Contents lists available at ScienceDirect

Chemical Engineering Research and Design

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

Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models

Weiming Shao, Xuemin Tian[*]

College of Information and Control Engineering, China University of Petroleum, 66#, Changjiang West Road, Huangdao District, Qingdao, 266580, China

ARTICLE INFO

Article history:
Received 24 September 2014
Received in revised form 23 December 2014
Accepted 18 January 2015
Available online 28 January 2015

Keywords:
Adaptive soft sensor
Adaptive localization
Local partial least squares
Selective ensemble learning
Bayesian inference

ABSTRACT

This paper proposes an adaptive soft sensing method based on selective ensemble of local partial least squares models, referring to as the SELPLS, for quality prediction of nonlinear and time-varying chemical processes. To deal with the process nonlinearity, we partition the process state into local model regions upon which PLS models are constructed, through a statistical hypothesis testing based adaptive localization procedure. Two main delightful advantages of this localization strategy are that, redundant local models can be effectively detected and deleted and the local model set can be easily augmented online without retraining from scratch. In addition, a local model weighting mechanism is proposed to adaptively differentiate the contributions of local models by explicitly quantifying their generalization abilities for the current process dynamics. Finally, the selective ensemble learning strategy combines partial local models instead of all available models through Bayesian inference, which is able to reach a good equilibrium between the prediction bias and variance. The proposed SELPLS based soft sensor is applied to a simulated continuous stirred tank reactor and a real-life industrial sulfur recovery unit. Extensive simulation results demonstrate the effectiveness of the proposed scheme in contrast with several state-of-the-art adaptive soft sensing approaches.

1. Introduction

In chemical processes, the product quality is often related to some key variables, such as the melt index in the process of polypropylene production (Xu and Liu, 2014). It is important and desirable to accurately obtain the value of those quality-related variables in real-time, since infrequent and inaccurate measurements of those variables may result in poor control performances, huge production losses and even safety hazards (Pani and Mohanta, 2011). Conventionally those variables are mostly measured through offline analysis or online analyzes. However, both of the two manners are unacceptably expensive and produce large measurement delay, which make them unsatisfactory to meet the increasing demands of modern industrial applications (Ge and Song, 2014; Kim et al., 2013a; Deng et al., 2013; Liu et al., 2012; Kano and Fujiwara, 2013).

As an alternative, soft sensor has been more and more reported being employed to deal with the above mentioned problems in recent years (Kadlec et al., 2009; Kano and Ogawa, 2010; Kano and Fujiwara, 2013). The essence of soft sensor is to construct a predictive model which can describe the mathematical relationship between the primary variables, i.e., hard-to-measure variables such as those quality-related variables, and the secondary variables, i.e., easy-to-measure variables such as flow rate, pressure and temperature. The

[*] Corresponding author. Tel.: +86 532 8698 3467.
E-mail addresses: swmupc@163.com (W. Shao), tianxm@upc.edu.cn, tianxm_upc@hotmail.com (X. Tian).

http://dx.doi.org/10.1016/j.cherd.2015.01.006
0263-8762/© 2015 The Institution of Chemical Engineers. Published by Elsevier B.V. All rights reserved.

values of primary variables are inferred by the predictive model using secondary variables. Therefore, soft sensor is easy to maintain and can deliver real-time estimation of the primary variables. In other words, soft sensor is money-saving and can significantly reduce the measurement delay.

During the past two decades, a variety of data-driven algorithms have been applied to develop soft sensors, with the aid of wide popularization of the distributed control system (DCS) which can collect a huge number of process data in modern chemical processes (Ge et al., 2014). These soft sensing approaches include linear modeling methods such as principal component regression (PCR) (Ge and Song, 2011; Tang et al., 2012a) and partial least squares (PLS) (Galicia et al., 2011; Shao et al., 2012; Tang et al., 2012b), and nonlinear modeling techniques such as artificial neural networks (ANN) (Himmelblau, 2008; Wu and Chai, 2010), support vector machines (SVMs) (Yu, 2012a; Zhang et al., 2012), Gaussian process regression (GPR) (Ni et al., 2012a; Grbić et al., 2013), etc. Although these nonlinear soft sensing methods have been extensively reported in the literature since two decades ago, the linear modeling methods seem more practically useful. A recent questionnaire survey in Japan indicates that linear modeling approaches account for more than 90% of the soft sensors used in chemical plants (Kano and Ogawa, 2010; Kim et al., 2013a). Among these linear modeling methods, PLS is becoming more and more popular in terms of practical application. The reason is not complicated. On one hand, PLS has transparent model structure that can be easily understood by engineers and process operators, and can handle the commonly existed data co-linearity and measurement noise in industrial processes that the multiple linear regression (MLR) cannot. On the other hand, compared with PCR, PLS is a supervised feature extraction method, which is able to take advantage of the important output information. Nevertheless, the linear PLS fails to perform well when modeling complex systems with strong nonlinearities.

Another critical problem that needs to be solved in soft sensor development is the performance deterioration of soft sensors after they are deployed into real-life operation due to drifts of process characteristics (Kadlec et al., 2011; Kim et al., 2013a; Kano and Fujiwara, 2013). These drifts may result from catalyst deactivation, mechanical aging, change of operating conditions, variation of feed properties, or even climatic change, etc. Therefore, developing adaptive soft sensors to adapt them to new process dynamics automatically is necessary for prolonging their life time in industrial applications. Moving window (MW) models (Kaneko et al., 2009; Zhang et al., 2013; Liu et al., 2010), recursive models (Dayal and MacGregor, 1997; Qin, 1998; Tang et al., 2012a; Shao et al., 2012) and just-in-time learning (JITL) models (Chen et al., 2011; Kim et al., 2013b; Liu et al., 2012; Liu and Chen, 2013; Fujiwara et al., 2009) are commonly applied to achieve such target and successful applications of these methods have been reported. However, there are some limitations associated with these methods that need to be analyzed. For example, MW and recursive models suffer from difficulty in dealing with abrupt variations such as change of set point value, because they usually could not track the process dynamics until sufficient new samples from the new operational condition have been collected. In addition, most recursive methods belong to the category of fitting a single 'global' model, which may not perform well due to the strong nonlinearities in most complex chemical processes (Fujiwara et al., 2009; Liu et al., 2012). Those JITL models are potential to solve the shortcomings of MW models and recursive models, yet their prediction performance is not always satisfactory, as the distance based JITL models ignore the correlation between process variables, while the correlation based JITL models may sometimes fail to select appropriate models as analyzed by Shao et al. (2014).

In addition to MW, recursive and JITL models, time difference (TD) models were also proposed recently to tackle performance deterioration of soft sensors (Kaneko and Funatsu, 2011a). Instead of directly modeling the relationship between the secondary variables and primary variables, it models the relationship between the time difference values of them. One most distinct advantage of TD models is that it can cope with certain drifts of process variables without online updating of the soft sensor model. However, conventional TD model cannot account for nonlinear processes, unless it combines the physical model or some nonlinear modeling techniques (Kaneko and Funatsu, 2011b). Yet in some applications accurate physical model of the underlying process is not available. Deep discussions and analyses about MW, JITL and TD models as well as their applicable scenarios can be found in Kaneko and Funatsu (2013).

Compared with the global model based soft sensors, local learning based soft sensors employ the philosophy of 'divide and conquer' and train a set of local models, each of which serves as the expert for one specific operating region of the process such that the process nonlinearity can be modeled (Kadlec and Gabrys, 2011). In addition, the time-varying characteristics, including abrupt changes, can also be handled through the online adaptations of local models and their contributions to the final estimation. Therefore, this paper focuses on developing adaptive soft sensor under the local learning framework.

The first key task in local learning is the localization which partitions the entire process state into local model regions so that the local models can be constructed on the corresponding data subsets. The most commonly used solutions in the soft sensing field to this problem are the clustering based methods, such as fuzzy C-means (FCM) (Liu, 2007; Fu et al., 2008; Liu et al., 2014) and expectation maximization (EM) based finite mixture models (Grbić et al., 2013; Khatibisepehr et al., 2012; Yu, 2012b). The main problems of these methods are that appropriate number of clusters are not easy to determine and it is difficult to add new clusters online when new process modes appear. Although the latter issue can be alleviated by updating the offline constructed local models with recursive (Khatibisepehr et al., 2012) or moving window (Grbić et al., 2013) techniques, 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.

Recently, another class of approaches was 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. For example, in Fujiwara et al. (2009), Xie et al. (2014) and Kaneko and Funatsu (2014a), the process state is partitioned by a moving window with a fixed moving width. In the work of Ni et al. (2012b, 2014), the entire dataset was repeatedly partitioned several times whenever an adaptation is triggered. Although manually setting the window size is avoided, this approach may become time-consuming as newly measured samples are continuously accumulated. Kadlec and Gabrys (Kadlec and

Gabrys, 2011) proposed an adaptive way for splitting the entire data set into local 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. Shao et al. (Shao et al., 2014) extended this approach to the multi-output process and continued to extract new local regions online. However, the two *t*-test based partitioning strategies ignore the negative influence of the variance of the predicted residual. As a result, the null hypothesis may remain valid even when the process characteristics have changed. In order to overcome this problem, Shao et al. (Shao et al., 2013) used not only *t*-test but also χ^2-test to detect the performance deterioration and the simulation results demonstrated the functionality of the consideration of the variance information. Nevertheless, a potential problem associated with this approach is that the number of local models is ever increasing as the plant is continuously operating, since no mechanism for detecting and discarding redundant local models was considered.

The subsequent part of developing local learning-based soft sensors is the prediction combination using the constructed local models in the previous step. Generally there are two strategies that can be applied at the prediction level. The first one is the best model selection way, which selects an 'optimal' local model to predict the output of the query sample, i.e., the unknown sample according to some criterions such as fuzzy membership (Fu et al., 2008), posterior probability (Liu and Chen, 2013; Yu et al., 2013), correlation index (Fujiwara et al., 2009), predicted errors (Ni et al., 2012b, 2014; Shao et al., 2014) and so on. However, this strategy may sometimes have large prediction variance because the secondary variables may be contaminated with large measurement noise and therefore exactly choosing the proper local model for the query sample is quite difficult. In contrast, the other prediction combination mechanism, which is focused in this paper, is to build an ensemble of all local models with convex weights. Such an ensemble learning strategy has been proven theoretically and practically useful to reduce the prediction variance in the machine learning field (see Kadlec, 2009 for a review). The fuzzy membership and posterior probability are also popular for assigning combination weights to local models for developing ensemble based soft sensors under local learning framework (Liu, 2007; Lv et al., 2013; Liu et al., 2014; Yu, 2012b; Khatibisepehr et al., 2012; Kadlec and Gabrys, 2011; Ge et al., 2014; Kaneko and Funatsu, 2014a).

Although these weighting strategies have shown their effectiveness, the prediction performance of this sort of soft sensors can be further improved from two aspects. On one hand, most weighting strategies do not fully make use of the output information of historical samples or explicitly quantify each local model's ability of describing the current process dynamics. So overcoming this issue will help to weight local models more reasonably. On the other hand, while combining all local models can be theoretically expected to decrease the prediction variance (Kadlec, 2009), the prediction bias has the risk to increase. Therefore, selective ensemble learning which combines only a subset of all local models at hand may be an advisable choice to reach an acceptable balance between the two terms, i.e., the bias and variance. Zhou et al. have proved that ensemble many may be better than ensemble all (Zhou et al., 2002). Zhao et al. (Zhao et al., 2012) and Tang et al. (Tang et al., 2013) have developed selective ensemble learning based soft sensors for estimating the effluent quality in wastewater treatment plants and load parameters of ball mill in grinding processes, respectively. The results of their case studies showed the benefits of selective ensemble learning. Hence we believe that incorporating the selective ensemble learning strategy into the local learning framework to develop adaptive soft sensors is also promising. However, the above mentioned selective learning strategies are not suitable for adaptive soft sensing and literature survey indicates that there is a rare publication carried out towards such an objective.

Motivated to solve the above summarized issues remaining in the localization step and the model combination step in local learning, in this paper, based on our previous work (Shao et al., 2013, 2014), we propose an adaptive soft sensing approach for time-varying and nonlinear chemical processes. As the core of the proposed soft sensor is the selective ensemble of local partial least squares models, it is referred to as the SELPLS. Compared with other local learning based adaptive soft sensors, our novel contributions lie in the following aspects.

(1) An adaptive localization procedure which improves the work of Shao et al. (2013) is presented for partitioning the process state into local model regions, where redundant local models are automatically detected and discarded. Such localization approach accounts for the effects of both the variance and mean of the predicted residuals. Moreover, new unique local modes can be easily identified online without retraining from scratch.
(2) A novel selective ensemble learning mechanism is proposed and incorporated into the framework of developing local learning based adaptive soft sensors. Specifically, local models' abilities of describing the current process dynamics are quantified by fully exploiting the sample information. Then, the local model with low generalization ability is filtered by a threshold value which is limited to the range of 0–1, and the selected ones are combined through the Bayesian inference to give a final estimation of the query sample.
(3) An extensive performance evaluation of the proposed SELPLS based soft sensor is conducted and comparisons between SELPLS and several other state-of-the-art PLS based adaptive soft sensing methods are carried out, both of which use a simulated continuous stirred tank reactor and a real sulfur recovery unit.

The remaining of this paper is organized as follows. Section 2 presents a brief review of the local PLS model. In Section 3, the proposed SELPLS based soft sensor development is detailedly described and analyzed, including the adaptive localization strategy and the selective ensemble learning strategy. In Section 4, case studies on two chemical processes are conducted and performance comparisons between the SELPLS and other state-of-the-art PLS based adaptive soft sensing approaches are reported. Finally, our work is concluded in Section 5.

2. Review of local partial least squares

The partial least squares (PLS) is adopted to construct local models, due to its previously discussed merits and wide population (Kano and Fujiwara, 2013; Kim et al., 2013a). The problem of PLS in dealing with nonlinear processes can be tackled by the local learning strategy, i.e., developing locally valid PLS (LPLS) models.

Denote the n-th samples of input and output variables as $\mathbf{x}_n = [\mathbf{x}_{1,n}\ \mathbf{x}_{2,n} \cdots \mathbf{x}_{m,n}]^{\mathrm{T}} \in R^m$ and $\mathbf{y}_n = [\mathbf{y}_{1,n}\ \mathbf{y}_{2,n} \cdots \mathbf{y}_{p,n}]^{\mathrm{T}} \in R^p$, respectively, where m is the number of secondary variables, p is the number of primary variables, and $()^{\mathrm{T}}$ denotes the vector or matrix transpose operator. Assume we have a local model region with data set $\{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X} = (\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_N)^{\mathrm{T}} \in R^{N\times m}$ and $\mathbf{Y} = (\mathbf{y}_1\ \mathbf{y}_2 \cdots \mathbf{y}_N)^{\mathrm{T}} \in R^{N\times p}$ are the input and output matrices with N samples, respectively. Note that we omitted the local model index for notational simplicity. For example, for the l-th local model, the associated data set is $\{\mathbf{X}^{(l)}, \mathbf{Y}^{(l)}\}$ with N_l samples. Without causing any confusion, we drop the index l here for simplifying notations.

The PLS algorithm projects $\mathbf{X}$ and $\mathbf{Y}$, which have been mean-centered and appropriately scaled, onto respective latent variables according to

$$\mathrm{X} = \mathrm{TP}^{\mathrm{T}} + E_{\mathrm{X}} \qquad (1)$$

$$\mathrm{Y} = \mathrm{UQ}^{\mathrm{T}} + E_{\mathrm{Y}} \qquad (2)$$

where, $\mathbf{T} = (\mathbf{t}_1\ \mathbf{t}_2\ \cdots\ \mathbf{t}_A) \in R^{N\times A}$ and $\mathbf{U} = (\boldsymbol{u}_1\ \boldsymbol{u}_2\ \cdots\ \boldsymbol{u}_A) \in R^{N\times A}$ are the score matrices, $\mathbf{P} = (\boldsymbol{p}_1\ \boldsymbol{p}_2\ \cdots\ \boldsymbol{p}_A) \in R^{m\times A}$ and $\mathbf{Q} = (\boldsymbol{q}_1\ \boldsymbol{q}_2\ \cdots\ \boldsymbol{q}_A) \in R^{P\times A}$ are the loading matrices, of $\mathbf{X}$ and $\mathbf{Y}$, respectively, while A represents the number of latent variables, and $\mathbf{E}_{\mathrm{X}} \in R^{N\times m}$ and $\mathbf{E}_{\mathrm{Y}} \in R^{N\times P}$ are the input and output residual matrices. While the PLS model' external relationship is shown by Eqs. (1) and (2), its linear internal relationship is given by

$$U = \mathrm{TB} + \mathrm{F} \qquad (3)$$

where, $\mathbf{B} = \mathrm{diag}\{b_1\ b_2\ \cdots\ b_A\}$ is a diagonal matrix with $b_i = \mathbf{t}_i^{\mathrm{T}}\mathbf{u}_i/\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i$ $(i = 1, 2, \cdots, A)$, and $\mathbf{F}$ is the residual matrix. The estimate of $\mathbf{Y}$ can be represented by $\mathbf{Y}^* = \boldsymbol{TBQ}^{\mathrm{T}}$. Usually the explicit regression coefficient matrix of the PLS model, $\mathbf{C}^{\mathrm{PLS}}$ that satisfies $\mathbf{Y} = \mathbf{XC}^{\mathrm{PLS}} + \mathbf{E}$ where $\mathbf{E}$ is the residual matrix, is computed by iteratively decomposing $\mathbf{X}$ and $\mathbf{Y}$ and a particularly popular method for this purpose is the nonlinear iterative PLS (NIPALS) algorithm. The detailed procedure of the NIPALS algorithm can be found in Qin (1998). The only parameter in PLS model that needs to be pre-set is the number of latent variables, A. Usually the optimal value of A can be established by the well-known k-fold cross validation (CV) procedure, for example $k = 10$ (Kadlec and Gabrys, 2011).

In order to cope with the time-varying characteristics, recursive PLS (RPLS) can be formulated by "merging" the old model represented by $\mathbf{P}$, $\mathbf{B}$ and $\mathbf{Q}$ with the newly measured data sample $\{\mathbf{x}_{\mathrm{new}}, \mathbf{y}_{\mathrm{new}}\}$. A new PLS model can be obtained by applying PLS algorithm to the following expanded input and output matrices (Qin, 1998):

$$\mathrm{X}_{\mathrm{new}} = \begin{bmatrix} \lambda\mathbf{P}^{\mathrm{T}} \\ \mathbf{x}_{\mathrm{new}} \end{bmatrix}, \quad \mathrm{Y}_{\mathrm{new}} = \begin{bmatrix} \lambda\mathbf{BQ}^{\mathrm{T}} \\ \mathbf{y}_{\mathrm{new}} \end{bmatrix} \qquad (4)$$

where, λ is called the forgetting factor that specifies the adaptation speed. Generally speaking, proper λ relates to finding an optimal trade-off between learning new and forgetting old information (Kadlec et al., 2011), which is an intractable work without deep insight of the process. In addition, a necessary condition able to perform this recursive operation is that A must be large enough such that $\|E_{\mathrm{X}}\|_{\mathrm{F}}$ is sufficiently small (Qin, 1998), where $\|\ \|_{\mathrm{F}}$ refers to the matrix Frobenius norm. Therefore, the required number of latent variables of RPLS may differ from the one obtained by the cross validation procedure.

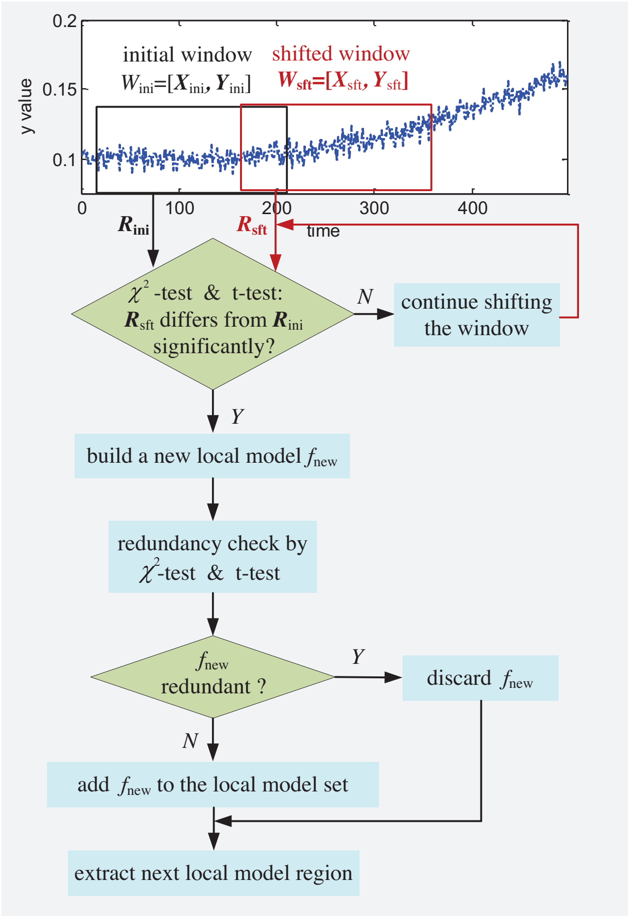

0.2
0.15
0.1
y value
initial window
W_{ini}=[X_{ini}, Y_{ini}]
shifted window
W_{sft}=[X_{sft}, Y_{sft}]
0
100
200
300
400
time
R_{ini}
R_{sft}
χ^2-test & t-test: R_{sft} differs from R_{ini} significantly?
N
continue shifting the window
Y
build a new local model f_{new}
redundancy check by χ^2-test & t-test
f_{new} redundant ?
Y
discard f_{new}
N
add f_{new} to the local model set
extract next local model region

Fig. 1 – Schematic diagram of the adaptive localization based on statistical hypothesis testing and moving window.

3. Proposed adaptive soft sensor development

The proposed SELPLS based adaptive soft sensor development methodology consists of two operation stages, namely the offline operation stage and the online operation stage. At the offline operation stage, an adaptive localization method splits the entire process state into local model regions and as a result, concise local PLS model set can be obtained. Besides no need of presetting the number of local models, another delightful advantage of such localization manner is that as the plant is operating continuously to provide consecutive-time samples, local regions that represent new process modes can be easily extracted without retraining from scratch. At the online operation stage, for estimating the output of each query sample, the offline-prepared local models are selectively combined. The one to select as well as its contribution to the final model output depends on its ability to describe the current process state, which is quantified in this paper. Now we detail the adaptive localization method and selective ensemble learning strategy.

3.1. *Adaptive localization*

This adaptive localization strategy uses the statistical hypothesis testing and moving window as its tools and considers the variances of both first-order and second-order information of the predicted residuals, which is depicted in Fig. 1. In this paper, we consider single-output processes only, that is,

$p = 1$. Initially, a data window W_{ini} is set with W consecutive-time samples denoted by $W_{ini} = \{\mathbf{X}_{ini}, \mathbf{Y}_{ini}\}$, upon which a local PLS model f_{ini} has been constructed, where $\mathbf{X}_{ini} \in R^{W \times m}$ and $\mathbf{Y}_{ini} \in R^{W \times 1}$ representing the input and output matrix, respectively. Assume at this point we have already identified and stored L $(L \geq 1)$ local models $\{f_l\}_{l=1}^{L}$, which represent L local model regions $\{W_l\}_{l=1}^{L}$, each containing W consecutive-time samples. Without loss of generality, the local model f_L is identified at the previous local model region extraction, and we have $W_{ini} = W_L$ and $f_{ini} = f_L$. Subsequently, W_{ini} is shifted one sample step ahead and a shifted window W_{sft} is obtained with the data set $W_{sft} = \{\mathbf{X}_{sft}, \mathbf{Y}_{sft}\}$.

The predicted residuals for $\mathbf{Y}_{ini}$ and $\mathbf{Y}_{sft}$ based on f_{ini}, i.e. $\mathbf{R}_{ini}$ and $\mathbf{R}_{sft}$, can be respectively calculated according to

$$R_{ini} = f_{ini}(X_{ini}) - Y_{ini} \tag{5}$$

$$R_{sft} = f_{ini}(X_{sft}) - Y_{sft} \tag{6}$$

If $\mathbf{R}_{sft}$ are not significantly different from $\mathbf{R}_{ini}$, the performance of f_{ini} over W_{sft} can be regarded as the same as that over W_{ini}. That is, the samples in W_{ini} and W_{sft} come from the same local process state. Consequently, there is no need to build a new local model based on W_{sft} and the window will be continuously shifted forward and new $\mathbf{R}_{sft}$ is calculated. Once $\mathbf{R}_{sft}$ significantly deviates from $\mathbf{R}_{ini}$, it indicates that a new local process state that is different from the one represented by W_{ini} is identified. Then a new local PLS model f_{new} can be built based on the latest W_{sft}. Now the critical issue that needs to be solved is how to judge whether $\mathbf{R}_{sft}$ evidentially differs from $\mathbf{R}_{ini}$ or not. In our previous work (Shao et al., 2013), this problem is transformed into inspecting if the means and variances of the two residuals are significantly different or not based on t-test and χ^2-test. In this paper we also adopt these two statistical tests. However, the process state partition strategy of Shao et al. (2013) is further improved in this paper.

Assuming that both $\mathbf{R}_{ini}$ and $\mathbf{R}_{sft}$ follow normal distribution, we first construct T statistic and χ^2 statistic as follows

$$T = \sqrt{W}\left(\bar{R}_{sft} - \bar{R}_{ini}\right)/\sigma_{sft} \tag{7}$$

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

where, $\bar{R}_{ini}$ and σ_{ini} are the mean and standard deviation of the distribution from which $\mathbf{R}_{ini}$ are drawn, while $\bar{R}_{sft}$ and σ_{sft} are the mean and standard deviation of the distribution from which $\mathbf{R}_{sft}$ are drawn. $\bar{R}_{ini}$ and σ_{ini} can be estimated using the samples of W_{ini}. Normally, $\bar{R}_{ini}$ is essentially 0. Similarly, $\bar{R}_{sft}$ and σ_{sft} can be estimated using the samples of W_{sft}. According to the standard statistical theory, when the hypothesis

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

holds, the T statistic follows t distribution with the freedom degree $W - 1$. Likewise, the χ^2 statistic follows χ^2 distribution with freedom degree of $W - 1$ if the hypothesis

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

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

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

the hypothesis (9) and hypothesis (10) are valid according to the t-test and χ^2-test, respectively. That is, from the statistical point of view, $\bar{R}_{sft}$ does not significantly differ from $\bar{R}_{ini}$ and σ_{sft} and σ_{ini} are considered identical as long as the conditions in Eq. (11) are satisfied. The λ_t in Eq. (11) represents the threshold value of the T statistic for the given significance level α_t, i.e. 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, i.e. Prob $\{\chi^2 < \lambda_\chi\} = 1 - \alpha_\chi$.

Thus, $\mathbf{R}_{sft}$ does not significantly differ from $\mathbf{R}_{ini}$ only when both of the conditions in Eq. (11) are fulfilled. Otherwise, a new local process state represented by W_{sft} and different from the one represented by W_{ini} is identified. A new local model denoted as f_{new} can be constructed based on the newly identified local data set $W_{sft} = \{\mathbf{X}_{sft}, \mathbf{Y}_{sft}\}$. Then, we set $W_{ini} = W_{sft}$ and $f_{ini} = f_{new}$ and repeat the above process to identify the next local model.

The above localization process is continued at the online operation stage such that novel local model regions that are not covered by the historical data can be extracted. Moreover, one can see that the online augmentation of local model set requires no retraining from scratch. However, a potential problem associated with this approach is that the number of local models is ever increasing as the plant is continuously operating and provides sequential data every day. Simply removing the oldest local model may not be the best to avoid this problem, as the oldest local model may represent important local process state which is different from the newly extracted one. Actually, the growing local model set may contain many similar local models, because the violation of Eq. (11) only means the newly identified local process state $\{W_{sft}, f_{new}\}$ differs from $\{W_L, f_L\}$, but whether $\{W_{sft}, f_{new}\}$ is different from other preserved local models, i.e., $\{W_l, f_l\}$ for $1 \leq l \leq L - 1$, is not checked. If $\{W_{sft}, f_{new}\}$ also differs from $\{W_l, f_l\}$ for $1 \leq l \leq L - 1$, then it represents a truly new local process state and $\{W_{sft}, f_{new}\}$ is added to the local model set, which causes the number of local models is increased by one. However, if there exists one local process state $\{W_l, f_l\}$ similar to $\{W_{sft}, f_{new}\}$, where $l \in \{1, 2, \cdots, L - 1\}$, then either $\{W_{sft}, f_{new}\}$ or $\{W_l, f_l\}$ can be regarded as redundant. From the perspective of model maintenance, $\{W_{sft}, f_{new}\}$ is discarded while $\{W_l, f_l\}$ is retained to save the plant from constantly replacing old models, which makes the local model number do not increase. In this paper, a mechanism is developed to perform this task of detecting and discarding redundant local models, which is also based on statistical hypothesis testing.

The predicted errors for $\mathbf{Y}_{sft}$ based on f_{new} and f_l are respectively calculated by

$$R_{new} = f_{new}(X_{sft}) - Y_{sft} \tag{12}$$

$$R_l = f_l(X_{sft}) - Y_{sft} \tag{13}$$

where, $l \in \{1, 2, \cdots, L - 1\}$. Again, under the assumption that $\mathbf{R}_{new}$ and $\mathbf{R}_l$ follow normal distribution, the T statistic and χ^2

statistic for the l-th local model can be constructed according to

$$T_l = \sqrt{W}\left(\bar{R}_l - \bar{R}_{\text{new}}\right)/\sigma_l \tag{14}$$

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

where, $\bar{R}_{\text{new}}$ and σ_{new} are the mean and standard deviation of the distribution where $\mathbf{R}_{\text{new}}$ is drawn, which can be estimated using the samples of $\mathbf{R}_{\text{new}}$, while $\bar{R}_l$ and σ_l represent the mean and standard deviation of the distribution where $\mathbf{R}_l$ is drawn, which are also estimated. If the following hypotheses

$$H_{\text{mean}}^l : \bar{R}_l = \bar{R}_{\text{new}} \tag{16}$$

$$H_{\text{std}}^l : \sigma_l = \sigma_{\text{new}} \tag{17}$$

are valid, T_l and χ_l^2 follow the t-distribution and χ^2-distribution, respectively, both of which have the freedom degree $W-1$. Thus, if there exists an $l \in \{1, 2, \cdots, L-1\}$ that satisfies

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

the two hypotheses (16) and (17) are both valid, and $\mathbf{R}_{\text{new}}$ and $\mathbf{R}_l$ are considered identical. In this scenario, f_{new} is discarded. Otherwise, we add f_{new} to the local model set. The detailed procedure for implementing this localization strategy is as follows.

Initialization: Collect an initial dataset W_{ini} with W consecutive-time samples, and construct a PLS model f_{ini} based on W_{ini}. Calculate $\mathbf{R}_{\text{ini}}$, and estimate $\bar{R}_{\text{ini}}$ and σ_{ini}. Set $L=1$ and $\{W_L, f_L\} = \{W_{\text{ini}}, f_{\text{ini}}\}$. Then set $W_{\text{sft}} = W_L$ and go to Step (1).

***Step* (1)**: Collect a new data sample from the plant[1], shift W_{sft} one sample step ahead to obtain new W_{sft}, calculate $\mathbf{R}_{\text{sft}}$ with Eq. (6) and f_{ini}, and estimate the corresponding $\bar{R}_{\text{sft}}$ and σ_{sft}.

***Step* (2)**: Construct the T statistic and χ^2 statistic according to Eqs. (7) and (8), respectively.

***Step* (3)**: Check if both the conditions of Eq. (11) hold. If yes, return to step (1); otherwise go to the next step.

***Step* (4)**: Build a new PLS model f_{new} upon W_{sft}, compute $\mathbf{R}_{\text{new}}$ by Eq. (12), and estimate the associated $\bar{R}_{\text{new}}$ and σ_{new}.

***Step* (5)**: Calculate $\mathbf{R}_l$ with Eq. (13) and estimate $\bar{R}_l$ and σ_l for $1 \le l \le L-1$.

***Step* (6)**: Construct T_l statistic and χ_l^2 statistic according to Eqs. (14) and (15), respectively, for $1 \le l \le L-1$.

***Step* (7)**: Check if there exists an l that satisfies both conditions of Eq. (18). If yes, discard $\{W_{\text{new}}, f_{\text{new}}\}$; otherwise let $L = L+1$ and add $\{W_{\text{sft}}, f_{\text{new}}\}$ as $\{W_L, f_L\}$ to the local model set.

***Step* (8)**: Set $W_{\text{ini}} = W_{\text{sft}}, f_{\text{ini}} = f_{\text{new}}, \bar{R}_{\text{ini}} = \bar{R}_{\text{new}}$ and $\sigma_{\text{ini}} = \sigma_{\text{new}}$ and return to Step (1) to identify the next local model.

Remark: in practice, according to (Kadlec and Gabrys, 2011) the significance levels α_t and α_χ in the statistical hypotheses can be set to small values such as 0.05, and their effects can be compensated partly by adjusting the window size W. Besides, it is worth pointing out that even though new local model regions are continuously extracted at the online operation stage, all the operations involved at this stage can be implemented 'offline'. Consequently, the online computational efficiency is not really an issue here.

3.2. *Selective ensemble of local partial least squares models*

In the traditional ensemble learning, the outputs of all local models built at the offline stage need to be combined to get the predicted y-value given a query $\mathbf{x}_q$ at the online operation stage. That is,

$$f(x_q) = \sum\nolimits_{l=1}^{L} w_l(x_q) f_l(x_q) \tag{19}$$

where, $f_l(\mathbf{x}_q)$ is the output of the l-th local model given $\mathbf{x}_q$, $w_l(\mathbf{x}_q)$ is the combination weight assigned to the l-th local model which satisfies $w_l(\mathbf{x}_q) \ge 0$ and $\sum\nolimits_{l=1}^{L} w_l(\mathbf{x}_q) = 1$, L is the number of identified local models and $f(\mathbf{x}_q)$ is the final estimation of the true output of $\mathbf{x}_q$, i,e, y_q. In what follows, without causing any confusion, $f(\mathbf{x}_q)$, $f_l(\mathbf{x}_q)$ and $w_l(\mathbf{x}_q)$ are simplified as f, f_l and w_l, respectively. The expected generalization error for $\mathbf{x}_q$, Err $(\mathbf{x}_q)$, based on f can be expressed as

$$\text{Err}(\mathbf{x}_q) = E\left[(f - y_q)^2\right] \tag{20}$$

where, $E[]$ represents expectation with respect to all possible realizations of the training data set of constant size. According to the bias/variance decomposition (Geman et al., 1992), Err$(\mathbf{x}_q)$ can be decomposed as

$$\begin{aligned}\text{Err}(\mathbf{x}_q) &= \underbrace{\left(E\left[f - y_q\right]\right)^2}_{(\text{Bias}(f))^2} + \underbrace{E\left[(f - E[f])^2\right]}_{\text{Variance}(f)} \\ &= \underbrace{\left(E\left[\sum\nolimits_{l=1}^{L} w_l f_l - y_q\right]\right)^2}_{(\text{Bias}(f))^2} + E\underbrace{\left[\left(\sum\nolimits_{l=1}^{L} w_l f_l - E\left[\sum\nolimits_{l=1}^{L} w_l f_l\right]\right)^2\right]}_{\text{Variance}(f)}\end{aligned} \tag{21}$$

Since the localization strategy detects and discards redundant local models, we assume that the correlation between any two local models is very small. Then the variance term of Err$(\mathbf{x}_q)$ can be approximated as

$$\begin{aligned}&\text{Variance}(f) \approx \sum\nolimits_{l=1}^{L} w_l^2\left(E[f_l^2] - E^2[f_l]\right) \\ &= \sum\nolimits_{l=1}^{L} w_l^2 \text{Variance}(f_l) \\ &\le \sum\nolimits_{l=1}^{L} w_l \text{Variance}(f_l)\end{aligned} \tag{22}$$

We can see that the ensemble variance is no more than the weighted mean of the variances of the ensemble members. Further, if the combination weights are identical, the ensemble variance is only 1/L of the averaged individual ensemble member variance. These analyses indicate that the ensemble learning indeed can help to reduce the predicted variance. However, the bias term in the decomposition of Eq. (21) does not imply that the predicted bias can be reduced by ensemble. On the contrary, the predicted bias may be increased by building an ensemble of all local models constructed at the offline stage. The reason lies in the fact that these local models are specialized on their own model regions due to the localization strategy in Section 3.1. Therefore, we could not expect each local model to provide unbiased estimation for $\mathbf{x}_q$. As a result, the bias term in Eq. (21) may be much larger than zero, which

[1] If the plant's operational data have already been collected and stored, simply shift W_{sft} one sample step ahead.

may still cause high generalization error. Therefore, in order to obtain a low estimation error for $\mathbf{x}_q$, neither of the bias or variance term in Eq. (21) could be high. To this end, we propose a selective ensemble strategy, which builds an ensemble of a few individual models that have good generalization ability for the current process dynamics instead of combing all local models at hand. In this paper, based on the work of Shao et al. (2014), the generalization ability of the l-th local model for the current process dynamics, $g(l)$, is quantified as

$$\begin{aligned} g(l) &= \frac{1}{\gamma e_0^{(l)} + (1-\gamma)\sum_{k=1}^{K} s_k e_k^{(l)} / \sum_{k=1}^{K} s_k} \\ &= \frac{1}{\gamma J_1^{(l)} + (1-\gamma) J_2^{(l)}} \end{aligned} \tag{23}$$

In Eq. (23), K is the number of the query sample $\mathbf{x}_q$'s nearest neighbors according to the Euclidean distance, which are denoted as $\{\mathbf{x}_k, y_k\}$ for $1 \le k \le K$, and

$$e_0^{(l)} = (f_l(\mathbf{x}_0) - y_0)^2 \tag{24}$$

$$e_k^{(l)} = (f_l(\mathbf{x}_k) - y_k)^2, \ 1 \le k \le K \tag{25}$$

where, $(\mathbf{x}_0, y_0)$ represents the latest measured sample in the historical dataset, $0 < \gamma < 1$ is the weighting parameter which is computed as

$$\gamma = \exp\left(-\rho d(\mathbf{x}_q, \mathbf{x}_0)\right) \tag{26}$$

where, $d(\cdot,\cdot)$ denotes the Euclidean distance metric and $\rho \ge 0$ is the scaling parameter, while s_k defines the similarity of $\mathbf{x}_q$ and $\mathbf{x}_k$, which is also computed using the exponential way, i.e.,

$$s_k = \exp\left(-\psi d\left(\mathbf{x}_q, \mathbf{x}_k\right)\right) \tag{27}$$

where, $\psi \ge 0$ is the scaling parameter.

The utility and importance of considering both $J_1^{(l)}$ and $J_2^{(l)}$ in the denominator of the right side of Eq. (23) has already been demonstrated in Shao et al. (2014). An improvement for γ which balances the influence of $J_1^{(l)}$ and $J_2^{(l)}$ in this paper, is that γ is not fixed but adaptively determined by Eq. (26). We can infer that when the query sample $\mathbf{x}_q$ is located faraway from $\mathbf{x}_0$, γ becomes small and $J_2^{(l)}$ gets more emphasis, and vice versa. This is helpful for more effectively handling the abrupt change and transition process. Similarly, according to Eq. (27), the effects of distant neighbors of $\mathbf{x}_q$ are forced to be very small. Therefore, the neighborhood size, K, can be set to be sufficiently large and its influence can be compensated by adjusting the scaling parameter ψ.

Since the denominator of Eq. (23) represents predicted errors, larger value of g means better generalization ability. Consequently, local models with high g values are preferable. In order to facilitate the implementation of selective ensemble, $g(l)$ for $1 \le l \le L$ is normalized as

$$\tilde{g}(l) = \frac{g(l)}{\max\{g(1), g(2), \cdots, g(L)\}} \in (0, 1] \tag{28}$$

where, $\tilde{g}(l)$ is the normalization form of $g(l)$. For $\mathbf{x}_q$, the selective ensemble learning strategy selects a few local models that have better generalization abilities than a certain level. Thus, if we denote the index set of the selected local models as $L_S = \{l_1, l_2, \cdots, l_S\}$, L_S can be determined as

$$L_S = \left\{ l | \tilde{g}(l) \ge \delta \right\} \tag{29}$$

where, S represents the number of selected local models and δ is a preset threshold value. δ is an important parameter for the proposed scheme, as it determines how many local models are combined. The larger the δ is, the less local models are selected and vice versa. Fortunately, the normalization in Eq. (28) facilitates the selection of δ to some extent, as it can be limited to $[0\,1]$. If δ is set to 0 or 1, this selective ensemble learning strategy degenerates to the conventional ensemble learning strategy that combines all local models or the best model selection strategy that selects only one 'optimal' local model, respectively.

Actually, the selective ensemble learning strategy forces those local models which are not selected to be zeros to filer out their negative influences on the final prediction. Other weighting methods that can fulfill the same task are also allowed. However, finding an optimal weight for each local model in every prediction round is usually a very tough work.

By building an ensemble of the selected local models, we can compute the predicted y-value for the query sample $\mathbf{x}_q$ as

$$\begin{aligned} \hat{y}_q &= \sum_{j=1}^{S} w_{l_j} f_{l_j}(\mathbf{x}_q) \\ &= \sum_{j=1}^{S} P\left(f_{l_j} | \mathbf{x}_q\right) f_{l_j}(\mathbf{x}_q) \end{aligned} \tag{30}$$

Here, the weight of f_{l_j}, w_{l_j}, is set to $P\left(f_{l_j} | \mathbf{x}_q\right)$, which means the posterior probability of the l_j-th local model, f_{l_j}, given the query sample $\mathbf{x}_q$. According to the Bayesian inference (Khatibisepehr et al., 2012), $P\left(f_{l_j} | \mathbf{x}_q\right)$ is computed as

$$P\left(f_{l_j} | \mathbf{x}_q\right) = \frac{P(f_{l_j}) P\left(\mathbf{x}_q | f_{l_j}\right)}{\sum_{j=1}^{S} P(f_{l_j}) P\left(\mathbf{x}_q | f_{l_j}\right)} \tag{31}$$

where, $P(f_{l_j})$ is the prior probability that the current process dynamics can be characterized by f_{l_j}, and $P(\mathbf{x}_q | f_{l_j})$ is the likelihood that $\mathbf{x}_q$ is generated by the l_j-th local model region. In this paper, the prior probability of each f_{l_j} is set equal as processed in Kaneko and Funatsu (2014a), while $P(\mathbf{x}_q | f_{l_j})$ is determined as the generalization ability of f_{l_j}, that is

$$P\left(f_{l_j}\right) = 1/S, \quad P\left(\mathbf{x}_q | f_{l_j}\right) = \tilde{g}\left(l_j\right) \tag{32}$$

Finally, after some manipulations, we obtain a simple formula for computing $\hat{y}_q$:

$$\begin{aligned} \hat{y}_q &= \sum_{j=1}^{S} w_{l_j} f_{l_j}(\mathbf{x}_q) \\ &= \sum_{j=1}^{S} \tilde{g}(l_j) f_{l_j}(\mathbf{x}_q) / \sum_{j=1}^{S} \tilde{g}(l_j) \end{aligned} \tag{33}$$

The detailed procedures of implementing such selective ensemble learning to obtain the estimated y-value for the query sample $\mathbf{x}_q$ are summarized as follows.

Step (1): Select K nearest neighbors of $\mathbf{x}_q$ according to Euclidean distance metric from the historical data set, denote them as $\left\{\mathbf{x}_k, y_k\right\}_{k=1}^{K}$.

Step (2): Compute s_k $(1 \le k \le K)$ and γ according to Eqs. (27) and (26), respectively.

Step (3): Calculate $e_0^{(l)}$ and $e_k^{(l)}$ with the l-th local model $(1 \le l \le L)$ according to Eqs. (24) and (25), respectively. Here L means the total number of stored local models.

Step (4): Calculate $g(l)$ $(1 \le l \le L)$ according to Eq. (23) and then normalize it to $\tilde{g}(l)$ $(1 \le l \le L)$ according to Eq. (28).

Step (5): Determine the selected local models via Eq. (29), and calculate their weights according to Eq. (31).

Step (6): Estimate the y-value of $\mathbf{x}_q$ according to Eq. (33).

After presenting the principle and detailed realization procedures of the proposed soft sensing method, let us analyze the influence of model parameters which will be helpful for parameter tuning. In addition to the significant levels α_t and α_χ and the neighborhood size K, which can be set to default values as discussed before, there are four parameters that need to be manually determined, namely the window size W, the two scaling parameters ψ and ρ and the threshold value δ. Generally speaking, 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, a too small window size may cause model instability and unreliable 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). Therefore, too small window size should be avoided.

The function of the scaling parameter ψ in Eq. (27) is to differentiate the importance of the query sample's neighbors. When ψ is set to zero, all the selected neighbors have the same weights, while only the nearest neighbor takes effect with $\psi \to \infty$. And ρ decides how important the neighborhood information is. $\rho = 0$ means that $\gamma \equiv 1$ and the neighborhood information is ignored, while $\rho \to \infty$ implies that $\gamma \approx 0$ and the neighborhood information takes full responsibility for weighting the local models. It should be emphasized that the threshold value δ in Eq. (29) is an important parameter in our selective ensemble learning strategy, because it determines the degree of selective ensemble. When $\delta = 0$, all local models at hand are combined, while only one local model is selected for the quality prediction with $\delta = 1$. The selective degree will affect the equilibrium between the bias term and variance term demonstrated in Eq. (21). Normally the multi-tuning of those parameters is an intractable task, which may take too much effort and time if it is manually done. Therefore, we suggest completing such a task automatically, through some intelligent optimization methods such as the Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA).

4. Case studies

The effectiveness of our proposed SELPLS based adaptive soft sensor was verified through two chemical processes, namely a simulated continuous stirred tank reactor (CSTR) and a real-life sulfur recovery unit (SRU) process. For comparison purpose, the performance of several state-of-the-art and commonly used PLS based adaptive soft sensing methods was also provided and analyzed. These benchmark methods consist of the recursive PLS (RPLS) (Qin, 1998), the locally weighted PLS (LWPLS) (Kano and Fujiwara, 2013), the moving window PLS (MWPLS) (Liu et al., 2010) and the localized adaptive soft sensor (LASS) (Ni et al., 2014). The estimation accuracy is evaluated on the test dataset by several commonly used error indexes including the root mean squares error (RMSE), the relative RMSE (RRMSE) and the maximum absolute error (MAE), which are defined as

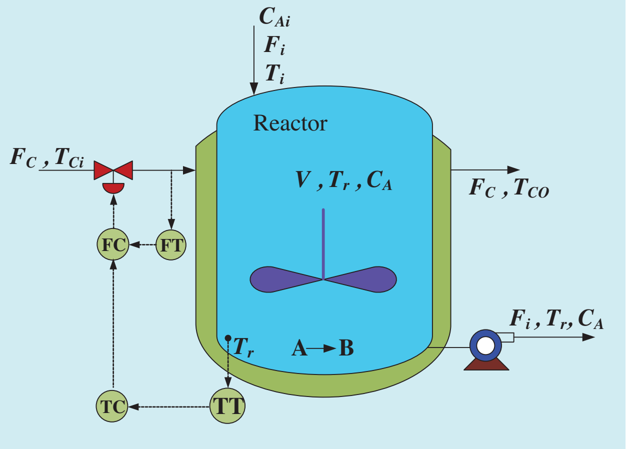


Fig. 2 – Schematic diagram of the CSTR with its control structure.

$$\mathrm{RMSE} = \sqrt{\sum_{t=1}^{N_t} (\hat{y}_t - y_t)^2 / N_t} \tag{34}$$

$$\mathrm{RRMSE} = \sqrt{\sum_{t=1}^{N_t} ((\hat{y}_t - y_t)/y_t)^2 / N_t} \times 100\% \tag{35}$$

$$\mathrm{MAE} = \max\left\{ \left| y_t - \hat{y}_t \right|, t = 1, 2, \cdots, N_t \right\} \tag{36}$$

where, y_t and $\hat{y}_t$ represent the true output and its predicted value of the t-th query sample, respectively; N_t denotes the number of the test samples. Additionally, the offline CPU time spent on the optimization process for parameter determination ($\mathrm{CPT}^{\mathrm{opt}}$) and the online consumed CPU time ($\mathrm{CPT}^{\mathrm{online}}$), averaged over 10 independent simulations, were utilized to evaluate the offline and online computational efficiency of each 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. *Continuous stirred tank reactor* (CSTR)

The schematic structure of an exothermic first-order CSTR, which has been widely used for testing adaptive soft sensors' performance (Khatibisepehr et al., 2012; Liu and Chen, 2013), is shown in Fig. 2.

In this CSTR, an irreversible reaction A → B takes place, whose dynamics can be mathematically described by

$$\frac{dC_A(t)}{dt} = \frac{F_i}{V}(C_{Ai} - C_A(t)) - k_0 C_A(t) \exp\left(-\frac{E}{RT_r}\right) \tag{37}$$

$$\frac{dT_r(t)}{dt} = \frac{F_i}{V}(T_i - T_r(t)) - \frac{\Delta H k_0 C_A(t)}{\rho C_p} \exp\left(-\frac{E}{RT_r}\right) + \frac{\rho_c C_{pc}}{\rho C_p V} F_c(t) \times \left(1 - \exp\left(\frac{-hA}{F_c(t)\rho C_p}\right)\right)(T_{ci} - T_r(t)) \tag{38}$$

The detailed description of this CSTR and its steady state operating conditions as well as the explanations of Eqs. (37) and (38) can be found in Khatibisepehr et al. (2012) which also showed that this CSTR is highly nonlinear. The residual

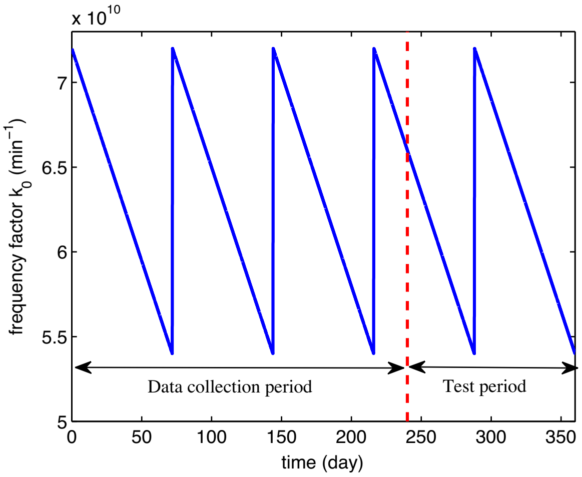


Fig. 3 – Periodical change of the frequency factor k_0.

concentration of A, i.e. $C_A(t)$, was the quality variable and thus 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. Similar to the settings in Fujiwara et al. (2009), throughout the entire simulation process, we periodically altered the set point of $T_r(t)$ to ±2 (K) every 10 days and meanwhile, in order to simulate the periodical decrease and recovery of the catalyst activation, the frequency factor k_0 was changing as shown in Fig. 3. These problem settings make this CSTR process not only highly nonlinear but also time-varying.

The sampling interval of the secondary variables, i.e. $F_c(t)$ and $T_r(t)$, was set to 1 min, but we assumed the quality variable, i.e. $C_A(t)$, was analyzed in the laboratory every 12 h and the analyzed value was referred to as the 'analytical value'. Therefore the sampling period for collecting the 'labeled' samples is half a day, and between every two adjacent labeled samples there are 719 'unlabeled' samples with the sampling period of 1 min. For capturing the process dynamics, the input vector was augmented with the sample at the previous sampling instant and thus the soft sensor model structure was determined as

$$C_A(k) = f\left(F_c\left(k-1\right), F_c\left(k-2\right), T_r\left(k-1\right), T_r\left(k-2\right)\right) \tag{39}$$

The entire simulation period was one year containing 720 analytical samples, among which the 480 samples of the first 240 days were used as training dataset and the rest 240 analytical samples in the test period of 120 days were employed as the validation data set for determining the algorithmic parameters of the soft sensor by cross validation. During the test period, there were large number of 'unlabeled' samples (approximately 34,5600) obtained at the sampling period of 1 min, which together with their actual output values formed the test dataset for evaluating the performance of these online soft sensors. Note that in practical situations, these 'unlabeled' samples' output values of C_A were actually unknown. In addition, both the primary variable and secondary variables were corrupted by Gaussian noises, and were scaled to within the range of [0.0, 1.0].

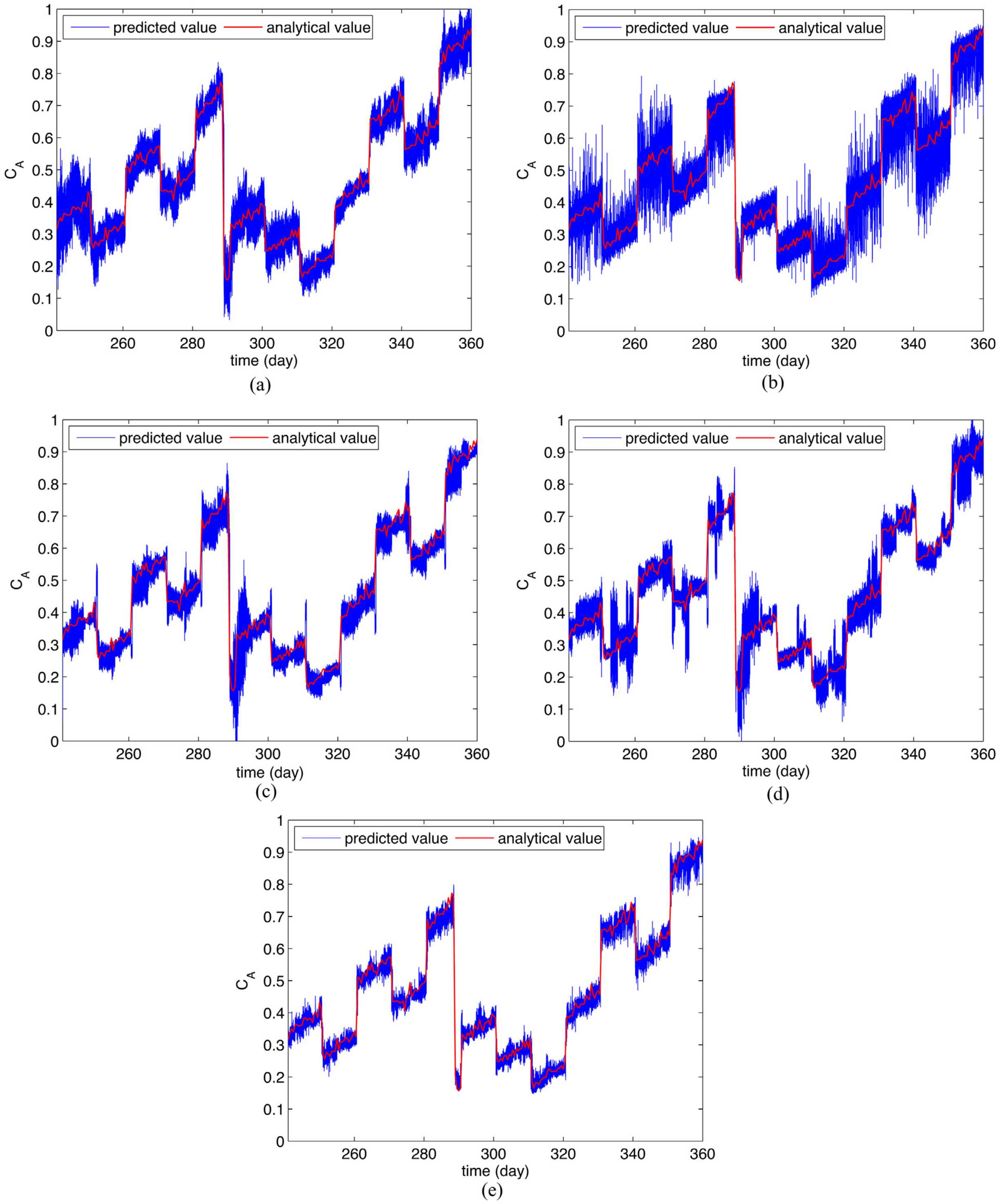
In the SELPLS, the latent variable number of PLS models (i.e., A) is determined by the 5-fold cross validation (CV), while other parameters are optimized by PSO, which minimized the predicted RMSE over the validation dataset. In the RPLS, both A and the forgetting factor λ are optimized by PSO. In the MWPLS, A is determined by 5-fold CV, and the window size is determined by enumeration method. In the LWPLS, the neighborhood size, A and the scaling parameters are optimized by PSO. In the LASS, A is determined by 5-fold CV. The candidates for those parameters are determined either by trial-and-error or by their physical interpretations. Note that in the PSO parameter setting, the iteration time is set to 50, and the number of particles is 10 times of the number of the optimized parameters. Specifically, the candidates of those parameters are listed as:

- RPLS: $A \in [1\,4]$, $\lambda \in (0\,1)$;
- LWPLS: $A \in [1\,4]$, neighborhood size $\in [10\,100]$, scaling parameter $\in [0.01\,10]$.
- MWPLS: $A \in [1\,4]$, window size $\in [8\,50]$;
- LASS: $A \in [1\,4]$;
- SELPLS: $A \in [1\,4]$, $W \in [8\,50]$, $\psi \in [0\,100]$, $\rho \in [0\,5]$, and $\delta \in [0\,1]$. These optimized algorithmic parameters of different soft sensors are summarized as follows.
- RPLS: $A = 2$ and $\lambda = 0.971$.
- LWPLS: The neighborhood size was 90, $A = 1$ and the scaling parameter was set to 0.115.
- MWPLS: The window size was set to 10 and A was determined by 5-fold CV.
- LASS: A was determined by 5-fold CV
- SELPLS: $W = 12$, $\psi = 39.0$, $\rho = 1.31$, $\delta = 0.426$. A was automatically determined by 5-fold CV. The neighborhood size was set to a sufficiently large value in advance, i.e. $K = 40$. Besides, the significance levels for both t-test and χ^2-test, i.e. α_t and α_χ, were set to 0.05 as suggested in Kadlec and Gabrys (2011).

In the LASS, other parameters such as the threshold value for the re-localization could be automatically determined. Thus there involved no manual work for parameter tuning in LASS. However, for avoiding unnecessary computational load, according to our insight into this CSTR process we set the maximum and minimum window size in each localization process of the LASS to 50 and 10, respectively. Note that the computer programs for realizing the PLS algorithm as well as the cross validation in the above five soft sensing methods all came from the PLS Toolbox which can be downloaded from http://www.eigenvector.com/software/pls_toolbox.htm.

Now the performance of the above mentioned adaptive soft sensors is ready to be compared. The estimated values for C_A obtained by these soft sensors based on the RPLS, the LWPLS, the MWPLS, the LASS and the proposed SELPLS are depicted in Fig. 4(a)–(e), respectively, where the analytical values of C_A are also shown for comparison. The abrupt change of C_A in Fig. 4(a)–(e) at around day 288 is caused by the catalyst activity recovery, while other step changes of C_A are caused by the change of set point value of the reaction temperature. Meanwhile, the estimation errors of these soft sensors are visualized in Fig. 5. Here the estimation errors are obtained by subtracting the true values from the corresponding predicted values of C_A. In addition, Table 1 quantitatively compares the performance of these soft sensors over the test data set with several performance measures. Note that the historical data set was augmented with the analytical values every 12 h at the online operation stage.

As can be seen from Fig. 4(a), the global method RPLS struggles to model this nonlinear and time-varying process well in most operation periods, especially when the catalyst activity was recovered as revealed in Fig. 5(a). The data in



(a)

(b)

(c)

(d)

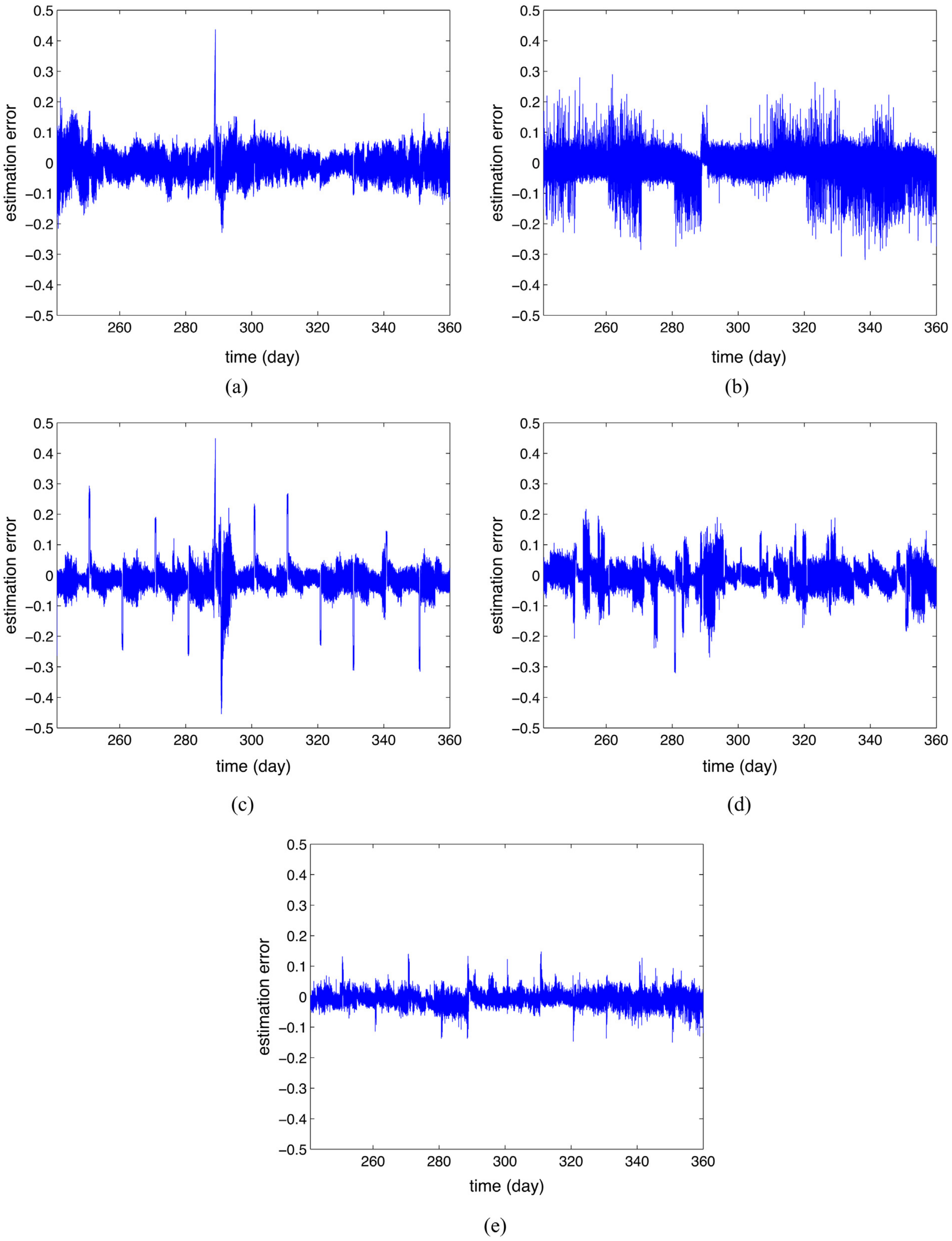
(e)

Fig. 4 – Predicted C_A over the test dataset achieved by various soft sensors based on: (a) the RPLS, (b) the LWPLS, (c) the MWPLS, (d) the LASS and (e) the SELPLS.

Table 1 – Quantitative performance over the test data set based on various soft sensors for the CSTR.

Soft sensor	RMSE	RRMSE (%)	MAE	$\mathrm{CPT}^{\mathrm{opt}}$(s)	$\mathrm{CPT}^{\mathrm{online}}$(s)
RPLS	0.0367	10.6	0.437	379.2	36.6
LWPLS	0.0285	7.4	0.318	633.4	272.8
MWPLS	0.0501	14.9	0.454	18.75	36.7
LASS	0.0393	10.2	0.320	0	36.7
SELPLS	0.0185	4.8	0.149	3078.8	147.6

Table 1 indicate that the LWPLS could improve the predicted accuracy a lot compared with RPLS, however, the predicted values of C_A and the estimation errors respectively depicted in Fig. 4(b) and Fig. 5(b) indicate that the performance of LWPLS is intuitively poor. Fig. 4(c) and Fig. 5(c) demonstrate that the MWPLS achieves higher estimation accuracy than the LWPLS does in relative steady-state operating conditions where the catalyst activity is decreasing slowly, implying that the MWPLS can handle the nonlinear and slow-changing process. Nevertheless, its prediction performance deteriorates significantly when the operation condition is undergoing a step change, such as the alteration of set point value of the



(a) (b) (c) (d) (e)

Fig. 5 – Comparison of the predicted errors for C_A over the test data set by various soft sensors based on: (a) the RPLS, (b) the LWPLS, (c) the MWPLS, (d) the LASS and (e) the SELPLS.

reacting temperature and the catalyst activity recovery. This is because the query sample is strange to the current PLS model under such a circumstance and only after a few analytical samples have been accumulated for the new operational state, can the newly updated PLS model recover from such a catastrophe. Such limitation of the MWPLS made it generate groups of large estimation errors and consequently receive the largest predicted RMSE, RRMSE and MAE among all soft sensing methods, which can been seen from Fig. 5(c) and Table 1.

We can infer from Fig. 4(d) and Fig. 5(d) that the estimation performance of LASS is unstable. When local models were selected properly, for example at the period around 286 day, the prediction accuracy was very high, but in many time periods such as that around 275 day, the estimation errors of the LASS are large, which indicates that inappropriate model adaptations occurred. The limitation of the model selection mechanism of the LASS as analyzed in Shao et al. (2014) may be responsible for the bad performance of the LASS.

By comprehensively analyzing the prediction results in Fig. 4, Fig. 5 and Table 1, we can conclude that the proposed SELPLS achieves the best overall prediction performance among all various soft sensing approaches. Specifically, the prediction accuracy of the SELPLS is consistently very satisfactory over all relative steady states compared with that of the LASS and LWPLS. In addition, although its prediction estimation fluctuates when the operating conditions were changed abruptly, the increase of the estimation error is rather smaller compared with that of the MWPLS. And the proposed soft sensor can very quickly adapt to the new process state without the need of accumulating several analytical samples of the new state. The good performance of the proposed soft sensor can be explained from three aspects. First, the nonlinearity of the CSTR can be handled by the proposed local learning strategy, which identified 88 locally valid PLS models. Secondly, the time-varying issues including both abrupt and slow changes are able to be solved by the reasonable quantification of each local model's generalization ability for the current process dynamics in Eq. (23). Third, the great decrease of prediction variance should give credit to the ensemble learning.

It is interesting to notice that comparisons between Fig. 4(a)–(d) indicate the estimation performance of the LWPLS seems worse than those of the RPLS, the MWPLS and the LASS, while the quantitative estimation errors in Table 1 demonstrate the LWPLS seems to achieve higher estimation accuracy than the RPLS, the MWPLS and the LASS. This can be explained from two aspects. On one hand, some y-values predicted by the LWPLS deviate from their true values too much, like 'pulses' protruding from the curve of 'predicted value', and meanwhile there are too many data points (about 60 mins × 24 h × 120days = 172,800 samples) and Fig. 4(b) is not wide enough, those predicted y-values that have large deviances (i.e., the 'pulses') are squeezed together and the gaps between them disappear in Fig. 4(b). As a result, the predicted performance of LWPLS shown in Fig. 4(b) seems bad in some time period, for example around day 345. On the other hand, the LWPLS can avoid large estimation errors during changes of operation conditions, i.e., the alteration of set-point of the reaction temperature and the recovery of catalyst activity. In contrast, clusters of large errors are generated during the changes of those operation conditions by the RPLS, the LASS and the MWPLS, which result in high estimation errors. That is why the estimation errors of the LWPLS listed in Table 1 are smaller than those of the RPLS, the LASS and the MWPLS.

The penultimate column of Table 1 shows the time spent on parameter optimization by each soft sensor. Note that the CPT^{opt} consumed by LASS is marked with zero, because the parameters of the LASS are automatically determined, which requires no offline parameter tuning. Apparently, the SELPLS requires much more time on parameter optimization than other soft sensors, which is a deficiency of the proposed soft sensor. There are two main reasons resulting in such phenomenon. On one hand, the SELPLS keeps extracting new local models and detecting and deleting redundant local models, which is not necessary in other soft sensing methods. One the other hand, for SELPLS, there are four parameters that need to be optimized by PSO, so the number of particles is more than those of other methods. Fortunately, the process of parameter optimization can be implemented offline, and the online computational efficiency of the SELPLS is barely affected, as shown in the last column of Table 1.

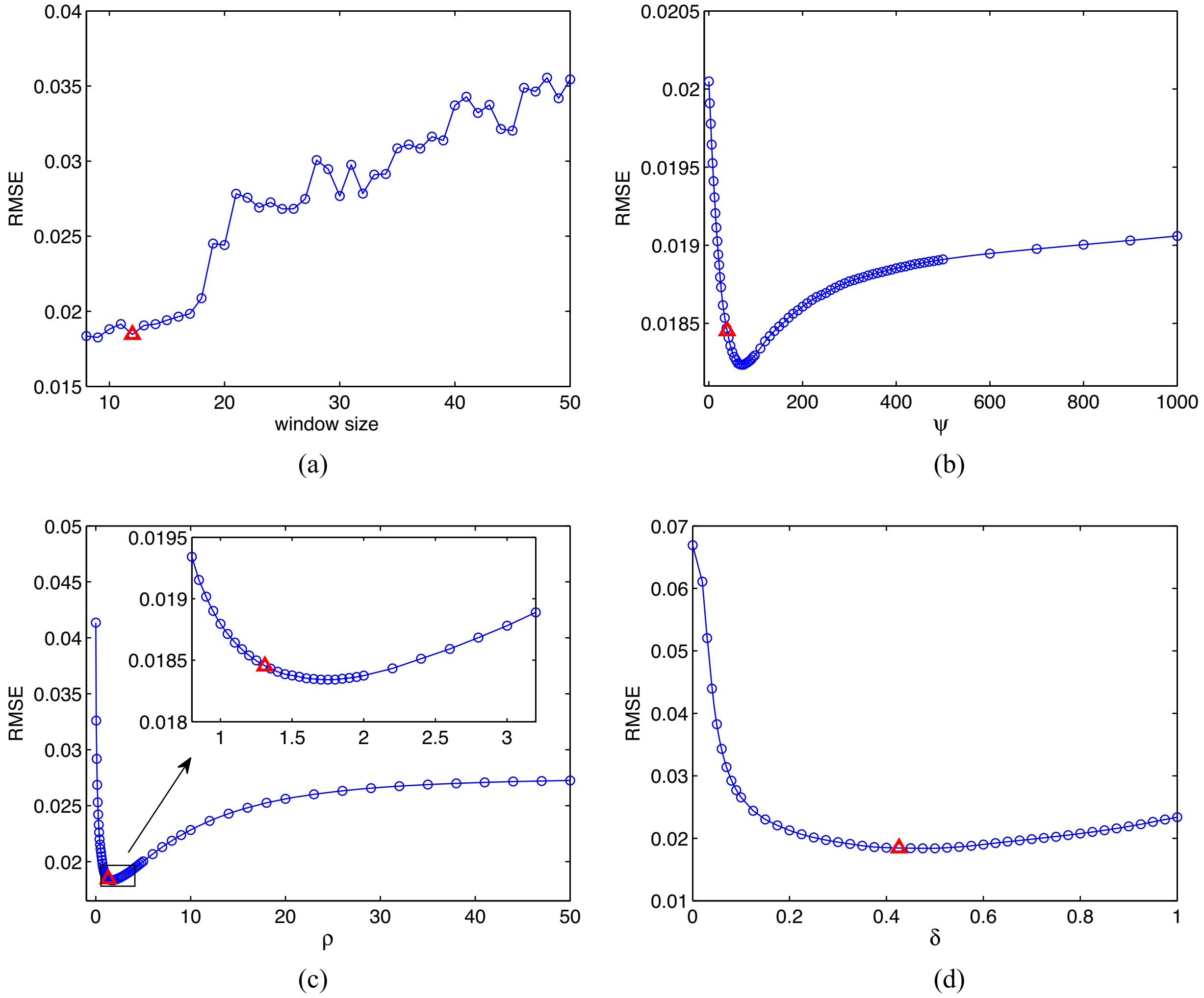
In the last column of Table 1 we can infer that the proposed soft sensor is still much less computationally efficient than those based on the RPLS, the MWPLS and the LASS. This is because compared with these three methods the SELPLS involves extra computations of local models' weights and search of query samples' neighbors at the online operation stage. However, the former is not the key factor that increases the computational load. We did the experiment using SELPLS where the redundant local model detection and deletion mechanism was removed from the localization procedure. As a result, a total of 155 local models were reserved and the predicted RMSE and the average online consumed time became 0.0180 and 153.1 s, respectively. We can see that even if the local model number grows 76% than 88, the online consumed CPU time raises only 3.7%. Meanwhile, the predicted RMSE decreased only 2.8% than 0.0180, which implies that the proposed redundant model detection and deletion mechanism can effectively remove large number of unnecessary local models with slight loss of the estimation accuracy.

The search of query samples' neighbors from the historical data set accounts for most of the online computational load in the SELPLS based soft sensor. If necessary, some database monitoring techniques such as the one proposed in Kaneko and Funatsu (2014b) which generates sparse database can be employed for enhancing the online computational efficiency. However, this topic goes beyond the scope of this paper. Despite of this drawback of the SELPLS compared with the RPLS, the MWPLS and the LASS, the SELPLS has a delightful advantage compared with the LWPLS, that is, online constructing local PLS models is avoided, which makes the SELPLS much more computationally efficient than the LWPLS. The data listed in the last column of Table 1 shows that the online CPT consumed by the SELPLS is almost half of that consumed by LWPLS. This advantage of SELPLS over the LWPLS by avoiding online modeling will be particularly significant in the scenario where the PLS model requires a large number of latent variables.

In what follows, we will investigate the influences of model parameters on the estimation accuracy. Fig. 6(a)–(d) show the predicted RMSE over the test data set as the function of the W, ψ, ρ and δ, respectively, where in each case, the rest of the algorithmic parameters were set to the values obtained by minimizing the index of RMSE over the validation data set using the PSO.

As can be seen from Fig. 6, the optimized values of the four parameters over the validation data set, i.e., $W = 12$, $\psi = 31.0$, $\rho = 1.31$ and $\delta = 0.426$ which are marked with red triangles in the corresponding four subplots of Fig. 6, achieve good generalization performance for the unseen test data set. In addition, except for W, the predicted RMSE is the uni-modal function of each of the rest three parameters and there are relative large intervals for those parameters where acceptable estimation accuracy can be obtained. These phenomena can facilitate the determination of the algorithmic parameters of the proposed soft sensor.

More specifically, the influence of W shown in Fig. 6(a) indicates small window size is preferable while the W greater than 18 results in very bad performance. As analyzed before, too small window size is not recommended, thus we suggest choosing relatively large W presupposing



(a)

(b)

(c)

(d)

Fig. 6 – Predicted RMSE over the test data set as a function of the SELPLS' parameters: (a) W, (b) ψ, (c) ρ and (d) δ.

that satisfactory performance can be achieved. Fig. 6(b) shows that the predicted RMSE seems much more insensitive on ψ than on other parameters, because the predicted RMSE remains under 0.0202 when ψ rises from 0 to a sufficiently large value, 1000. Nevertheless, it doesn't mean that the neighborhood information of the query sample can be negligible, which will be demonstrated later when Fig. 6(c) is analyzed.

Fig. 6(c) illustrates that the predicted RMSE first decreases and then increases rapidly when ρ varies from 0 to 10, which demonstrates the estimation accuracy is very sensitive to the ρ less than 10. Proper values of ρ for the test data set are located within the range of [1 2.5] and the 'optimal' value obtained by PSO over the validation data set is 1.31, with which the variation of γ on the test data set is shown in Fig. 7(a). Meanwhile, in order to explore the relationship between γ and the estimation accuracy, we fix γ as a constant in each simulation and then draw the γ-test RMSE curve over the test data set as shown in Fig. 7(b), where W, ψ and δ are kept identical to their corresponding optimized values.

Fig. 7(a) shows the adaptive change of γ according to the operating conditions of the CSTR. In the relative steady states, γ is very large, meaning that analytical samples get more emphasis. While when the process state is undergoing abrupt changes, i.e., the step change of the set point value of the reaction temperature and the catalyst activity recovery, the query samples are located far away from the corresponding newest 'labeled' analytical samples, γ becomes smaller, implying that we should trust the query samples' neighborhood information more. Fig. 7(b) shows that if γ were fixed to a proper constant such as 0.7, the estimation performance would still be acceptable compared with those of the RPLS, the LWPLS, the MWPLS and the LASS. However, the proposed way of adaptively determining γ on the basis of Eq. (26) can achieve higher estimation accuracy. In addition, the adaptive way of calculating γ by Eq. (26) is intuitively reasonable compared with fixing γ as a constant according to the analysis about Fig. 7(a). Therefore, making γ adaptively changed with the operation condition using Eq. (26) is preferred in this paper.

It is also worth to reveal the predicted results with $\rho = 0$, as shown in Fig. 8.

As analyzed before, $\rho = 0$ implies $\gamma \equiv 1$ and the neighborhood information of the query sample is ignored. As a result, the predicted performance is rather disappointing during those periods when abrupt changes happened, indicating that abandoning the neighborhood information of the query sample is not wise for handling abrupt changes. However, in those periods where the process characteristics are varying slowly, the prediction accuracy of the SELPLS with $\rho = 0$ is very high. Therefore, the SELPLS with $\rho = 0$ has the potential of modeling slow time-varying processes and satisfactory performance can be expected.

Fig. 6(d) also demonstrates with neither $\delta = 0$ nor $\delta = 1$ can the SELPLS achieve the best estimation performance.

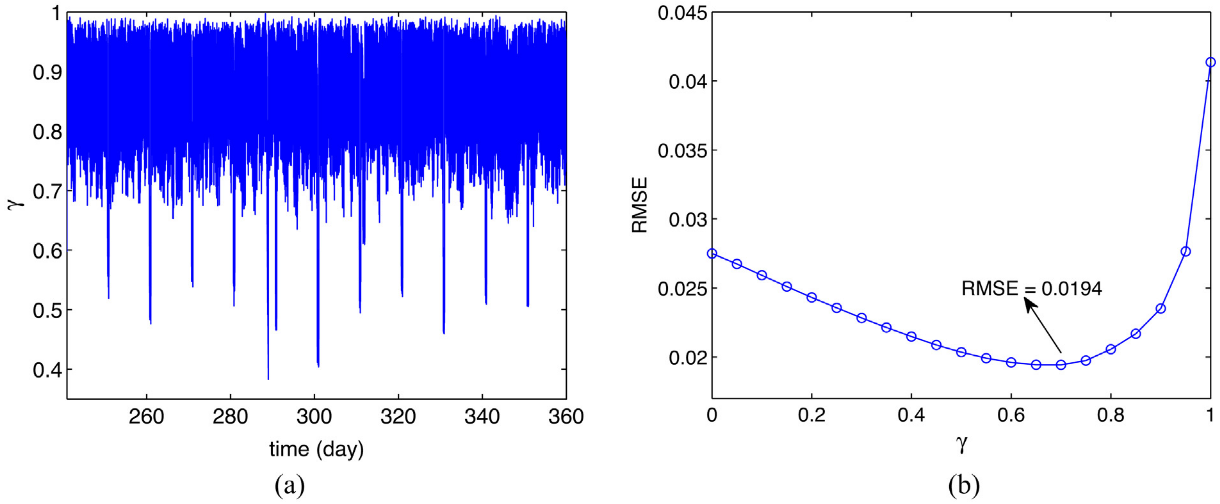


Fig. 7 – (a) Variation of γ on the test data set with $\rho = 1.31$; (b) influence of fixed γ on the estimation accuracy.

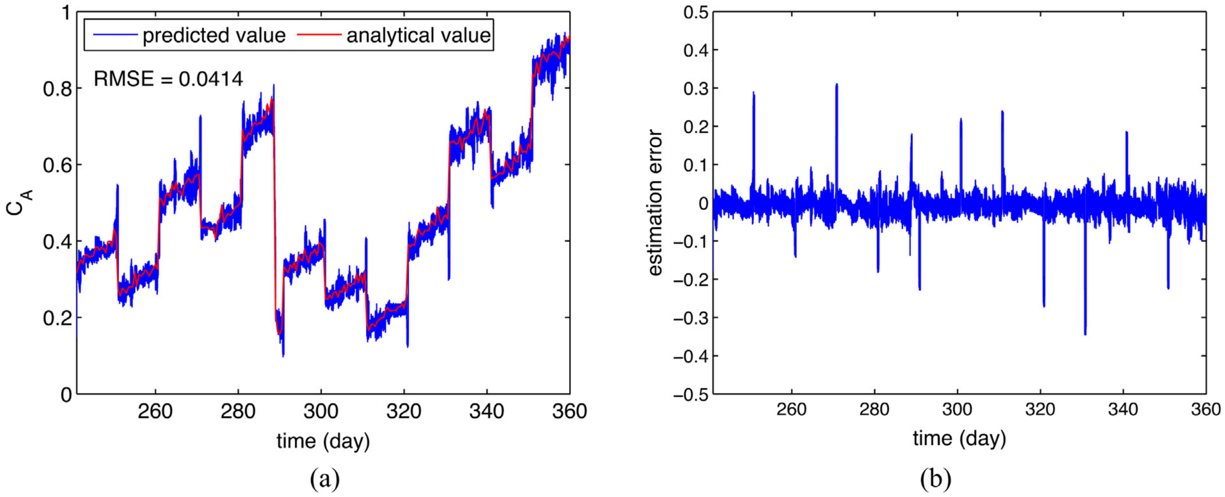


Fig. 8 – Predicted results of the SELPLS with $\rho = 0$: (a) predicted C_A, (b) estimation error. The rest of the algorithmic parameters were set to the optimized values over the validation data set.

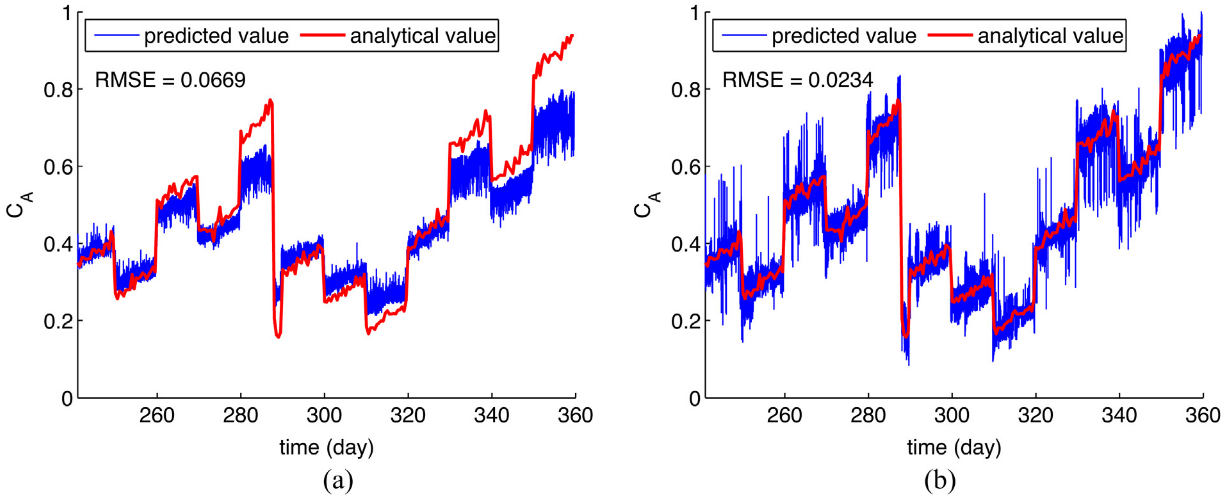


Fig. 9 – (a) Predicted C_A of the SELPLS based soft sensor with: (a) $\delta = 0$, (b) $\delta = 1$. The rest of the algorithmic parameters were set to the optimized values over the validation data set.

Especially with $\delta = 0$, although the prediction variance is decreased, the prediction error (RMSE = 0.0669) is rather high due to the severe estimation bias as shown in Fig. 9(a). On the contrary, the estimation bias can be removed with $\delta = 1$, but the predicted error (RMSE = 0.0234) is still not low enough, because the prediction variance increases as shown in Fig. 9(b). In contrast, the proposed selective ensemble learning that sets δ to 0.426 combines a few local models and reaches a good equilibrium of the estimation bias and variance. Consequently, the estimation performance with $\delta = 0.426$ is much better

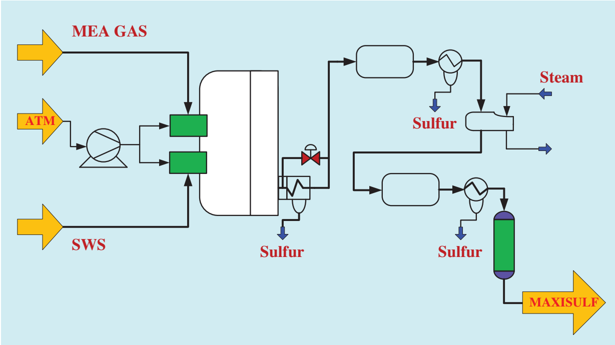


Fig. 10 – The simplified block scheme of the SRU process.

Table 2 – Descriptions of input and output variables of the soft sensor for SRU.

Variables	Description
x_1	MEA gas flow
x_2	First air flow
x_3	Second air flow
x_4	Gas flow in SWS zone
x_5	Air flow in SWS zone
y_1	Concentration of H_2S
y_2	Concentration of SO_2

than the cases with $\delta = 0$ and $\delta = 1$, as demonstrated in Fig. 4(e) and Fig. 6(d). The predicted results shown in Fig. 9 verify the previous analysis about the predicted bias and variance of ensemble learning in Section 3.2. In addition, since Fig. 6 shows that the predicted RMSE is consistently under 0.0194 (corresponding to 5% increment of 0.0185) with δ ranging from 0.300 to 0.625, we can infer that proper δ that generates satisfactory estimation performance is not difficult to find.

4.2. *Sulfur recovery unit (SRU)*

In this subsection, application results of various adaptive soft sensors to a real life industrial chemical process, i.e. the sulfur recovery unit (SRU) process, is provided. A simplified block scheme of this SRU process is shown as Fig. 10. The SRU is normally utilized to remove environmental pollutants, which are harmful to the atmosphere and human body, from acid gas streams. Two kinds of acid gas are taken as the input of SRU, namely the MEA gas that is rich in hydrogen sulfide (H_2S) and the SWS gas that is rich in H_2S and ammonia (NH_3). In SRU, H_2S is transformed into pure sulfur via a partial oxidation with air, and sulfur dioxide (SO_2) is formulated. The non converted gas (less than 5%) is fed to the Maxisulfur plant for a final conversion phase. The tail gas from the SRU contains residual H_2S and SO_2, whose concentrations need to be monitored before they are released into the atmosphere. However, these two kinds of acid gas damage hardware sensors by corrosion and consequently hardware instruments are frequently removed and maintained, which greatly increases the cost of production. Thus, soft sensors are required to estimate the concentrations of H_2S and SO_2. More detailed description of the SRU process can be found in Fortuna et al. (2007).

Five process variables and the concentrations of H_2S and SO_2 tabulated in Table 2 are considered as the inputs and outputs of the required soft sensors, respectively.

The SRU dataset contains 10,081 samples, which can be downloaded from http://www.springer.com/engineering/control/book/978-1-84628-479-3. It came from a real industrial chemical process, and has become a benchmark dataset for evaluating the performance of adaptive soft sensors (Kadlec et al., 2011). We selected 2500 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. Although the SRU is a multi-output process, we treat it as two single-output processes in this paper. By the analysis of expert knowledge and consideration of process dynamics (Fortuna et al., 2007), the PLS model structure is determined as

$$\hat{y}_1(k) = f_1\left[x_1(k), x_1(k-5), x_1(k-7), x_1(k-9), \cdots, x_5(k), x_5(k-5), x_5(k-7), x_5(k-9)\right] \tag{40}$$

$$\hat{y}_2(k) = f_2\left[x_1(k), x_1(k-5), x_1(k-7), x_1(k-9), \cdots, x_5(k), x_5(k-5), x_5(k-7), x_5(k-9)\right] \tag{41}$$

where, $\hat{y}_1(k)$ and $\hat{y}_2(k)$ represent the predicted value of y_1 and y_2, respectively, for the k-th query sample in the test data set.The candidates of parameters of various soft sensors are summarized as (for both y_1 and y_2):

- RPLS: $A \in [1\ 20]$, $\lambda \in (0\ 1)$;
- LWPLS: $A \in [1\ 20]$, neighborhood size $\in [50\ 200]$, scaling parameter $\in [0.01\ 10]$.
- MWPLS: $A \in [1\ 20]$, window size $\in [50\ 300]$;
- LASS: $A \in [1\ 20]$;
- SELPLS: $A \in [1\ 20]$, $W \in [50\ 200]$ with an interval of 5, $\psi \in [0\ 50]$, $\rho \in [0\ 5]$, and $\delta \in [0\ 1]$. The parameters of the RPLS, the LWPLS, the MWPLS and the proposed SELPLS based soft sensors were also optimized by the PSO over the validation data set, which are listed as follows.
- RPLS: For y_1, $A = 4$ and $\lambda = 0.966$, while for y_2, $A = 4$ and $\lambda = 0.947$.
- LWPLS: For y_1, the neighborhood size was 126, $A = 5$ and the scaling parameter was 0.138; for y_2, the neighborhood size was 118, $A = 4$ and the scaling parameter was 0.078;
- MWPLS: The window size for y_1 and y_2 were set to 190 and 210, respectively, and A for both y_1 and y_2 were determined by 10-fold CV.
- LASS: for both y_1 and y_2, A was chosen by 10-fold CV
- SELPLS: For y_1, $W = 145$, $\psi = 4.378$, $\rho = 1.824$, $\delta = 0.305$; for y_2, $W = 140$, $\psi = 5.706$, $\rho = 1.526$, $\delta = 0.099$. A for both y_1 and y_2 were determined by 10-fold CV. K was set to 100 and α_t and α_χ were set to 0.05 for both y_1 and y_2.

In the LASS, the maximum and minimum window size in each localization procedure was set to 300 and 50, respectively. The PLS toolbox was also utilized for all the five soft sensors.

Performance comparisons in terms of scatter plots between the SELPLS and the other methods for y_1 and y_2 are illustrated in Fig. 11 and Fig. 12, respectively, while Table 3 quantitatively lists the estimation errors of all the adaptive soft sensors.

As can be seen from Fig. 11 and Fig. 12, the data points of the SELPLS based soft sensor are overall located more tightly along the diagonal line for both y_1 and y_2 than those of other soft sensors, which means that the proposed soft sensor outperforms

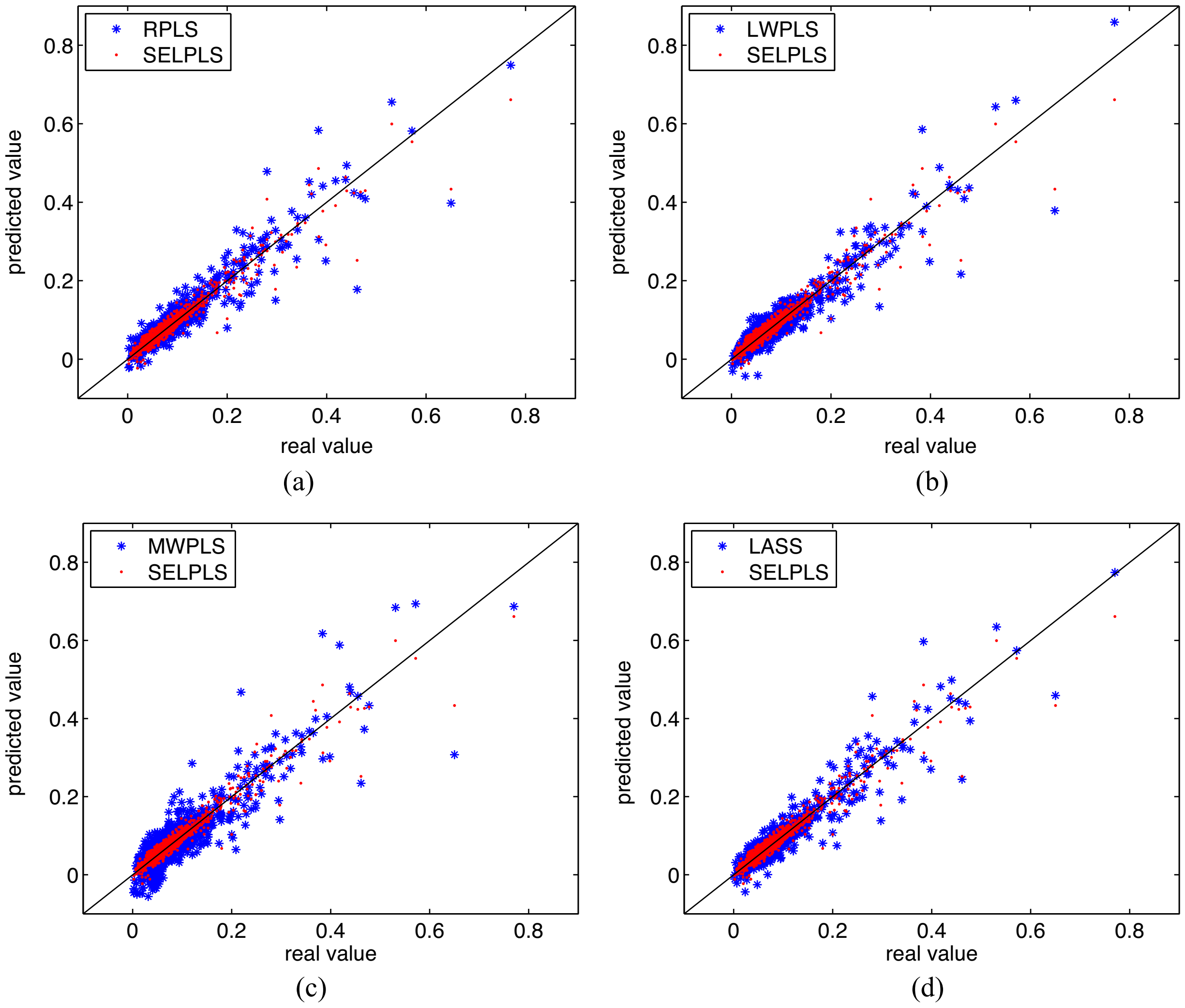


Fig. 11 – Scatter plot comparisons for y_1 between the SELPLS and other methods.

Table 3 – Predicted errors of various soft sensors for y_1 and y_2.

Soft sensor	For y_1			For y_2		
	RMSE	RRMSE (%)	MAE	RMSE	RRMSE (%)	MAE
RPLS	0.0169	43.0	0.284	0.0152	18.7	0.196
LWPLS	0.0181	43.1	0.271	0.0179	16.3	0.151
MWPLS	0.0246	75.0	0.343	0.0230	29.2	0.175
LASS	0.0172	30.7	0.217	0.0168	18.3	0.151
SELPLS	0.0130	21.7	0.217	0.0119	11.7	0.108

the others. The quantified estimation errors also demonstrate that the SELPLS can improve the prediction performance a lot than other methods. Particularly, in the SRU process, prediction accuracy for large values (the peak value, for example) of the target variables, which may probably cause severe consequences, should be paid much attention to. In this respect, even though the sample density for both y_1 and y_2 in high-value area is much sparser than that in low-value area, the proposed soft sensor achieves better performance. Table 3 also indicates that the SELPLS can help to reduce the MAE, which usually appears in high-value areas of the H_2S and SO_2.

In contrast, the MWPLS based soft sensor performs worst among these soft sensors. One possible reason maybe that the globally optimal window size is not suitable for all local process states. Thus adaptively changing the window size for the MWPLS according to the process characteristics may be preferable. In addition, one potential issue associated with the LASS as mentioned in Section 1 should be pointed out. That is, as newly measured samples are continuously accumulated, the re-localization procedure in the LASS may be quite time consuming. For example, in this case study, the historical data set initially contained 2500 samples and was gradually augmented at the online operation stage by the validation data set containing 2536 samples. Throughout the simulation, the re-localization happened 726 and 685 times in the LASS based soft sensor for y_1 and y_2, and the total simulation time was 8.5 and 6.2 h, respectively. Consequently, if the LASS is applied to those processes with large amount of analytical samples, improvement on the computational efficiency is also important.

Fig. 11 and Fig. 12 also show that the data points of SELPLS are distributed symmetrically on the two sides of the diagonal line, that is, the SELPLS based soft sensor nearly provides unbiased estimation results for both y_1 and y_2. If we set the threshold value δ which decides the degree of the selective ensemble to 0 for both y_1 and y_2, the predicted results would be as shown in Fig. 13. Other parameters were kept invariant as their optimized values over the validation data set.

Comparing the data points' distribution of the SELPLS exhibited in Fig. 13 with those appeared in Fig. 11 and Fig. 12, we can readily draw a conclusion that predicted bias occurred with all local models combined, which results in high predicted RMSE. On the other hand, the predicted RMSE based

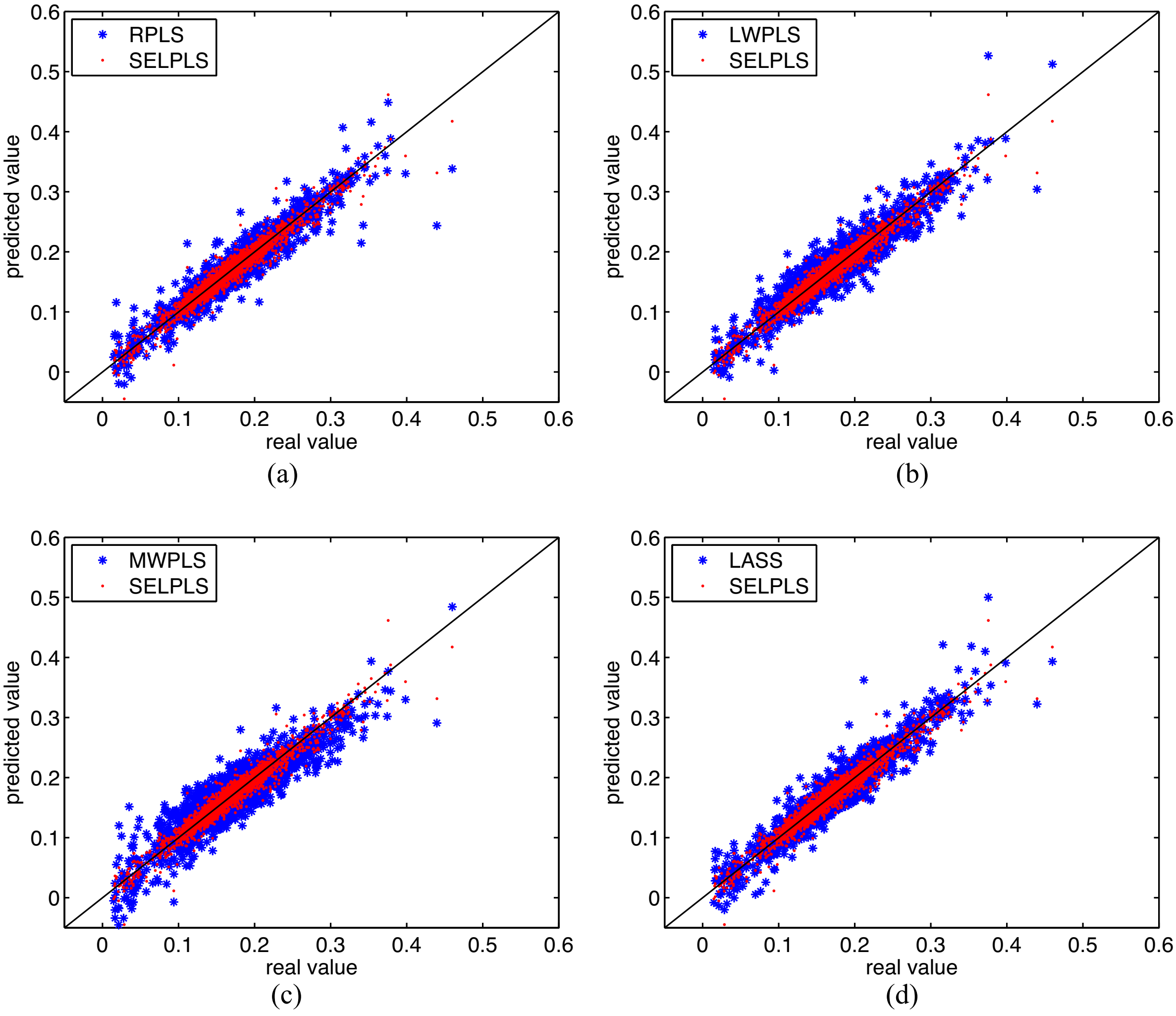


Fig. 12 – Scatter plot comparisons for y_2 between the SELPLS and other methods.

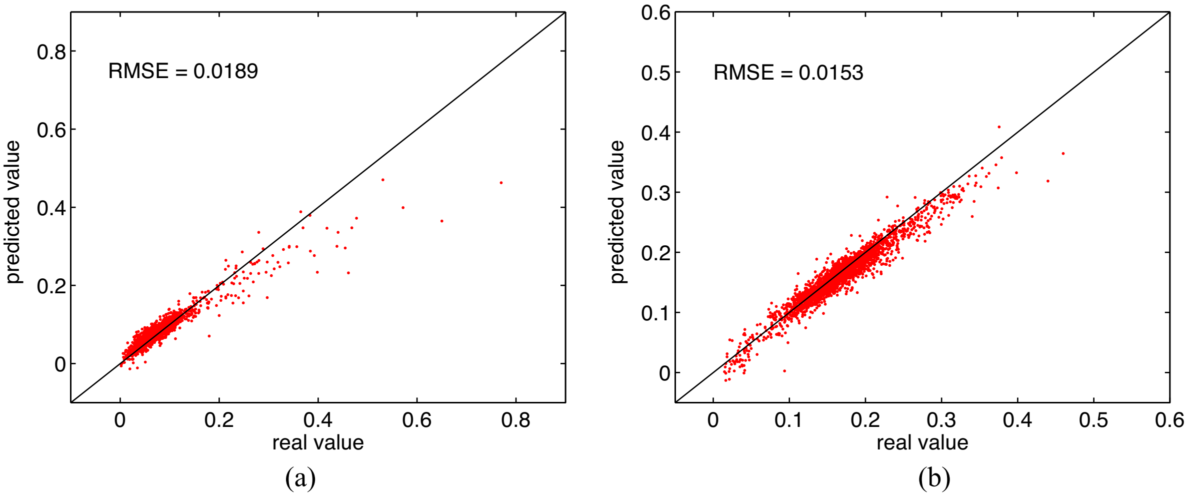
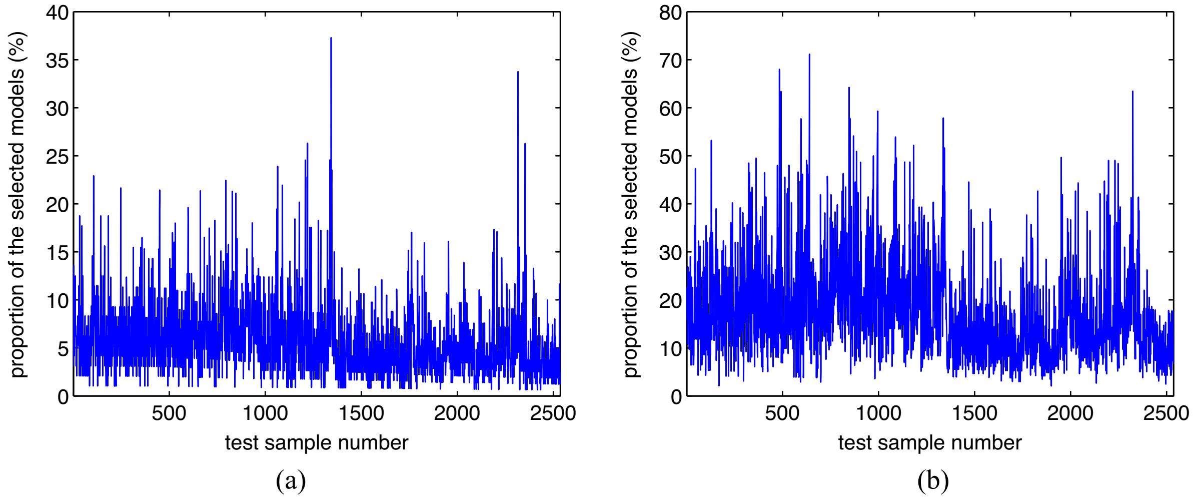


Fig. 13 – Predicted results of the SELPLS with $\delta = 0$: (a) for y_1, (b) for y_2.

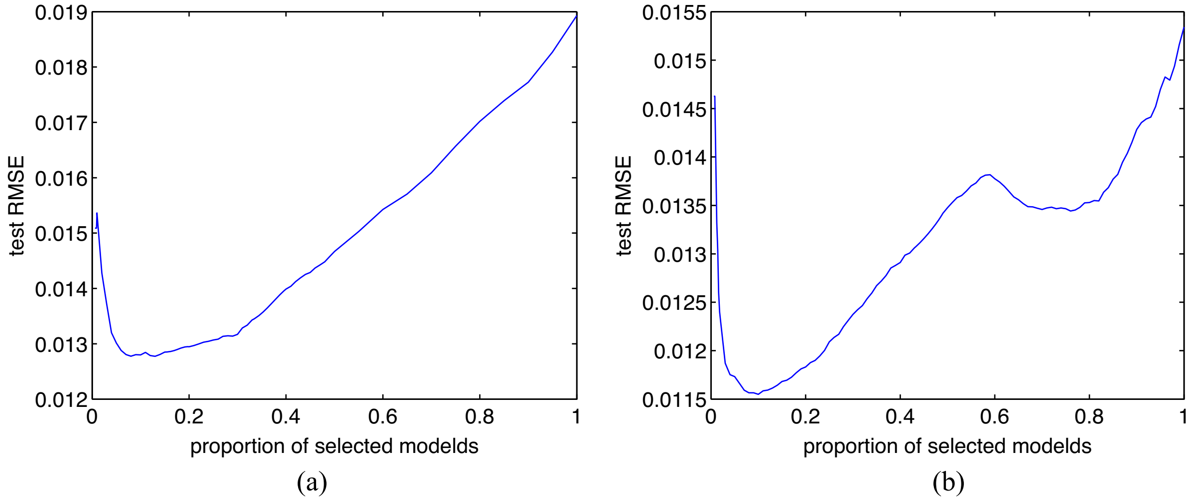
on SELPLS with $\delta = 1$ would change to 0.0151 and 0.0146 for y_1 and y_2, respectively. These results confirm the effectiveness and utility of the proposed selective ensemble learning strategy. For better insight of the selective ensemble's mechanism, the ratio of the selected local models to all local models available for test samples is visualized in Fig. 14. As can be seen, even though the threshold value δ was fixed, the proportion of the selected models changed adaptively, since each local model's generalization ability for the query samples were not invariant. Actually, despite the quantity of the available local models was changing during the online operation stage, the number of selected local models also varied.

It is interesting to reveal the relationship between the proportion of the selected models and the estimation accuracy, as illustrated in Fig. 15. Moreover, in order to further compare



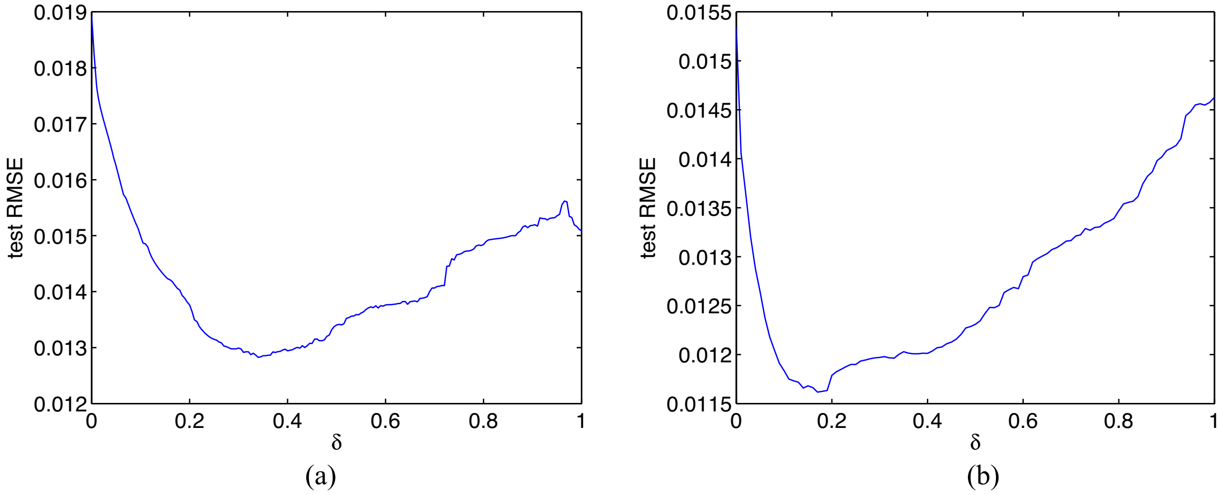
(a) (b)

Fig. 14 – Ratio of the selected local models to all local models available for test samples in the case of (a) y_1, (b) y_2.



(a) (b)

Fig. 15 – Relationship between predicted RMSE and proportion of selected models in the case of (a) y_1, (b) y_2.



(a) (b)

Fig. 16 – Relationship between predicted RMSE and δ in the case of (a) y_1, (b) y_2.

the characteristics between fixed proportion and adaptive proportion of selected models, we also present the relationship between δ and the estimation error as shown in Fig. 16, where fixed δ equals to adaptive proportion of the selected models. Figs. 15 and 16 seem to imply that both fixed and adaptive proportions of the selected models are able to achieve satisfactory performance. And there exists some relationship between the fixed proportion of selected models and δ. That is, when the proportion of selected models is fixed as 1, it is equal to δ that is set to 0, i.e., all local models are selected; while when the fixed proportion of selected models is close to 0 (usually could not be zero to ensure at least one local model is selected), it equals to δ that approaches 1, i.e., very few models are selected.

However, further comparison indicates that there is an advantage of using δ to control the selective degree over that using fixed proportion of selected models. That is, there is wider space for δ to achieve high-accuracy performance. For example, in Fig. 16(a), the range of δ that makes the test RMSE lower than 0.0140 is [0.185 0.685], while in Fig. 15(a), the range of the proportion of selected models that meets the same requirement is [0.03 0.4]. In Fig. 16(b) and Fig. 15(b), the two ranges that make the test RMSE lower than 0.0125 change to [0.06 0.54] and [0.018 0.32], respectively. In addition, deep analysis reveals a potential limitation of the strategy using fixed proportion of selective local models to determine the selective degree. That is, as the amount of local models grows at the online operation stage, the number of local models also increases if the proportion of selected models stays invariant, which may cause unexpected prediction bias when the number of selected models exceeds certain value in some scenarios.

We also investigated the running result of the localization approach of the SELPLS in the SRU process. In the case of y_1, a total of 162 local models were preserved, 67 of which were added at the online operation stage. As for y_2, the numbers changed to 164 and 70, respectively. Furthermore, if the redundant model detection and deletion mechanism were stripped, which was similar to the way used in Shao et al. (2013), the total model number would rise to 239 and 257 and the predicted RMSE over the test data set would turn into 0.0128 and 0.0114 in the case of y_1 and y_2, respectively. Through comparison we can find that the redundant model detection and deletion mechanism could remove 48% and 57% unnecessary local models for y_1 and y_2, respectively. And meanwhile, the estimation performance deteriorated only 1.6% and 4.4% for y_1 and y_2, respectively, in terms of the index of RMSE.

5. Conclusions

Aiming at handling the process nonlinearity and the time-varying issue in chemical processes, we have developed an adaptive soft sensing method, referred to as the SELPLS, which combines the local learning strategy and selective ensemble learning strategy together. An adaptive localization procedure has been first presented to construct concise local model set. Then we have derived a selective ensemble learning strategy, which combines parts of the preserved local models via the Bayesian inference. During the selective ensemble process, an effective way of qualifying local models' generalization performance for the current process dynamics has been formulated, which is the basis of our selective ensemble learning and the Bayesian inference. Extensive simulation investigations and discussions based on two chemical processes have demonstrated the priorities of our proposed soft sensor over several existing adaptive soft sensors.

Acknowledgments

This work was supported by the National Science Foundation of China (grant nos. 61273160, 61403418) and the Fundamental Research Funds for the Central Universities (grant nos. 14CX06067A, 13CX05021A).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.cherd.2015.01.006.

References

Chen, K., Ji, J., Wang, H.Q., Liu, Y., Song, Z.H., 2011. Adaptive local kernel-based learning for soft sensor modeling of nonlinear processes. Chem. Eng. Res. Design 89 (10), 2117–2124.

Dayal, B.S., MacGregor, J.F., 1997. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. J. Process Control 7 (3), 169–179.

Deng, J., Xie, L., Chen, L., Khatibisepehr, S., Huang, B., Xu, F.W., Espejo, A., 2013. Development and industrial application of soft sensors with on-line Bayesian model updating strategy. J. Process Control 22 (3), 317–325.

Fu, Y.F., Su, H.Y., Zhang, Y., Chu, J., 2008. Adaptive soft-sensor modeling algorithm based on FCMISVM and its application in PX adsorption separation process. Chin. J. Chem. Eng. 16 (5), 746–751.

Fujiwara, K., Kano, M., Hasebe, S., Takinami, A., 2009. Soft-sensor development using correlation-based just-in-time modeling. AIChE J. 55 (7), 1754–1764.

Fortuna, L., Graziani, S., Rizzo, A., Xibilia, G.M., 2007. Soft Sensors for Monitoring and Control of Industrial Processes. Springer-Verlag, London.

Galicia, H.J., He, Q.P., Wang, J., 2011. A reduced order soft sensor approach and its application to continuous digester. J. Process Control 21 (4), 489–500.

Ge, Z.Q., Song, Z.H., 2011. Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples. AIChE J. 57 (8), 2109–2118.

Ge, Z.Q., Song, Z.Q., 2014. Ensemble independent component regression models and soft sensing application. Chemom. Intell. Lab. Syst. 130 (15), 115–122.

Ge, Z.Q., Song, Z.H., Wang, P.L., 2014. Probabilistic combination of local independent component regression model for multimode quality prediction in chemical processes. Chem. Eng. Res. Design 92 (3), 501–512.

Geman, S., Bienenstock, E., Doursat, R., 1992. Neural networks and the bias/variance dilemma. Neural Comput. 4, 1–58.

Grbić, R., Slišković, D., Kadlec, P., 2013. Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models. Comput. Chem. Eng. 58 (11), 84–97.

Himmelblau, D.M., 2008. Accounts of experiences in the application of artificial neural networks in chemical engineering. Ind. Eng. Chem. Res. 47 (16), 5782–5796.

Kadlec., 2009. On Robust and Adaptive Soft Sensors. Bournemouth University, Poole (Ph.D. dissertation).

Kadlec, P., Gabrys, B., Strandt, S., 2009. Data-driven soft sensors in the process industry. Comput. Chem. Eng. 33 (4), 795–814.

Kadlec, P., Grbić, R., Gabrys, B., 2011. Review of adaptation mechanisms for data-driven soft sensors. Comput. Chem. Eng. 35 (1), 1–24.

Kadlec, P., Gabrys, B., 2011. Local learning based adaptive soft sensor for catalyst activation prediction. AIChE J. 57 (5), 1288–1301.

Kaneko, H., Arakawa, M., Funatsu, K., 2009. Development of a new soft sensor method using independent component analysis and partial least squares. AIChE J. 55 (1), 87–98.

Kaneko, H., Funatsu, K., 2011a. Maintenance-free soft sensor models with time difference of process variables. Chemom. Intell. Lab. Syst. 107 (2), 312–317.

Kaneko, H., Funatsu, K., 2011b. Development of soft sensor models based on time difference of process variables with accounting for nonlinear relationship. Ind. Eng. Chem. Res. 58 (18), 10643–10651.

Kaneko, H., Funatsu, K., 2013. Classification of the degradation of soft sensor models and discussion on adaptive models. AIChE J. 59 (7), 2339–2347.

Kaneko, H., Funatsu, K., 2014a. Adaptive soft sensor based on online support vector regression and Bayesian ensemble learning for various states in chemical plants. Chemom. Intell. Lab. Syst. 137 (15), 57–66.

Kaneko, H., Funatsu, K., 2014b. Database monitoring index for adaptive soft sensors and the application to industrial process. AIChE J. 60 (1), 160–169.

Kano, M., Ogawa, M., 2010. The state of the art in chemical process control in Japan: good practice and questionnaire survey. J. Process Control 20 (9), 969–982.

Kano, M., Fujiwara, K., 2013. Virtual sensing technology in process industries: trends and challenges revealed by recent industrial applications. J. Chem. Eng. Jpn. 46 (1), 1–17.

Khatibisepehr, S., Huang, B., Xu, F.W., Espejo, A., 2012. A Bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry. J. Process Control 22 (10), 1913–1929.

Kim, S., Kano, M., Hasebe, S., Takinami, A., Seki, T., 2013a. Long-term industrial applications of inferential control based on just-in-time soft sensors: economical impact and challenges. Ind. Eng. Chem. Res. 52 (35), 12346–12356.

Kim, S., Okajima, R., Kano, M., Hasebe, S., 2013b. Development of soft sensor using locally weighted PLS with adaptive similarity measure. Chemom. Intell. Lab. Syst. 124 (5), 43–49.

Liu, J.L., 2007. On-line soft sensor for polyethylene process with multiple production grades. Control Eng. Pract. 15 (7), 769–778.

Liu, J.L., Chen, D.S., Shen, J.F., 2010. Development of self-validating soft sensors using fast moving window partial least squares. Ind. Eng. Chem. Res. 49 (22), 11530–11546.

Liu, Y., Gao, Z.L., Li, P., Wang, H.Q., 2012. Just-in-time kernel learning with adaptive parameter selection for soft sensor modeling of batch processes. Ind. Eng. Chem. Res. 51 (11), 4313–4327.

Liu, Y., Chen, J.H., 2013. Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes. J. Process Control 33 (6), 793–804.

Liu, Y., Li, C.L., Gao, Z.L., 2014. A novel unified correlation model using ensemble support vector regression for prediction of flooding velocity in randomly packed towers. J. Ind. Eng. Chem. 22 (3), 1109–1118.

Lv, Y., Liu, J.Z., Yang, T.T., Zeng, D.L., 2013. A novel least squares support vector machine ensemble model for NO_x emission prediction of a coal-fired boiler. Energy 55 (15), 319–329.

Ni, W.D., Tan, S.K., Ng, W.J., Brown, S.D., 2012a. Moving-window GRP for nonlinear dynamic system modeling with dual updating and dual preprocessing. Ind. Eng. Chem. Res. 51 (8), 6416–6428.

Ni, W.D., Tan, S.K., Ng, W.J., Brown, S.D., 2012b. Localized, adaptive recursive partial least squares regression for dynamic system modeling. Ind. Eng. Chem. 55 (23), 8025–8039.

Ni, W.D., Brown, S.D., Man, R.L., 2014. A localized adaptive soft sensor for dynamic system modeling. Chem. Eng. Sci. 111 (24), 250–363.

Pani, A.K., Mohanta, H.K., 2011. A survey of data treatment techniques for soft sensor design. Chem. Product Process Model. 6 (1), 1–21.

Qin, S.J., 1998. Recursive PLS algorithms for adaptive data modeling. Comput. Chem. Eng. 22 (4-5), 503–514.

Shao, W.M., Tian, X.M., Wang, P., 2012. Online learning soft sensor method based on recursive kernel algorithm for PLS. J. Chem. Ind. Eng. Soc. China 63 (9), 2887–3289.

Shao, W.M., Tian, X.M., Chen, H.L., 2013. Adaptive anti-over-fitting soft sensing method based on local learning. Prepr. 10th IFAC Int. Symp. Dyn. Control Process Syst. 10 (1), 415–420 ((Mumbai, India), Dec. 18–20).

Shao, W.M., Tian, X.M., Wang, P., 2014. Local partial least squares based online soft sensing method for multi-output processes with adaptive process states division. Chin. J. Chem. Eng. 22 (7), 828–836.

Tang, J., Yu, W., Chai, T.Y., Zhao, L.J., 2012a. On-line principal component analysis with application to process modeling. Neurocomputing 82 (1), 167–178.

Tang, J., Chai, T.Y., Zhao, L.J., Yu, W., Yue, H., 2012b. Soft sensor for parameters of mill load based on multi-spectral segments PLS sub-models and on-line adaptive weighted fusion algorithm. Neurocomputing 78 (1), 38–47.

Tang, J., Chai, T.Y., Zhao, L.J., 2013. Modeling load parameters of ball mill in grinding process based on selective ensemble multisensory information. IEEE Trans. Autom. Sci. Eng. 10 (3), 726–740.

Wu, F.H., Chai, T.Y., 2010. Soft sensing method for magnetic tube recovery ratio via fuzzy systems and neural networks. Neurocomputing 73 (13–15), 2489–2497.

Xie, L., Zeng, J.S., Gao, C.H., 2014. Novel just-in-time learning-based soft sensor utilizing non-Gaussian information. IEEE Trans. Control Syst. Technol. 22 (1), 360–369.

Xu, S.Q., Liu, X.G., 2014. Melt index prediction by fuzzy functions with dynamic fuzzy neural networks. Neurocomputing 142 (22), 191–198.

Yu, J., 2012a. A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. Comput. Chem. Eng. 41 (11), 134–144.

Yu, J., 2012b. 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 (12), 22–30.

Yu, J., Chen, K.L., Rashid, M.M., 2013. 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 (19), 96–109.

Zhang, S.N., Wang, F.L., He, D.K., Jia, R.D., 2012. Real-time product quality control for batch processes based on stacked least squared support vector regression models. Comput. Chem. Eng. 36 (10), 217–226.

Zhang, S.N., Wang, F.L., He, D.K., Jia, R.D., 2013. Online quality prediction for cobalt oxalate synthesis process using least squares support vector regression approach with dual updating. Control Eng. Pract. 21 (10), 1267–1276.

Zhao, L.J., Chai, T.Y., Yuan, D.C., 2012. Selective ensemble extreme learning machine modeling of effluent quality in wastewater treatment plants. 9(6), 627–633.

Zhou, Z.H., Wu, J.X., Tang, W., 2002. Ensembling neural networks: many could be better than all. Artif. Intell. 137 (1–2), 239–263.