International Journal of Innovative
Computing, Information and Control
Volume **12**, Number **2**, April **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 543–556

CALIBRATION METHOD OF SOFT SENSOR BASED ON BAYESIAN GAUSSIAN PROCESS REGRESSION

Huan Min and Xionglin Luo

Department of Automation
China University of Petroleum – Beijing
No. 18, Fuxue Road, Changping District, Beijing 102249, P. R. China
luoxl@cup.edu.cn; billmin881220@gmail.com

Received August 2015; revised December 2015

Abstract. *Soft sensor model is basically an approximation of the actual objective process model. As the process model is almost time-variant, soft sensor shall be calibrated regularly such that it keeps pace with the changes of the process. However, the sampling interval of the hard-to-measure variable is usually far longer than that of the easy-to-measure variables. Consequently soft sensor cannot make calibration timely due to the lack of estimation error, and the performance of soft sensor inevitably degrades. To solve this problem, we proposed a soft sensor calibration method based on the Bayesian Gaussian process regression (GPR). When soft sensor deteriorates, the target variable is estimated by GPR-based interpolation over the sparse history data. Then we obtain a missing data zone of the target variable. By selecting several datasets from the data zone, we can train several candidate models based on the soft sensor model. The soft sensor is finally calibrated by weighted combination of the trained candidate models. The feasibility and effectiveness of the proposed calibration method is verified by experiments on a pH neutralization facility and comparative simulation experiments on a continuous stirred tank reactor with a Kalman filter based calibration method.*

Keywords: Deterioration, Calibration, Gaussian process regression, Multiple candidate models

1. **Introduction.** In chemical processes, the main difficulty for effective quality control lies in the lack of realtime measurements of the critical variables, due to lag time and some other technical or economic reasons. As an effective solution, soft sensor technology is proposed and developed rapidly in the past two decades [1]. Soft sensor mainly builds predictive models for the hard-to-measure critical variables by using the available stored measurement data or/and mechanism information.

Soft sensor model is commonly built by statistical or mechanistic modeling approaches. There are various modeling methods [2]. As every technique has its limitations, the practical application of soft sensor technology is also troubled by the technique's limitation. The significant problem of the soft sensor application is that soft sensor will inevitably run into the performance deterioration because of the inevitable model mismatch between soft sensor and the process. When soft sensor's performance starts to deteriorate, the prediction of the soft sensor becomes unreliable due to the decreasing precision. Usually the built soft sensor can only serve reliably within certain application domain [3]. Because the model mismatch within the application domain is acceptable, the sensor's accuracy is considered reliable. The actual processes are usually time-variant and nonlinear. And the model behaviors of the industrial processes are usually changed constantly due to process variation such as changes of temperatures, pressures, reactant concentrations, and catalyst activity [4-7]. However, the soft sensor model is not updated during a working

period. As a consequence, the model mismatch tends to diverge, which can weaken the estimation accuracy of the soft sensor, especially when the degree of the process's nonlinearity is considerably high. Under that circumstance the model mismatch problem will become significant if soft sensor fails to adapt itself to the changing model behaviors of the processes. To avoid the performance deterioration of the soft sensor during its service, the regular calibration of soft sensor is a must. In practical applications of soft sensors, there are mainly two directions on soft sensor calibration: the output compensation method and the model self-adjustment method. The output calibration method is usually used by field operators because of its simplicity and low cost [8]. When the operating condition does not change greatly, this method can usually get good performance. Apparently the output compensation method cannot solve the problem essentially and the effectiveness is highly constrained. By contrast, the model self-adjustment method has drawn more and more attention from academia [9-14]. The implementation of the model self-adjustment method varies. Generally the difference can be mainly summarized as modeling with model parameters updated regularly and modeling with model(s) rebuilt regularly. In this paper we only discuss modeling with parameters updated regularly. Basically it is that the soft sensor model continuously stays in service after being built while the model parameters are updated regularly. Under this framework, the used model is usually complicated and nonlinear to meet the demand of effectively describing the behaviors of the complex processes. Therefore, it is significant to update the model parameters timely to avoid the performance deterioration. However, when the feedback of target variable is far less frequent, it is difficult to update the model parameters of the soft sensor due to the lack of estimation error. In such situations the soft sensor cannot be calibrated timely to avoid performance deterioration. To solve this problem, a calibration method based on Kalman filtering was proposed [15]. However, the proposed method requires a considerably reliable mechanism model of the process since it uses Kalman filter. The difficulty to obtain the required mechanism model limits the method's application.

In this paper we proposed a calibration method based on Gaussian process regression. The schematic diagram of the proposed soft sensor calibration method is given in Figure 1. For simplicity, the process variables and the target variable are denoted as $\boldsymbol{x}$ and y, respectively. Assuming that the sampling cycle of the target variable y is T. The objective variable y is available on a less frequent basis than the process variables x. It is desirable to estimate the missing target variables $\{y_{ij}\}_{i=1}^{m} \mid_{j=k}$ within the latest sampling cycle T_k so as to update the parameters of the soft sensor model. According to the proposed calibration method, when the behavior of the process changes, we use the technology of Bayesian Gaussian process regression (GPR) to estimate the missing values of the target variable y by interpolation over the sparse history data $\{y_{ij}\}_{j=k-n}^{k}$. Thereby we can obtain a missing data zone as shown in Figure 3. Then a proper number of datasets are obtained from the missing data zone. By using the datasets several candidate models are obtained based on the soft sensor model. Finally the soft sensor is calibrated by a weighted combination of multiple candidate models.

The rest of the paper is organized as follows. Section 2 gives an introduction of the Bayesian Gaussian process regression. Section 3 presents the mechanism of the proposed calibration method. Section 4 gives cases which verify the effectiveness of the proposed calibration method. Section 5 is the summarization part.

2. Bayesian Gaussian Process Regression.

Bayesian GPR is a probabilistic regression. Suppose that we have a training data set D consisting of m pairs of n-dimensional inputs vectors $\{\boldsymbol{x}_i\}$ and m noisy observed scalar outputs $\{y_i\}$:

$$D = \{(\boldsymbol{x}_i, y_i) \mid i = 1, \cdots, m\} = \{\boldsymbol{X}, \boldsymbol{y}\}$$

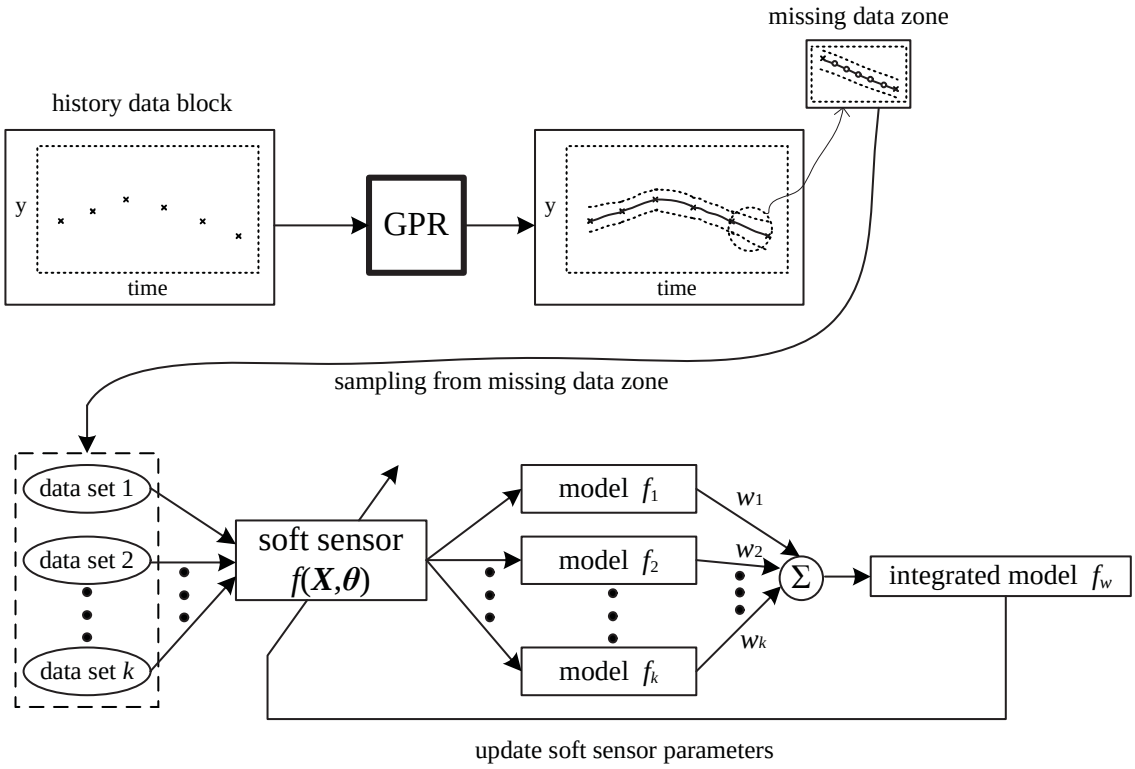


FIGURE 1. Schematic of the proposed soft sensor calibration method

where, $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^m$ and $\boldsymbol{y} = \{y\}_i^m$, respectively.

For the purpose of building a statistical model for the training data set, the observed output y_i can be handled as the combination of the real value $f(\boldsymbol{x}_i)$ and the additive observed noise ε_i. For simplicity, we assume that the noise ε_i is independent and identically distributed Gaussian noise, i.e., $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. Hence, we have

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i \tag{1}$$

where, f is the latent function and $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ is the noise.

Basically Bayesian GPR mechanism learns about the latent function f from the given training set D and infers the potential values $f(\boldsymbol{x}^*)$ at the test location $\boldsymbol{x}^*$. Bayesian GPR aims to crack the latent function f by using Bayesian rule. The Bayesian framework requires three basic components: prior, likelihood and evidence. In a Gaussian process, a Gaussian process prior is assumed over latent function f so that we can obtain the prior component as $p(\boldsymbol{f} \mid \boldsymbol{X}, \theta_p)$. As required, the likelihood component and the evidence component can be expressed by $p(\boldsymbol{y} \mid \boldsymbol{f}, \theta_l)$ and $p(D \mid \theta_p, \theta_l)$, respectively. Therefore, according to Bayesian rule [16,17], the posterior can be computed from likelihood and the prior with information of empirical observed data D integrated to update our belief by

$$p(\boldsymbol{f} \mid D, \theta_l, \theta_p) = \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \theta_l) p(\boldsymbol{f} \mid \boldsymbol{X}, \theta_p)}{p(D \mid \theta_p, \theta_l)} \tag{2}$$

where, $p(\boldsymbol{f} \mid D, \theta_l, \theta_p)$ is the posterior, θ_l and θ_p denote hyper parameters of distributions of the likelihood and the prior, respectively; and $p(D \mid \theta_p, \theta_l)$ presents the normalization constant.

The expression of Equation (2) gives a solution of learning the latent function f from given training data set D and infers the unknown output value at a new point $\boldsymbol{x}^*$. Therefore, the main difficulty lies in the solution of Equation (2), mainly the computation of

the prior and the likelihood. A Gaussian process is a stochastic process, finite numbers of which have a consistent joint Gaussian distribution [18]. The assumed Gaussian process prior over the latent functions is presented by

$$p(\boldsymbol{f} \mid \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m) = N(\boldsymbol{m}(\boldsymbol{X}), \boldsymbol{K})$$

where, $\boldsymbol{f} = [f_1, f_2, \cdots, f_m]$ is the vector of latent functions, $\boldsymbol{m}(\boldsymbol{X})$ is the vector of mean function and $\boldsymbol{K}$ is the kernel matrix. Basically a Gaussian process can totally be determined by the mean function and this kernel matrix. The elements k_{ij} of matrix $\boldsymbol{K}$ are calculated by kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. A valid kernel function shall be capable to guarantee a positive semi definite $\boldsymbol{K}$. A commonly used kernel function is Gaussian kernel function

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \nu^2 \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\lambda^2}\right) \tag{3}$$

where ν^2 denotes the prior variance and λ controls rate of decay of the kernel function value. An understanding of the kernel function is that kernel function defines the relationship between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ by their distance, the idea behind which is close to the automatic relevance determination (ARD) [19]. The parameters ν and λ are also termed hyper parameters. As previously assumed, the observed outputs $\{y_i\}$ are mixed with Gaussian noise. Therefore, the likelihood $p(\boldsymbol{y} \mid \boldsymbol{f}, \theta_l)$ can be computed by

$$p(\boldsymbol{y} \mid \boldsymbol{f}, \theta_l) = p(\boldsymbol{y} \mid \boldsymbol{f}, \sigma) = \prod_{i=1}^{m} N(\boldsymbol{f}_i, \sigma^2) = N(\boldsymbol{f}, \sigma^2 \mathbf{I}_{m\times m}) \tag{4}$$

For simplicity, the mean function $m(\boldsymbol{x})$ is set to be zero. According to Bayesian decision theory [20], the posterior distribution can be obtained by

$$\begin{aligned} p(\boldsymbol{f} \mid D, \sigma, \boldsymbol{K}) &\propto N(\boldsymbol{f}, \sigma^2 \mathbf{I}_{m\times m}) N(0, \boldsymbol{K}) \\ &\propto N(\boldsymbol{K}(\boldsymbol{K} + \sigma^2 \mathbf{I}_{m\times m})^{-1}\boldsymbol{y}, \boldsymbol{K}^{-1} + \sigma^{-2}\mathbf{I}_{m\times m}) \end{aligned} \tag{5}$$

The outcome of Equation (5) can further be used to compute $f(\boldsymbol{x}^*)$ at the test location $\boldsymbol{x}^*$ by

$$\begin{aligned} p(f^*|\boldsymbol{f}, \boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{K}) &\propto N(\boldsymbol{k}^* \boldsymbol{K}^{-1} \boldsymbol{f}, \boldsymbol{k}^{**} - (\boldsymbol{k}^*)^{\mathrm{T}} \boldsymbol{K}^{-1} \boldsymbol{k}^*) \\ p(f^*|D, \boldsymbol{x}^*, \sigma, \boldsymbol{K}) &= \int p(f^*|\boldsymbol{f}, \boldsymbol{x}^*) p(\boldsymbol{f}|D, \sigma^2, \boldsymbol{K}) \mathrm{d}\boldsymbol{f} \\ &\propto N((\boldsymbol{k}^*)^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{y}, k^{**} - (\boldsymbol{k}^*)^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{k}^*) \end{aligned} \tag{6}$$

where $\boldsymbol{k}^* = [k(\boldsymbol{x}^*, \boldsymbol{x}_1), k(\boldsymbol{x}^*, \boldsymbol{x}_2), \cdots, k(\boldsymbol{x}^*, \boldsymbol{x}_m)]^{\mathrm{T}}$ and $k^{**} = k(\boldsymbol{x}^*, \boldsymbol{x}^*)$, respectively.

Therefore,

$$\begin{aligned} f^* &= (\boldsymbol{k}^*)^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{y} \\ \sigma^2_{f^*} &= k^{**} - (\boldsymbol{k}^*)^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{k}^* \end{aligned} \tag{7}$$

where $\boldsymbol{C} = \boldsymbol{K}(\boldsymbol{X}) + \sigma^2 \mathbf{I}_{m\times m}$ is the kernel matrix which contains all the potential information of the given data set D. The expression of Equation (7) is used for the interpolation over the sparse history data of the objective variable y, which is introduced in Section 3.

3. Mechanism of the Proposed Calibration Method.

3.1. Interpolation over the sparse history data of actual output y.

The situation is that there are only some sparsely sampled y values and lots of densely sampled explanatory variables $\boldsymbol{x}$, which is illustrated in Figure 2. This is also mentioned and described as the multirate property [15]. In order to make it tractable to update the soft sensor model sufficiently, those missing values of objective variable y, due to the considerably slow sampling rate, shall be estimated such that x corresponds to y pair wise. In the proposed calibration method, the interpolation of missing y is handled via Bayesian Gaussian process regression.

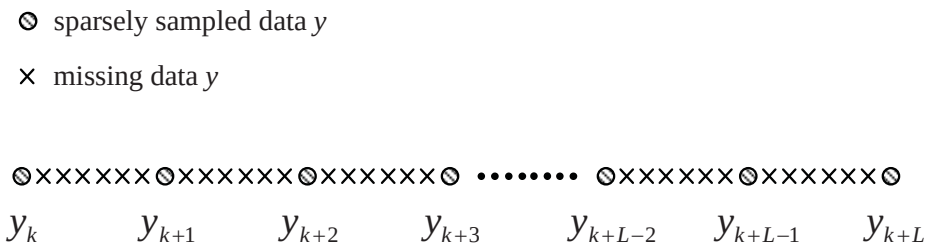


FIGURE 2. Illustration of sampled and missing data of actual output y

Before the interpolation, the window width which represents the number of the sampled history data of y is selected properly. Suppose that the window width is L, and then we have $\boldsymbol{y}_{sparse} = \{y_i\}_{i=k}^{k+L}$. Apparently the obtained $\boldsymbol{y}_{sparse}$ can be considered as a time series and the interpolation over $\boldsymbol{y}_{sparse}$ is actually the imputation of a time series. Similar to Equation (1), we can formulate the relationship between real y values and the noisy y values by

$$y_i = \hat{y}_i + \varepsilon_i \tag{8}$$

As previously mentioned the obtained $\boldsymbol{y}_{sparse}$ is handled as a time series regardless of the explanatory variables $\boldsymbol{x}$, because it becomes complex and unnecessary to involve the explanatory variables $\boldsymbol{x}$ to formulate Equation (8) exactly the same as Equation (1). Instead, the $\boldsymbol{x}$ is replaced by the time variable t_i which is simplified as subscript i. For instance, the above $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is formulated as $k(i, j)$. Suppose that the ratio between sampling cycle of y and that of $\boldsymbol{x}$ is r. Then we can use $\{k+i\}_{i=0}^{L}$ and $\{\{k+i+\frac{s}{r}\}_{s=0}^{r}\}_{i=0}^{L}$ to denote the time variables of sampled $\boldsymbol{y}_{sparse}$ and that of the missing y values, respectively. According to Equation (7), the expectations of the missing data of y are obtained as well as the corresponding variances by

$$\begin{aligned} \hat{\boldsymbol{y}} &= \{\{\hat{y}_{k+i+\frac{s}{r}}\}_{s=0}^{r}\}_{i=0}^{L} \\ \hat{\sigma} &= \{\{\hat{\sigma}_{k+i+\frac{s}{r}}\}_{s=0}^{r}\}_{i=0}^{L} \end{aligned} \tag{9}$$

When the GPR interpolation over the sparse history y data is handled, as we can see in Figure 3, a zone of missing data is obtained. The zone between the two dash lines describes that the real y values are distributed in this zone with a certain high possibility. Although we have obtained the expectations of those missing data, data involved in the dash circle is selected to update the soft sensor, because these selected data can describe the latest behavior of the process.

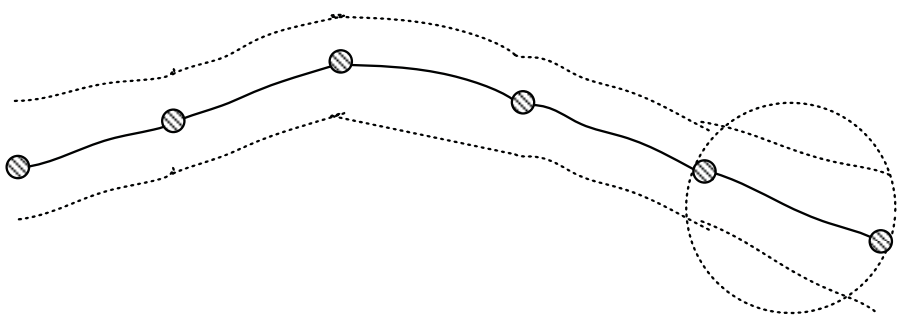


FIGURE 3. Missing data zone by GPR-based interpolation over history data y

3.2. **Training candidate models to update soft sensor.** In this section a proper number of candidate models of the soft sensor are trained by using the above selected datasets to update the original soft sensor model. Note that the thought of ensemble is an effective and widely used solution to incorporate multiple sources of information [21-24]. The soft sensor model is ultimately calibrated by an ensemble of the multiple candidate models.

By using Bayesian Gaussian process regression over the sparse history data y within a data window, a missing data zone of target variable y can be obtained. As shown in Figure 3, only the part of the data zone within the dash circle is chosen to update the soft sensor. Properly the width of this zone is chosen to be the 95% confidence interval, i.e., $\mu \pm 1.96\sigma$. For the purpose of better understanding, this local data zone is presented in Figure 5. The two dash lines represent the boundaries of $\mu \pm 1.96\sigma$. Then a proper number of datasets are selected from this data zone to train candidate models. We evenly divide this data zone into several sub-zones. As we can see in Figure 5, the data zone is divided into N sub-zones and correspondingly we can obtain N datasets. We assign a weight to each dataset because the values of each dataset correspond to a probability of being the actual values. Given the obtained datasets $\{D_i\}_{i=1}^N$, a total of N candidate models $\{f_i(\boldsymbol{x}, \theta)\}_{i=1}^N$ are trained based on the soft sensor model $f(\boldsymbol{x}, \theta)$ as shown in Figure 1. The N trained models are weighted by the probability of the used dataset. For simplicity, we use w_{prior} to denote the prior weight of the candidate model. It makes sense that values near μ have the larger w_{prior} than those far away from μ. The sum of all w_{prior} shall equal one. Assuming that N equals 7 for simplicity, the weights of the candidate models are illustrated in Figure 4.

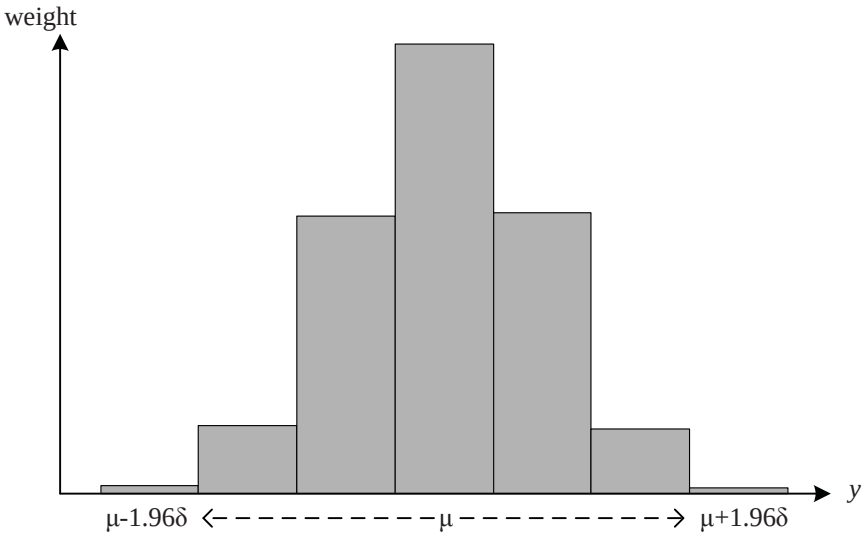


FIGURE 4. Prior weights of candidate models

Furthermore, we also use a validation point to validate the accuracy of the candidate models. An additional weight is assigned to each of the candidate models according to their estimation errors at the validation point. Let us use w_{add} to denote the additional weight. As designed, smaller estimation error is awarded with larger w_{add}. The final weight of the candidate models is the product of the prior weight and additional weight, i.e., $w_{prior} \times w_{add}$. The next important thing is to integrate the candidate models so as to calibrate the soft sensor model. By using the final weight the integration is handled by weighted sum of the candidate models. The algorithm for this process is given as below.

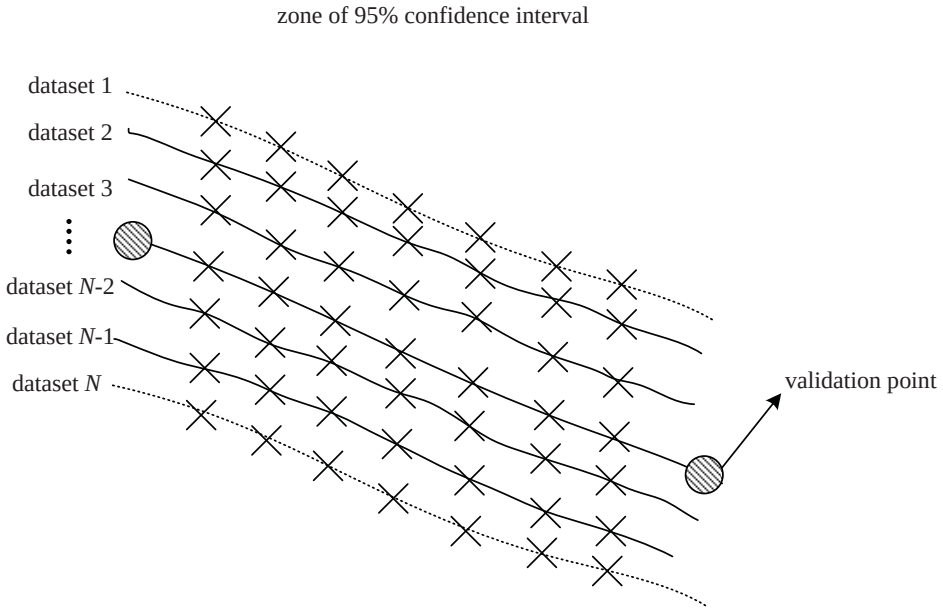


FIGURE 5. Select data points to obtain multiple datasets

Algorithm of updating the soft sensor

1. According to results of Equation (9), obtain multiple datasets

$$\{D_i\}_{i=1}^{N} = \left\{ \hat{\boldsymbol{y}} \pm \frac{1.96c}{(N-1)/2} \hat{\sigma} \right\}_{c=0}^{(N-1)/2};$$

2. Based on the soft sensor model $f(\boldsymbol{x}, \theta_k)$, use $\{D_i\}_{i=1}^{N}$ to update soft sensor model so that multiple candidate models are obtained by $\{f_i(\boldsymbol{x}, \theta_{k+L,i})\}_{i=1}^{N}$ and assign prior weight $\{w_{prior}(i)\}_{i=1}^{N}$;

3. Use the obtained $\{f_i(\boldsymbol{x}, \theta_{k+L,i})\}_{i=1}^{N}$ to predict the value of the validation point, according to the estimation error $\{e_i\}_{i=1}^{N}$, calculate the additional weight by $w_{add}(i) = \dfrac{1}{e_i \sum_{i=1}^{N} \frac{1}{e_i}}$ and the final weights of the candidate model are calculated by $\{w_i = w_{add}(i) \times w_{prior}(i)\}_{i=1}^{N}$;

4. The parameters of the soft sensor model are then updated by integration of the parameters of the multiple candidate models: $\theta_k \to \theta_{k+L} = \sum_{i=1}^{N} w_i \theta_{k+L,i}$.

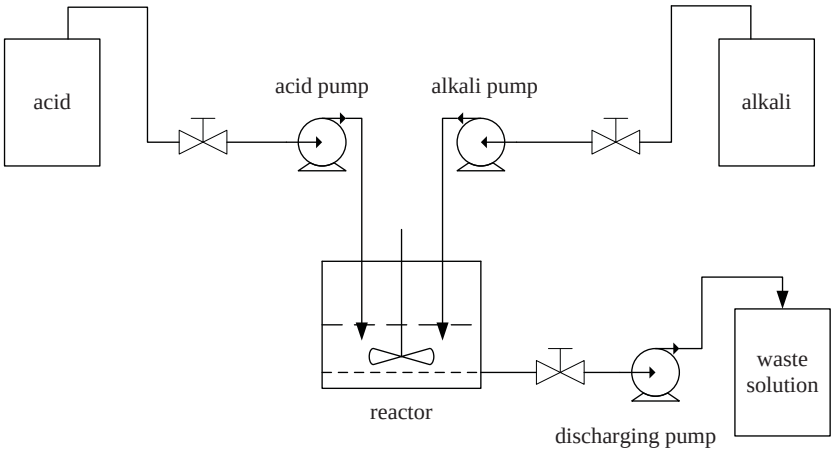
4. **Case Studies.** Previously we have given the implementation of the calibration method. Experiments on a real pH neutralization reactor were conducted to test the validation and the results are given.

Since the process of pH neutralization has high nonlinearity [25], it has been widely used by many people to test their proposed methods. First a basic introduction of the facility used in the experiment is given.

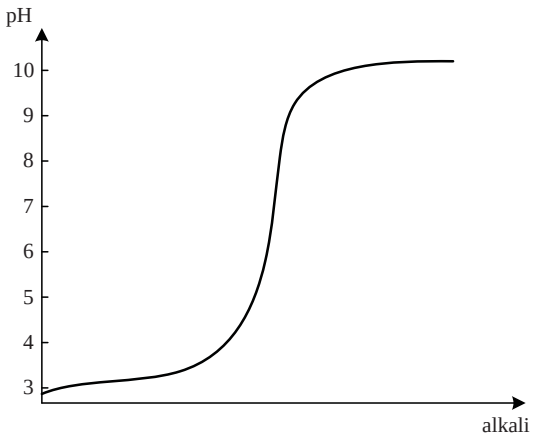
As shown in Figure 6(a), the facility has three plastic barrels, four pulse pumps and a reaction tank in the upper right. The three barrels are for acid solution (H_2SO_4), alkali solution (NaOH) and waste solution, respectively. Only three pulse pumps are used for pumping acid, alkali and waste solution with the fourth pump spared. The reaction tank is

where pH neutralization proceeds. In Figure 6(b), we can see that acid solution and alkali solution are pumped into the reaction tank where a stirrer keeps stirring to facilitate the neutralization reaction. When the level of the waste solution is high, the waste solution pump starts to work. The pH neutralization curve is given in Figure 6(c). This curve shows that the pH neutralization has severe nonlinearity.

(a)



(b)



(c)

FIGURE 6. Experiment facility and pH neutralization property: (a) pH neutralization reactor; (b) schematic diagram of the pH neutralization facility; (c) pH neutralization curve

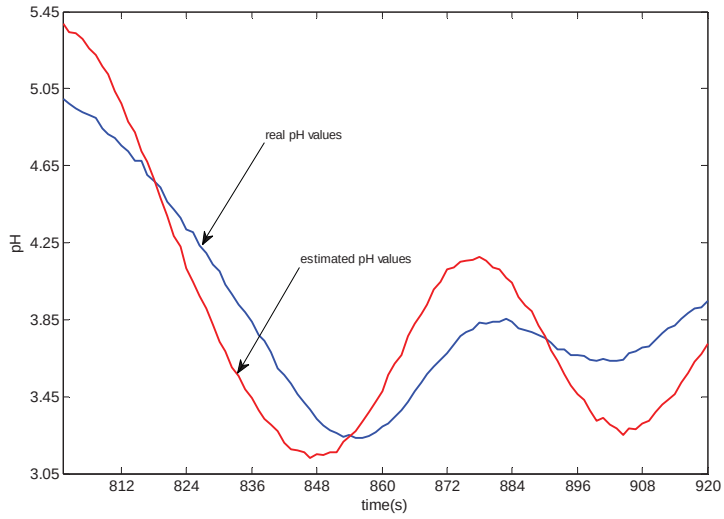
We give the mechanism model of pH neutralization reaction as below.

$$\begin{cases} A\dfrac{\mathrm{d}[h(t)y(t)]}{\mathrm{d}t} = bu(t) - aF(t) - p(t)y(t) \\ A\dfrac{\mathrm{d}h(t)}{\mathrm{d}t} = u(t) + F(t) - p(t) \\ \mathrm{pH}(t) = \lg \dfrac{y(t) + \sqrt{y^2(t) + 4K_w}}{2K_w} \end{cases} \tag{10}$$

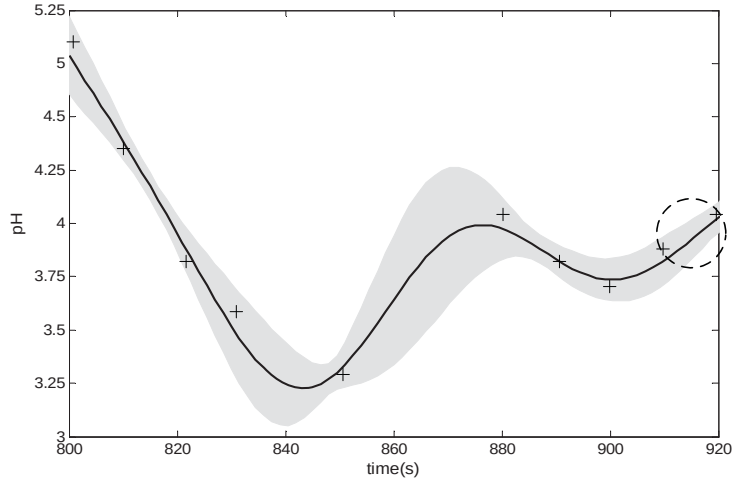
where $\mathrm{pH}(t)$ is the pH value; $h(t)$ is the level of the reactor; $y(t)$ is defined as $y(t) = [\mathrm{OH}^-] - [\mathrm{H}^+]$; $u(t)$ is the alkali pump flow; $F(t)$ is acid pump flow; $p(t)$ is the effluent flow. A, a, b and k_w are constants: A is base area of the reactor; a and b are concentrations of acid and alkali solutions; $k_w = 10^{-14}$ is water equilibrium constant.

In this experiment the soft sensor model is built by using the mechanism of Equation (10). Also the concentration of the alkali solution i.e., b is considered as the changeable parameter, which influences the process behavior of the neutralization reaction. During the experiment, b was manually changed so that the model mismatch between the soft sensor model and the real process model becomes considerably large. At the beginning of the experiment the soft sensor was programmed not to calibrate itself timely. Then the performance of soft sensor deteriorated and the estimation of the soft sensor became unreliable as shown in Figure 7(a). Then the proposed calibration method was then launched to update the soft sensor. The symbol '+' denotes a sampled data which is used for Bayesian GPR interpolation. As we can see that the selected sampled data are not evenly distributed which is intentionally caused by 'data loss', and this usually occurs during practical industrial processes. Fortunately when handling GPR, it is not strictly required to make the input x evenly distributed. In Figure 7(b), we can see in the region where the sampled data are denser the corresponding variances are smaller and that means the estimation in the region is more precise. The grey area is the obtained missing data zone. As shown in Figure 7(b), we only used the data zone which is drawn in the dash circle and ranges from 910s to 920s on the time line. By using the method presented in Section 3.2, the soft sensor model can update the distorted parameter so as to decrease the model mismatch. After the soft sensor model was calibrated, from 920s to 925s, the estimated pH value was drastically regulated from about 3.7 to about 4.26 while the real pH value shifted from about 3.95 to about 4.03. Although the regulation was first overshot, the estimated pH value was then gradually regulated to trace the trajectory of the real pH value from 925s to 1024s as shown in Figure 7(c). Therefore, from the experiment results the proposed calibration method for soft sensor model can be proved feasible.

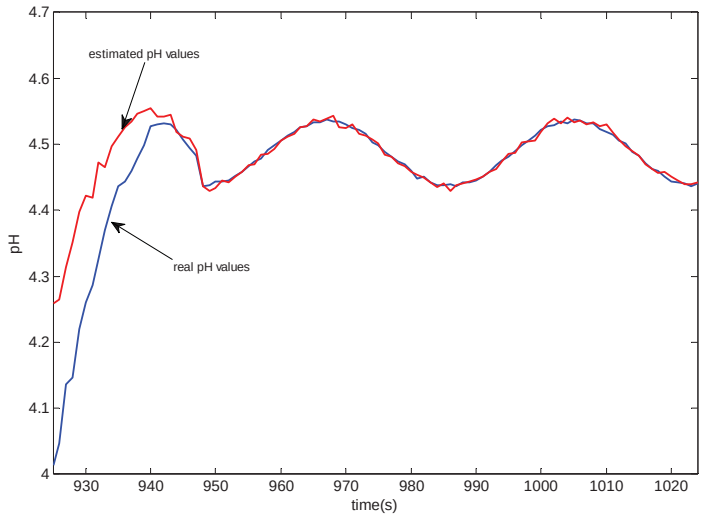
In order to further validate the proposed calibration method, we also conducted a comparative experiment with another calibration method based on Kalman filter. This comparative calibration method is a dual-rate Kalman filter based calibration method [15]. The idea behind the method is: when the real target variable is available, the method uses the real value to obtain innovation so as to process the computation of Kalman filter; otherwise, the method uses the estimated values of the soft sensor to get innovation so that the Kalman filter model can be computed. In such a way, the soft sensor is calibrated. In the above individual experiment, we have preliminarily proved the feasibility of the proposed calibration method. For a better comparison, the experiment shall be conducted under the same condition. However, this is practically difficult. Also note that the ultimate goal of solving the sensor drift problem is to calibrate the estimated target values as close to the real values as possible and that a simulation experiment is quite perfect to provide the same condition for comparative experiments. Thus, we



(a)



(b)



(c)

FIGURE 7. Experiment results: (a) soft sensor performance deterioration; (b) GPR-based interpolation of history data of pH values; (c) calibration effect curve

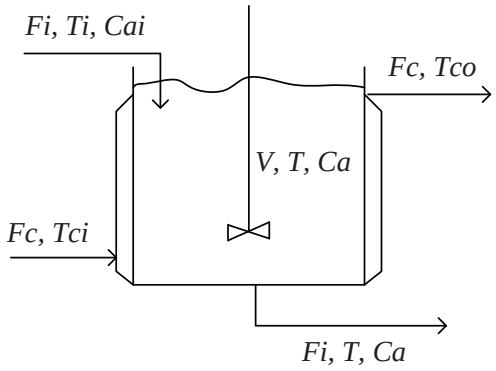


FIGURE 8. Continuous stirred tank reactor

decided to simulate on a continuous stirred tank reactor (CSTR). As shown in Figure 8, the simulation facility used is a CSTR where an irreversible and exothermic reaction takes place with the reactants mixed perfectly. The governing equations of this CSTR are given as below [26].

$$\begin{aligned}\frac{\mathrm{d}T(t)}{\mathrm{d}t} &= \frac{F_i}{V}(T_i - T(t)) - \frac{\Delta H k_0 C_a(t)}{\rho C_p}\exp\left(\frac{-E}{\mathrm{R}T}\right) \\ &\quad + \frac{\rho_c C_{pc}}{\rho C_p V}F_c(t)\left(1 - \exp\left(\frac{-hA}{F_c(t)\rho C_p}\right)\right)(T_{ci} - T(t))\end{aligned} \tag{11}$$

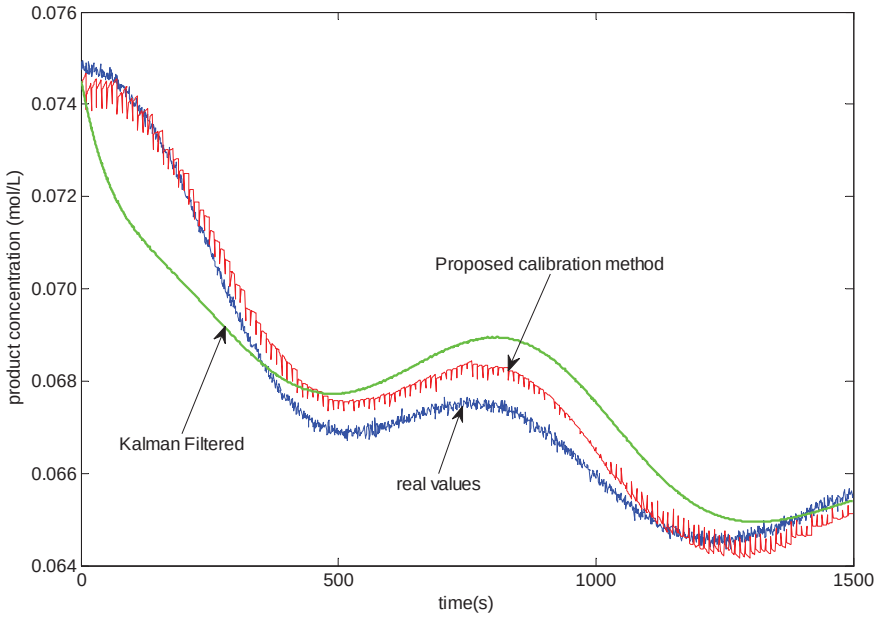
$$\frac{\mathrm{d}C_a(t)}{\mathrm{d}t} = \frac{F_i}{V}(C_{ai} - C_a(t)) - k_0 C_a(t)\exp\left(\frac{-E}{\mathrm{R}T}\right) \tag{12}$$

where C_a is the product concentration, T is the product temperature and F_c is the coolant flow rate. The parameters of the CSTR and some steady state operating conditions are given in Table 1.

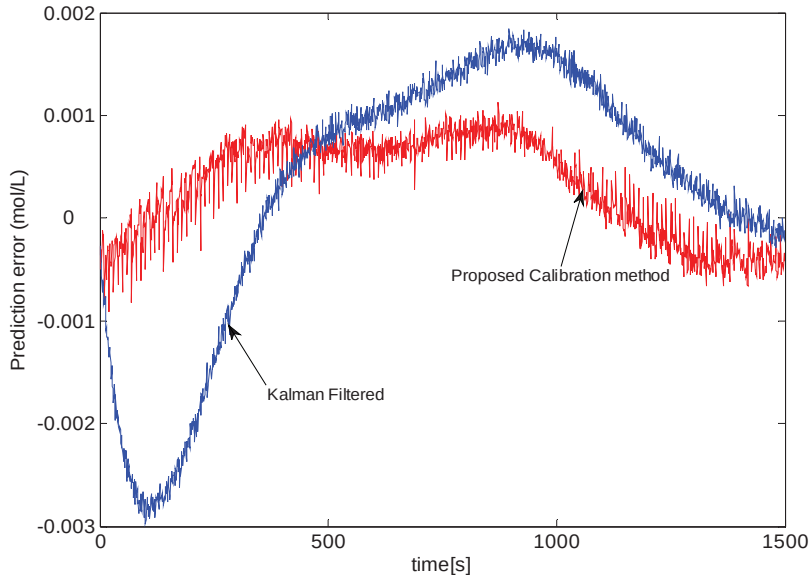
TABLE 1. The parameters of the CSTR and some steady state operating conditions

Parameters	Symbol	Value
Feed flow-rate	F_i	100 L/min
Feed concentration	C_{ai}	1 mol/L
Feed temperature	T_i	350 K
Reactor volume	V	100 L
Reactor rate constant	k_0	7.2×10^{10} min^{-1}
Activation energy term	E/R	1×10^4 K
Heat of reaction	ΔH	-2×10^5 cal/mol
Reactant density	ρ	1000 g/L
Reactant specific heat	C_p	1 cal/g/K
Heat transfer term	$h\ A$	7×10^5 cal/min/K
Coolant inlet temperature	T_{ci}	350 K
Coolant density	ρ_c	1000 g/L
Coolant specific heat	C_{pc}	1 cal/g/K

For simplicity, during the simulation temperature $T(t)$ was selected as the process variable, and product concentration $C_a(t)$ was chosen as the objective variable. The reactor rate constant k_0 is usually affected by factors such as temperature, and activation energy. Note that the reactor rate constant k_0 greatly affects the reaction. Thereby it was selected as the process's volatile parameter. During the simulation experiment,



(a)



(b)

FIGURE 9. Comparative experiment results: (a) comparative results of calibration performance; (b) comparative results of prediction error

the reactor rate constant k_0 was programmed to vary such that the real values of the product concentration $C_a(t)$ varied as shown in Figure 9(a). The relevant parameters of the working point are specified: the coolant flow rate was fixed at 97 L/min and other parameters referred to those corresponding values in Table 1. As we can see in Figure 9(a), the calibration performance of the proposed calibration method is better than that of the dual-rate Kalman filter based calibration method. If the prediction error of the soft sensor model after calibration is large, the corresponding calibration method is obviously not that effective. The conclusion that the proposed soft sensor calibration method is effective can be verified by the comparative results of prediction error in Figure 9(b).

Through results of the individual experiment on a pH neutralization facility and comparative simulation experiments on a CSTR with the chosen Kalman filter based calibration method proposed by other scholars, the proposed calibration method based on Gaussian process regression is proved feasible and effective.

5. **Conclusions.** In this paper, we proposed a calibration method based on Gaussian process regression. By handling interpolation over finite number of sparse history target values, the missing data zone of the target variable is obtained. A proper number of datasets are selected from the missing data zone. By using the datasets several candidate models are trained based on the soft sensor model. Finally the soft sensor is calibrated by a weighted combination of multiple candidate models. The experiment results show that the proposed soft sensor calibration method is feasible. However, there is still an important issue which is not discussed in the paper. The background of the proposed calibration method is that the sampling cycle of the target variable is far larger than that of the process variables. The cycle ratio between the two sampling cycles directly affects the performance of the calibration method. Our future work will involve the study of the relationship between the cycle ratio and the calibration performance.

Acknowledgment. This work was supported by the National Basic Research Program of China (2012CB720500). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] L. Fortuna, S. Graziani, A. Rizzo and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*, Springer, London, 2007.

[2] P. Kadlec, B. Gabrys and S. Strandt, Data-driven soft sensors in the process industry, *Comput. Chem. Eng.*, vol.33, no.4, pp.795-814, 2009.

[3] K. Hiromasa, A. Masamoto and F. Kimito, Applicability domains and accuracy of prediction of soft sensor models, *AIChE Journal*, vol.57, no.6, pp.1506-1513, 2011.

[4] G. Martinelli and M. C. Carotta, Thick-film gas sensors, *Sens. Actuators B*, vol.23, pp.157-161, 1995.

[5] H. Meixner and U. Lampe, Metal oxide sensors, *Sens. Actuators B*, vol.33, pp.198-202, 1996.

[6] F. Hossein-Babaei and V. Ghafarinia, Compensation for the drift-like terms caused by environmental fluctuations in the responses of chemoresistive gas sensors, *Sens. Actuators B*, vol.143, pp.641-648, 2010.

[7] A. Vergara, K. D. Benkstein, C. B. Montgomery and S. Semancik, Demonstration of fast and accurate discrimination and quantification of chemically similar species utilizing a single cross-selective chemiresistor, *Anal. Chem.*, vol.86, no.14, pp.6753-6757, 2014.

[8] X. Zuo, F. Tu, H. Y. Qing and X. L. Luo, Advanced control of acetylene hydrogenation reactor (II) – Soft sensor and its engineering practice, *Control and Instruments in Chemical Industry (China)*, vol.30, no.2, pp.19-21, 2003.

[9] R. Feng, Y. J. Zhang, Y. Z. Zhang and H. H. Shao, Drifting modeling method using weighted support vector machines with application to soft sensor, *Acta Automat. Sin.*, vol.30, pp.436-441, 2004.

[10] J. J. Macias, P. Angelov and X. W. Zhou, A method for predicting quality of the crude oil distillation, *Proc. of the Int. Symp. Evolving Fuzzy Syst.*, Lake, United Kingdom, pp.214-220, 2006.

[11] X. Wang, U. Kruger and G. W. Irwin, Process monitoring approach using fast moving window PCA, *Ind. Eng. Chem. Res.*, vol.44, pp.5691-5702, 2005.

[12] P. Kadlec and B. Gabrys, Adaptive local learning soft sensor for inferential control support, *Proc. of the Int. Confer. Comput. Intel. Modelling Contr. Auto.*, Vienna, Austria, pp.243-248, 2008.

[13] H. Jin, X. Chen, J. Yang and L. Wu, Adaptive soft sensor modeling framework based on just-in-time learning and kernel partial least squares regression for nonlinear multiphase batch processes, *Computers and Chemical Engineering*, vol.71, pp.77-93, 2014.

[14] S. Khatibisepehr, B. Huanga, F. Xu and A. Espejo, A Bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry, *Journal of Process Control*, vol.22, pp.1913-1929, 2012.

[15] Y. Wu and X. L. Luo, A novel calibration approach of soft sensor based on multirate data fusion technology, *Journal of Process Control*, vol.20, no.10, pp.1252-1260, 2010.
[16] J. M. Bernardo and A. F. Smith, *Bayesian Theory*, 2nd Edition, Wiley, Chichester, 2006.
[17] A. O'Hagan and J. J. Forster, *Bayesian Inference (Kendall's Advanced Theory of Statistics 2B)*, 2nd Edition, Edward Arnold, London, 2004.
[18] J. Quinonero-Candela, C. E. Rasmussen and C. K. I. Williams, Approximation methods for Gaussian process regression, *Large-Scale Kernel Machines*, pp.203-223, 2007.
[19] D. J. C. MacKay, Bayesian nonlinear modeling for the prediction competition, *Ashrae Transactions*, vol.100, no.2, pp.1053-1062, 1994.
[20] M. Kuss, *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*, Ph.D. Thesis, TU Darmstadt, Germany, 2006.
[21] T. Sun, L. Jiao, F. Liu, S. Wang and J. Feng, Selective multiple kernel learning for classification with ensemble strategy, *Pattern Recognition*, vol.46, pp.3081-3090, 2013.
[22] L. Li, R. Stolkin, L. Jiao, F. Liu and S. Wang, A compressed sensing approach for efficient ensemble learning, *Pattern Recognition*, vol.47, pp.3451-3465, 2014.
[23] Q. Hu, L. Li, X. Wu, G. Schaefer and D. Yu, Exploiting diversity for optimizing margin distribution in ensemble learning, *Knowledge-Based Systems*, vol.67, pp.90-104, 2014.
[24] C. Tong, A. Palazoglu and X. Yan, Improved ICA for process monitoring based on ensemble learning and Bayesian inference, *Chemometrics and Intelligence Laboratory Systems*, vol.135, pp.141-149, 2014.
[25] X. L. Luo, *Chemical Process Dynamics*, Chemical Industry Press, Beijing, 2005 (in Chinese).
[26] Z. Xu, J. Zhao and J. Qian, Nonlinear MPC using an identified LPV model, *Industrial and Engineering Chemistry Research*, vol.48, pp.3043-3051, 2009.