ADCONIP 2014 Hiroshima

Co-learning with a locally weighted partial least squares for soft sensors of nonlinear processes

Yoshiyuki Yamashita[1] and Kaoru Sasagawa[1]

***Abstract*— A method to improve adaptivity of soft sensors is investigated in this paper. Soft sensors have become very important in the chemical industry to achieve a highly efficient, high-quality and safe production system. Among the various methods, partial least squares (PLS) method is the most used for soft sensors. In this research, a co-learning style locally weighted PLS method which utilizes a semi-supervised regression is proposed to estimate a process value. The method is applied to a simulated reactor process, and the results clearly show an improvement in the estimation accuracy compare with the conventional method.**

I. INTRODUCTION

In process industry, soft sensors have been widely used to estimate important variables[1], [2], [3]. The most popular methods for soft sensors are data-driven approaches such as multi-variate regression, partial least squares (PLS) and neural networks[4], [5]. They are using single model for relatively wide operating regions.

To increase the adaptivity of soft sensors, several types of just-in time methods have been shown to be useful[6]. To continuously update the estimation model of the soft-sensor, recursive PLS has been proposed[7]. Recently, to cope with changes in process characteristics as well as nonlinearity, locally weighted partial least squares (LW-PLS) was developed[8], [10].

In LW-PLS, data containing measurements of variables must be corrected to estimate and compile the database in advance. In the sparse region of the database, estimation performance of the method becomes poor. In this paper, a co-training style semi-supervised LW-PLS algorithm named co-learning style locally weighted PLS (COPLS) is proposed. This algorithm is a method to combine LW-PLS with co-training style self training to use unabeled samples in the sparce region. By combining self training with LW-PLS, the chances for online updates of the database are increased. As the result, the new method will improve adaptivity of the soft sensor.

The rest of this paper is constructed as follows. In section II, the co-training style LW-PLS algorithm COPLS is explained. In section III, a case study of the algorithm is discussed in comparison with the LW-PLS. Finally, this research is concluded in section IV.

II. METHOD

A. *Locally weighted PLS*

Typical data-driven soft sensors are constructed using a regression model. Among the various types of regression models, PLS model has been widely used for soft sensors[9], [4].

Once the model is built, usual PLS does not change its parameters even if the system changes as time elapses. This means that model maintenance is required to keep the model matched to the process. Therefore, various adaptive methods have been developed[1]. LW-PLS was proposed[8], which introduced a just-in-time learning concept to PLS. LW-PLS constructs a local PLS model by prioritizing samples in a database according to their similarity with a query sample. The similarity is usually defined based on the Euclidian distance or Mahalanobis distance. In this paper, Euclidian distance is considered.

Let the i-th sample of input and output variables be

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{iM}]^{\mathrm{T}} \tag{1}$$

$$\mathbf{y}_i = [y_{i1}, y_{i2}, \cdots, y_{iL}]^{\mathrm{T}} \tag{2}$$

where M and L are the number of input and output variables. In this paper, the number of output variables is assumed to be one.

The similarity ω_i between a query sample $\mathbf{x}_{\mathrm{q}}$ and $\mathbf{x}_i$ is measured using the normal Euclidian distance d_i.

$$d_i = \sqrt{(\mathbf{x}_i - \mathbf{x}_{\mathrm{q}})^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_{\mathrm{q}})} \tag{3}$$

$$\omega_i = \exp(-\frac{d_i}{\sigma_d \varphi}) \tag{4}$$

where σ_d is the standard deviation of $d_i (i = 1, 2, \cdots, N)$ and φ is a localization parameter, N is the number of samples.

For a query sample $\mathbf{x}_{\mathrm{q}}$, the similarity ω_i is calculated using Eq. 4 and a local PLS model is constructed with a similarity matrix $\mathbf{\Omega}$.

$$\mathbf{\Omega} = \mathrm{diag}(\omega_1, \omega_2, \cdots, \omega_N) \tag{5}$$

In an extreme case, if the similarity matrix $\mathbf{\Omega}$ is an identity matrix, the LW-PLS becomes the same as usual PLS.

[1] Yoshiyuki Yamashita is with Department of Chemical Engineering, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan yama_pse@cc.tuat.ac.jp

B. Self Training and Regression

Most data-driven soft sensors use the machine learning method. Traditionally, there have been two fundamentally different approaches in machine learning. The first approach is supervised learning, which requires labeled data. The second is unsupervised learning, which does not require labeled data. Semi-supervised learning is halfway between supervised and unsupervised learning. It uses unlabeled data in addition to labeled data. In this study, we focused on the semi-supervised approach.

Semi-supervised learning algorithms have been eagerly studied during the past few years[11]. Research on semi-supervised learning mainly studies classification. Although semi-supervised regression is very important, not many studies have been investigated on the regression. Among some algorithms for semi-supervised regression, this study is based on the algorithms proposed by Zhou & Li[12]. The algorithm utilizes the idea of co-training, which trains two classifiers separately on two sufficient and redundant views. It employs two k-nearest neighbor regressors with different distance metrics. The influence of the labeling of unlabeled examples on the labeled examples is analyzed to choose appropriate unlabeled examples to label.

For each unlabeled example $\mathbf{x}_u$, its k-nearest labeled examples R_u is identified. The most confidently labeled example is identified by maximizing the following value

$$\delta_{\mathbf{x}_u} = \sum_{\mathbf{x}_i \in R_u} (\mathbf{y}_i - h(\mathbf{x}_i))^2 - \sum_{\mathbf{x}_i \in R_u} (\mathbf{y}_i - h'(\mathbf{x}_i))^2, \tag{6}$$

where h is the original regressor, and h' is the one refined with $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$. Note that $\hat{\mathbf{y}}_u = h(\mathbf{x}_u)$.

The output of the final regressor is the average of both regressors.

$$h^*(\mathbf{x}) = \frac{1}{2}(h(\mathbf{x}) + h'(\mathbf{x})) \tag{7}$$

C. COPLS

The basic idea of our proposed method is to combine the self-training method with LW-PLS for the modeling of soft sensors. To choose appropriate unlabeled examples to label, a labeling confidence is estimated. The labeling confidence is calculated by consulting the influence of the labeling of labeled and unlabeled examples on the labeled examples. The mechanism for estimating the labeling confidence is the key to the algorithm. The original self-training algorithm is modified in several ways for this purpose.

In this study, both absolute and relative evaluation of the confidence was employed. We also added conditions to activate the self-learning. We did not repeat the addition of the data.

Let D_S be a selected k-nearest local subset of measured values, that is real value labeled data. Similar to Eq. 6, the following values were used as a confidence evaluation in this algorithm.

$$\delta e = \sum_{\mathbf{x}_i \in D_S} (y_i - \hat{y}_i(\mathbf{x}_i))^2 - \sum_{\mathbf{x}_i \in D_S} (y_i - \hat{y}'_i(\mathbf{x}_i))^2 \tag{8}$$

where $\hat{y}_i(\mathbf{x}_i)$ is the estimate of y for the sample $\mathbf{x}_i$ by the locally weighted PLS model, and $\hat{y}'_i(\mathbf{x}_i)$ is the one refined with the query sample and its estimates $(\mathbf{x}_q, \hat{y}_q)$.

$$\Delta e_{\max} = \max_{\mathbf{x}_i \in D_S} |y_i - \hat{y}_i| \tag{9}$$

$$\Delta e'_{\max} = \max_{\mathbf{x}_i \in D_S} |y_i - \hat{y}'_i| \tag{10}$$

Assuming that the datasets $\mathbf{x}_i$ and y_i are normalized to have zero average and unit standard deviation. The COPLS estimates the value y_q by the following procedure.

1) Calculate the Euclidian distances between the samples $\mathbf{x}_i$ and $\mathbf{x}_q$.
2) Select y_i corresponding to k-nearest $\mathbf{x}_i$ from $\mathbf{x}_q$ and generate a subset of the database.
3) Build a local PLS model for the selected subset D_S of the dataset.
4) Calculate $\hat{y}_q$ from $\mathbf{x}_q$ based on the local PLS model.
5) Calculate $\Delta e_{\max}$ for the selected subset, and only if $\Delta e_{\max} > e^*$, the following self training is activated.

When $\Delta e_{\max}$ is small, the data subset D_S is considered to be dense enough, and the refinement by self-training is not necessary. The following procedure describes the self training for sparse data subset.

1) Temporarily add $(\mathbf{x}_q, \hat{y}_q)$ to the database.
2) Select k-nearest subset from $\mathbf{x}_q$ and build a local PLS model, and calculate the estimate $\hat{y}''_q$ from the model.
3) Calculate $\Delta e''_{\max}$ and δe.
4) If $\Delta e''_{\max} < e^*$ and $\delta e > 0$ then finally add $(\mathbf{x}_q, \hat{y}_q)$ to the database. Otherwise eliminate the temporary added dataset from the database.

COPLS repeats this procedure for every new query samples and updates the database. In this procedure, e^* is an adjusting parameter to be fixed.

III. A CASE STUDY

A. Problem Definition

A schematic of the case study process is shown in Fig. 1. In this process, raw material A is continuously fed into the reactor. In the reactor, liquid product B is produced by an irreversible reaction from A to B. Reactor temperature T is

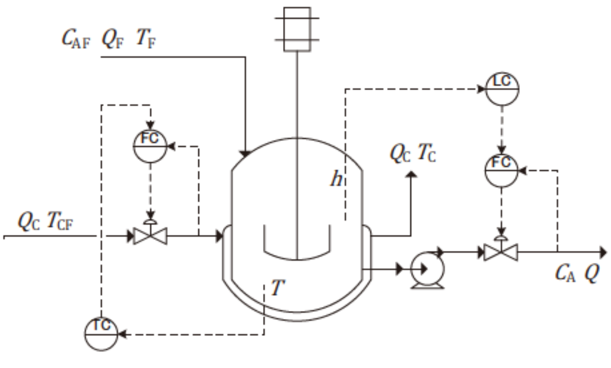

Fig. 1. Schematic of the case study plant

TABLE I

PROCESS VARIABLES

$Q = 0.1\ \mathrm{m^3/min}$	$A = 0.1666\ \mathrm{m^2}$
$Q_C = 6.6 \times 10^{-3}\ \mathrm{m^3/min}$	$k_0 = 3.0 \times 10^{10}\ \mathrm{min^{-1}}$
$T_{CF} = 300\ \mathrm{K}$	$\Delta H = -5.0 \times 10^4\ \mathrm{J/mol}$
$T = 430\ \mathrm{K}$	$\rho C_P = 2.39 \times 10^5\ \mathrm{J/(m^3 \cdot K)}$
$T_C = 384.8\ \mathrm{K}$	$\rho_C C_{PC} = 4.175 \times 10^6\ \mathrm{J/(m^3 \cdot K)}$
$T_F = 320\ \mathrm{K}$	$E/R = 8.75 \times 10^3\ \mathrm{K}$
$C_{AF} = 1.0 \times 10^3\ \mathrm{mol/m^3}$	$UA_C = 5.0 \times 10^4\ \mathrm{J/(min \cdot K)}$
$C_A = 22.4\ \mathrm{mol/m^3}$	$V_C = 0.356\ \mathrm{m^3}$
$h = 0.6\ \mathrm{m}$	

controlled by coolant flow rate Q_C to the jacket. Liquid level h in the reactor is controlled by the outlet flow rate Q_F with cascaded controller.

The mathematical model used for the simulation is summarized as follows:

$$\begin{aligned}
\frac{dC_A}{dt} &= -k_0 e^{-E/RT} C_A + \frac{Q_F C_{AF} - Q C_A}{Ah} && (11)\\
\frac{dT}{dt} &= \frac{-k_0 e^{-E/RT} C_A(-\Delta H)}{\rho C_p} && (12)\\
&\quad + \frac{Q_F T_F - QT}{Ah} + \frac{UA_C(T_C - T)}{\rho C_p Ah} &&\\
\frac{dT_C}{dt} &= \frac{Q_C(T_{CF} - T_C)}{V_c} + \frac{UA_C(T - T_C)}{\rho_C C_{PC} V_C} && (13)\\
\frac{dh}{dt} &= \frac{Q_F - Q}{A} && (14)
\end{aligned}$$

Parameters in the model are shown in Table I.

Valve characteristics of the coolant flow and outlet follows:

$$Pv(s) = \frac{K}{\tau s + 1} \qquad (15)$$

where $K = 1/16$ and $\tau = 2$ s. Both controllers were PI controllers with 1 min sampling intervals, and the control parameters were designed based on the IMC method[13].

TABLE II

PI PARAMETERS OF THE CONTROLLERS

	K_P [-]	T_I [min]
temperature	-0.01	-0.000001
coolant flow rate	0.53	0.03
liquid level	-1	0.0001
outlet flow rate	0.53	0.03

The abovementioned model was simulated for 180 days with 5 minutes sampling. During the simulation, values of six variables, T, Q, Q_F, h and Q_C, were recorded with measurement noises. System noises were also added to Q_F. Figure 2 shows the time series of the generated dataset. In this simulation, initial temperature was set to 425 K. The temperature was increased in 10 K steps several times after 100 days. The concentration C_A was assumed to be measured daily. The dataset from the first 90 days was used for training and that from the last 90 days was used for evaluation of the method.

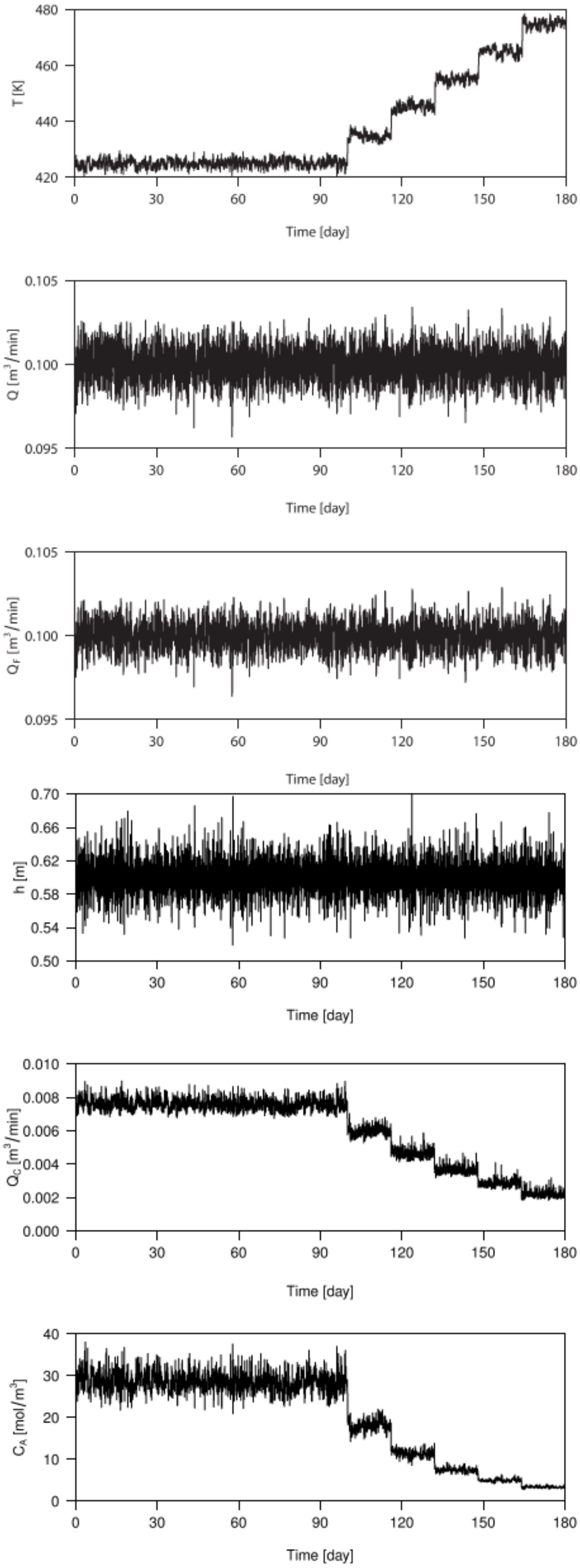

Fig. 2. Time series of the simulated dataset

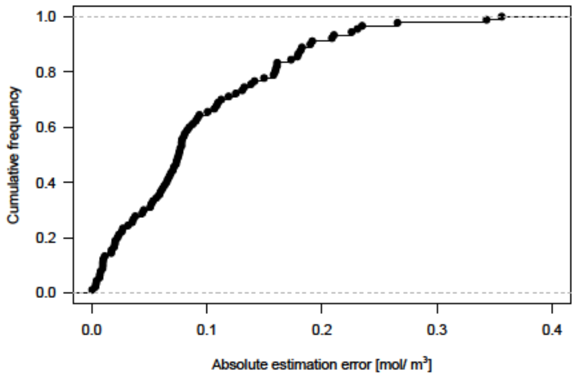

Fig. 3. Cumulative frequency of the absolute estimation error in LOOCV

TABLE III

DETERMINATION OF THE NUMBER OF LATENT VARIABLES

number of latent variables	square sum of the estimation error
1	164.465
2	142.74
3	5.27
4	5.94
5	6.09

B. Building a Soft Sensor

Based on the leave-one-out cross-validation (LOOCV), number of latent variables are determined to maximize the estimation accuracy. Table III shows the result of the estimation error for each number of latent variables. From this table, the number of latent variables are determined to three.

To investigate th effect of the tuning parameter e^* in the COPLS method, we evaluated the COPLS estimation with $e^* = 0.15, 0.20, 0.25, 0.30$ and 0.35 in this case study.

C. Estimation Results

The estimation results of C_A by LW-PLS for the last 90 days are shown in Fig. 4. Between 90 and 100 days, the operating conditions were the same as the first 90 days and the estimation was very close to the true value, where red lines cannot be observed because they overlap the black lines.. After 100 days, the estimation accuracy decreased because the operating conditions were different from the original dataset. In this region, the dataset became sparse and was hard to model by PLS linear approximation.

Figures 6 and 5 show the estimation result of C_A using the proposed COPLS method. In these figures, samples between 90 and 100 days have the same operating conditins as the trainig dataset and can be estimated very well by all methods. After 100 days, the operating conditions became gradually different from the training datasets, and the estimation accuracy of the LW-PLS method is reduced because the corresponding region of the samples in the dataset became sparse. By comparing Figs. 4 with 6 and 5, it is clear that

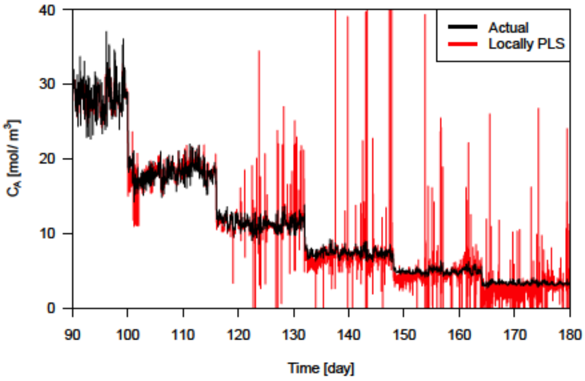

Fig. 4. Estimation using the LW-PLS method

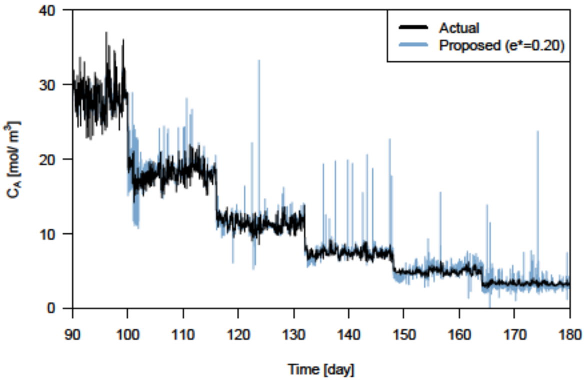

Fig. 5. Estimation using the COPLS method ($e^* = 0.20$)

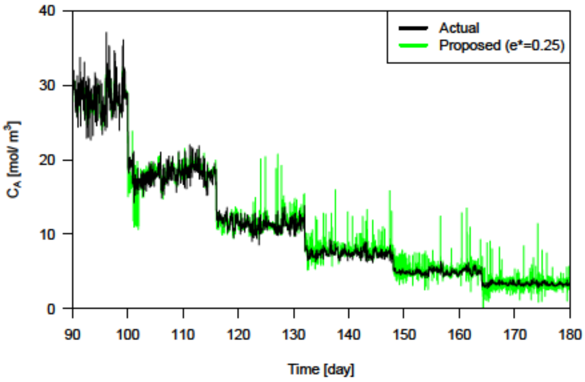

Fig. 6. Estimation using the COPLS method ($e^* = 0.25$)

TABLE IV

SUMMARY OF THE ESTIMATION ACCURACY

method	RMSE	R^2
LWPLS	2.148	0.934
COPLS ($e^* = 0.15$)	0.925	0.977
COPLS ($e^* = 0.20$)	0.907	0.987
COPLS ($e^* = 0.25$)	0.855	0.988
COPLS ($e^* = 0.30$)	0.998	0.984
COPLS ($e^* = 0.35$)	0.886	0.987

the proposed method is more accurate than the LW-PLS estimation.

To evaluate the result qualitatively, root mean square error (RMSE) and the coefficient of determination R^2 of the estimation was calculated and is summarized in Table IV for estimations using the LW-PLS and COPLS methods. As can be observed from the RSME values, the COPLS method ($e^* = 0.25$) had the best estimation accuracy among these methods. In this case, the RMSE became 39.8% of the LW-PLS estimation. The value of R^2 using the LW-PLS method came closer to that using the LW-PLS method. This also shows that the COPLS estimation is accurate.

About the computation time of COPLS method, the CPU time for one sample was at most 600 m second, where the calculation was executed as a single processor on a 3.0GHz Intel Core 2 Duo processor with 4GB of memory. When the self-training part of the procedure was not activated, the CPU time was about 200 m second.

IV. CONCLUSIONS

A co-learning style locally weighted partial least squares method was proposed to estimate process values of a process system. The original measured dataset for the regression was extended by adding unmeasured data based on the evaluation of estimated confidence values.

The method was applied to a simulation of a reactor process and the estimation accuracy was improved by more than double in RMSE compared with the original LWPLS method. The improvement in the accuracy in the sparse data region, in particular, was significant.

V. NOMENCLATURE

Symbol	Contents	Unit
A	Cross section area of the reactor	[m^2]
A_{C}	Contact area of the cooling jacket	[m^2]
C_{A}	Outlet concentration of A	[mol/m^3]
C_{AF}	Inlet concentration of A	[mol/m^3]
T	Reactor temperature	[K]
T_{C}	Outlet coolant temperature	[K]
T_{CF}	Inlet Coolant temperature	[K]
T_{F}	Feed temperature	[K]
h	Liquid level in the reactor	[m]
$-\Delta H$	Reaction heat	[J/mol]
Q	Reactor outlet flowrate	[m^3/min]
Q_{C}	Coolant flowrate	[m^3/min]
Q_{F}	Feed flowrate	[m^3/min]
U	Overall heat transfer coefficient	[W/m^{2}K]

ACKNOWLEDGMENTS

This work was supported by Grant-in-Aid for Science Research (C) 23560522 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] Kadlec P, Grbic R, Gabrys B. Review of Adaptation Mechanisms for Data-driven Soft Sensors, Comput. Chem. Eng., vol. 35, pp. 1–24, 2011.

[2] Damour C, Benne M. Soft-sensor for Industrial Sugar Crystallization: On-line Mass of Crystals, Concentration and Purity Measurement, Control Eng. Pract., vol. 18, pp. 839–844, 2010.

[3] Ram Chandra Poudel, Tatsuhiko Sakaguchi and Yoshiaki Shimizu, A Selective Approach on Data Based Quality Prediction for Quenched and Tempered Steel Reinforcement Bars, J. Chemical Engineering of Japan, vol. 46, pp. 294-301, April 2013.

[4] Kadlec P, Gabrys B, Strandt S. Data-driven Soft Sensors in the Process Industry, Comput. Chem. Eng., vol. 33, pp. 795–814,, 2009.

[5] B. Lin, B. Recke, J.K.H. Knudsen and S.B. Jørgensen, A systematic Approach for Soft Sensor Development, Computers and Chemical Engineering, vol. 31, pp. 419–425, 2007.

[6] M. Kano and K. Fujiwara, Virtual Sensing Technology in Process Industries: Trends and Challenges Revealed by Recent Industrial Applications, J. Chemical Engineering of Japan, vol. 46, pp. 1-17, January 2013.

[7] S. J. Qin, Recursive PLS algorithms for adaptive data modeling, Computers and Chemical Engineering, vol. 22, pp. 503–514, 1998.

[8] S. Kim, M. Kano, H. Nakagawa and S. Hasebe, Estimation of Active Pharmaceutical Ingredients Content Using Locally Weighted Partial Least Squares and Statistical Wavelength Selection, International Journal of Pharmaceutics, vol. 421, pp. 269-274, 2011.

[9] S. Wold, M. Sjöström and L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometrics and Intelligent Laboratory Systems, vol. 55, pp. 109–130, 2001.

[10] S. Kim, R. Okajima, M. Kano and S. Hasebe, Development of soft-sensor using locally weighted PLS with adaptive similarity measure, Chemometrics and Intelligent Laboratory Systems, vol. 124, pp. 43-49, 2013.

[11] X. Zu and A.B. Goldberg, Introduction to Semi-Supervised Learning, Morgan & Claypool Pub., 2009.

[12] Z.H. Zhou and M. Li, Semisupervised Regression with Cotraining-style Algorithms, IEEE Trans. Knowledge and Data Engineering, vol. 19, pp. 1479–1493, 2007.

[13] D.E.Seborg, T.F.Edgar, D.A.Mellichamp and F.J.Doyle III, Process Dynamics and Control, 3rd ed., Wiley 2011.