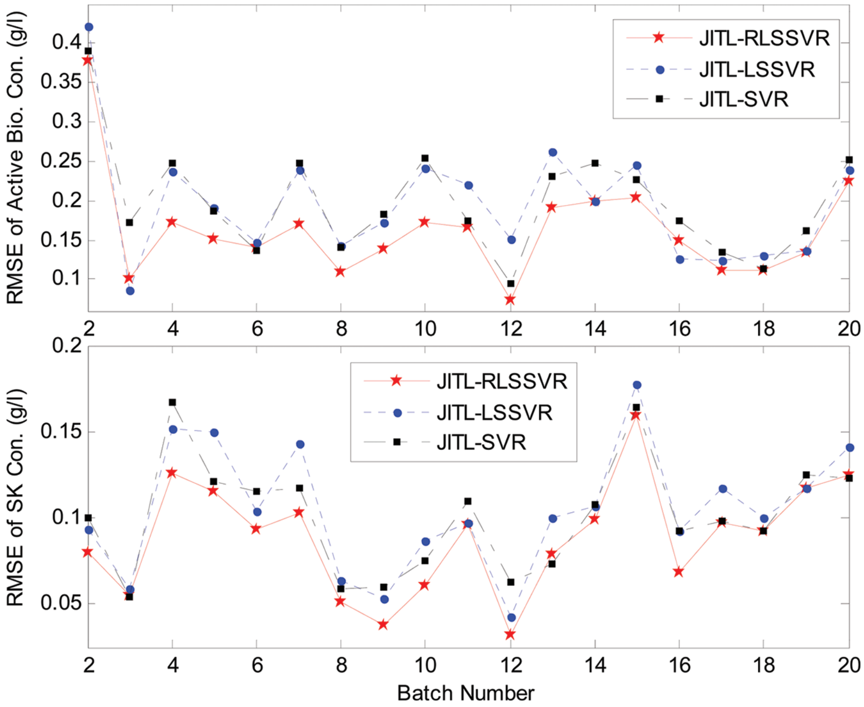

Figure 8. RMSE comparison of JITL-RLSSVR, JITL-LSSVR, and JITL-SVR methods for online prediction of active biomass and SK concentrations of the second to 20th batches.

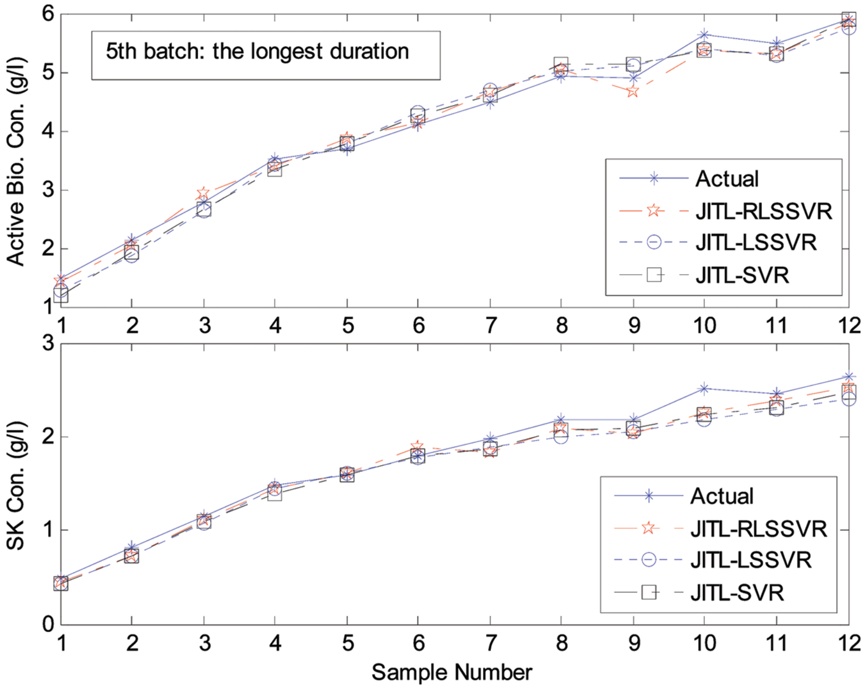

Figure 9. Comparison of JITL-RLSSVR, JITL-LSSVR, and JITL-SVR methods for online prediction of active biomass and SK concentrations of the fifth batch.

can be obtained from 52 samples using the SF criterion. Similarly, as is shown in Figure 7, at most 58 similar samples can be obtained from 105 samples. This indicates that the similar samples for different batches are generally not the same. Consequently, the range of $[l_{\min}, l_{\max}]$ is difficult to determine for different batches. Also, using a fixed value of similar samples is infeasible.

As is shown in Figures 6 and 7, the maximal SF values and their distributions are different. For the last query samples of the fifth batch, the maximal SF value is less than 0.8 and only 2/52 (about 4%) of the samples are larger than 0.6. Different from that, for the last query samples of the tenth batch, the maximal SF value is larger than 0.9 and about 10% of the samples are larger than 0.6. This is because the fifth batch is the longest batch and shows characteristics different from those of the previous four batches, which are shown in Figure 4. Therefore, as can be drawn from the definition of SF in eq 12

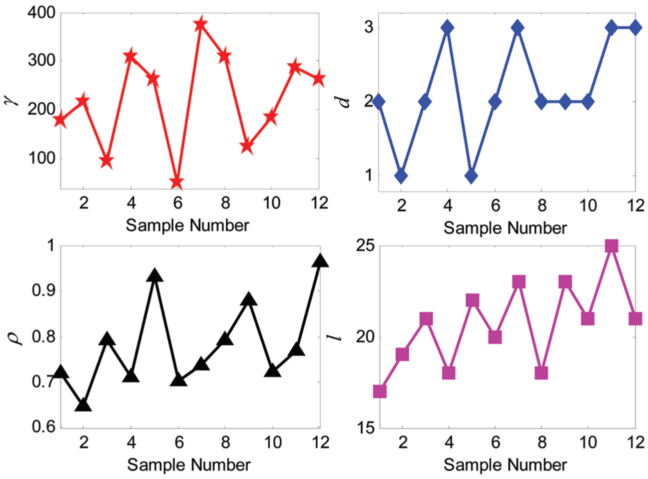

Figure 10. Modeling process of JITL-RLSSVR with adaptive parameter selection strategy of the fifth batch.

and the results in Figures 6 and 7, the SF index focuses on the similarity of the individual sample.

In contrast to SF, the proposed CSF criterion denotes the cumulative contributions of the most similar samples and thus seems more reasonable for batch processes. As is shown in Figures 6 and 7, for example, more than half of the similar samples can be absorbed to construct the JITL model with $s_l = 0.8$. Additionally, the range of $[l_{\min}, l_{\max}]$ can be selected adaptively by setting $0.8 \leq s_l \leq 1$ for different batches.

3.3. Adaptive Parameter Selection for JITL-RLSSVR Model. In this section, the adaptive parameter selection strategy proposed in section 2 is implemented to demonstrate the effect of JITL-RLSSVR and meanwhile to compare with two existing JITL kernel methods aforementioned, i.e., JITL-LSSVR and JITL-SVR, without online optimization of model parameters. Here, all 20 batches are investigated to make a justifiable comparison of performances. In addition to the RMSE index, two more performance indices are considered, i.e., relative RMSE (RE = $[\sum_{i=1}^{k}((\hat{y}_{i,m} - y_{i,m})/y_{i,m})^2/k]^{1/2}$) and maximal absolute error (MAE = max $|\hat{y}_{i,m} - y_{i,m}|$, for $i = 1, ..., k$), where $\hat{y}_{i,m}$ denotes the prediction term of $y_{i,m}$.

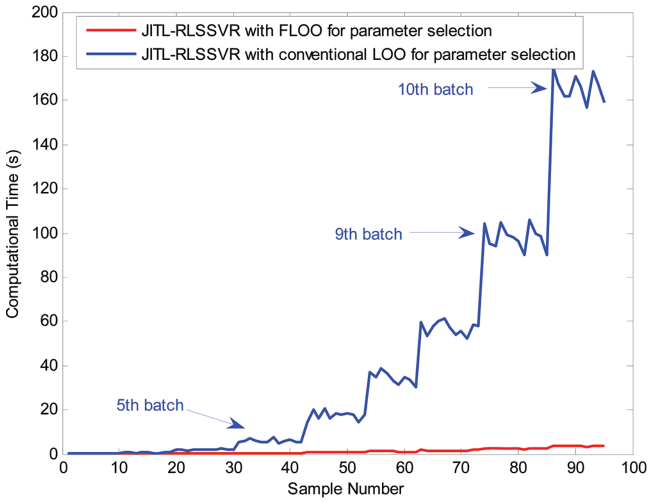

Figure 12. Comparison of the computational time of JITL-RLSSVR with FLOO and conventional LOO approaches for parameter selection for online modeling of the second to tenth batches.

As analyzed previously, the range of $[l_{\min}, l_{\max}]$ for all the batches can be chosen by setting $0.8 \leq s_l \leq 1$. Also, the value of Δl is selected as 2 to reduce the computational load. The initial optimal parameters $[\gamma_{\text{opt}}, d_{\text{opt}}, \rho_{\text{opt}}]$ can be determined using the modeling samples of the first batch. From the first query sample of the second batch, the online modeling procedure starts and

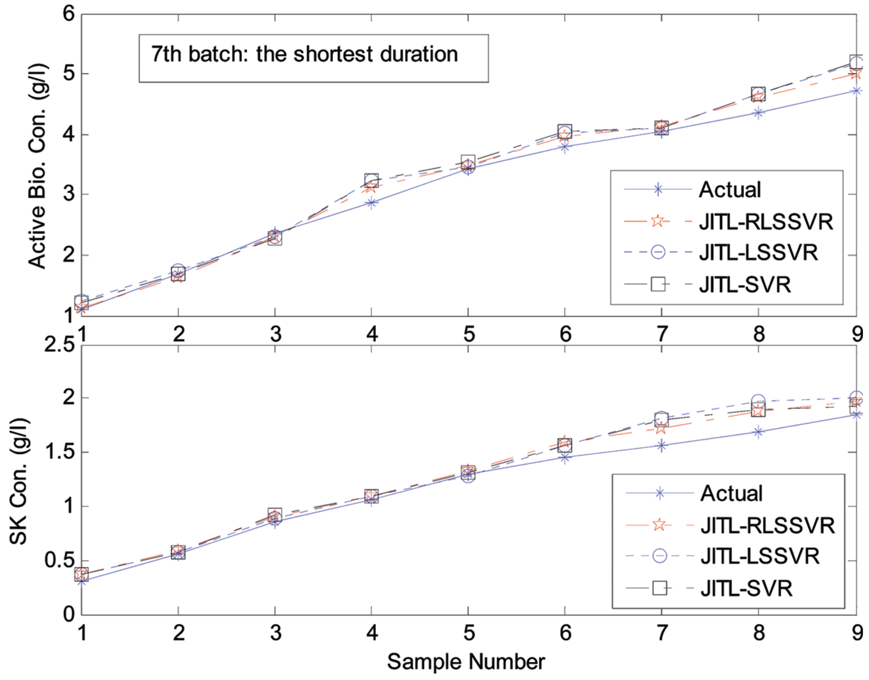

Figure 11. Comparison of JITL-RLSSVR, JITL-LSSVR, and JITL-SVR methods for online prediction of active biomass and SK concentrations of the seventh batch.

the parameter set $S_{\text{par}} = [\gamma_S, d_S, \rho_S]$ can be updated automatically:

$$S_{\text{par}} = [\gamma_S, d_S, \rho_S] = \begin{cases} 0.8\gamma_{\text{opt}}, \gamma_{\text{opt}}, 1.2\gamma_{\text{opt}} \\ \max(d_{\text{opt}} - 1, 1), d_{\text{opt}}, d_{\text{opt}} + 1 \\ 0.9\rho_{\text{opt}}, \rho_{\text{opt}}, \min(1.2\rho_{\text{opt}}, 1) \end{cases} \quad (27)$$

Consequently, the adaptive parameter selection strategy can be implemented to select the optimal parameters $[\gamma_{q,\text{opt}}, d_{q,\text{opt}}, \rho_{q,\text{opt}}, l_{q,\text{opt}}]$ in the range of S_{par} to establish the JITL-RLSSVR model for $\mathbf{x}_{k,q}$ using "step 2, batch-to-batch online optimization" described in section 2.4.

The main characteristics of thee methods for parameter selection are tabulated in Table 2. Correspondingly, the detailed prediction performance indices of three JITL methods for all the batches are tabulated in Table 3. As an illustrative example, the RMSE comparison of three JITL methods, i.e., JITL-RLSSVR, JITL-LSSVR, and JITL-SVR, for online prediction of the active biomass and SK concentrations for each batch (from the second to the 20th batches) is shown in Figure 8. From all the results shown in Figure 8 and Table 3, one can conclude that JITL-RLSSVR with adaptive parameter selection can achieve better prediction performance than both JITL-LSSVR and JITL-SVR without online optimization of model parameters for every batch.

Actually, batch processes always have uneven durations. In this case study, the fifth batch has the longest duration and shows batch-to-batch variation from previous batches. Online prediction comparison results of this batch using JITL-RLSSVR, JITL-LSSVR, and JITL-SVR are shown in Figure 9. The detailed modeling procedure with adaptive parameter selection of $[\gamma_{q,\text{opt}}, d_{q,\text{opt}}, \rho_{q,\text{opt}}, l_{q,\text{opt}}]$ for every query sample is shown in Figure 10. The optimal parameters using the FLOO-CV criterion can be adjusted query to query. Consequently, due to the online optimization of model parameters, the JITL-RLSSVR method can improve the prediction performance for these query samples. As for the shortest batch, i.e., the seventh batch, online prediction comparison results of this batch using three JITL methods are shown in Figure 11. During the end of the batch, the prediction performance can also be enhanced using online optimization of model parameters.

At last, the comparison of computational time of two JITL-RLSSVR methods is also investigated. One has adopted the FLOO-CV strategy and the other is still using the traditional LOO-CV strategy for parameter selection. The comparison results of computational time are shown in Figure 12. For online modeling of a query of the tenth batch with a similar set of about 60 samples (also shown in Figure 7), the JITL-RLSSVR method with FLOO-CV only needs about 3.5 s. However, without FLOO-CV, the modeling time is about 170 s. If the modeling set is larger, the computational time saved will be much more. Therefore, as can be drawn from Figure 12, the FLOO-CV strategy can improve the computational efficiency greater than traditional LOO-CV for parameter selection.

4. CONCLUSIONS

Traditional global soft sensors are insufficient for modeling of batch processes since most of them have no steady-state operating point and often exhibit batch-to-batch variations. This paper has addressed the subject of developing a suitable nonlinear JITL method for online modeling of batch processes. A JITL-RLSSVR approach is proposed and applied to construct soft sensors for batch processes. Its distinguished characteristics can be summarized in three main aspects:

1. A CSF criterion is presented to describe the similarity and select the relevant samples for the query sample.
2. The parameters of the nonlinear JITL-RLSSVR model can be determined adaptively using an efficient FLOO strategy with low computational load, resulting in little stress on the model parameter tuning.
3. The soft sensor implement algorithm for batch processes has been developed by taking both the batch-to-batch similarity and variation characteristics into consideration to make the modeling algorithm more applicable.

The obtained results show that JITL-RLSSVR generally outperforms existing global methods and related JITL methods without online optimization of model parameters. Therefore, the benefits of better prediction performance and practical implement procedure make JITL-RLSSVR suitable for online modeling of batch processes. Finally, as can be pointed out, the method can be further improved to adapt itself to larger batch-to-batch variation. It can be extended using different kernel functions combined with prior knowledge to better describe process characteristics. Additionally, its application to multistage/multiphase batch processes is our future research.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The phenomenological model of the SK fed-batch fermentation process is given in eqs S1−S8. The nominal kinetic parameter values and the initial conditions of the SK process are listed in Tables S1 and S2.[32,55] A graphical user interface of the SK fed-batch fermentation simulator is shown in Figure S1. The detailed simulation settings of the SK simulator for this study are listed in Table S3. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: +86-571-88871060. E-mail: yliuzju@zjut.edu.cn.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant 61004136) and Zhejiang Provincial Natural Science Foundation of China (Grant Y4100457). The authors would also like to thank the reviewers for helpful comments and valuable suggestions in greatly improving the quality of this paper.

■ ABBREVIATIONS

CSF = cumulative similarity factor
CV = cross-validation
FLOO = fast leave-one-out
JITL = just-in-time learning
KL = kernel learning
LOO = leave-one-out

LSSVR = least-squares support vector regression
MAE = maximal absolute error
MIMO = multi-input multi-output
MSR = multivariate statistical regression
NN = neural networks
PLS = partial least-squares
PCR = principal component regression
RPLS = recursive partial least-squares
RLSSVR = recursive LSSVR
RE = relative RMSE
RMSE = root-mean-square error
SF = similarity factor
SK = streptokinase
SVR = support vector regression

REFERENCES

(1) Dochain, D. State and parameter estimation in chemical and biochemical processes: a tutorial. *J. Process Control* **2003**, *13*, 801−818.

(2) Henson, M. A. Biochemical reactor modeling and control. *IEEE Control Syst. Mag.* **2006**, *26*, 54−62.

(3) Yao, Y.; Gao, F. R. A survey on multistage/multiphase statistical modeling methods for batch processes. *Annu. Rev. Control* **2009**, *33*, 172−183.

(4) Russell, S. A.; Kesavan, P.; Lee, J. H.; Ogunnaike, B. A. Recursive data-based prediction and control of batch product quality. *AIChE J.* **1998**, *44*, 2442−2458.

(5) Zhao, C. H.; Wang, F. L.; Mao, Z. Z.; Lu, N. Y.; Jia, M. X. Quality prediction based on phase-specific average trajectory for batch processes. *AIChE J.* **2008**, *54*, 693−705.

(6) Mangold, M.; Bück, A.; Schenkendorf, R.; Steyer, C.; Voigt, A.; Sundmacher, K. Two state estimators for barium sulfate precipitation in a semi-batch reactor. *Chem. Eng. Sci.* **2009**, *64*, 646−660.

(7) Facco, P.; Bezzo, F.; Barolo, M. Nearest-neighbor method for the automatic maintenance of multivariate statistical soft sensors in batch processing. *Ind. Eng. Chem. Res.* **2010**, *49*, 2336−2347.

(8) Undey, C.; Cinar, A. Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Syst. Mag.* **2002**, *22*, 40−52.

(9) Chen, J. H.; Song, C. M.; Hsu, T. Y. On-line monitoring of batch processes using IOHMM based MPLS. *Ind. Eng. Chem. Res.* **2010**, *49*, 2800−2811.

(10) Alford, J. S. Bioprocess control: Advances and challenges. *Comput. Chem. Eng.* **2006**, *30*, 1464−1475.

(11) Lee, J. S.; Lee, K. S. Iterative learning control applied to batch processes: An overview. *Control Eng. Pract.* **2007**, *15*, 1306−1318.

(12) Nagy, Z. K.; Braatz, R. D. Robust nonlinear model predictive control of batch processes. *AIChE J.* **2003**, *49*, 1776−1786.

(13) Flores-Cerrillo, J.; MacGregor, J. F. Within-batch and batch-to-batch inferential-adaptive control of semibatch reactors: a partial least squares approach. *Ind. Eng. Chem. Res.* **2003**, *42*, 3334−3345.

(14) Xiong, Z.; Zhang, J. A batch-to-batch iterative optimal control strategy based on recurrent neural network models. *J. Process Control* **2005**, *15*, 11−21.

(15) Bonvin, D.; Srinivasan, B.; Hunkeler, D. Control and optimization of batch processes. *IEEE Control Syst. Mag.* **2006**, *26*, 34−45.

(16) Fortuna, L.; Graziani, S.; Rizzo, A.; Xibilia, M. G. *Soft Sensors for Monitoring and Control of Industrial Processes*; Springer-Verlag: New York, 2007.

(17) Himmelblau, D. M. Accounts of experiences in the application of artificial neural networks in chemical engineering. *Ind. Eng. Chem. Res.* **2008**, *47*, 5782−5796.

(18) Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795−814.

(19) Kano, M.; Ogawa, M. The state of the art in chemical process control in Japan: Good practice and questionnaire survey. *J. Process Control* **2010**, *20*, 969−982.

(20) Vapnik, V. N. *The Nature of Statistical Learning Theory*; Springer-Verlag: New York, 1995.

(21) Schölkopf, B.; Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, 2002.

(22) Taylor, J. S.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, U.K., 2004.

(23) Suykens, J. A. K.; Van Gestel, T.; De Brabanter, J.; De Moor, B.; Vandewalle, J. *Least Squares Support Vector Machines*; World Scientific: Singapore, 2002.

(24) Suykens, J. A. K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293−300.

(25) Suykens, J. A. K.; De Brabanter, J.; Lukas, L.; Vandewalle, J. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing* **2002**, *48*, 85−105.

(26) Yan, W. W.; Shao, H. H.; Wang, X. F. Soft sensing modeling based on support vector machine and Bayesian model selection. *Comput. Chem. Eng.* **2004**, *28*, 1489−1498.

(27) Lee, D. E.; Song, J. H.; Song, H. O.; Yoon, E. S. Weighted support vector machine for quality estimation in the polymerization process. *Ind. Eng. Chem. Res.* **2005**, *44*, 2101−2105.

(28) Gandhi, A. B.; Joshi, J. B.; Jayaraman, V. K.; Kulkarni, B. D. Data-driven dynamic modeling and control of a surface aeration system. *Ind. Eng. Chem. Res.* **2007**, *46*, 8607−8613.

(29) Chitralekha, S. B.; Shah, S. L. Application of support vector regression for developing soft sensors for nonlinear processes. *Can. J. Chem. Eng.* **2010**, *88*, 696−709.

(30) Zhang, X.; Yan, W. W.; Shao, H. H. Nonlinear multivariate quality estimation and prediction based on kernel partial least squares. *Ind. Eng. Chem. Res.* **2008**, *47*, 1120−1131.

(31) Dubey, A.; Realff, M. J.; Lee, J. H.; Schork, F. J.; Butte, A.; Olle, B.; Kizer, L. E. Modeling and inferential control of the batch acetylation of cellulose. *AIChE J.* **2006**, *52*, 2149−2160.

(32) Desai, K.; Badhe, Y.; Tambe, S. S.; Kulkarni, B. D. Soft-sensor development for fed-batch bioreactors using support vector regression. *Biochem. Eng. J.* **2006**, *27*, 225−239.

(33) Li, Y. F.; Yuan, J. Q. Prediction of key state variables using support vector machines in bioprocesses. *Chem. Eng. Technol.* **2006**, *29*, 313−319.

(34) Wang, J. L.; Yu, T.; Jin, C. Y. On-line estimation of biomass in fermentation process using support vector machine. *Chin. J. Chem. Eng.* **2006**, *14*, 383−388.

(35) Jain, P.; Rahman, I.; Kulkarni, B. D. Development of a soft sensor for a batch distillation column using support vector regression techniques. *Chem. Eng. Res. Des.* **2007**, *85*, 283−287.

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

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

(38) Wang, X.; Kruger, U.; Irwin, G. W. Process monitoring approach using fast moving window PCA. *Ind. Eng. Chem. Res.* **2005**, *44*, 5691−5702.

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

(40) Liu, X. Q.; Kruger, U.; Littler, T.; Xie, L.; Wang, S. Q. Moving window kernel PCA for adaptive monitoring of nonlinear processes. *Chemom. Intell. Lab. Syst.* **2009**, *96*, 132−143.

(41) Liu, Y.; Hu, N. P.; Wang, H. Q.; Li, P. Soft chemical analyzer development using adaptive least-squares support vector regression with selective pruning and variable moving window size. *Ind. Eng. Chem. Res.* **2009**, *48*, 5731−5741.

(42) Atkeson, C. G.; Moore, A. W.; Schaal, S. Locally weighted learning. *Artif. Intell. Rev.* **1997**, *11*, 11−73.

(43) Bontempi, G.; Birattari, M.; Bersini, H. Lazy learning for local modeling and control design. *Int. J. Control* **1999**, *72*, 643−658.

(44) Cheng, C.; Chiu, M. S. A new data-based methodology for nonlinear process modeling. *Chem. Eng. Sci.* **2004**, *59*, 2801−2810.

(45) Cheng, C.; Chiu, M. S. Nonlinear process monitoring using JITL-PCA. *Chemom. Intell. Lab. Syst.* **2005**, *76*, 1−13.

(46) Pan, T. H.; Li, S. Y.; Cai, W. J. Lazy learning-based online identification and adaptive PID control: a case study for CSTR process. *Ind. Eng. Chem. Res.* **2007**, *46*, 472−480.

(47) Fujiwara, K.; Kano, M.; Hasebe, S.; Takinami, A. Soft-sensor development using correlation-based just-in-time modeling. *AIChE J.* **2009**, *55*, 1754−1765.

(48) Fujiwara, K.; Kano, M.; Hasebe, S. Development of correlation-based clustering method and its application to software sensing. *Chemom. Intell. Lab. Syst.* **2010**, *101*, 130−138.

(49) Liu, Y.; Wang, H. Q.; Li, P. Local least squares support vector regression with application to online modeling for batch processes. *J. Chem. Ind. Eng. (China)* **2007**, *58*, 2846−2851.

(50) Ge, Z. Q.; Song, Z. H. A comparative study of just-in-time-learning based methods for online soft sensor modeling. *Chemom. Intell. Lab. Syst.* **2010**, *104*, 306−317.

(51) Cawley, G. C.; Talbot, N. L. C. Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. *J. Mach. Learn. Res.* **2007**, *8*, 841−861.

(52) Cawley, G. C.; Talbot, N. L. C. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks* **2004**, *17*, 1467−1475.

(53) Golub, G. H.; van Loan, C. F. *Matrix Computations*, 3rd ed.; The John Hopkins University Press: Baltimore, MD, 1996.

(54) Liu, Y.; Wang, H. Q.; Yu, J.; Li, P. Selective recursive kernel learning for online identification of nonlinear systems with NARX form. *J. Process Control* **2010**, *20*, 181−194.

(55) Patnaik, P. R. A heuristic approach to fed-batch optimization streptokinase fermentation. *Bioprocess Eng.* **1995**, *13*, 109−112.

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