Chemometrics and Intelligent Laboratory Systems 144 (2015) 108–121

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Online soft sensor design using local partial least squares models with adaptive process state partition

Weiming Shao [a], Xuemin Tian [a], Ping Wang [a], Xiaogang Deng [a], Sheng Chen [b,c,*]

[a] College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China
[b] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[c] King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 28 November 2014
Received in revised form 7 April 2015
Accepted 8 April 2015
Available online 17 April 2015

Keywords:
Soft sensor
Process state partition
Local models
Statistical hypothesis testing
Partial least squares

ABSTRACT

We propose a soft sensing method using local partial least squares models with adaptive process state partition, referring to as the LPLS-APSP, which is capable of effectively handling time-varying characteristics and nonlinearities of processes, the two major adverse effects of common industrial processes that cause low-performance of soft sensors. In our proposed approach, statistical hypothesis testing is employed to adaptively partition the process state into the unique local model regions each consisting of certain number of consecutive-time data samples, and partial least squares is adopted to construct local models. Advantages of this adaptive strategy are that the number of local models does not need to be pre-defined and the local model set can be augmented online without retraining from scratch. Moreover, to improve the prediction accuracy, a novel online model adaptation criterion is proposed, which not only takes the current process dynamics into account, but also enables mining the information contained in the neighborhood of the query sample. The guidelines for tuning the model parameters are also presented. The LPLS-APSP scheme is applied to develop the dynamic soft sensors for a simulated continuous stirred tank reactor and a real industrial debutanizer column, and the results obtained demonstrate the effectiveness of this proposed approach, in comparison to several existing state-of-the-art methods, for online soft sensor design.

1. Introduction

Hardware sensors are ubiquitous in industrial plants to measure process variables in order to deliver data for process monitoring and control [1,2]. Many important quality-related process variables, such as product concentration and octane number, are very difficult to obtain. Although hardware based analysers can provide online measurements for these difficult-to-measure variables, they are expensive and have several limitations, including long analyzing delay, inaccuracy and difficult to maintain, which make them an unsatisfactory choice for industrial process sensing applications [3,4]. Moreover, some key process variables, for example, the melt index of polypropylene, can be mostly obtained through laboratory analysis [4,5], which may take hours to complete. Infrequent and inaccurate estimation of these key process variables may result in poor control performances, huge production losses and even cause safety hazards [6].

Fortunately, the above-mentioned problems may be eliminated by applying soft sensors, which are easy to maintain and can deliver real-time estimates of those hard-to-measure primary process variables with low cost. Compared to the first principle based modeling, a procedure that is typically time-consuming and costly, data-driven soft sensors are widely applied because they can be quickly developed without the need to first gain the substantial insights into the complex mechanisms of industrial plants. Therefore, a variety of algorithms have been proposed to develop data-driven soft sensors for industrial processes, such as multivariate statistical regression (MSR) techniques that include principal component analysis (PCA) [7,8], partial least squares (PLS) [9,10], artificial neural networks (ANN) [11,12], and support vector machines (SVMs) [13,14]. However, the performance of soft sensors often deteriorate over time due to a predominant difficulty encountered in most industrial processes, namely, time-varying characteristics caused for example by catalyst deactivation, mechanical aging, etc. [4,15].

Consequently, online updating soft sensor models is essential in practice. Moving window and recursive methods are commonly adopted to adapt soft sensors to new process dynamics [16–20]. But it is well known that these adaptive methods have difficulty in coping with abrupt changes such as change of set point value, unless sufficient samples from the new operational condition have been collected. Most recursive methods belong to the category of fitting a single 'global' model, which may fail to perform well owing to the strong nonlinearities

* Corresponding author. Tel.: +44 23 8059 6660.
E-mail addresses: swmupc@163.com (W. Shao), tianxm@upc.edu.cn (X. Tian), wangpingupc@163.com (P. Wang), dengxg2002@gmail.com (X. Deng), sqc@ecs.soton.ac.uk (S. Chen).

http://dx.doi.org/10.1016/j.chemolab.2015.04.003
0169-7439/

existed in most complex industrial processes [4,21], as a single global model has the difficulties in adequately representing all the operating points of an industrial process. Compared to the global model based learning, local learning based soft sensors that employ the philosophy of 'divide and rule' construct several local models, each of which serves as the expert for one specific operating region of the process. The time-varying characteristics, including abrupt changes, can also be handled when the local models or their contributions to the query sample are updated online.

The first key task in the development of soft sensors under the local learning framework is to divide the process state into the local model regions by partitioning the entire data set into an appropriate number of the data subsets so that the local models can be constructed on the corresponding local regions or data subsets. In the soft sensing field, clustering based methods, such as fuzzy C-means (FCM) [22–25] and expectation maximization (EM) based finite mixture models [10, 26–28], are commonly used solutions to this problem. However, appropriate number of clusters is not easy to determine and it is difficult to add new clusters online for coping with the time-varying characteristics of a process. Although the latter issue can be alleviated by updating the offline constructed local models using recursive or moving window techniques [27,28], it is more desirable to construct a new local model upon receiving the newly emerged data samples located in the previously unexploited region of the sample space. Besides, the EM based methods may encounter difficulty when the dimensionality of the process variables is very high, because in such a scenario accurate estimation of probability density function demands considerable amount of training samples. Just-in-time learning (JITL) [29–34] is another popular technique for partitioning the process state into the local data subsets, and it also has the potential to address both process nonlinearities and time-varying characteristics. However, the performance of JITL based methods is not always satisfactory because the correlation between the process variables is ignored [21].

Recently, some approaches were proposed for identifying the local models, each of which is constructed based on a data subset consisting of some consecutive time samples. The advantages of these methods are that the correlation between process variables is considered and the number of local models does not need to be pre-defined. In [21,35, 36], the process state is partitioned by a moving window with a fixed length. Normally, high prediction accuracy requires a small moving window width, leading to a great amount of local models and a large online computational load. The work [37,38] proposed to repeatedly partition the entire dataset whenever an adaptation is triggered as newly measured samples are accumulated, and thus this approach may be time consuming. In the incremental local learning soft sensing algorithm (ILLSSA) [39], an adaptive way was presented for splitting the entire data set into the local consecutive-time-sample regions based on the *t*-test. This method considers the relationship of modeling function and process characteristics better than the above-mentioned other two methods. However, the variance in the denominator of the *T* statistic may cause negative effects on the performance of the ILLSSA, and as a result, the null hypothesis may remain valid even when the process characteristics have changed. It can be seen that adaptively and appropriately partitioning the process state into the local consecutive-time-sample regions is a promising but challenging task.

The other critical part of developing local learning based soft sensors is to formulate the estimated value of the target variable based on the constructed local models. There are two strategies to fulfill such a task. The first strategy computes the final model output as the weighted combination of all the local models' outputs. The weights can be either equal for all the local models [31] or different for different local models [22,23, 25–27,39]. The second strategy is to switch to different local models online according to some model adaptation criteria, such as fuzzy membership [24] or posterior probability [10,40,41]. However, both the fuzzy membership and the posterior probability are distance based similarity metrics, and they are not very suitable for the local models built from the consecutive-time-sample data. In the correlation-based just-in-time learning (CoJIT) [21], the correlation indexes among the process variables are utilized for local model adaptation, which improves the prediction performance, and this ensures that the CoJIT often outperforms the conventional JITL. However, it neglects the mapping relationship between the target variable and the secondary variables, which may sometimes lead to inappropriate model adaptation. Moreover, it requires the massive memory space for storing the loading matrices of the PCA models, which may not be available in some applications [42]. In the works [35,36], a similarity measure, which is constructed by the support vector data description (SVDD) and independent components (ICs), was proposed as the model adaptation criterion. But the online computational load of the quadratic programming introduced by the SVDD can be large. In the localized adaptive recursive PLS (LARPLS) [37] and the localized adaptive soft sensor (LASS) [38], the prediction error for the newest one sample was selected as the local model adaptation criterion. Although this approach overcomes the shortcomings of the CoJIT, the estimate produced may not be sufficiently accurate. The reason is simply because it too greedily pursues the error minimization for the single newest sample, while disregarding the information about the query sample. This drawback severely limits the generalization ability of the adapted local models. It can be seen that although the second strategy of computing the final model output is promising owing to its potential ability to enhance the adaptive performance of soft sensor models, many critical issues remain unsolved by the existing techniques.

Against the above background, in this paper, we propose an online soft sensor design which is capable of dealing effectively with the time-varying characteristics and nonlinearities encountered in most real-life industrial processes. As our proposed soft sensing method adopts the local learning framework that utilizes local partial least squares models with adaptive process state partition, it is referred to as the LPLS-APSP. Our soft sensor design also adopts the approach of adaptive switching to different local models online. In comparison to the existing adaptive soft sensing methods, however, our novel contributions are as follows.

- We develop an adaptive process state partition based on effective statistical hypothesis testing. Specifically, local model regions are adaptively defined, while redundant local models are automatically detected and discarded, both based on the χ^2-test and *t*-test, which account for the effects of the variance and mean of the predicted residuals, respectively.
- We propose a novel criterion for online model adaptation by exploiting the predicted error for the newest measured sample as well as mining the neighborhood information of the query sample, which significantly enhances the online performance of our LPLS-APSP based soft sensor.
- We conduct an extensive evaluation of the proposed LPLS-APSP soft sensor design using a simulated continuous stirred tank reactor and a real industrial debutanizer column, and demonstrate its superior performance over several existing benchmark online soft sensor designs.

This paper is structured as follows. Section 2 very briefly reviews the local PLS model. In Section 3, our proposed online LPLS-APSP based soft sensor design is detailed, which includes adaptive process state partition, online model adaptation and the design of the LPLS-APSP based soft sensor. In Section 4, two chemical processes are employed to demonstrate the superior performance of the LPLS-APSP soft sensor design over several existing state-of-the-art designs. In addition, in this section, how the design parameters of our proposed soft sensor influence the online sensing performance is thoroughly investigated and the guidelines for tuning these parameters are presented. Our conclusions and remarks for future work are given in Section 5.

2. Review of local partial least squares

Under the local learning framework, linear local models are normally constructed to avoid huge computational complexity. However, process data used for soft sensing are typically strongly co-linear which may be caused for example by the partial redundancy in the sensor arrangement, such as two adjacent temperature meters, and so on [1]. A commonly used way of eliminating data co-linearity is to transform the secondary variables onto the latent space by means of PLS, which has gained popularity and becomes a most preferable modeling method by practical engineers [43].

Let $\boldsymbol{x}_k = [x_{1,k}\, x_{2,k} \cdots x_{m,k}]^{\mathrm{T}} \in \mathbb{R}^m$ be the kth sample vector of the m secondary variables and $\boldsymbol{y}_k = [y_{1,k}\, y_{2,k} \cdots y_{p,k}]^{\mathrm{T}} \in \mathbb{R}^p$ be the kth sample vector of the p primary variables, respectively, where $(\cdot)^{\mathrm{T}}$ denotes the transpose operator. Assume that the process state has been partitioned into a number of local model regions, and each local model region is associated with a consecutive-time-sample time series data set $\{\boldsymbol{X}, \boldsymbol{Y}\}$, where $\boldsymbol{X} = [\boldsymbol{x}_1\, \boldsymbol{x}_2 \cdots \boldsymbol{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N \times m}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1\, \boldsymbol{y}_2 \cdots \boldsymbol{y}_N]^{\mathrm{T}} \in \mathbb{R}^{N \times p}$ are the input and output matrices with N samples, respectively. For notational simplicity, we have omitted the local model index. For example, for the lth local model, the associated data set is $\{\boldsymbol{X}^{(l)}, \boldsymbol{Y}^{(l)}\}$ with N_l samples but the index l is dropped.

In PLS modeling, the linear relationship between $\boldsymbol{Y}$ and $\boldsymbol{X}$ is described by $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{C}^{\mathrm{PLS}} + \boldsymbol{E}$, where all the data samples in $\boldsymbol{X}$ and $\boldsymbol{Y}$ have been mean-centered, and $\boldsymbol{C}^{\mathrm{PLS}} \in \mathbb{R}^{m \times p}$ is the regression coefficient matrix defined by $\boldsymbol{C}^{\mathrm{PLS}} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{+}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$ in which $(\cdot)^{+}$ denotes the generalized inverse operator, while $\boldsymbol{E} \in \mathbb{R}^{N \times p}$ is the residual matrix. Thus, the estimate of $\boldsymbol{Y}$ is defined by $\hat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{C}^{\mathrm{PLS}}$. PLS projects $\boldsymbol{X}$ and $\boldsymbol{Y}$ onto respective latent variables according to

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \boldsymbol{E}_X, \tag{1}$$

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}} + \boldsymbol{E}_Y, \tag{2}$$

where $\boldsymbol{T} = [\boldsymbol{t}_1\, \boldsymbol{t}_2 \cdots \boldsymbol{t}_A] \in \mathbb{R}^{N \times A}$ and $\boldsymbol{U} = [\boldsymbol{u}_1\, \boldsymbol{u}_2 \cdots \boldsymbol{u}_A] \in \mathbb{R}^{N \times A}$ are the score matrices, $\boldsymbol{P} = [\boldsymbol{p}_1\, \boldsymbol{p}_2 \cdots \boldsymbol{p}_A] \in \mathbb{R}^{m \times A}$ and $\boldsymbol{Q} = [\boldsymbol{q}_1\, \boldsymbol{q}_2 \cdots \boldsymbol{q}_A] \in \mathbb{R}^{p \times A}$ are the loading matrices, of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, while A is the number of the latent variables, and $\boldsymbol{E}_X \in \mathbb{R}^{N \times m}$ and $\boldsymbol{E}_Y \in \mathbb{R}^{N \times p}$ are the input and output residual matrices. The columns of the score matrix $\boldsymbol{T}$ are usually orthogonal to each other, but those of $\boldsymbol{U}$ are generally not.

Eqs. (1) and (2) represent the external relationship of the PLS model. The linear internal relationship of the PLS model is given by

$$\boldsymbol{U} = \boldsymbol{T}\boldsymbol{B} + \boldsymbol{F}, \tag{3}$$

where $\boldsymbol{B} = \mathrm{diag}\{b_1, b_2, \cdots, b_A\}$ is a diagonal matrix, and the regression weights $\{b_1, b_2, \cdots, b_A\}$ are determined by minimizing the residuals $\boldsymbol{F}$. The estimate of $\boldsymbol{Y}$ can alternatively be represented by $\hat{\boldsymbol{Y}} = \boldsymbol{T}\boldsymbol{B}\boldsymbol{Q}^{\mathrm{T}}$.

A PLS model can be computed by either the nonlinear iterative PLS (NIPALS) algorithm [44] or the kernel algorithm for PLS [45]. To improve the prediction performance, a dynamic PLS model can also be formulated by augmenting the input matrix with the past samples of the secondary variables. Moreover, in order to cope with time-varying characteristics, the PLS model can be recursively updated by "merging" the old model with the new data sample $\{\boldsymbol{x}_{\mathrm{new}}, \boldsymbol{y}_{\mathrm{new}}\}$. Specifically, the new regression coefficient matrix is updated from $\boldsymbol{C}^{\mathrm{PLS}}$ according to

$$\boldsymbol{C}^{\mathrm{PLS}}_{\mathrm{new}} = \left(\begin{bmatrix} \lambda\boldsymbol{P}^{\mathrm{T}} \\ \boldsymbol{x}_{\mathrm{new}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \lambda\boldsymbol{P}^{\mathrm{T}} \\ \boldsymbol{x}_{\mathrm{new}} \end{bmatrix} \right)^{+} \begin{bmatrix} \lambda\boldsymbol{P}^{\mathrm{T}} \\ \boldsymbol{x}_{\mathrm{new}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \lambda\boldsymbol{B}\boldsymbol{Q}^{\mathrm{T}} \\ \boldsymbol{y}_{\mathrm{new}} \end{bmatrix}, \tag{4}$$

where the forgetting factor $0 \le \lambda \le 1$ specifies the adaptation strength. A necessary condition to perform this recursive updating is that the number of the selected latent variables A is sufficiently large such that $\|\boldsymbol{E}_X\|_F$ is sufficiently small [19], where $\|\cdot\|_F$ denotes the matrix Frobenius norm.

3. Proposed online soft sensor design

The operations of the LPLS-APSP based online soft sensor can be viewed as consisting of "offline" operations and online operations. At the offline operation stage, the process state is adaptively partitioned into local model regions using statistical hypothesis testing and a new local PLS model is constructed upon the identified new local model region. Moreover, the newly added local model is checked with the existing local models using statistical hypothesis test, and if an existing local model is found to be very similar to this new local model, the existing old local model is discarded. Note that during this adaptive process state partitioning stage, the plant is operating online continuously to provide consecutive-time samples of the measured process variables. The term "offline" really means that the computations for this stage can be carried out offline. At the online operation stage, a query sample arises and the online soft sensor must response. In this online stage, model adaptation occurs based on some specific criterion and the selected local model is responsible for predicting the output of the query sample. Our proposed criterion for online model adaptation not only exploits the predicted error for the newest measured sample but also mines the neighborhood information of the query sample We now detail these two components of our proposed LPLS-APSP based online soft sensor design.

3.1. Adaptive process state partition based on statistical hypothesis testing

For the sake of illustrating the basic concepts as well as for the purpose of simplifying notations, we restrict the discussions to the single output case, namely, $p = 1$. A rational local model region should contain a period of consecutive-time samples during which the model has the same performance [39]. Our local model region partition approach is depicted in Fig. 1, which considers the variances of both first-order and second-order information in the measured process variables. Initially, a data window $\mathcal{W}_{\mathrm{ini}}$ is set with the W consecutive-time samples denoted by $\mathcal{W}_{\mathrm{ini}} = \{\boldsymbol{X}_{\mathrm{ini}}, \boldsymbol{Y}_{\mathrm{ini}}\}$, upon which a local expert or PLS model $\boldsymbol{f}_{\mathrm{ini}}$ has been constructed, where $\boldsymbol{X}_{\mathrm{ini}} \in \mathbb{R}^{W \times m}$, $\boldsymbol{Y}_{\mathrm{ini}} \in \mathbb{R}^{W \times 1}$ and $\boldsymbol{f}_{\mathrm{ini}}$ maps $\boldsymbol{X}_{\mathrm{ini}}$ onto the predictions of $\boldsymbol{Y}_{\mathrm{ini}}$. It is assumed that at this point the local model set consists of L local models $\{\boldsymbol{f}_l\}_{l=1}^{L}$ with $L \ge 1$, which indicates that the process state has been partitioned into the L local model regions which are represented by the L data subsets $\{\mathcal{W}_l\}_{l=1}^{L}$, each containing W consecutive-time samples. Without loss of generality, the local model $\boldsymbol{f}_L$ is identified at the previous local model region extraction, and we have $\mathcal{W}_{\mathrm{ini}} = \mathcal{W}_L$ and $\boldsymbol{f}_{\mathrm{ini}} = \boldsymbol{f}_L$.

Subsequently, the window is shifted one sample step ahead and a shifted window $\mathcal{W}_{\mathrm{sft}}$ is obtained with the data set $\mathcal{W}_{\mathrm{sft}} = \{\boldsymbol{X}_{\mathrm{sft}}, \boldsymbol{Y}_{\mathrm{sft}}\}$. For $\boldsymbol{Y}_{\mathrm{ini}}$ and $\boldsymbol{Y}_{\mathrm{sft}}$, the predicted residuals based on $\boldsymbol{f}_{\mathrm{ini}}$ are calculated, respectively, according to

$$\boldsymbol{R}_{\mathrm{ini}} = \boldsymbol{Y}_{\mathrm{ini}} - \boldsymbol{f}_{\mathrm{ini}}(\boldsymbol{X}_{\mathrm{ini}}), \tag{5}$$

$$\boldsymbol{R}_{\mathrm{sft}} = \boldsymbol{Y}_{\mathrm{sft}} - \boldsymbol{f}_{\mathrm{ini}}(\boldsymbol{X}_{\mathrm{sft}}). \tag{6}$$

If $\boldsymbol{R}_{\mathrm{ini}}$ and $\boldsymbol{R}_{\mathrm{sft}}$ are not significantly different, the performance of $\boldsymbol{f}_{\mathrm{ini}}$ over $\mathcal{W}_{\mathrm{sft}}$ can be regarded as the same as that over $\mathcal{W}_{\mathrm{ini}}$. Then, the samples in $\boldsymbol{R}_{\mathrm{ini}}$ and $\boldsymbol{R}_{\mathrm{sft}}$ can both be regarded as coming from the same local process state. Consequently, $\mathcal{W}_{\mathrm{sft}}$ is continuously shifted and the new $\boldsymbol{R}_{\mathrm{sft}}$ is calculated. Once $\boldsymbol{R}_{\mathrm{sft}}$ significantly deviates from $\boldsymbol{R}_{\mathrm{ini}}$, a local process state is identified, which is different from the one represented by $\mathcal{W}_{\mathrm{ini}}$, and the shifting process is stopped. A new local mode $\boldsymbol{f}_{\mathrm{new}}$ can then be constructed based on $\mathcal{W}_{\mathrm{sft}}$. Thus, how to judge whether $\boldsymbol{R}_{\mathrm{sft}}$ evidently differs from $\boldsymbol{R}_{\mathrm{ini}}$ is a critical issue for this adaptive process state partition. In our previous work [46], this problem is converted into examining whether the means and variances of the two residual sequences, $\boldsymbol{R}_{\mathrm{ini}}$ and $\boldsymbol{R}_{\mathrm{sft}}$, are significantly different or not based on the t-test and χ^2-test, respectively. In this paper, we also adopt these two statistical tests.

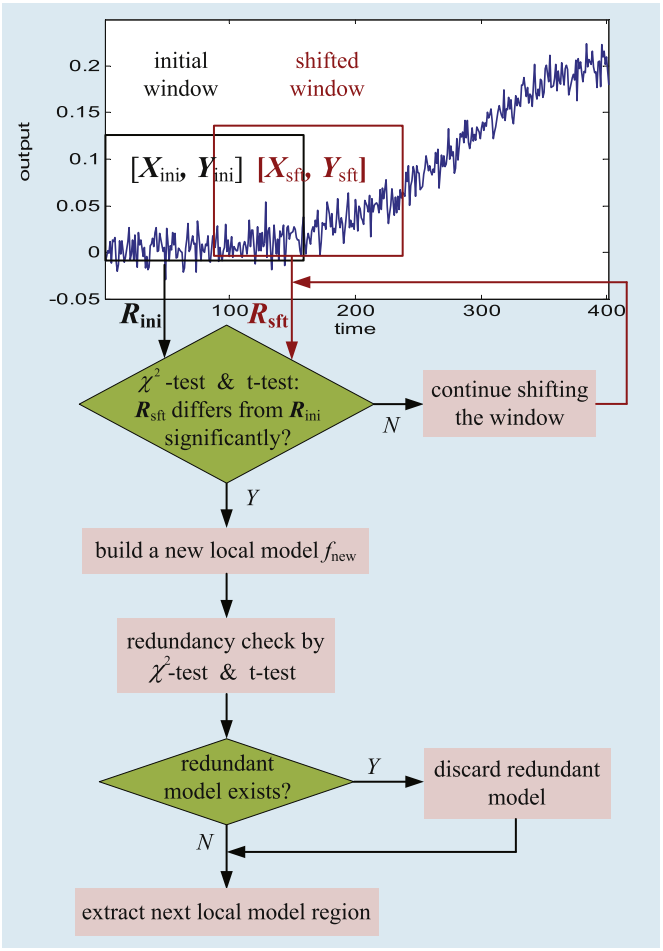

Fig. 1. Schematic of adaptive process state partition based on statistical hypothesis testing.

Under the assumption that both $\boldsymbol{R}_{ini}$ and $\boldsymbol{R}_{sft}$ follow normal distribution, the T statistic and χ^2 statistic can be constructed as follows

$$T = \sqrt{W}(\overline{R}_{sft} - \overline{R}_{ini})/\sigma_{sft}, \tag{7}$$

$$\chi^2 = (W-1)\sigma_{sft}^2/\sigma_{ini}^2, \tag{8}$$

where $\overline{R}_{ini}$ and σ_{ini} are the mean and standard deviation of the distribution from which the samples of $\boldsymbol{R}_{ini}$ are drawn, while $\overline{R}_{sft}$ and σ_{sft} are the mean and standard deviation of $\boldsymbol{R}_{sft}$. $\overline{R}_{ini}$ and σ_{ini} can be estimated using the samples of $\mathcal{W}_{ini}$ provided that W is sufficiently large. Normally, $\overline{R}_{ini}$ is essentially 0. Similarly, $\overline{R}_{sft}$ and σ_{sft} can be estimated based on the samples of $\mathcal{W}_{sft}$. According to the standard statistical theory, when the hypothesis

$$H_{mean} : \overline{R}_{sft} = \overline{R}_{ini} \tag{9}$$

is valid, the T statistic (Eq. (7)) follows the t distribution with the degree of freedom $W - 1$. Likewise, the χ^2 statistic (Eq. (8)) follows the χ^2 distribution with the degree of freedom $W - 1$, if the hypothesis

$$H_{std} : \sigma_{sft} = \sigma_{ini} \tag{10}$$

holds. Consequently, the t-test and χ^2-test can be utilized to test whether the above two hypotheses are valid or not. Specifically, if

$$|T| < \lambda_t \text{ and } \chi^2 < \lambda_\chi \tag{11}$$

hypothesis (9) is valid by the t-test and hypothesis (10) is valid according to the χ^2-test, that is, the mean of $\boldsymbol{R}_{sft}$ does not deviate significantly from that of $\overline{R}_{ini}$, and the variances of $\boldsymbol{R}_{sft}$ and $\overline{R}_{ini}$ are considered to be the same. In Eq. (11), λ_t is the threshold value of the T statistic for the given significance level α_t, i.e., $\text{Prob}\{|T| < \lambda_t\} = 1 - \alpha_t$, while λ_χ is the threshold value of the χ^2 statistic for the given significance level α_χ, namely, $\text{Prob}\{\chi^2 < \lambda_\chi\} = 1 - \alpha_\chi$.

Thus, $\boldsymbol{R}_{sft}$ does not differ significantly from $\overline{R}_{ini}$ when both conditions of Eq. (11) are fulfilled. Otherwise, $\boldsymbol{R}_{sft}$ is judged to deviate from $\overline{R}_{ini}$ significantly, and we have identified a local process state that is different from the one represented by $\{\mathcal{W}_{ini}, \boldsymbol{f}_{ini}\}$. Let us denote the newly constructed local model based on the newly identified data subset $\mathcal{W}_{sft} = \{\boldsymbol{X}_{sft}, \boldsymbol{Y}_{sft}\}$ as $\boldsymbol{f}_{new}$. In the previous work [46], $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ is simply added to the local model set, and the number of the local model regions is increased by one. However, a potential problem associated with this approach is that the number of local models is even increasing as the plant is continuously operating. Simply removing the 'oldest' local model may not be the correct way of avoiding this problem, as the oldest local model may actually be different from the newly constructed one, and it represents a different local process state. The more serious problem associated with the approach of [46] is however that the growing local model set may contain many similar local models, and therefore some of them are redundant.

Let us analyze the root of this problem. When we identify a new local process state $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ based on the t-test and χ^2-test of Eq. (11), we only show that it differs from $\{\mathcal{W}_{ini}, \boldsymbol{f}_{ini}\} = \{\mathcal{W}_L, \boldsymbol{f}_L\}$ in the existing old local model set $\{\mathcal{W}_l, \boldsymbol{f}_l\}_{l=1}^{L}$. If $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ also differs from $\{\mathcal{W}_l, \boldsymbol{f}_l\}$ for $1 \le l \le L-1$, then it represents a truly new local process state. In this case, $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ is added to the local model set and the number of the local models is increased by 1, i.e., $L = L + 1$. However, if there exists an old local model $\{\mathcal{W}_{l'}, \boldsymbol{f}_{l'}\}$ which is similar to $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$, where $l' \in \{1, 2, \cdots, L-1\}$, then either $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ or $\{\mathcal{W}_{l'}, \boldsymbol{f}_{l'}\}$ can be regarded as redundant. Since $\{\mathcal{W}_{l'}, \boldsymbol{f}_{l'}\}$ is 'older' than $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$, after adding $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ to the local model set, we should discard $\{\mathcal{W}_{l'}, \boldsymbol{f}_{l'}\}$ and the number of the local models does not increase. In this paper, we propose to use statistical hypothesis testing for performing this task of detecting and discarding a redundant local model during the adaptive process state partition.

The predicted residuals of $\{\boldsymbol{X}_{sft}, \boldsymbol{Y}_{sft}\}$ based on $\boldsymbol{f}_{new}$ and $\boldsymbol{f}_l$ are defined respectively by

$$\boldsymbol{R}_{new} = \boldsymbol{Y}_{sft} - \boldsymbol{f}_{new}(\boldsymbol{X}_{sft}), \tag{12}$$

$$\boldsymbol{R}_l = \boldsymbol{Y}_{sft} - \boldsymbol{f}_l(\boldsymbol{X}_{sft}), \quad 1 \le l \le L-1. \tag{13}$$

Again, under the assumption that $\boldsymbol{R}_{new}$ and $\boldsymbol{R}_l$ follow normal distribution, the T statistic and χ^2 statistic can be constructed according to

$$T_l = \sqrt{W}(\overline{R}_l - \overline{R}_{new})/\sigma_l, \tag{14}$$

$$\chi_l^2 = (W-1)\sigma_l^2/\sigma_{new}^2, \tag{15}$$

where $\overline{R}_{new}$ and σ_{new} are the mean and standard deviation of $\boldsymbol{R}_{new}$, which can be estimated using the samples of $\boldsymbol{R}_{new}$, while $\overline{R}_l$ and σ_l are the mean and standard deviation of $\boldsymbol{R}_l$, which can also be estimated. If the following two hypotheses

$$H_{mean_l} : \overline{R}_l = \overline{R}_{new}, \tag{16}$$

$$H_{std_l} : \sigma_l = \sigma_{new}, \tag{17}$$

are valid, T_l and χ_l^2 follow the t-distribution and χ^2-distribution, respectively. Therefore, if there exists an $l \in \{1, 2, \cdots, L-1\}$ such that

$$|T_l| < \lambda_t \text{ and } \chi_l^2 < \lambda_\chi, \tag{18}$$

hypotheses (16) and (17) are both valid, and $\overline{R}_{new}$ and $\boldsymbol{R}_l$ are regarded to be identical. In this case, we discard $\{\mathcal{W}_l, \boldsymbol{f}_l\}$ and add $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ to the local model set. Otherwise, we add $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ to the local model set and increase the number of the local models by 1.

In practice, the significance levels α_t and α_χ can be set to small values, such as 0.05, and their effects can be compensated partly by adjusting the window size W. It is worth pointing out again that although new local model regions may be continuously extracted at the plant is operating online continuously, all the operations involved at this stage can be implemented 'offline'. Therefore, the online computational efficiency is not really an issue here.

3.1.1. Offline stage

We are now ready to summarize the procedure of adaptively extracting the unique local model regions through the proposed statistical hypothesis testing, which constitutes the offline stage of our proposed LPLS-APSP based online soft sensor.

3.1.2. Initialization

Collect an initial data set $\mathcal{W}_{ini}$ with W consecutive-time samples, and build a PLS model $\boldsymbol{f}_{ini}$ upon $\mathcal{W}_{ini}$. Then calculate $\boldsymbol{R}_{ini}$, and estimate $\overline{R}_{ini}$ and σ_{ini}.

Set $L = 1$ and $\{\mathcal{W}_L, \boldsymbol{f}_L\} = \{\mathcal{W}_{ini}, \boldsymbol{f}_{ini}\}$. Then set $\mathcal{W}_{sft} = \mathcal{W}_L$, and go to **Step 1)**.

Step 1): Extract a new local model region.

Step 1a): Collect a new data sample from the plant[1], shift $\mathcal{W}_{sft}$ one sample ahead to obtain the new $\mathcal{W}_{sft}$, calculate $\boldsymbol{R}_{ini}$ of Eq. (6) using $\boldsymbol{f}_{ini}$, and estimate the corresponding $\overline{R}_{sft}$ and σ_{sft}.

Step 1b): Construct the T statistic and χ^2 statistic of Eqs. (7) and (8), respectively.

Step 1c): Check the conditions of (11).

If both the conditions hold

return to **Step 1a)**;

otherwise

go to **Step 1d)**.

Step 1d): Build a new PLS model $\boldsymbol{f}_{new}$ upon $\mathcal{W}_{sft}$, compute $\boldsymbol{R}_{new}$ of Eq. (12), and estimate the associated $\overline{R}_{new}$ and σ_{new}. Then go to **Step 2)**.

Step 2): Detect and discard redundant local model. for ($l = 1$ to $L - 1$) ***loop***

Step 2a): Calculate $\boldsymbol{R}_l$ of Eq. (13), estimate $\overline{R}_l$ and σ_l.

Step 2b): Construct the T_l statistic and χ^2_l statistic of Eqs. (14) and (15), respectively.

Step 2c): Check the conditions of Eq. (18).

If both the conditions hold: i) discard $\{\mathcal{W}_l, \boldsymbol{f}_l\}$, renumber the remaining $L - 1$ old local models as 1, 2, ···, $L - 1$, and add $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ as $\{\mathcal{W}_L, \boldsymbol{f}_L\}$. ii) Then break **loop** and go to **Step 2d)**

otherwise: continue until **end of loop**.

end of loop: Set $L = L + 1$, add $\{\mathcal{W}_{sft}, \boldsymbol{f}_{new}\}$ as $\{\mathcal{W}_L, \boldsymbol{f}_L\}$, and go to **Step 2d)**.

Step 2d): Set $\mathcal{W}_{sft} = \mathcal{W}_L$, and return to **Step 1)**.

3.2. Criterion for online model adaptation

For the LPLS-APSP based online soft sensor, a single local model is responsible for estimating unknown samples. Thus, an effective model adaptation criterion is of vital importance in selecting an appropriate local model to estimate the output of query sample. Before introducing our proposed model selection criterion, we examine three existing approaches to learn from their strength and weakness. In the JITL modeling [29–34], the samples that are located within a neighborhood of the query sample, represented by the dashed circle in Fig. 2a, are used to train a temporal local model $\boldsymbol{f}_{tmp}$ to estimate the output of the query sample. In the case illustrated in Fig. 2a, the JITL criterion selects neither local model 1 nor local model 2. Given the newest input and output measurement $\{\mathbf{x}_0, y_0\}$[2], the CoJIT modeling [21] calculates the Q-statistics Q_l of $\{\boldsymbol{x}_0, y_0\}$ to the input–output subspaces $\mathcal{W}_l$, $1 \le l \le L$, and selects the local model $\boldsymbol{f}_{l^*}$, which has the minimum Q-statistic value Q_{l^*}, to predict the output of query sample. However, the Q-statistic, which is the square distance between $\{\boldsymbol{x}_0, y_0\}$ and the subspace of data set $\mathcal{W}_l$, does not represent the prediction accuracy for y_0. For the example shown in Fig. 2b, Q_1 is smaller than Q_2, and thus, the model $\boldsymbol{f}_1$ is selected. But the actual predicted error of $\boldsymbol{f}_1$ for $\{\boldsymbol{x}_0, y_0\}$, $e_1 = y_0 - \boldsymbol{f}_1(\boldsymbol{x}_0)$, is larger than $e_2 = y_0 - \boldsymbol{f}_2(\boldsymbol{x}_0)$ of the model $\boldsymbol{f}_2$. In the LARPLS [37] and the LASS [38], the local model that minimizes the prediction error for the newest measured sample $\{\mathbf{x}_0, y_0\}$ is selected to provide the estimate of the output for query sample. For the case illustrated in Fig. 2c, the local model $\boldsymbol{f}_2$ is selected as it provides a smaller prediction error e_2 for $\{\boldsymbol{x}_0, y_0\}$ than e_1 offered by the local model $\boldsymbol{f}_1$. However, the prediction accuracy of $\boldsymbol{f}_2$ for the query sample is rather poor in this case. This is because $\boldsymbol{x}_0$ and the query sample may not belong to the same local process state.

In addition to capturing the latest process dynamics, the principle of 'similar inputs resulting in similar outputs' is also important, and thus model adaptation should take into account the following factors:

1) The latest available process's operating condition, which is normally contained in the newest labeled measurement sample, should not be ignored.
2) An appropriate local model should also provide good estimates for the neighbors of the query sample.
3) Furthermore, since the query sample is similar to its neighbors, for a good local model, the estimates for the query sample and its neighbors should not differ too much. In other words, the differences between the predicted error of the query sample and those of its neighbors are expected to be small.

By taking into account the above three considerations, we propose the following novel criterion for online local model adaptation, where $J^{(l)}$ denotes the value of the proposed cost function for the lth local model,

$$\begin{aligned} J^{(l)} &= \alpha e_0^{(l)} + (1-\alpha)\left(\sum_{i=1}^{K} s_i e_i^{(l)} / \sum_{i=1}^{K} s_i + (1-\beta)\sum_{i=1}^{K} s_i \Delta e_i^{(l)} / \sum_{i=1}^{K} s_i\right) \\ &= \alpha J_1^{(l)} + (1-\alpha) J_2^{(l)}. \end{aligned} \tag{19}$$

In criterion (19), K is the neighborhood size of the query sample $\boldsymbol{x}_q$, namely, the neighbors of $\boldsymbol{x}_q$ are $\{\boldsymbol{x}_i^{\{q\}}, y_i^{\{q\}}\}$ for $1 \le i \le K$, $0 < \alpha, \beta < 1$ are the two weighting parameters, and

$$e_0^{(l)} = |\boldsymbol{f}_l(\boldsymbol{x}_0) - y_0|, \tag{20}$$

$$e_i^{(l)} = \left|\boldsymbol{f}_l\left(\boldsymbol{x}_i^{(q)}\right) - y_i^{(q)}\right|, \quad 1 \le i \le K, \tag{21}$$

$$\Delta e_i^{(l)} = \left|\boldsymbol{f}_l\left(\boldsymbol{x}_i^{(q)}\right) - \mathbf{f}_l\left(\boldsymbol{x}_q\right)\right|, \quad 1 \le i \le K, \tag{22}$$

[1] If the plant operational data have already been collected and stored, simply shift $\mathcal{W}_{sft}$ one sample step ahead.

[2] As we consider $p = 1$, the single-output case, the scalar notation is used for the output.

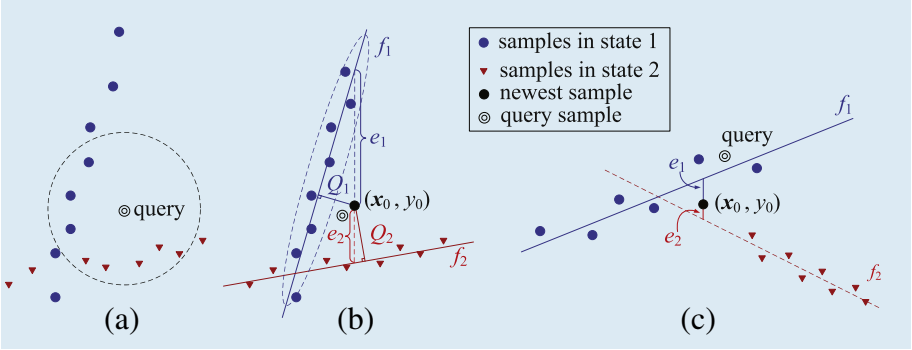

Fig. 2. Illustration of three existing approaches for selecting local model for estimating the output of query sample: (a) JITL, (b) CoJIT, and (c) LARPLS and LASS.

while s_i defines the similarity of $\boldsymbol{x}_i^{(q)}$ and $\boldsymbol{x}_q$. Several similarity measures are defined in [29,47,48]. The following commonly used, simple yet effective similarity measure is employed

$$s_i = exp\left(-\phi d\left(\boldsymbol{x}_i^{(q)}, \boldsymbol{x}_q\right)\right), \tag{23}$$

where $d(\cdot,\cdot)$ denotes the Euclidean distance metric and $\phi > 0$ is a scaling factor. When $0 < \phi < 1$, the influence of distance is attenuated, while when $\phi > 1$, the influence of distance is amplified. Later, we will discuss how to determine an appropriate value of ϕ, which depends on application.

The first item in the right hand side of Eq. (19), $J_1^{(l)}$, considers the current operating condition, while the second term $J_2^{(l)}$ enables the mining of the information contained in the query sample and its neighbors. These two components of the cost function together ensure that the selected local model $\boldsymbol{f}_{\widetilde{l}}$, where

$$\widetilde{l} = arg \min_{1 \le l \le L} J^{(l)}, \tag{24}$$

has the ability to generalize well for the query sample. For the case shown in Fig. 2c, for example, the proposed model selection criterion will select the local model $\boldsymbol{f}_1$, rather than $\boldsymbol{f}_2$. Note that if we set $\alpha = 1$, then $J^{(l)} = J_1^{(l)}$, and the proposed model selection criterion degenerates to the one that is used in the LARPLS [37] and the LASS [38]. We are now ready to summarize the online stage of our proposed LPLS-APSP based online soft sensor. Given a query sample $\boldsymbol{x}_q$:

Step 1): Search for the K nearest neighbors of $\boldsymbol{x}_q$ according to the Euclidean distance metric, and denote the results by $\{\boldsymbol{x}_i^{(q)}, y_i^{(q)}\}_{i=1}^K$.
Step 2): Calculate the similarity of $\boldsymbol{x}_q$ and $\boldsymbol{x}_i^{(q)}$ using Eq. (23) for $1 \le i \le K$.
Step 3): According to Eqs. (20) to (22), calculate $e_0^{(l)}$, $e_i^{(l)}$ and $\Delta e_i^{(l)}$ with $\boldsymbol{f}_l$ for $1 \le i \le K$ and $1 \le l \le L$.
Step 4): Calculate $J^{(l)}$ according to Eq. (19) for $1 \le l \le L$.
Step 5): Find the optimal local model $\boldsymbol{f}_{\widetilde{l}}$ according to Eq. (24), and estimate the output of $\boldsymbol{x}_q$ using $\boldsymbol{f}_{\widetilde{l}}$.

It is worth pointing out that compared to the JITL based online soft sensor, online constructing local models is avoided in our LPLS-APSP based online soft sensor, since all the local models have been prepared 'offline'. Therefore, our LPLS-APSP is computationally more efficient than the JITL at the online operation stage.

4. Two case studies

The performance of our LPLS-APSP based online soft sensor was investigated using a simulated continuous stirred tank reactor (CSTR) and a real industrial debutanizer column process. For comparison purpose, the results of several state-of-the-art methods for online soft sensor design were also provided and analyzed. These benchmark methods included the least squares SVM (LSSVM) based design [49], the conventional distance based JITL PLS (JITLPLS) [47], the recursive PLS (RPLS) [19], the moving window PLS (MWPLS) [18] and the CoJIT [21]. Except for the LSSVM based design which is a global nonlinear modeling method, the other ones investigated are all adaptive and employ a linear PLS algorithm to construct the soft sensor model. The estimation accuracy was evaluated by the root mean squares error (RMSE), the relative RMSE (RR) and the maximum absolute error (MAE) defined respectively by

$$\text{RMSE} = \sqrt{\sum_{t=1}^{N_T} (y_t - \hat{y}_t)^2 / N_T}, \tag{25}$$

$$\text{RR} = 100\% \times \sqrt{\sum_{t=1}^{N_T} ((y_t - \hat{y}_t)/y_t)^2 / N_T}, \tag{26}$$

$$\text{MAE} = \max\{|y_t - \hat{y}_t|, t = 1, 2, \cdots, N_T\}, \tag{27}$$

where y_t and $\hat{y}_t$ denote the true output value and its predicted value of the tth test sample, respectively, while N_T is the number of test samples. Additionally, assuming that each element of matrix occupies one byte's memory space, the local model number (MN) and the occupied memory space (MS) were also employed to measure the requirements on storage devices. Furthermore, the online consumed CPU time ($\text{CPT}^{\text{online}}$), averaged over 10 independent simulations, was utilized to evaluate the online computational efficiency of a soft sensor. The computations of all the experiments were carried out on a Core i5 (2.6 GHz × 2) with 4 GB RAM, and with Windows 7 and MATLAB version R2010a.

4.1. Continuously stirred tank reactor

The schematic structure of an exothermic irreversible first-order CSTR, which are widely used for testing soft sensors' performance [27,40], is shown in Fig. 3. In the CSTR, an irreversible reaction $A \rightarrow B$ takes place, which transforms cyclopentadiene A to the product cyclopentenol B. This reaction can be described by

$$\frac{dC_A(t)}{dt} = \frac{F_i}{V}(C_{Ai} - C_A(t)) - k_0 C_A(t) \exp\left(-\frac{E}{RT_r(t)}\right), \tag{28}$$

$$\begin{aligned}\frac{dT_r(t)}{dt} = {} & \frac{F_i}{V}(T_i - T_r(t)) - \frac{\Delta H k_0 C_A(t)}{\rho C_p} \exp\left(-\frac{E}{RT_r(t)}\right) \\ & + \frac{\rho_c C_{pc}}{\rho C_p V} F_c(t)\left(1 - \exp\left(-\frac{hA}{F_c(t)\rho C_p}\right)\right)(T_{ci} - T_r(t)).\end{aligned} \tag{29}$$

The detailed description of this CSTR's nonlinearity and its steady state operating conditions as well as the model parameters can be

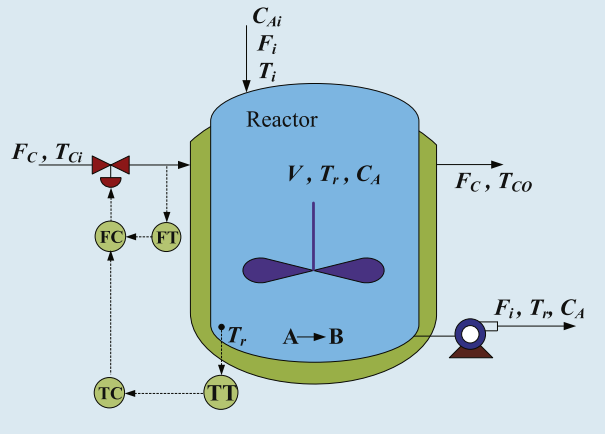

Fig. 3. Schematic diagram of the CSTR with its control structure.

found in [27]. The residual concentration of A, denoted as $C_A(t)$, was the target variable for soft sensing. The flow rate of the cooling water, $F_c(t)$, and the reactor temperature, $T_r(t)$, were chosen as the secondary variables.

Throughout the entire simulation process, we periodically alter the set point of $T_r(t)$ to ± 2 (K) every 60 h. The sampling interval was set to 1 min. But we assumed that $C_A(t)$ was analyzed in the laboratory every 2 h and the analyzed value was referred to as the 'analytical value'. Thus the sampling period for collecting the 'labeled' samples was 2 h, and between the two labeled samples, there were 119 'unlabeled' samples with the sampling period of 1 min. To capture the process dynamics, the input vector was augmented with the samples at the previous sampling instant, and therefore the model structure for developing soft sensors was set to

$$C_A(k) = f(F_c(k-1), F_c(k-2), T_r(k-1), T_r(k-2)). \tag{30}$$

To simulate the periodical decrease and recovery of catalyst activation, the frequency factor k_0 was changing as shown in Fig. 4, which made this CSTR process time-varying and highly nonlinear. The whole simulation period was 1800 h containing 900 analytical (labeled) samples, among which the 600 analytical samples of the first 1200 h were used as the training dataset for model construction and the rest 300 analytical samples in the test period of 600 h were employed as the validation data set for determining the algorithmic or design parameters of the soft sensor by cross validation. During the test period, there were large number of 'unlabeled' samples (35,700) obtained at the sampling

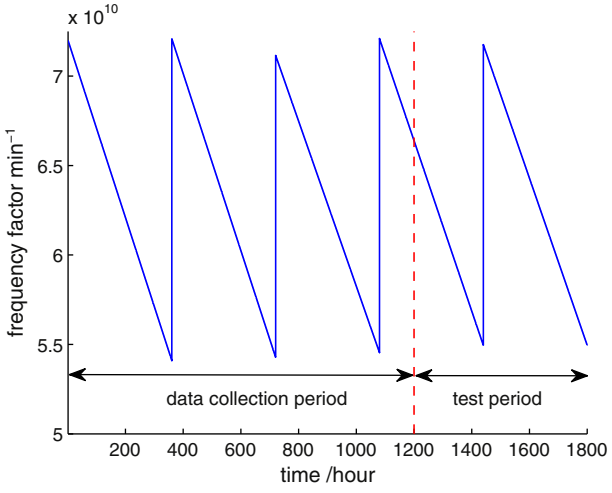

Fig. 4. Periodical change of the frequency factor.

period of 1 min, for which the corresponding output values of C_A were actually known. These unlabeled data samples together with their actual output values formed the test dataset for evaluating the performance of an online soft sensor. In the simulations, all the process variables, $C_A(k)$, $F_c(k)$ and $F_r(k)$, were corrupted by Gaussian noises. Additionally, all the variables were scaled to within the range of [0.0, 1.0].

4.1.1. Online soft sensor designs

The number of local models found by the CoJIT was MN = 881, while the number of local models determined by the APSP procedure of our proposed soft sensor was MN = 98, which was much smaller than that of the CoJIT. The design parameters of all the methods investigated were optimized by a particle swarm optimization (PSO) technique to minimize the predicted RMSE over the validation dataset. Appendix A explains how the search ranges for the parameters of each soft sensor were determined for th PSO algorithm, which produced the following optimized algorithmic parameters for each soft sensor design.

a) LSSVM: the Gaussian kernel width was 0.28, and the regularization parameter was 998.5.
b) JITLPLS: 8 nearest samples were used as the query sample's neighbors, and the number of latent variables in the PLS model, LV_{PLS}, was $LV_{PLS} = 1$.
c) RPLS: the forgetting factor was 0.965 and $LV_{PLS} = 3$.
d) MWPLS: the moving window length was 10 and $LV_{PLS} = 2$
e) CoJIT: the window size was 20 with the window moving width set to 1, and the coefficient for constructing the 7.14×10^{-5}, while $LV_{PLS} = 3$ and the number of latent variables in the PCA model, LV_{PCA}, was $LV_{PCA} = 3$.
f) LPLS-APSP: the window size was $W = 13$, $LV_{PLS} = 1$, $K = 15$, $\phi = 47.1$, $\alpha = 0.216$, and $\beta = 0.483$. Both the significance levels for the t-test and χ^2-test, namely, α_t and α_χ, were set to 0.05, as suggested in [39].

As an example, we illustrate why the obtained design parameters of the LPLS-APSP are appropriate. Fig. 5a to d shows the predicted RMSE over the test data set as the function of the window width W, the neighborhood size K, the scaling factor ϕ, and the two weighting parameters α and β, respectively, where in each case, the rest of the algorithmic parameters were set to the designed values given above, obtained by optimizing the predicted RMSE over the validation data set using the PSO. It can be seen from Fig. 5a that the true optimal value of $W = 12$ for the best test performance is close to the optimized $W = 13$ based on the validation data set. Fig. 5b shows that when K is greater than 10, small test RMSE can be obtained and, therefore, the optimized $K = 15$ obtained based on the validation set is also a good choice. Fig. 5c indicates that the test RMSE is a uni-modal function of ϕ and the optimal value of ϕ is about 30 with the minimum test RMSE value of 0.208. Note that the test RMSE for the designed $\phi = 47.1$ is 0.0210, which is very close to the optimal value of 0.0208, and therefore the optimized $\phi = 47.1$ based on cross validation is appropriate. Lastly, Fig. 5d provides a three dimensional plot to illustrate the influence of α and β on the test RMSE, which reveals that the optimized value ($\alpha = 0.216$, $\beta = 0.483$) based on cross validation is located within the 'acceptable area' of high test performance, where the test RMSE is less than 0.0215. The results of Fig. 5 confirm that the algorithmic parameters obtained by minimizing the predicted RMSE over the validation data set achieve a good generalization performance for the unseen test data set and, therefore, are appropriate. Space limitation precludes us from detailing the procedure of optimizing the algorithmic parameters by using the PSO to minimize the predicted RMSE over the validation data set, which is basically a standard cross-validation approach.

In addition, some other useful information can be extracted from Fig. 5. For example, it can be seen from Fig. 5a that large window size W results in poor test performance while small window size is

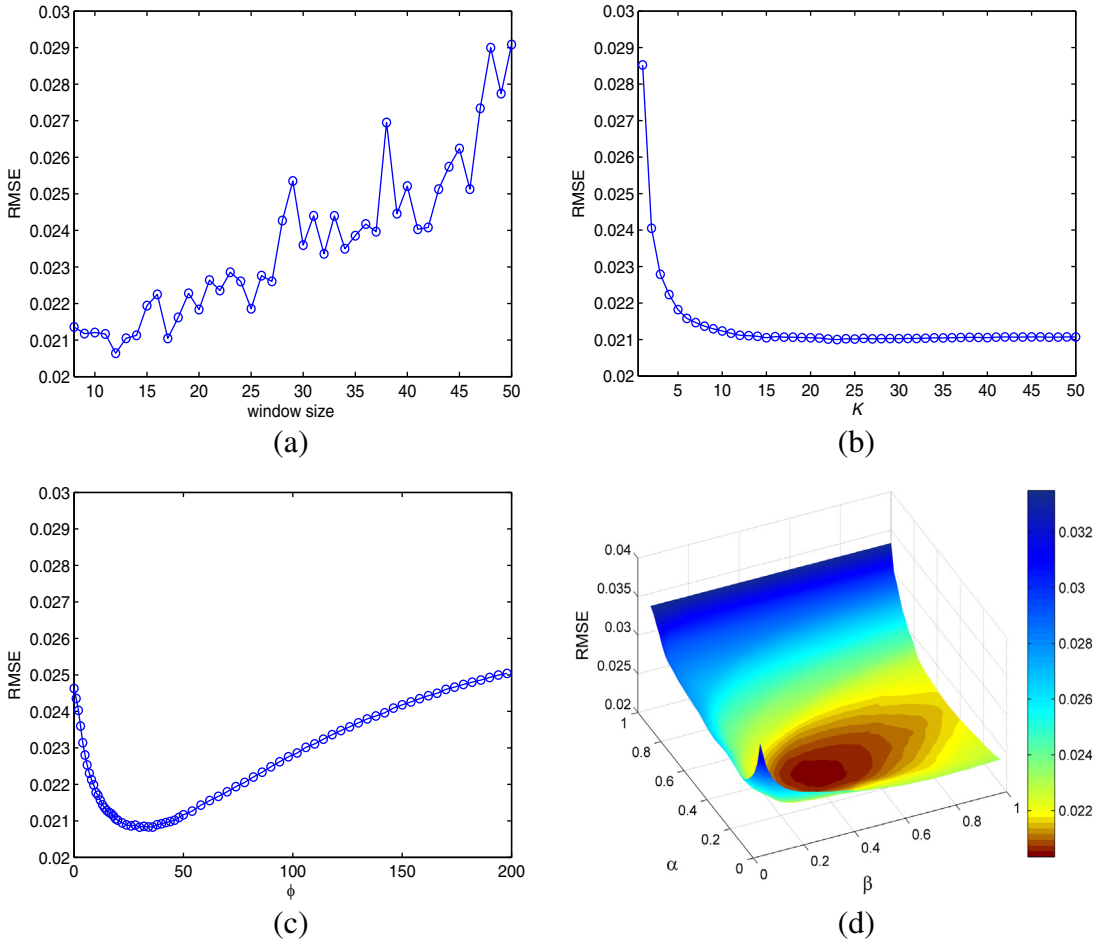

Fig. 5. Predicted RMSE over the test data set as a function of the LPLS-APSP design parameters: (a) W, (b) K, (c) ϕ, and (d) α and β. In each case, the rest of the algorithmic parameters were set to the designed values.

preferable. This is because the larger W is, the lower the 'localization degree' is, i.e., the weaker the ability of handling the process nonlinearity by local linear models is, and vice versa. However, too small window size may cause model instability and incorrect estimation of the predicted residuals' mean value and variance, which are required in calculating the T and χ^2 statistics of Eqs. (7) and (8) as well as Eqs. (14) and (15). Thus the value of W should be set to achieve a satisfactory cross-validation performance.

The effect of the query sample's neighbors, depicted in Fig. 5b, indicates that the test RMSE decreases as K increases. However, when K is larger than 10, the test RMSE remains almost constant. This is because the influence of distant neighbors abates sharply to nearly zero through the weighting strategy given in Eq. (23) for the given scaling parameter of $\phi = 47.1$. When ϕ is set to 0, all the neighbors have the same importance and the test accuracy is poor as can be seen from Fig. 5c. This confirms that differentiating the influence of neighbors is necessary, meaning we should choose $\phi > 0$. But if ϕ is excessively large, only the first neighbor of the query sample counts, which is equivalent to setting $K = 1$ and is not appropriate either. Thus, the influences of K and ϕ should be considered jointly, and we suggest setting K to a relative large value together with a carefully selected scaling parameter value ϕ.

Additionally, when α is set to 1, the model adaptation criterion $J^{(l)}$ defined by Eq. (19) degenerates to the one used in the LARPLS [37] and the LASS [38], and the test RMSE is very large as can be seen from Fig. 5d, which indicates incorrect model adaptation. By contrast, as soon as α decreases away from 1, meaning that the neighbors of the query sample start to take effect, the test RMSE decreases. Therefore, mining neighborhood information of query samples to form a model adaptation criterion is necessary and effective.

4.1.2. Test performance comparison

We are now ready to compare the performance of various online soft sensor designs. Fig. 6a to f depicts the predicted values for C_A over the test data set obtained by the various online soft sensor designs based on the LSSVM, the JITLPLS, the RPLS, the MWPLS, the CoJIT and the proposed LPLS-APSP, respectively, where the actual values as well as the analytical values of C_A are also shown for comparison. Table 1 quantitatively compares the test performance of these online soft sensor designs over the test data set with several performance measures. Note that the historical data set, i.e., the training data set, was augmented with the analytical values every two hours at the online operation stage.

Fig. 6a indicates that the global modeling based LSSVM design does not behave well for this CSTR process with strong time-varying nonlinearity, and its predictions of C_A are visually biased over the time intervals of 1300 h to 1350 h, 1400 h to 1450 h, and 1650 h to 1700 h. As a local learning strategy, the JITLPLS design can improve the estimation accuracy to some extent compared to the LSSVM, but its performance is still unsatisfactory because its prediction errors are still large and the biases in the time intervals of 1300 h to 1350 h, 1400 h to 1450 h, and 1650 h to 1700 h are still visible, as can be seen from Fig. 6b. The prediction accuracy of the RPLS design is very poor, particularly from the time period of 1200 h to 1550 h, as clearly shown in Fig. 6c, where the RPLS struggles to track the process dynamics in time when the set point value of the reaction temperature changes, leading to large

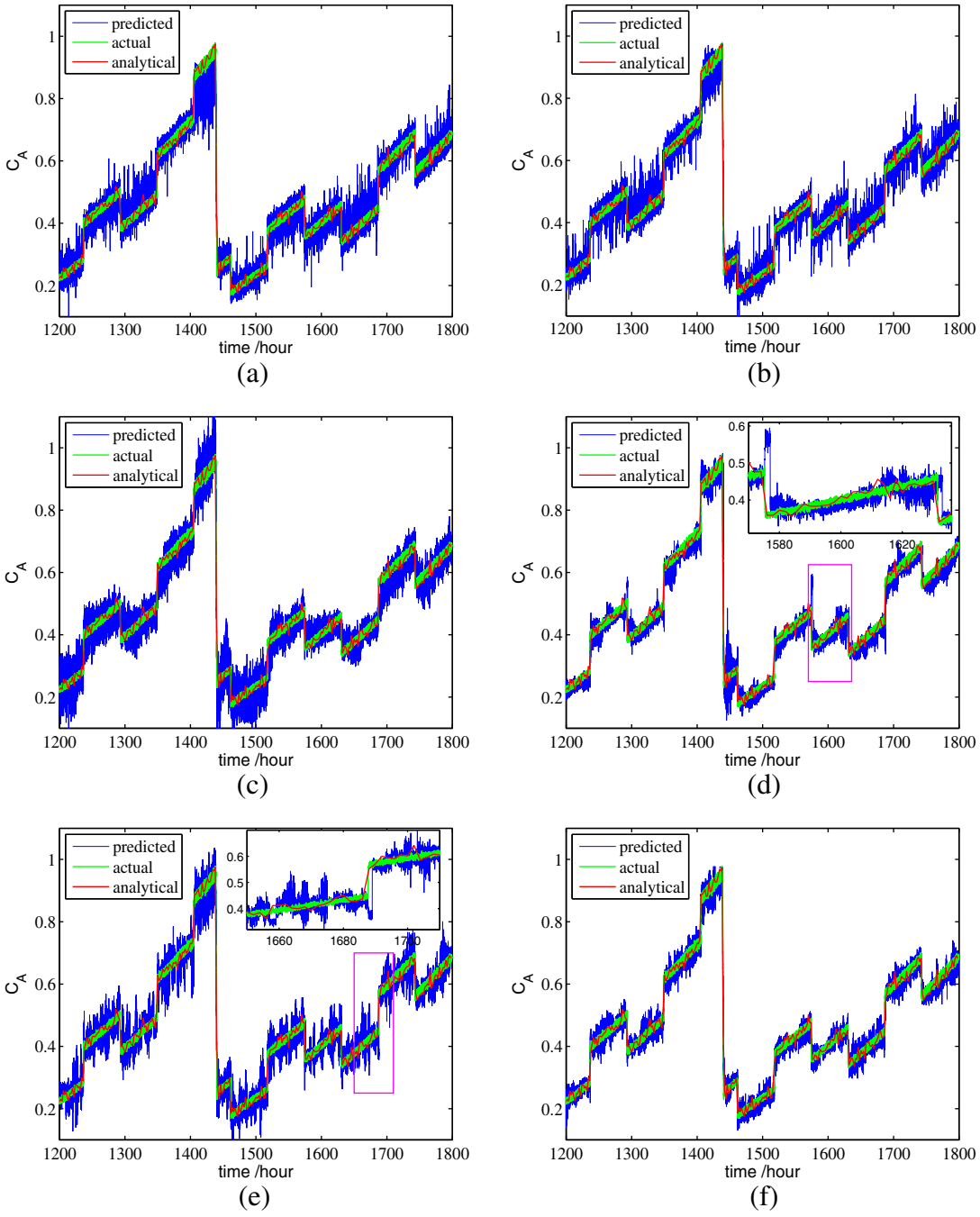

Fig. 6. Comparison of the prediction results over the test data set for the CSTR achieved by various online soft sensors based on: (a) the LSSVM, (b) the JITLPLS, (c) the RPLS, (d) the MWPLS, (e) the CoJIT and (f) the LPLS-APSP.

prediction errors. The MWPLS based design achieves much higher prediction accuracy, particularly in the relative steady-state operating conditions. As can be seen from Fig. 6d, however, its performance deteriorates significantly when the operation condition changes sharply. For example in the enlarged plot shown in Fig. 6d, when the actual plant state is undergoing a step change, the prediction of the MWPLS based soft sensor goes wild. This is because the query sample is strange to the current PLS model under such a circumstance. Only after a few labeled analytical samples have been accumulated for the new operational state, can the newly updated PLS model recovers from such a catastrophe. Similar problem also appears in Fig. 6e, because the CoJIT does not switch local model unless the labeled analytical value is available [21]. Besides, it can be seen from Fig. 6e that when the local model is selected properly, for example at the time period around 1680 h showing in the enlarged plot, the estimation performance of the CoJIT is excellent, but in many time periods, the estimation variances of the CoJIT are large, which indicates that inappropriate model adaptations occur.

Table 1
Performance comparison of various online soft sensors over the test data set for the CSTR.

Soft sensor	RMSE	RR (%)	MAE	MN	MS (byte)	CPT^{online} (s)	CPT^{opt} (s)
LSSVM	0.0294	6.53	0.280	1	–	5.72	329.2
JITLPLS	0.0245	5.91	0.206	1	–	27.91	247.1
RPLS	0.0386	10.5	0.217	1	–	1.11	266.7
MWPLS	0.0335	8.03	0.491	1	–	1.15	292.6
CoJIT	0.0327	7.94	0.214	881	20,263	1.14	14,556.5
LPLS-APSP	0.0210	5.26	0.159	98	490	13.57	2273.7

By contrast, it can be seen from Fig. 6f that the overall prediction performance of the proposed LPLS-APSP is the best. To be more specific, although the prediction variance of the LPLS-APSP design is slightly larger than that of the MWPLS in relative steady states, its prediction accuracy is consistently very good over all the states. In particular, the proposed LPLS-APSP based soft sensor is highly effective to prevent large prediction error when the process state undergoes abrupt changes, such as the change of the set point value of the reaction temperature and the recovery of catalyst activity. In fact, the MWPLS model may approximately be regarded as the 'optimal' local model for the newest 10 'labeled' samples, and mining neighborhood information in the LPLS-APSP is similar to that of the JITL. Therefore, the proposed method, in a sense, performs a tradeoff between the MWPLS and the JITLPLS, but it is capable of alleviating the negative effects of both the MWPLS and JITLPLS. The prediction accuracy measures listed in Table 1 also confirm that the LPLS-APSP based soft sensor attains the overall best performance. This demonstrates that the proposed model adaptation criterion defined in Eq. (19) is highly effective. However, like the JITLPLS, online model switching occurs in the proposed soft sensor each time a query sample is produced, which may result in high model adaptation frequency too.

The selected local model for the LPLS-APSP soft sensor during the online operation stage is illustrated in Fig. 7. It can be seen that the latest model is only selected for about 16% of all the queries, and for about 84% of all the queries other models are selected. This is simply because the latest model may not be optimal for minimizing the objective defined in Eq. (19), especially when abrupt changes occur, as the latest model does not take the query sample's information into consideration. Actually, if the latest model were always selected, the proposed method would degrade into the MWPLS, whose performance is far inferior to the proposed method, as shown in Fig. 6d and Table 1.

In terms of MN, the LSSVM adopts a global nonlinear model, while the JITLPLS, RPLS and MWPLS adapt a single linear PLS model. By contrast, as can be seen from Table 1, the LPLS-APSP adapts among 98 local models, among which 28 local models are actually identified during the online operation stage. This clearly shows that the proposed soft sensor is capable of handling the plant's time-varying dynamics. Furthermore, the MN of the LPLS-APSP is only 11% of the total local models employed by the CoJIT, and the window size of the LPLS-APSP is also smaller than that of the CoJIT. Consequently, the required MS of the LPLS-APSP is significantly smaller than that needed by the CoJIT. More specifically, for the CoJIT, the Q and T^2 statistics of all the local PCA models need to be calculated in each adaptation and thus the loading matrices and eigenvalues of all the PCA models have to be stored. For this CSTR process, a total of $\mathrm{MN} \times (m+2) \times \mathrm{LV}_{\mathrm{PCA}}$ byte memory space are required for storing the local PCA models' parameters,

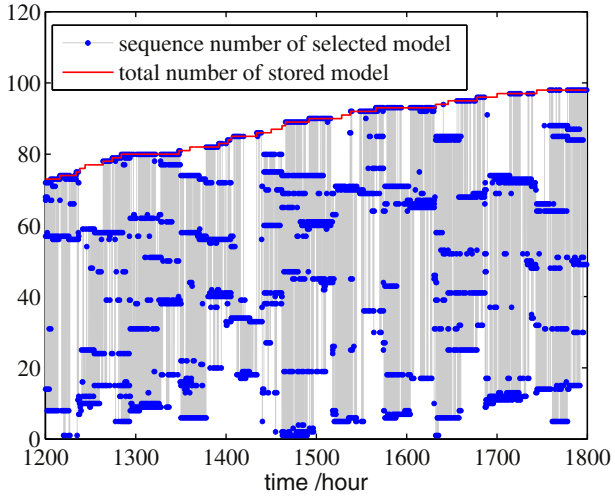

Fig. 7. The selected local model for the LPLS-APSP soft sensor during the online operation stage.

where m is the dimension of the input vector. Furthermore, the local PLS models' regression coefficients need to be stored too, and this requires the additional $\mathrm{MN} \times (m+1)$ byte memory space. Thus, the CoJIT requires a total of 20,263 byte MS. By contrast, the LPLS-APSP design only needs the $\mathrm{MN} \times (m+1) = 490$ byte MS for storing its local PLS models' regression coefficients. Since the LSSVM, JITLPLS, RPLS and MWPLS designs just need to store a single model for online operation, their MS requirements are negligible, which are indicated in Table 1 by the symbol '–'.

In terms of online computational complexity, Table 1 shows that the RPLS, MWPLS and CoJIT all consume very little online $\mathrm{CPT}^{\mathrm{online}}$. This is because these three soft sensors require neither online modeling nor searching for the query sample's nearest neighbors. By comparison, the online $\mathrm{CPT}^{\mathrm{online}}$ consumed by the LPLS-APSP soft sensor is significantly larger, as it requires us to perform a model adaptation to choose a most appropriate local model. However, the online $\mathrm{CPT}^{\mathrm{online}}$ consumed by the LPLS-APSP is only half of the online $\mathrm{CPT}^{\mathrm{online}}$ required by the JITLPLS, which need to online construct a local model. The advantage of the LPLS-APSP over the JITLPLS by avoiding online constructing a local model is particular significant in the scenario where the PLS model requires a large number of latent variables.

The computational complexity of off-line soft sensor design was also investigated, and the last column of Table 1 lists the CPU time spent by each soft sensor design on the offline parameter optimization, denoted by $\mathrm{CPT}^{\mathrm{opt}}$, using the PSO algorithm which iterated 50 times with the number of particles set to ten times of the number of the parameters to be optimized. It can be seen that the LPLS-APSP requires considerably more offline $\mathrm{CPT}^{\mathrm{opt}}$ than the first four methods in offline parameter optimization. This is owing to the following two aspects. Firstly, the APSP continuously identifies newly emerged local model regions and detect redundant local models. As explained previously, this APSP does not require real-time implementation. Although it does not affect the online computational efficiency, it does increase offline computational requirements. Secondly, the number of parameters to be optimized is six, larger than those of the first four methods. However, the offline computational complexity of the proposed LPLS-APSP design is significantly lower than that of the CoJIT design. For the CoJIT method, in order to sufficiently localize the process, the window moving width is set to 1, and this results in a huge amount of local models. Note that in terms of real-time operation of a soft sensor, it is the online computational complexity, represented by $\mathrm{CPT}^{\mathrm{online}}$, that really matters. The offline computational complexity of $\mathrm{CPT}^{\mathrm{opt}}$ is only of secondary importance.

4.2. Industrial debutanizer column process

The debutanizer distillation column is a part of the desulfuring and naphtha splitter plant where propane and butane are removed as overheads from the naphtha steam [2] as shown in Fig. 8. Thus, one of its main tasks is to minimize butane content at the bottom of the column, and the butane content measurement is normally obtained by the gas chromatography with a large measurement delay. Therefore, a soft sensor for online estimating the concentration of butane is desirable. Several hardware sensors are installed in the debutanizer column for obtaining secondary variables, which are indicated with yellow circles in Fig. 8. The detailed descriptions of these input variables are given in Table 2.

The debutanizer column dataset containing 2394 samples, which is available at [50], came from a real industrial chemical process, and it has become a benchmark dataset for evaluating the performance of adaptive soft sensors [15]. We selected 600 samples evenly from the first half of the entire data set to form the historical training data set, while the second half of the data set was evenly divided into two parts, with one part served as the validation data set and the other as the test data set. The samples of the validation data set were also served as the measured samples at the online operation stage and added into the historical data set to simulate the real-life operation situation. The

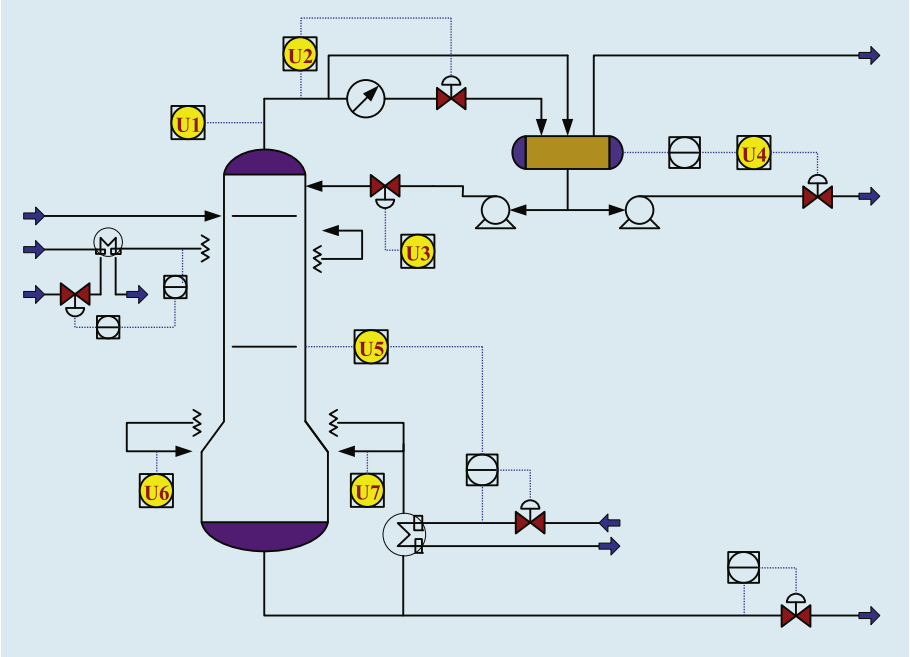

Fig. 8. Block diagram of the debutanizer column.

PLS model structure was determined by the expert knowledge and the consideration of process dynamics [2], and can be expressed as

$$\hat{y}(t-k) = f(u_1(t), u_2(t), \cdots, u_5(t), u_5(t-1), u_5(t-2), u_5(t-3), (u_6(t)+u_7(t))2, y(t-4), y(t-5), y(t-6), \quad (31)$$

where $\hat{y}(t)$ represents the predicted value of the butane concentration at time t, and $y(t-k)$ denotes the analytical value of the butane concentration measured by the gas chromatography at time $t-k$ with k being a positive integer.

The PSO was again utilized to optimize the algorithmic parameters of all the soft sensors based on the validation data set. With the search ranges for the parameters of each soft sensor as given in Appendix A, the obtained design parameters are as follows.

a) LSSVM: the Gaussian kernel width was 13.97, and the regularization parameter was 1.2×10^5.
b) JITLPLS: 150 nearest samples were chosen as the query sample's neighbors and the number of latent variables LV_{PLS} was 12.
c) RPLS: the forgetting factor was 0.963 and $LV_{PLS} = 12$.
d) MWPLS: the moving window length was 240 and $LV_{PLS} = 12$.
e) CoJIT: the window moving width was set to 1, the window size was 250, and the coefficient for constructing the correlation index was 9.38×10^{-7}. In addition, LV_{PLS} and LV_{PCA} were 12 and 10, respectively.
f) LPLS-APSP: $W = 90$, $LV_{PLS} = 10$, $K = 16$, $\phi = 69.9$, $\alpha = 0.691$ and $\beta = 0.235$. Additionally, α_t and α_χ were set to 0.05 in advance.

Table 2
Detailed descriptions of the input variables for soft sensing of the debutanizer column.

Variable	Description
u_1	Top temperature
u_2	Top pressure
u_3	Reflux flow
u_4	Flow to next process
u_5	6th tray temperature
u_6	Bottom temperature
u_7	Bottom temperature

Performance comparisons in terms of scatter plots between the LPLS-APSP and the other methods are illustrated in Fig. 9, while Table 3 quantitatively compares the performance of all the online soft sensors. From Fig. 9a, it can be seen that the prediction results of the LSSVM are biased when the target variable is less than 0.2 and greater than 0.6, due to the time-varying nonlinear characteristics of the debutanizer column process. By contrast, the data points of the LPLS-APSP are located more tightly and evenly along the diagonal line within the whole operating range, implying no estimation bias and a smaller estimation variance. Like the LPLS-APSP, the JITLPLS, RPLS, MWPLS and CoJIT are capable of avoiding the estimation bias, as can be seen from Fig. 9b to e. The scatter plots of Fig. 9b to e together with the prediction accuracy measures listed in Table 3 also show the advantage of the LPLS-APSP design over these other online soft sensors, in terms of prediction accuracy. In addition, Table 3 indicates that the proposed LPLS-APSP design constructs a total of 36 local models. Among these 36 local models, 10 are identified at the online operation stage. For this debutanizer column process, again the LPLS-APSP design requires a significantly smaller number of local models as well as considerably lower memory cost than the CoJIT based design.

As discussed in Section 3.1, in the proposed APSP strategy, we determine whether two predicted residuals are significantly different or not by considering both the mean and variance information based on the t-test and χ^2-test, which is different from the way of using only the mean value information proposed in [39]. We now illustrate why employing both the t-test and χ^2-test is better than only adopting the t-test. For this purpose, we denote the proposed LPLS-APSP considering both the mean and variance information as the LPLS-APSP1, while a similar LPLS-APSP strategy only considering the mean information is denoted as the LPLS-APSP2. With the significance levels fixed to 0.05, Fig. 10a and b investigate the influence of the window size to the achievable predicted RMSE on the test data set and the number of local models identified, respectively, for the both design strategies. Note that when the window size changes, other design parameters are fixed to their corresponding values optimized by the PSO based on the validation data set. Observe from Fig. 10a that the predicted RMSE performance over the test data set based on the LPLS-APSP1 design is consistently smaller than that obtained by the LPLS-APSP2 design across the whole range of window sizes investigated. As can be seen from Fig. 10b, for the design

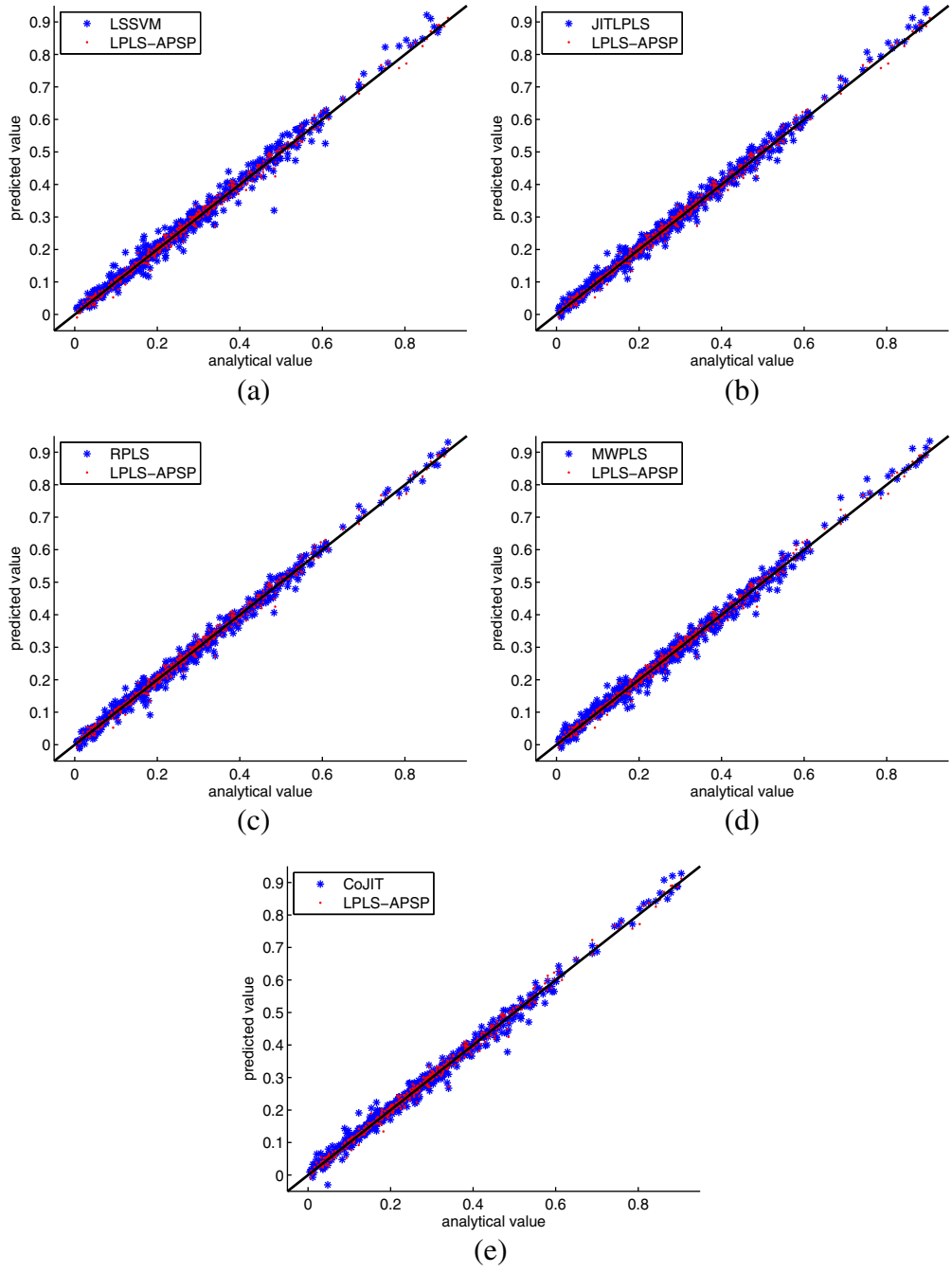

Fig. 9. Scatter plot comparison over the test data set of the debutanizer column process by the online soft sensors: (a) the LSSVM and LPLS-APSP, (b) the JITLPLS and LPLS-APSP, (c) the RPLS and LPLS-APSP, (d) the MWPLS and LPLS-APSP, and (e) the CoJIT and LPLS-APSP.

considering only the mean information, the window size has little influence to the number of local models needed. In addition, the local model size of the LPLS-APSP2 is always smaller than that of the LPLS-APSP1. This phenomena together with the predicted RMSE performance of Fig. 10a indicate that using only the mean value information is insufficient for detecting the deterioration of a model, since the hypothesis of H_{mean} may remain valid even thought the hypothesis H_{std} is already invalid. The results of Fig. 10 thus confirm that our APSP scheme of utilizing both the t-test and χ^2-test is more reliable and accurate in identifying the actual local models required.

Table 3
Performance comparison of various online soft sensors over the test data set of the debutanizer column process.

Soft sensor	RMSE	RR (%)	MAE	MN	MS (byte)
LLSVM	0.0215	23.4	0.163	1	–
JITLPLS	0.0175	13.5	0.067	1	–
RPLS	0.0163	19.1	0.091	1	–
MWPLS	0.180	23.8	0.081	1	–
CoJIT	0.0170	21.6	0.105	945	144,585
LPLS-APSP	0.0120	14.9	0.069	36	468

5. Conclusions

We have developed a novel adaptive soft sensing method under the generic local learning framework, referred to as the LPLS-APSP, which is capable of reliably and effectively handling highly nonlinear and time-varying industrial processes. Our original contribution has been twofold. Firstly, we have proposed a new APSP strategy based on both the t-test

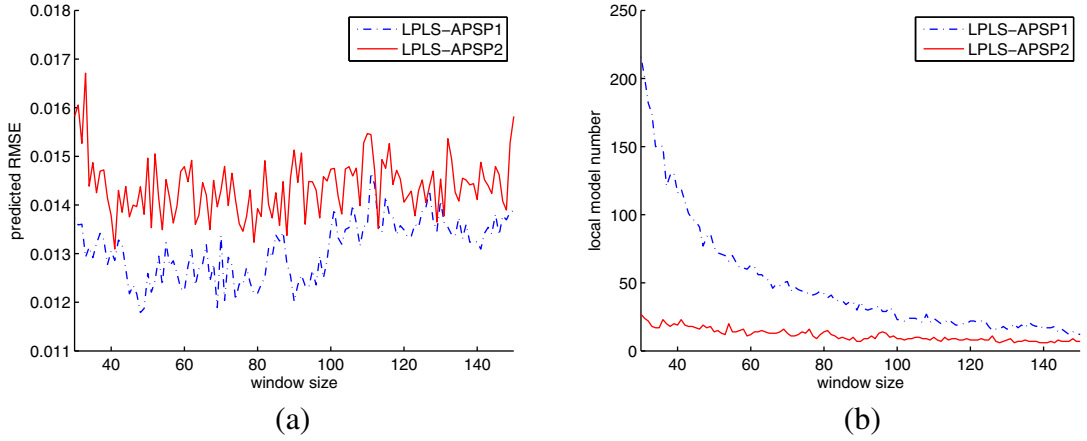

Fig. 10. Influence of the window size on: (a) the predicted RMSE and (b) the local model number, for the LPLS-APSP1 which employs both the t-test and χ^2-test and the LPLS-APSP2 which adopts the t-test only.

and χ^2-test by utilizing both the mean and variance information together, which is capable of reliably and efficiently identifying and updating the local PLS model set. Secondly, we have derived a novel model adaptation criterion for online local model switching, which effectively and efficiently exploits the prediction error information for the newest measured sample as well as mines the neighborhood information of the query sample. In the extensive simulation investigation involving two chemical processes, our proposed LPLS-APSP based soft sensor has been shown to significantly enhance prediction accuracy, while improving the online computational efficiency and maintaining low memory cost, in comparison to several existing adaptive soft sensors.

Conflict of interest

There is no conflict of interest.

Acknowledgment

This work was supported by National Nature Science Foundation of China (Grant No. 61273160) and the Fundamental Research Funds for the Central Universities of China (Grant Nos. 14CX06067A, 13CX05021A and 10CX04046A).

Appendix A. Search ranges for the parameters of each soft sensor

We determine the parameters of each soft sensor using the PSO to minimize the predicted RMSE over the validation dataset. Hence the search range of each parameter needs to be set properly for the PSO algorithm. Search ranges of some parameters can be directly obtained according to their physical meanings or interpretations. For example, according to Eq. (30), we know that the dimension of the input vector is 4 for the CSTR process, while through Eq. (31) we see that the dimension of the input vector is 12 for the debutanizer column process. Therefore, the search ranges for the numbers of the latent variables of the PLS model are obviously $\{1, 2, 3, 4\}$ and $\{1, 2, \cdots, 12\}$ for the CSTR process and the debutanizer column process, respectively. Search ranges of many parameters however are difficult to set properly just according to their physical meanings. We then set the search ranges of these parameters by trial-and-error. We now explain what we meant by trial-and-error. For example, consider setting the search range of the scaling factor $\phi > 0$ for our LPLS-APSP in the case of the CSTR. We initially set its search range to $(0, 10]$, and run a tentative optimization process by the PSO with a few iterations. When the tentative optimization process is finished, we find that the 'optimized' value of ϕ is 10, which reaches the upper bound of its search range. So we enlarge the search range to $(0\ 100]$ and repeat the above tentative optimization process. This time the 'optimized' value of ϕ is inside the search range. Therefore, we set the search range for ϕ to be $(0\ 100]$. We now detail how the search ranges for the parameters of each soft sensor are determined, specifically, by *physical interpretation* or by *trial-and-error*.

a) LSSVM: the search ranges for the Gaussian kernel width and regularization parameter are determined by trial-and-error.
b) JITPLS: the search range for the neighborhood size is determined by trial-and-error, while the search range for the number of latent variables in the PLS model, LV_{PLS}, is determined by physical interpretation.
c) RPLS: the search ranges for the forgetting factor and LV_{PLS} are determined by physical interpretation.
d) MWPLS: the search range for the window size is determined by trial-and-error, while the search range for LV_{PLS} is determined by physical interpretation.
e) CoJIT: the search range for the window size is determined by trial-and-error, while the search ranges for the other three parameters, the coefficient for constructing the correlation index, LV_{PCA} and the number of latent variables in the PCA model, LV_{PCA}, are determined by physical interpretation.
f) LPLS-APSP: the search ranges for the window size W, the neighborhood size of query sample K and the scaling factor ϕ are determined by trial-and-error, while the search ranges for the two weighting parameters, α and β, are determined by physical explanation.

The resulting search ranges for the parameters of each soft sensor are listed below. For the CSTR process:

a) LSSVM: the search ranges for the Gaussian kernel width and regularization parameter are $[0.05, 5]$ and $[1, 5000]$, respectively.
b) JITPLS: the search range for the neighborhood size is $\{6, 7, \cdots, 50\}$ and $LV_{PLS} \in \{1, 2, 3, 4\}$.
c) RPLS: the search range for the forgetting factor is $(0, 1]$, while $LV_{PLS} \in \{1, 2, 3, 4\}$.
d) MWPLS: the search range for the window size is $\{8, 9, \cdots, 50\}$, while $LV_{PLS} \in \{1, 2, 3, 4\}$.
e) CoJIT: the search ranges for the window size and correlation index coefficient are $\{8, 9, \cdots, 50\}$ and $[0, 1]$, respectively, while $LV_{PLS} \in \{1, 2, 3, 4\}$ and $LV_{PCA} \in \{1, 2, 3\}$.
f) LPLS-APSP: $W \in \{8, 9, \cdots, 50\}$, $LV_{PLS} \in \{1, 2, 3, 4\}$, $K \in \{1, 2, \cdots, 50\}$, $\phi \in (0, 100]$, $\alpha \in [0, 1]$, and $\beta \in [0, 1]$.

For the debutanizer column process:

a) LSSVM: the search ranges for the Gaussian kernel width and regularization parameter are $[0.05, 50]$ and $[1, 2 \times 10^5]$, respectively.
b) JITPLS: the search range for the neighborhood size is $\{20, 21, \cdots, 200\}$, while $LV_{PLS} \in \{1, 2, \cdots, 12\}$.

c) RPLS: the search range for the forgetting factor is (0, 1], while $LV_{PLS} \in \{1, 2, \cdots, 12\}$.
d) MWPLS: the search range for the window size is {20, 21, ···, 250}, while $LV_{PLS} \in \{1, 2, \cdots, 12\}$.
e) CoJIT: the search ranges for the window size and correlation index coefficient are {20, 21, ···, 300} and [0, 1], respectively, while $LV_{PLS} \in \{1, 2, \cdots, 12\}$ and $LV_{PCA} \in \{1, 2, \cdots, 11\}$.
f) LPLS-APSP: $W \in \{20, 21, \cdots, 150\}$, $LV_{PLS} \in \{1, 2, \cdots, 12\}$, $K \in \{1, 2, \cdots, 50\}$, $\phi \in (0, 100]$, $\alpha \in [0, 1]$, and $\beta \in [0, 1]$.

References

[1] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Comput. Chem. Eng. 33 (4) (2009) 795–814.
[2] L. Fortuna, S. Graziani, A. Rizzo, M.G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, London, Springer-Verlag, 2007.
[3] J. Deng, L. Xie, L. Chen, S. Khatibisepehr, B. Huang, F.W. Xu, A. Espejo, Development and industrial application of soft sensors with on-line Bayesian model updating strategy, J. Process Control 22 (3) (2013) 317–325.
[4] Y. Liu, Z.L. Gao, P. Li, H.Q. Wang, Just-in-time kernel learning with adaptive parameter selection for soft sensor modeling of batch processes, Ind. Eng. Chem. Res. 51 (11) (2012) 4313–4327.
[5] H.G. Tian, X.M. Tian, X.G. Deng, Soft sensor for polypropylene melt index based on improved orthogonal least squares, Proc. 8th WCICA (Jinan, China), July 6–9, 2010 2010, pp. 5881–5885.
[6] A.K. Pani, H.K. Mohanta, A survey of data treatment techniques for soft sensor design, Chem. Prod. Process. Model. 6 (1) (2011) 1–21.
[7] Z.Q. Ge, Z.H. Song, Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples, AIChE J. 57 (8) (2011) 2109–2118.
[8] K. Warne, G. Prasad, S. Rezvani, L. Maquire, Statistical and computational intelligence techniques for inferential model development: a comparative study evaluation and a novel proposition for fusion, Eng. Appl. Artif. Intell. 17 (8) (2004) 871–885.
[9] H.J. Galicia, Q.P. He, J. Wang, A reduced order soft sensor approach and its application to continuous digester, J. Process Control 21 (4) (2011) 489–500.
[10] J. Yu, Multiway Gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes, Ind. Eng. Chem. Res. 51 (40) (2012) 13227–13237.
[11] D.M. Himmelblau, Accounts of experiences in the application of artificial neural networks in chemical engineering, Ind. Eng. Chem. Res. 47 (16) (2008) 5782–5796.
[12] F.H. Wu, T.Y. Chai, Soft sensing method for magnetic tube recovery ratio via fuzzy systems and neural networks, Neurocomputing 73 (13–15) (2010) 2489–2497.
[13] J. Yu, A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses, Comput. Chem. Eng. 41 (2012) 134–144.
[14] S.N. Zhang, F.L. Wang, D.K. He, R.D. Jia, Real-time product quality control for batch processes based on stacked least-squares support vector regression models, Comput. Chem. Eng. 36 (2012) 217–226.
[15] P. Kadlec, R. Grbić, B. Gabrys, Review of adaptation mechanisms for data-driven soft sensors, Comput. Chem. Eng. 35 (1) (2011) 1–24.
[16] S.N. Zhang, F.L. Wang, D.K. He, R.D. Jia, Online quality prediction for cobalt oxalate synthesis process using least squares support vector regression approach with dual updating, Control. Eng. Pract. 21 (10) (2013) 1267–1276.
[17] J. Tang, W. Yu, T.Y. Chai, L.J. Zhao, On-line principal component analysis with application to process modeling, Neurocomputing 82 (2012) 167–178.
[18] J.L. Liu, D.-S. Chen, J.-F. Shen, Development of self-validating soft sensors using fast moving window partial least squares, Ind. Eng. Chem. Res. 49 (22) (2010) 11530–11546.
[19] S.J. Qin, Recursive PLS algorithms for adaptive data modeling, Comput. Chem. Eng. 22 (4–5) (1998) 503–514.
[20] W.M. Shao, X.M. Tian, P. Wang, Online learning soft sensor method based on recursive kernel algorithm for PLS, Chin. J. Chem. Eng. 63 (9) (2012) 2887–2891.
[21] K. Fujiwara, M. Kano, S. Hasebe, A. Takinami, Soft-sensor development using correlation-based just-in-time modeling, AIChE J. 55 (7) (2009) 1754–1764.
[22] Y. Liu, C.L. Li, Z.L. Gao, A novel unified correlation model using ensemble support vector regression for prediction of flooding velocity in randomly packed towers, J. Ind. Eng. Chem. 20 (3) (2014) 1109–1118.
[23] J.L. Liu, On-line soft sensor for polyethylene process with multiple production grades, Control. Eng. Pract. 15 (7) (2007) 769–778.
[24] Y.F. Fu, H.Y. Su, Y. Zhang, J. Chu, Adaptive soft-sensor modeling algorithm based on FCMISVM and its application in PX adsorption separation process, Chin. J. Chem. Eng. 16 (5) (2008) 746–751.
[25] Y. Lv, J.Z. Liu, T.T. Yang, D.L. Zeng, A novel least squares support vector machine ensemble model for NOx emission prediction of a coal-fired boiler, Energy 55 (2013) 319–329.
[26] J. Yu, Online quality prediction of nonlinear and non-Gaussian chemical processes with shifting dynamics using finite mixture model based Gaussian process regression approach, Chem. Eng. Sci. 82 (2012) 22–30.
[27] S. Khatibisepehr, B. Huang, F.W. Xu, A. Espejo, A Bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry, J. Process Control 22 (10) (2012) 1913–1929.
[28] R. Grbić, D. Slišković, P. Kadlec, Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models, Comput. Chem. Eng. 58 (2013) 84–97.
[29] C. Cheng, M.S. Chiu, A new data-based methodology for nonlinear process modeling, Chem. Eng. Sci. 59 (13) (2004) 2801–2810.
[30] K. Chen, J. Ji, H.Q. Wang, Y. Liu, Z.H. Song, Adaptive local kernel-based learning for soft sensor modeling of nonlinear processes, Chem. Eng. Res. Des. 89 (10) (2011) 2117–2124.
[31] Y.Q. Liu, D.P. Huang, Y. Li, Development of interval soft sensors using enhanced just-in-time learning and inductive confidence predictor, Ind. Eng. Chem. Res. 51 (8) (2012) 3356–3367.
[32] Y. Liu, Z.L. Gao, J.H. Chen, Development of soft sensors for online quality prediction of sequential-reactor-multi-grade industrial processes, Chem. Eng. Sci. 102 (2013) 602–612.
[33] S. Kim, R. Okajima, M. Kano, S. Hasebe, Development of soft-sensor using locally weighted PLS with adaptive similarity measure, Chemom. Intell. Lab. Syst. 124 (2013) 43–49.
[34] K. Chen, Y. Liu, Adaptive weighted relevant sample selection of just-in-time learning soft sensor for chemical processes, Proc. 10th IEEE ICCA (Hangzhou, China), June 12–14, 2013 2013, pp. 810–815.
[35] J.S. Zeng, L. Xie, C.H. Gao, J.J. Sha, Soft sensor development using non-Gaussian just-in-time modeling, Proc. CDC-ECC 2011 (Orlando, FL), Dec. 12–15, 2011 2011, pp. 5868–5873.
[36] L. Xie, J.S. Zeng, C.H. Gao, Novel just-in-time learning-based soft sensor utilizing non-Gaussian information, IEEE Trans. Control Syst. Technol. 22 (1) (2014) 360–369.
[37] W.D. Ni, S.K. Tan, W.J. Ng, S.D. Brown, Localized, adaptive recursive partial least squares regression for dynamic system modeling, Ind. Eng. Chem. 51 (23) (2012) 8025–8039.
[38] W.D. Ni, S.D. Brown, R.L. Man, A localized adaptive soft sensor for dynamic systems modeling, Comp. Chem. Sci. 111 (2014) 250–363.
[39] P. Kadlec, B. Gabrys, Local learning-based adaptive soft sensor for catalyst activation prediction, AIChE J. 57 (5) (2011) 1288–1301.
[40] Y. Liu, J.H. Chen, Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes, J. Process Control 23 (6) (2013) 793–804.
[41] J. Yu, K.L. Chen, M.M. Rashid, A Bayesian model averaging based multi-kernel Gaussian process regression framework for nonlinear state estimation and quality prediction of multiphase batch processes with transient dynamics and uncertainty, Chem. Eng. Sci. 93 (2013) 96–109.
[42] P. Kadlec, B. Gabrys, Adaptive on-line prediction soft sensing without historical data, Proc. 2010 IJCNN (Barcelona, Spain), July 18–23, 2010 2010, pp. 1–8.
[43] M. Kano, M. Ogawa, The state of the art in chemical process control in Japan: good practice and questionnaire survey, J. Process Control 20 (9) (2010) 969–982.
[44] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[45] B.S. Dayal, J.F. MacGregor, Improved PLS algorithms, J. Chemometr. 11 (1) (1997) 73–85.
[46] W.M. Shao, X.M. Tian, H.L. Chen, Adaptive anti-over-fitting soft sensing method based on local learning, Preprints of 10th IFAC Int. Symp. Dynamics and Control of Process Systems (Mumbai, India), Dec. 18–20, 2013 2013, pp. 415–420.
[47] Z.Q. Ge, Z.H. Song, A comparative study of just-in-time learning based methods for online soft sensor modeling, Chemom. Intell. Lab. Syst. 104 (2010) 306–317.
[48] C.G. Atkeson, A.W. Moore, S. Schaal, Locally weighted learning, Artif. Intell. Rev. 11 (1–5) (1997) 11–73.
[49] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.
[50] http://www.springer.com/engineering/control/book/978-1-84628-479-3.