Soft-Sensor Development Using Correlation-Based Just-in-Time Modeling

Koichi Fujiwara, Manabu Kano, Shinji Hasebe, and Akitoshi Takinami
Kyoto University, Dept. of Chemical Engineering, Nishikyo-Ku, Kyoto 615-8510, Japan;
and Showa Denko K.K., Engineering Dept. Nakanosu, Oita, 870-0819, Japan

DOI 10.1002/aic.11791
Published online June 8, 2009 in Wiley InterScience (www.interscience.wiley.com).

Soft-sensors have been widely used for estimating product quality or other key variables, but their estimation performance deteriorate when the process characteristics change. To cope with such changes, recursive PLS and Just-In-Time (JIT) modeling have been developed. However, recursive PLS does not always function well when process characteristics change abruptly and JIT modeling does not always achieve the high-estimation performance. In the present work, a new method for constructing soft-sensors based on a JIT modeling technique is proposed. In the proposed method, referred to as correlation-based JIT modeling (CoJIT), the samples used for local modeling are selected on the basis of the correlation among measured variables and the model can adapt to changes in process characteristics. The usefulness of the proposed method is demonstrated through a case study of a CSTR process, in which catalyst deactivation and recovery are taken into account. In addition, its industrial application to a cracked gasoline fractionator is reported. © 2009 American Institute of Chemical Engineers *AIChE J,* 55: 1754–1765, 2009

Keywords: soft-sensor, just-in-time modeling, recursive partial least squares, principal component analysis, estimation

Introduction

A soft-sensor, or a virtual sensor, is a key technology for estimating product quality or other important variables when online analyzers are not available.[1,2] In chemical processes, for example, soft-sensors have been widely used to estimate product quality of distillation columns, reactors, and so on. Partial least squares (PLS) regression and artificial neural network (ANN) have been widely accepted as a useful technique for soft-sensor design.[3–7] In addition, the application of subspace identification (SSID) to soft-sensor design has been reported.[8,9]

Generally, building a high performance soft-sensor is very laborious, as input variables and samples for model construction have to be selected carefully and parameters have to be tuned appropriately. Even if a good soft-sensor is developed successfully, its estimation performance deteriorates when process characteristics change. In chemical processes, for example, equipment characteristics are changed by catalyst deactivation or scale adhesion. Such a situation may bring to decline the product quality. Therefore, from the practical viewpoint, maintenance of soft-sensors is very important to keep their estimation performance. Soft-sensors should be updated as the process characteristics change, and manual and repeating construction of them should be avoided due to its heavy workload.[10]

To cope with changes in process characteristics and to update statistical models automatically, recursive methods, such as recursive PLS, were developed.[11] These methods can adapt models to new operating conditions recursively. However, when a process is operated within a narrow range for a certain period of time, the model will adapt excessively and will not function in a sufficiently wide range of operating condition. In addition, recursive methods cannot cope with abrupt changes in process characteristics.

On the other hand, Just-in-Time (JIT) modeling was proposed to cope with changes in process characteristics as well as nonlinearity, and it has been used for nonlinear process monitoring as well as soft-sensing.[12–14] In JIT modeling, a

Correspondence concerning this article should be addressed to K. Fujiwara at fujiwara@cheme.kyoto-u.ac.jp local model is built from past data around a query point only when an estimated value is requested. JIT modeling is useful when global modeling does not function well. However, its estimation performance is not always high because the samples used for local modeling are selected on the basis of the distance from the query point and the correlation among variables is not taken into account. How should we determine the samples used for local modeling to build a highly accurate statistical model? Distance is not the most important. A good model cannot be developed when correlation among input–output variables is weak even if the distance between samples is very small. Conversely, a very accurate model can be developed when the correlation is strong even if the distance is large. One approach proposed is to select samples used for local modeling on the basis of not only the distance but also the angle between two samples.[15] However, the angle does not always describe the correlation among variables adequately because there are pairs of samples that are orthogonal to each other even when they exist on the same subspace.

In the present work, a new method for building soft-sensors based on JIT modeling is proposed. In the proposed method, referred to as correlation-based JIT (CoJIT) modeling, the samples used for local modeling are selected on the basis of correlation together with distance, and the Q statistic is used as an index of the correlation dissimilarity. The Q statistic is derived from principal component analysis (PCA), and it is a measure of dissimilarity between the sample and the modeling data from the viewpoint of the correlation among variables.[16] The proposed CoJIT modeling can cope with abrupt changes of process characteristics and also achieve high-estimation performance. It can also cope with process nonlinearity.

The usefulness of the proposed method is demonstrated through a case study of a CSTR process in which catalyst deactivation and recovery are investigated as the changes in process characteristics. In addition, an application result of the proposed CoJIT modeling to an industrial chemical process is reported.

Conventional Modeling Method

In this section, conventional methods for soft-sensor design are briefly explained.

Dynamic PLS

PLS has been widely used for building soft-sensors because it can cope with a colinearity problem. The ith measurements of input variables and those of output variables are denoted by $\boldsymbol{x}_i = [x_1, x_2, \ldots, x_M]^{\mathrm{T}}$ ($i = 1, 2, \ldots, N$) and $\boldsymbol{y}_i = [y_1, y_2, \ldots, y_L]^{\mathrm{T}}$, respectively, where x_m and y_l are the mth input variable and the lth output variable, respectively. $\boldsymbol{X} \in \Re^{N \times M}$ and $\boldsymbol{Y} \in \Re^{N \times L}$ are matrices whose ith rows are $\boldsymbol{x}_i^{\mathrm{T}}$ and $\boldsymbol{y}_i^{\mathrm{T}}$, respectively. It is assumed here that each column of these matrices is mean-centered and scaled appropriately. In PLS, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are decomposed as follows:

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

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^{\mathrm{T}} + \boldsymbol{F} \tag{2}$$

where $\boldsymbol{T} \in \Re^{N \times R}$ is the latent variable matrix, $\boldsymbol{P} \in \Re^{M \times R}$ and $\boldsymbol{Q} \in \Re^{L \times R}$ are the loading matrices of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. R denotes the number of adopted latent variables. $\boldsymbol{E} \in \Re^{N \times M}$ and $\boldsymbol{F} \in \Re^{N \times L}$ are the error matrices.

The estimation performance of soft-sensors can be improved by taking into account process dynamics. For this purpose, the past information is used as inputs in addition to the present information. In such a case, the input variable vector z_i is written as

$$z_i = [\boldsymbol{x}_i^{\mathrm{T}}, \boldsymbol{x}_{i-k_1}^{\mathrm{T}}, \boldsymbol{x}_{i-k_2}^{\mathrm{T}}, \cdots]^{\mathrm{T}} \tag{3}$$

where $k_1, k_2, \cdots$ are i or less arbitrary natural numbers. This method is referred to as dynamic PLS.[5,17] This extension of inputs is useful not only for PLS modeling but aslo for JIT modeling.

Recursive PLS

The estimation performance of a statistical model will deteriorate when process characteristics change. Therefore, soft-sensors should be updated as process characteristics change. However, their redesign is very laborious and it is difficult to determine when they should be updated. To cope with these problems, recursive PLS updates the model by using

$$\boldsymbol{X}_{\mathrm{new}} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{x}_{\mathrm{new}}^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{Y}_{\mathrm{new}} = \begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{y}_{\mathrm{new}}^{\mathrm{T}} \end{bmatrix} \tag{4}$$

whenever both new input and output variables, $\boldsymbol{x}_{\mathrm{new}}$ and $\boldsymbol{y}_{\mathrm{new}}$, are measured. The output variables are estimated with the updated model until the next input and output variables are measured. As the size of each matrix increases as a new sample is stored, the computational load steadily increases in this approach. However, the PLS model obtained from the above data matrices is the same as the PLS model obtained from the following data matrices[11]:

$$\boldsymbol{X}_{\mathrm{new}} = \begin{bmatrix} \boldsymbol{P}^{\mathrm{T}} \\ \boldsymbol{x}_{\mathrm{new}}^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{Y}_{\mathrm{new}} = \begin{bmatrix} \boldsymbol{Q}^{\mathrm{T}} \\ \boldsymbol{y}_{\mathrm{new}}^{\mathrm{T}} \end{bmatrix}. \tag{5}$$

By defining the input and output data matrices as Eq. (5), both the size of each matrix and the computational load can be kept constant. In addition, a forgetting factor β can be used to adapt a model to changes in process characteristics more rapidly,

$$\boldsymbol{X}_{\mathrm{new}} = \begin{bmatrix} \beta\boldsymbol{P}^{\mathrm{T}} \\ \boldsymbol{x}_{\mathrm{new}}^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{Y}_{\mathrm{new}} = \begin{bmatrix} \beta\boldsymbol{Q}^{\mathrm{T}} \\ \boldsymbol{y}_{\mathrm{new}}^{\mathrm{T}} \end{bmatrix} \tag{6}$$

where $0 < \beta \leq 1$.

Just-in-Time modeling

In general, a global linear model cannot function well when a process has strong nonlinearity in its operation range, and it is difficult to construct a nonlinear model that is applicable to a wide operation range as nonlinear modeling usually requires a huge number of samples. Therefore, a method that divides a process operation region into small multiple regions and builds a local model in each small region has been proposed. In such a method, a process is expressed by a combination of local models. A piecewise affine (PWA) model is an example of such a model.[18,19] However, in the PWA model, the optimal division of the operation region is not always clear and the interpolation between the local models is complicated.

Another method for developing local models based on a database is JIT modeling. In comparison with the conventional modeling methods, JIT modeling has the following features:

- When new input and output data are available, they are stored into a database.
- Only when estimation is required, a local model is constructed from samples located in a neighbor region around the query point, and output variables are estimated.
- The constructed local model is discarded after its use for estimation.

JIT modeling can avoid the problems of region division and interpolation between the local models occurring in PWA modeling. However, in JIT modeling, samples for local modeling should be selected appropriately and online computational load becomes large.

Correlation-Based Just-in-Time Modeling

Conventional JIT modeling uses distance to define a neighbor region around a query point and selects samples regardless of the correlation among variables. In the present work, new JIT modeling that can select samples on the basis of the correlation is proposed. In the proposed method, referred to as correlation-based JIT (CoJIT) modeling, the data set that can most correctly describe the correlation fit for the query sample is selected for local modeling.

Evaluation of correlation similarity

Although several indices of similarity between data sets have been proposed,[20,21] the Q statistic is used as an index of correlation dissimilarity in CoJIT modeling. The Q statistic is derived from PCA, which is a tool for data compression and information extraction.[16] PCA finds linear combinations of variables that describe major trends in a data set.

In PCA, the loading matrix $\boldsymbol{V}_R \in \Re^{M \times R}$ is derived as the right singular matrix of a data matrix $\boldsymbol{X} \in \Re^{N \times M}$ whose ith row is $\boldsymbol{x}_i^T$, and the column space of $\boldsymbol{V}_R$ is the subspace spanned by principal components. Here, M, N, and $R(\leq M)$ denote the numbers of variables, samples, and principal components retained in the PCA model, respectively. All variables are mean-centered and appropriately scaled. The score is a projection of $\boldsymbol{X}$ onto the subspace spanned by principal components. The score matrix $\boldsymbol{T}_R \in \Re^{N \times R}$ is given by

$$\boldsymbol{T}_R = \boldsymbol{X}\boldsymbol{V}_R. \tag{7}$$

$\boldsymbol{X}$ can be reconstructed or estimated from $\boldsymbol{T}_R$ with linear transformation $\boldsymbol{V}_R$.

$$\hat{\boldsymbol{X}} = \boldsymbol{T}_R\boldsymbol{V}_R^{\mathrm{T}} = \boldsymbol{X}\boldsymbol{V}_R\boldsymbol{V}_R^{\mathrm{T}} \tag{8}$$

The information lost by the dimensional compression, that is, errors, is written as

$$\boldsymbol{E} = \boldsymbol{X} - \hat{\boldsymbol{X}} = \boldsymbol{X}(\boldsymbol{I} - \boldsymbol{V}_R\boldsymbol{V}_R^{\mathrm{T}}). \tag{9}$$

Using the errors, the Q statistic is defined as

$$Q = \sum_{m=1}^{M} (x_m - \hat{x}_m)^2. \tag{10}$$

The Q statistic is the distance between the sample and the subspace spanned by principal components. In other words, the Q statistic is a measure of dissimilarity between the sample and the modeling data from the viewpoint of the correlation among variables.

In addition, to guarantee that the sample is located in modeling data and to avoid extrapolation, Hotelling's T^2 statistic can be used. The T^2 statistic is defined as

$$T^2 = \sum_{r=1}^{R} \frac{t_r^2}{\sigma_{t_r}^2} \tag{11}$$

where σ_{t_r} denotes the standard deviation of the rth score t_r. The T^2 statistic expresses the normalized distance from the origin in the subspace spanned by principal components. When the T^2 statistic is small, the sample is close to the mean of the modeling data. The Q and T^2 statistics can be integrated into a single index for the data set selection as proposed by Raich and Cinar for a different purpose[22]:

$$J = \lambda T^2 + (1 - \lambda)Q \tag{12}$$

where $0 \leq \lambda \leq 1$.

Correlation-based Just-in-Time modeling

In the proposed CoJIT modeling, first, samples stored in the database are divided into several data sets. Although the method of generating data sets is arbitrary, each data set is generated so that it consists of successive samples included in a certain period of time in this work, because the correlation among variables in such a data set is expected to be very similar. To build a local model, the evaluation index J in Eq. (12) is calculated for each data set, and the data set that minimizes J is selected as the modeling data set when an estimated value is required.

Figure 1 shows the difference of sample selection for local modeling between JIT modeling and CoJIT modeling. The samples consist of two groups that have different correlation. In conventional JIT modeling, samples are selected regardless of the difference of correlation as shown in Figure 1 (left), as a neighbor region around the query point is defined only by distance. On the other hand, CoJIT modeling can select samples whose correlation is best fit for the query point as shown in Figure 1 (right), as the Q statistic is used for evaluating the dissimilarity between the query point and each data set.

Assume that the first through Sth input-output measurements are stored in the database and $z_i = [\boldsymbol{x}_i^{\mathrm{T}}, \boldsymbol{y}_i^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{M+L}$ $(i = 1, 2, \cdots, S)$. To cope with process dynamics, measurements at different sampling times can be included in z_i. The procedure of CoJIT modeling is as follows:

(1) A newly measured input–output sample z_{S+1} is stored in the database.

(2) The index J is calculated from z_{S+1} and a data set $\boldsymbol{Z}^{\{S\}}$ that was used for building the previous local model $f^{\{S\}}$, and $J_I = J$.

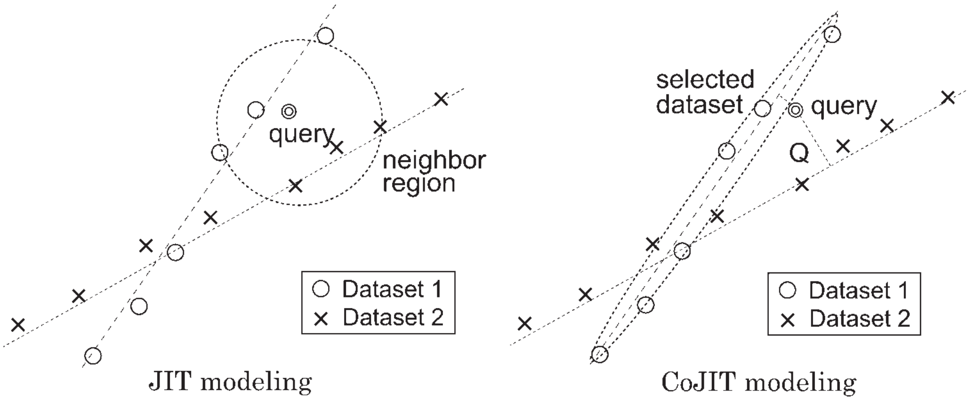

Figure 1. How to select samples in JIT modeling (left) and CoJIT modeling (right).

(3) If $J_I \leq \bar{J}_I$, then $f^{\{S+1\}} = f^{\{S\}}$ and $\mathbf{Z}^{\{S+1\}} = \mathbf{Z}^{\{S\}}$. $f^{\{S+1\}}$ is used for estimating the output variables until the next input and output measurements z_{S+2} are measured. When z_{S+2} is available, return to Step 1. If $J_I > \bar{J}_I$, $k = 1$, and go to the next step. Here, $\bar{\boldsymbol{J}}_{\boldsymbol{I}}$ is the threshold.

(4) The kth data set $\mathbf{Z}_k = [z_k, \cdots, z_{k+W-1}]^{\mathrm{T}} \in \Re^{(M+L)\times W}$ is extracted from the database, where W is the window size.

(5) The index J of the kth data set, J_k, is calculated from $\mathbf{Z}_k$ and z_{S+1}.

(6) $k = k + d$. If $k \leq S - W + 1$, then return to Step 5. If $k > S - W + 1$, then go to the next step. Here, d is the window moving width.

(7) The data set $\mathbf{Z}_K$ that minimizes J_k is selected and defined as $\mathbf{Z}^{\{S+1\}}$.

(8) A new local model $f^{\{S+1\}}$ whose input is $\boldsymbol{X}^{\{S+1\}} = [\boldsymbol{x}_K, \cdots, \boldsymbol{x}_{K+W-1}]^{\mathrm{T}}$ and output is $\boldsymbol{Y}^{\{S+1\}} = [\boldsymbol{y}_K, \cdots, \boldsymbol{y}_{K+W-1}]^{\mathrm{T}}$ is built.

(9) The updated model $f^{\{S+1\}}$ is used for estimating the output variables until the next input and output measurements z_{S+2} is available. When z_{S+2} is available, return to Step 1.

Any modeling method can be used for building a local model f, but principal component regression (PCR) is used in the proposed method because scores are already calculated in Step 5. In addition, Steps 2 and 3 control the model update frequency. When the threshold $\bar{J}_I$ is large, the update frequency becomes low.

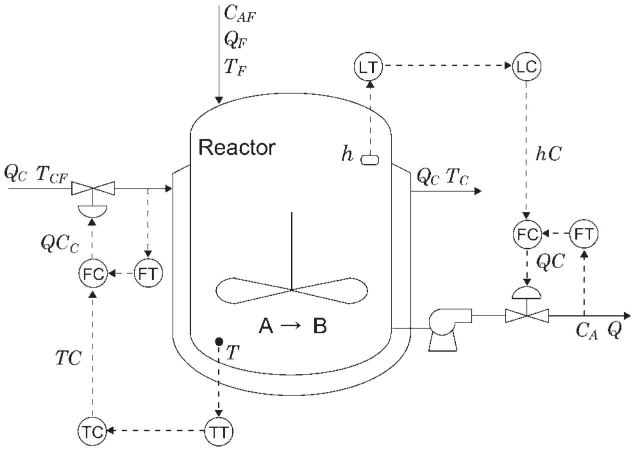

Figure 2. Schematic diagram of CSTR with cascade control systems.

In the above procedure, local models are updated only when outputs are measured. However, CoJIT modeling may select a modeling data set by using only input variables. In such a case, the model is updated whenever the input variables are measured.

In the implementation of the above algorithm, repeated use of the singular value decomposition (SVD) in Steps 2 and 5 should be avoided because SVD needs a large computational load. Therefore, the formula of a dataset stored in the database should be the loading matrix $\boldsymbol{V}_k$ insted of $\boldsymbol{Z}_k$. In addition, the number of stored data sets can be limited to reduce the computational time and the memory usage, as they steadily increase as a new sample is stored in the database. The angle θ between two subspaces can be used to determine whether the loading matrix $\boldsymbol{V}_{\mathrm{new}}$ of the newly defined data set $\boldsymbol{Z}_{\mathrm{new}} = [z_{S-W+2}, \cdots, z_{S+1}]^{\mathrm{T}}$ should be stored in the database or not. After a new local model $f^{\{S+1\}}$ is built in Step 9, the angle θ between the column space $\boldsymbol{V}_{\mathrm{new}}$ and the last loading matrix stored in the database $\boldsymbol{V}_{\mathrm{L}}$ can be calculated, and $\boldsymbol{V}_{\mathrm{new}}$ is stored in the database only when $\theta > \bar{\theta}$, where $\bar{\theta}$ is the threshold. On the other hand, when $\theta \leq \bar{\theta}$, $\boldsymbol{V}_{\mathrm{new}}$ is not stored since the column spaces of these

Table 1. Process Variables of the CSTR Process

Symbol	Variable
C_A	Reactant concentration (mol/m^3)
T	Reactor temperature (K)
T_{C}	Coolant temperature (K)
h	Reactor level (m)
Q	Reactor exit flow rate (m^3/min)
Q_{C}	Coolant flow rate (m^3/min)
Q_{F}	Reactor feed flow rate (m^3/min)
C_{AF}	Feed concentration (mol/m^3)
T_{F}	Feed temperature (K)
T_{CF}	Coolant feed temperature (K)
hC	Level controller instruction
QC	Outlet flow rate controller instruction
TC	Temperature controller instruction
QC_{C}	Colorant flow rate controller instruction
T_{set}	Reactor temperature setpoint (K)

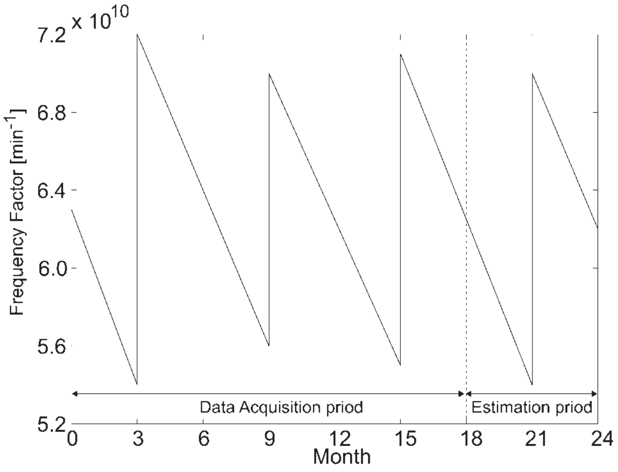

Figure 3. Change of a frequency factor.

loading matrices are similar and storing both $\boldsymbol{V}_{\mathrm{new}}$ and $\boldsymbol{V}_{\mathrm{L}}$ in the database is redundant. In addition, an old data set might be removed from the database to limit its maximum size. The size of a database should be carefully determined on the basis of process characteristics.

Case Study

In this section, the estimation performance of the proposed CoJIT modeling is compared with that of recursive PLS and conventional JIT modeling through their applications to product composition estimation for a CSTR process. The detailed CSTR model used in this case study is described in Appendix A.

Problem settings

A schematic diagram of the CSTR process with feedback control systems is shown in Figure 2. In this process, an irreversible reaction A → B takes place. The setpoint of the reactor temperature is changed between ±2 K every 10 days. Although 15 process variables listed in Table 1 are calculated in this simulation, measurements of only five variables T, h, Q, Q_{C}, Q_{F} are used for the analysis, and their sampling interval is 1 min. In addition, reactant concentration C_A is measured in a laboratory once a day. Therefore, a soft-sensor that can estimate C_A accurately in real time needs to be developed for efficient operation.

In this case study, to consider catalyst deactivation as the changes in process characteristics, the frequency factor k_0 is assumed to decrease with time. In addition, the catalyst is recovered every half year (180 days). Figure 3 shows the deterioration and recovery of the frequency factor k_0. The operation data for the past 1.5 years (540 days) were stored in the database. Although newly measured data are stored, the soft-sensor is updated in the next 180 days. A part of the operation data is shown in Figure 4.

Estimation by recursive PLS

A soft-sensor that estimates reactant concentration C_A is constructed by using recursive PLS, and it is updated every 24 h when C_A is measured. To take into account process dynamics, the input data consist of the present sample and the sample measured 1 min before. The number of latent variables used in the PLS model is determined by trial and error to maximize the prediction performance.

The estimation result is shown in Figure 5. The top figure shows the estimation result for 180 days. Although C_A is

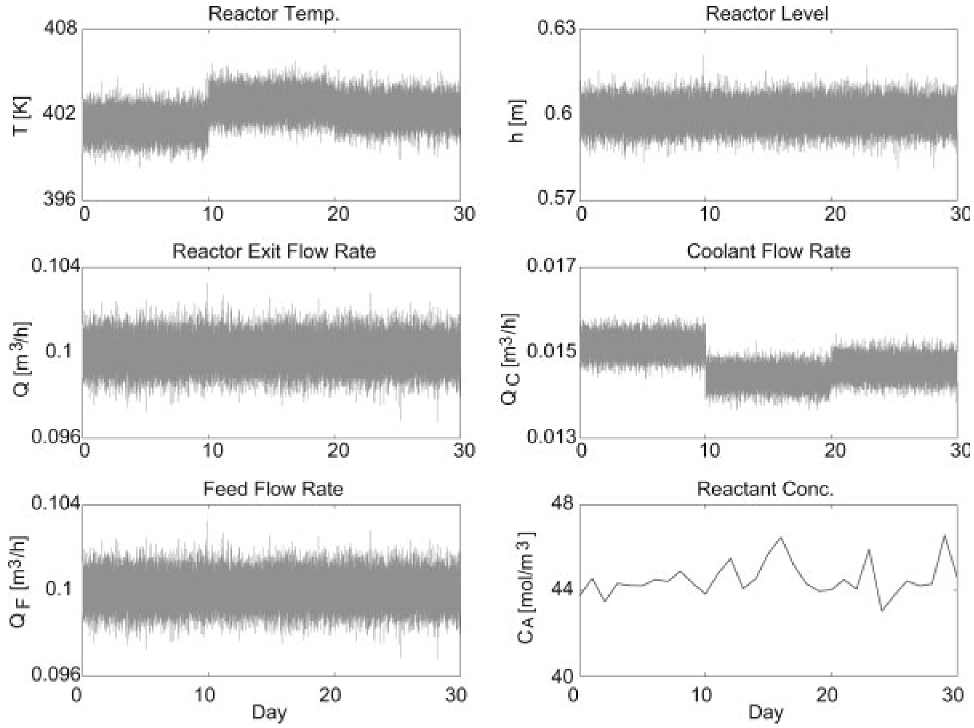

Figure 4. Operation data obtained from the CSTR process.

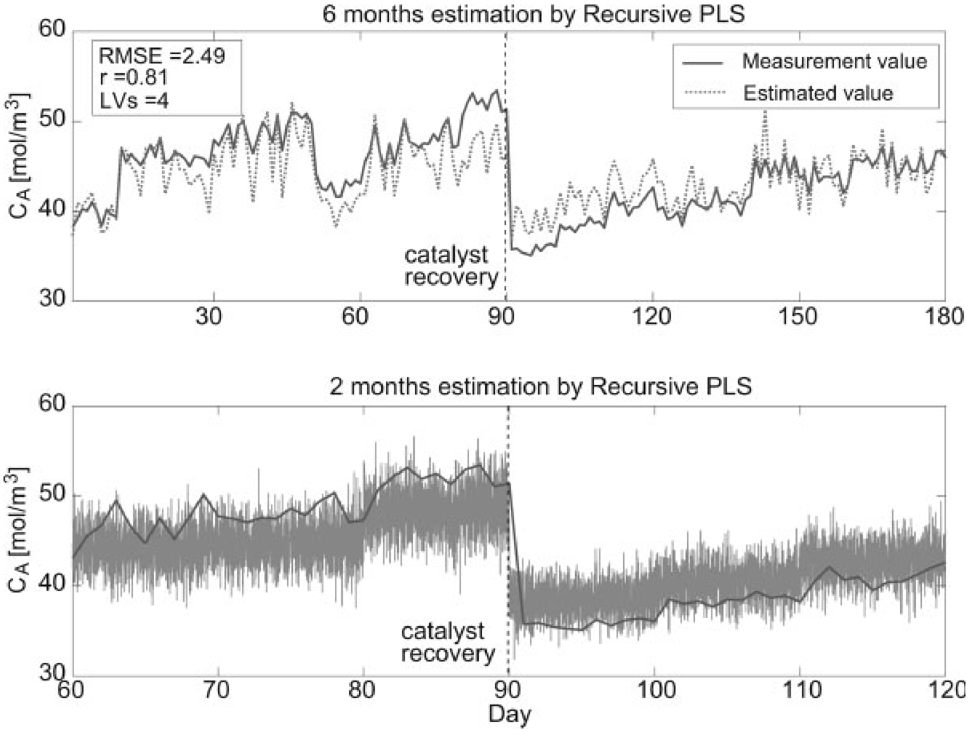

Figure 5. Prediction result of C_A by recursive PLS.

estimated every minute, only estimates corresponding to the measurements of C_A are plotted to compare the estimates with the measurements. The bottom figure shows the enlarged result for 2 months before and after the catalyst recovery. The estimates shown in the bottom figure are calculated every minute whenever the input variables are observed, and they are fluctuated by measurement noise. In this figure, r denotes the correlation coefficient between

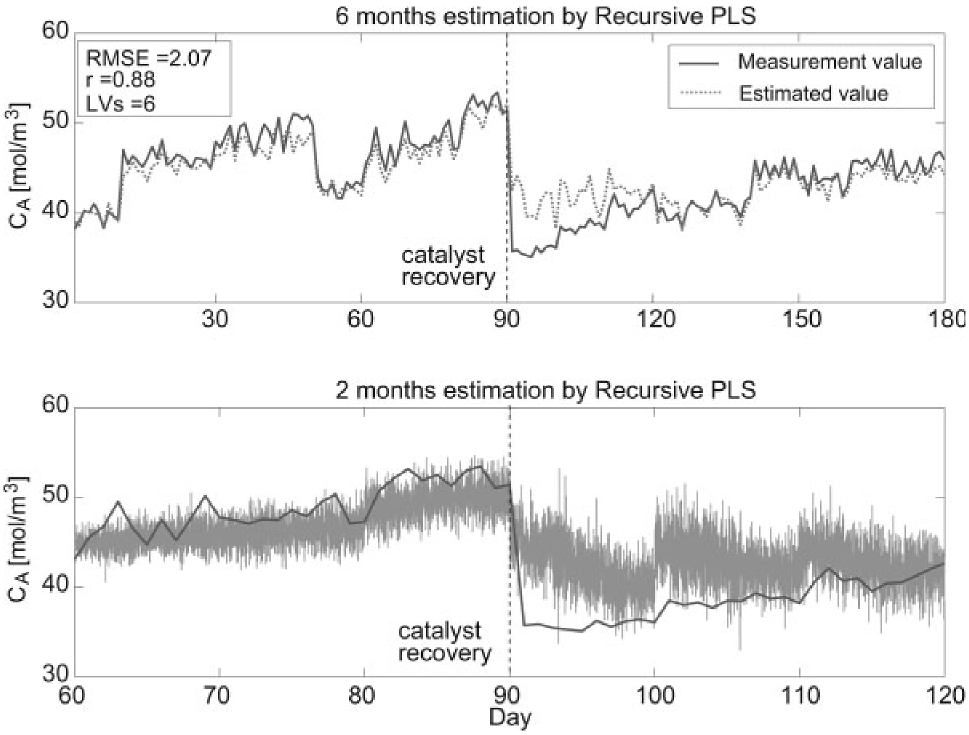

Figure 6. Prediction result of C_A by recursive PLS with forgetting factor β = 0.97.

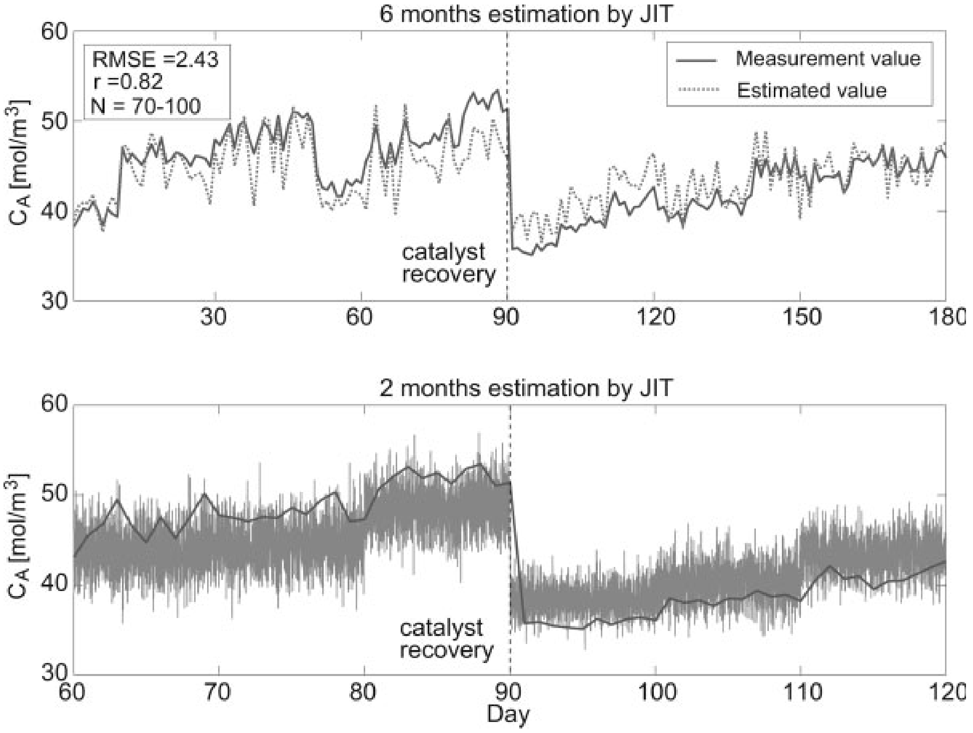

Figure 7. Prediction result of C_A by JIT modeling.

measurements and estimates, RMSE is the root-mean-squares error, and LVs is the number of latent variables.

The result shows that recursive PLS does not function well. In general, recursive PLS without the forgetting factor is not suitable for a process in which the output measurement interval is excessively long in comparison with changes in process characteristics, because this method uses all the past data for modeling. In other words, recursive PLS is suitable only for slow changes in process characteristics.

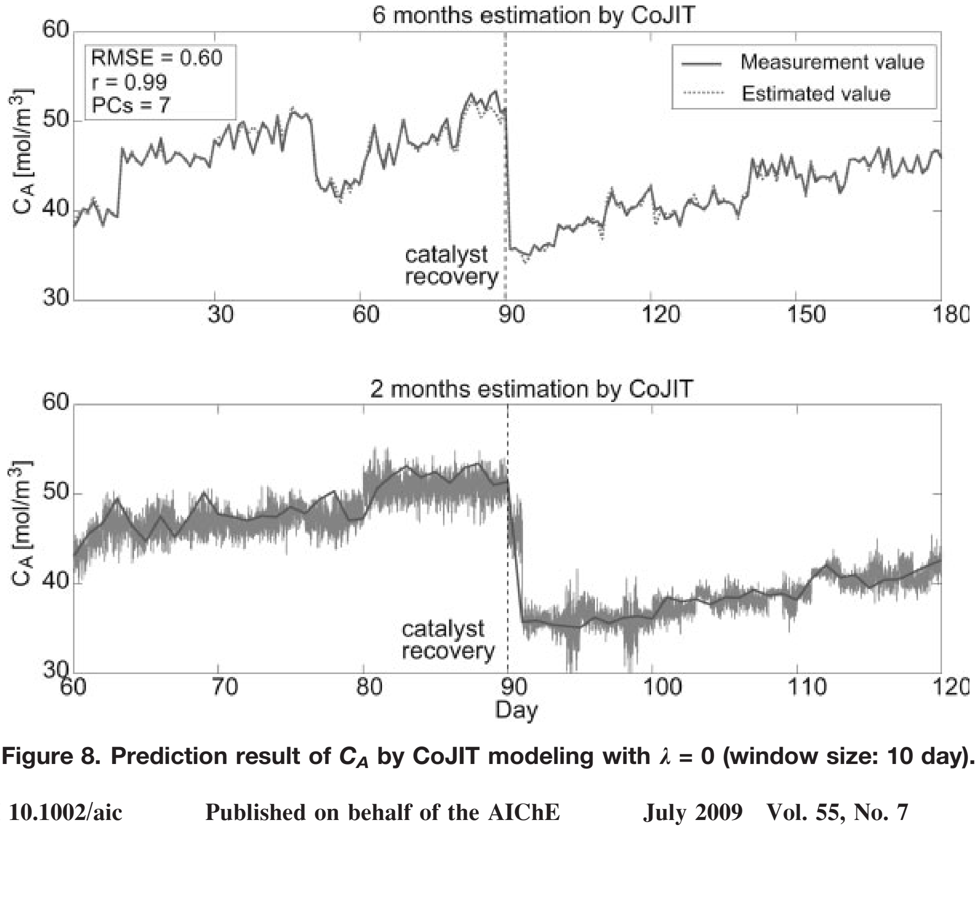

Figure 8. Prediction result of C_A by CoJIT modeling with λ = 0 (window size: 10 day).

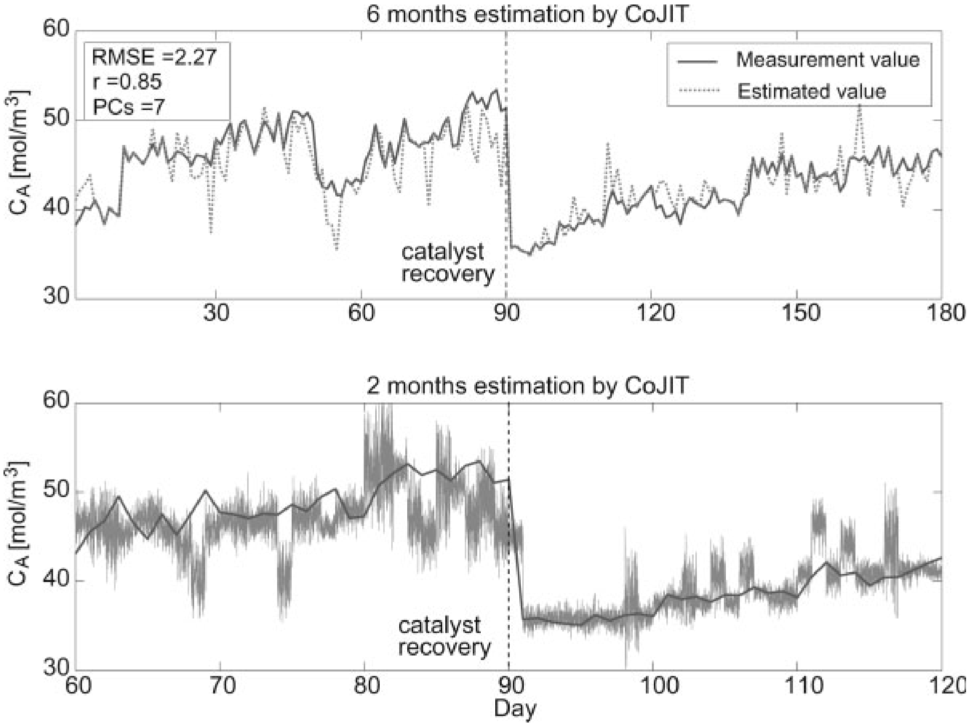

Figure 9. Prediction result of C_A by CoJIT modeling with λ = 0 (window size: 20 day).

Next, the forgetting factor β is used to adapt the model to the changes in process characteristics more rapidly. The estimation result with forgetting factor $\beta = 0.97$ is shown in Figure 6. The estimation accuracy before catalyst recovery is greatly improved by using the forgetting factor, but it deteriorates markedly after catalyst recovery. In particular, the estimation error is increased every 10 days when the setpoint of reactor temperature is changed because the model with the forgetting factor did not keep information when catalyst was recovered in the past. If β becomes less than 0.97, the estimates fluctuate more wildly, as the number of substantive samples that are used for modeling becomes very small.

Estimation by Just-in-Time modeling

Reactant concentration C_A is estimated by using JIT modeling. To take into account process dynamics, the input data consist of the present sample and the sample measured 1 min before. A local model is constructed from samples located in a neighbor region around the query point, whenever the input variables are measured. A new sample is stored only when C_A is measured. In this case study, linear local models are built and Euclidean distance is used as the measure for selecting samples to build local models. The MATLAB Lazy Learning Toolbox is used.[24]

The estimation result is shown in Figure 7. In this figure, N is the upper and lower limit of the number of samples for local modeling. This result shows that JIT modeling does not function well. The reason for the poor performance of JIT modeling seems to be that JIT modeling does not take account of correlation among variables when a local model is built. To validate this reasoning, CoJIT modeling is applied to the same problem.

Estimation by correlation-based Just-in-Time modeling

Reactant concentration C_A is estimated with the proposed CoJIT modeling. To take into account process dynamics, the input data consist of the present sample and the sample measured 1 min before. The kth data set $\boldsymbol{Z}_k$ of the CoJIT modeling is as follow:

$$\boldsymbol{Z}_k = \begin{bmatrix} \boldsymbol{x}_k^{\mathrm{T}} & \boldsymbol{x}_k^{*\mathrm{T}} & y_k \\ \boldsymbol{x}_{k+1}^{\mathrm{T}} & \boldsymbol{x}_{k+1}^{*\mathrm{T}} & y_{k+1} \\ \vdots & \vdots & \vdots \\ \boldsymbol{x}_{k+W-1}^{\mathrm{T}} & \boldsymbol{x}_{k+W-1}^{*\mathrm{T}} & y_{k+W-1} \end{bmatrix} \in \Re^{W \times 11} \qquad (13)$$

where $y_j \in \Re$ is the jth day measurement of C_A, $\boldsymbol{x}_j \in \Re^5$ is the sample that is measured simultaneously with y_j and $\boldsymbol{x}_j^* \in \Re^5$ is the sample measured one minute before $\boldsymbol{x}_j$ is measured.

The criterion for selecting a data set is to minimize the Q statistic, that is, $\lambda = 0$ in Eq. (12). The local model is updated every 24 h when C_A is measured. The window size W is 10 or 20 days, the window moving width is $d = 1$, the thresholds are $\bar{J}_I = 0$ and $\bar{\theta} = 0$ rad. The number of principal components used in CoJIT modeling is determined by trial and error.

The estimation results are shown in Figures 8 and 9, where PCs denote the number of principal components. For $W = 10$ days, these results show that the estimation performance is very high. In particular, the estimates can follow the measurements even shortly after catalyst recovery. On the other hand, the estimates fluctuate in the case of $W = 20$ days. Although the adequate W depends on the rate of change in process characteristics and the number of input variables, it is determined by trial and error mainly because it is difficult to relate the rate of change in process

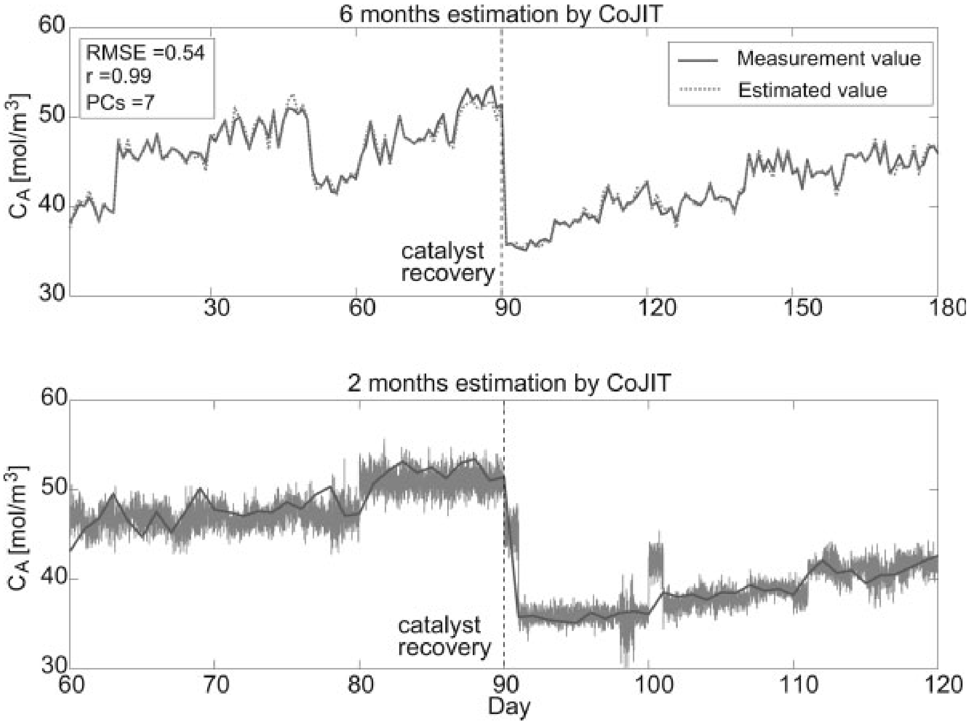

Figure 10. Prediction result of C_A by CoJIT modeling with λ = 0.01 (window size: 10 day).

characteristics with W quantitatively. In general, large W is preferable from the viewpoint of reliability of the model but it is not good from the viewpoint of adaptation of the model.

Next, a data set for local modeling is selected by using both the Q and T^2 statistics. The index J is used with $\lambda =$ 0.01. The estimation results are shown in Figures 10 and 11. These results show that the correlation coefficient r is

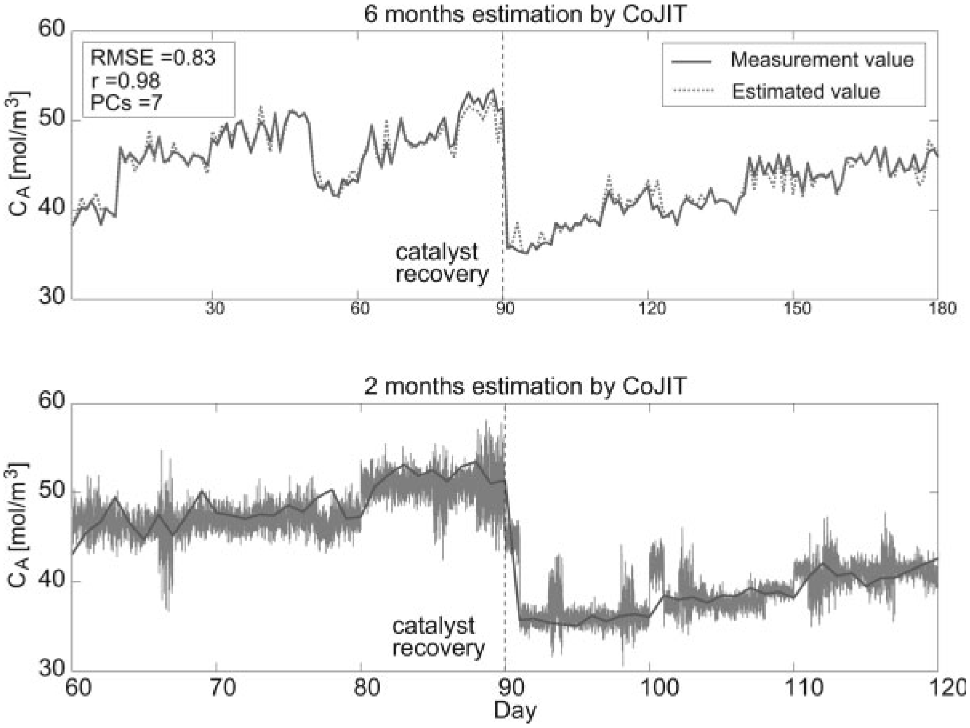

Figure 11. Prediction result of C_A by CoJIT modeling with λ = 0.01 (window size: 20 day).

improved by 15% in the case of $W = 20$ days, and RMSE is improved by 10% even in the case of $W = 10$ days. Therefore, it is beneficial for improving the estimation performance to use not only the Q statistic but also the T^2 statistic for data set selection, as it is avoidable to select the dataset that is distant from the query point.

On the other hand, the estimation performance deteriorates both in the case of $W = 10$ and 20 days when λ is larger than 0.01. For example, in the case of $W = 10$ days, RMSE with $\lambda = 0.02$ is 0.74 and RMSE with $\lambda = 0.1$ is 1.11. These results show that the estimation performance deteriorates when the contribution of the T^2 statistic is larger than necessary.

In addition, when plant operators are resistant to frequent model update, the threshold $\bar{J}_I$ can be used for controlling model update frequency. In the case of $W = 10$ days and $\lambda = 0.01$, the number of model update is reduced from 180 with $\bar{J}_I = 0$ to 66 with $\bar{J}_I = 0.15$, however, RMSE deteriorates from 0.56 with $\bar{J}_I = 0$ to 0.99 with $\bar{J}_I = 0.15$. It is obvious that there is a trade-off between model update frequency and estimation performance.

With the proposed CoJIT modeling, RMSE is improved by about 74% and 78% in comparison with recursive PLS and JIT modeling, respectively. These results of this case study clearly show that the proposed CoJIT modeling functions very successfully.

Application to an Industrial Chemical Process

In this section, an application result of the proposed CoJIT modeling to an industrial chemical process is reported. A soft-sensor for estimating the aroma concentration was constructed to realize highly efficient operation of the cracked gasoline fractionator of the ethylene production process at the Showa Denko K.K. (SDK) Oita plant in Japan.

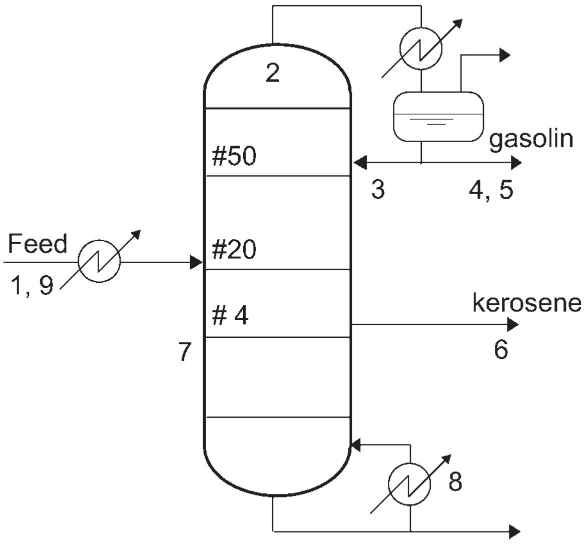

Figure 12. Schematic diagram of the cracked gasoline fractionator of the ethylene production process at the Showa Denko K.K. (SDK) Oita plant.

Table 2. Input Variables of the Soft-Sensor for the CGL Fractionator

No.	Variable
1	Feed flow rate
2	Tower top temperature
3	Reflux volume
4	Outlet cracked gasoline temperature
5	Outlet cracked gasoline flow rate
6	Outlet cracked kerosene flow rate
7	Tray #4 differential pressure
8	Reboiler flow rate
9	Cracked furnace coil outlet temperature

CGL fractionator

A schematic diagram of the cracked gasoline (CGL) fractionator of the ethylene production process is shown in Figure 12. The CGL fractionator is controlled by applying multivariable model predictive control (MPC) with an optimizer, and the aroma concentration in the CGL (aroma denotes the generic name for benzene, toluene, xylene, styrene, etc.) is used as one of the constraints in the optimizer. Although the operation data of the CGL fractionator are stored in the database every hour, the aroma concentration is analyzed in a laboratory usually once a day because of its long analysis time. For safety, the process must be operated in a condition that has a wide margin and is far from constraints. Therefore, a soft-sensor that can estimate the aroma concentration accurately in real time needs to be developed for realizing efficient operation.

Operation data

Although the number of measured variables in the CGL fractionator is 19, only eight variables were selected as the input variables of the soft-sensor on the basis of the physical process knowledge. The past information of the selected input variables is not used as the input variables of the soft-sensor because of its long measurement interval. In addition, the coil outlet temperature of the cracking furnace measured 4 h before was used together with the selected input variable, as the product composition is affected by the operating condition of the cracking furnace which is located in the upstream of the ethylene production process, and it takes about 4 h for materials to reach the CGL fractionator from the cracking furnace. Therefore, the number of input variables is nine. The selected input variables of the soft-sensor are listed in Table 2 and Figure 12. In the initial state, the operation data obtained from April 30, 2006, to February 23, 2007, were stored in the database. Then, the soft-sensor was updated and the aroma concentration was estimated for the next 300 days, February 24, 2007, to December 25, 2007.

Estimation results

The aroma concentration was estimated with recursive PLS. The number of latent variables used in recursive PLS was two, and the forgetting factor was $\beta = 0$. The model was updated every 24 h when the aroma concentration was analyzed in the laboratory. The estimation results are shown in Figure 13 (top). Although the aroma concentration is estimated every hour, the estimated values are shown every day

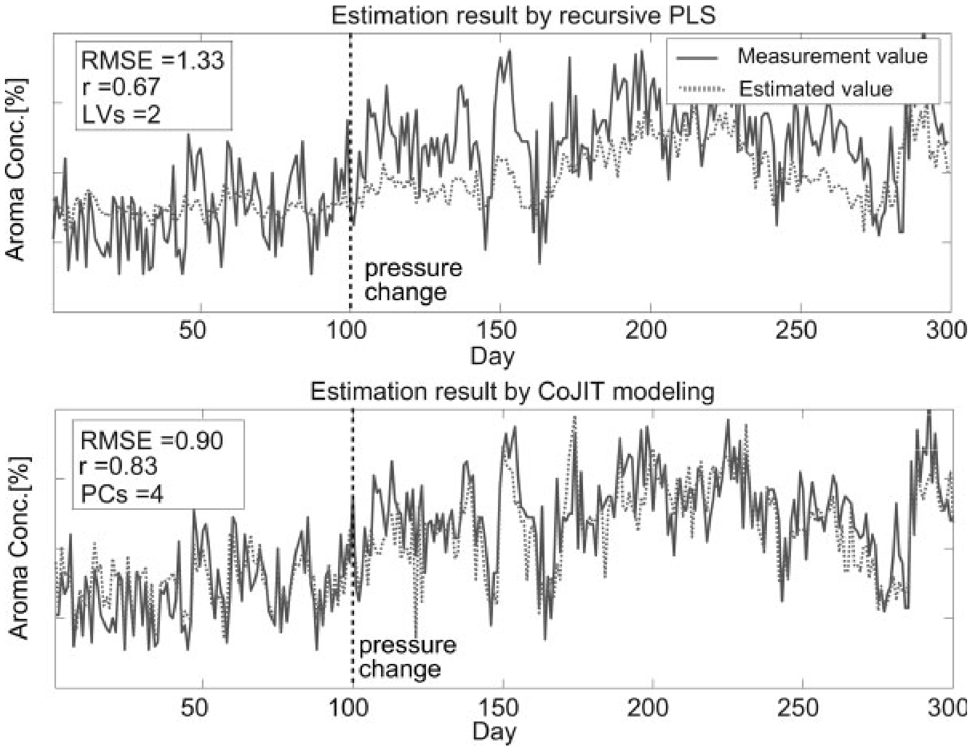

Figure 13. Prediction results of aroma concentration: recursive PLS (top) and CoJIT modeling (bottom).

to compare the estimates with the analysis value. This result shows that recursive PLS does not function well. In particular, there is a bias between the measurements and the estimates after the 100th day when the pressure of the compressor was changed. In addition, the estimation performance was not improved even when the forgetting factor was used.

Next, the aroma concentration was estimated with the proposed CoJIT modeling. The window size was $W = 20$ days, window moving width was $d = 1$ day, the thresholds were $\bar{J}_I = 0$ and $\bar{\theta} = 0$ rad, and the parameter in Eq. (12) was $\lambda = 0.25$. The estimation results are shown in Figure 13 (bottom). This result shows that the estimation performance of CoJIT modeling is high and RMSE is improved by about 28% in comparison with recursive PLS.

In CoJIT modeling, the average computational time for updating the model was 48 ms. Then, the number of the data sets stored in the database was limited to reduce the computational time. The thresholds were $\bar{\theta} = 0.1$, 0.3, and 0.5 rad. The computational results are shown in Table 3. The computer configuration was as follows: OS: Windows Vista Business (32 bit), CPU: Intel Core2 Duo 6300 (1.86 GHz × 2), RAM: 2G byte, and MATLAB 2007b is used. From these results, the computational time is greatly reduced when the threshold $\bar{\theta}$ becomes large. Although the computational time in the case of $\bar{\theta} = 0.5$ is reduced by about 53% in comparison with the case of $\bar{\theta} = 0$, RMSE deteriorates only 4%.

Table 3. Mean CPU Time and Estimation Performance

θ (rad)	CPU Time (ms)	r	RMSE (%)	Number of Stored Dataset
0	47.6	0.83	0.90	600
0.1	47.0	0.83	0.91	583
0.3	38.2	0.82	0.93	385
0.5	22.0	0.81	0.94	248

Conclusion

In the present work, to develop a soft-sensor that can cope with changes in process characteristics as well as nonlinearity, a new JIT modeling method that selects samples on the basis of correlation among variables is proposed. The usefulness of the proposed correlation-based JIT (CoJIT) modeling is demonstrated through a case study of a CSTR process and its application to an industrial chemical process. In recursive PLS and conventional JIT modeling, it is difficult to adapt models when the process characteristics change abruptly. On the other hand, the proposed CoJIT modeling can cope with abrupt changes in process characteristics and greatly improve the estimation performance, as it can select samples for local modeling by appropriately accounting for the correlation among variables. The proposed CoJIT modeling has a potential for realizing efficient maintenance of soft-sensors in the real world.

Literature Cited

1. Kano M, Nakagawa Y. Data-based process monitoring, process control, and quality improvement: recent developments and applications in steel industry. *Comput Chem Eng*. 2008;32:12–24.
2. Ohshima M, Tanigaki M. Quality control of polymer production processes. *J Proc Cont*. 2000;10:135–148.
3. Mejdell T, Skogestad S. Estimation of distillation compositions from multiple temperature measurements using partial-least-squares regression. *Ind Eng Chem Res*. 1991;30:2543–2555.
4. Kresta JV, Marlin TE, MacGregor JF. Development of inferential process models using PLS. *Comput Chem Eng*. 1994;18:597–611.

5. Kano M, Miyazaki K, Hasebe S, Hashimoto I. Inferential control system of distillation compositions using dynamic partial least squares regression. *J Proc Cont*. 2000;10:157–166.
6. Kamohara H, Takinami A, Takeda M, Kano M, Hasebe S, Hashimoto I. Product quality estimation and operating condition monitoring for industrial ethylene fractionator. *J Chem Eng Jpn*. 2004; 37:422–428.
7. Radhakrishnan V, Mohamed A. Neural networks for the identification and control of blast furnace hot metal quality. *J Proc Cont*. 2000;10:509–524.
8. Amirthalingam R, Lee J. Subspace identification based inferential control applied to a continuous pulp digester. *J Proc Cont*. 1999;9: 397–406.
9. Kano M, Lee S, Hasebe S. Two-stage subspace identification for softsensor design and disturbance estimation. *J Proc Cont*. 2009; 19:179–186.
10. Ogawa M, Kano M. Practice and challenges in chemical process control applications in Japan. *The 17th IFAC World Congress*, 2008; Paper WeC25.3.
11. Qin SJ. Recursive PLS algorithms for adaptive data modeling. *Comput Chem Eng*. 1998;22:503–514.
12. Bontempi G, Birattari M, Bersini H. Lazy learing for local modelling and control design. *Int J Cont*. 1999;72:643–658.
13. Atkeson CG, Moore AW, Schaal S. Locally weighted learning. *Artif Intel Rev*. 1997;11:11–73.
14. Cheng C, Chiu MS. Nonlinear process monitoring using JITL-PCA. *Chem Int Lab Sys*. 2005;76:1–13.
15. Cheng C, Chiu MS. A new data-based methodology for nonlinear process modeling. *Chem Eng Sic*. 2004;59:2801–2810.
16. Jackson JE, Mudholkar GS. Control procedures for residuals associated with principal component analysis. *Technometrics*. 1979;21: 341–349.
17. Ricker NL. Use of biased least-squares estimators for parameters in discrete-time pulse response models. *Ind Eng Chem Res*. 1998; 27:343–350.
18. Ferrari-Trecate G, Muselli M, Liberati D, Morari M. A clustering technique for the identification of piecewise affine system. *Automatica*. 2003;39:205–217.
19. Sakamoto M, Dong D, Hamaguchi T, Ota Y, Itoh T, Hashimoto Y. Nonlinear systems approximation using a piecewise affine model based on a radial basis functions network. *J Chem Eng Jpn*. 2006; 39:1078–1084.
20. Kano M, Ohno H, Hasebe S, Hashimoto I. A new multivariate statistical process monitoring method using principal component analysis. *Comput Chem Engng*. 2001:25;1103–1113.
21. Kano M, Hasebe S, Hashimoto I, Ohno H. Statistical process monitoring based on dissimilarity of process data. *AIChE J*. 2002;48: 1231–1240.
22. Raich A, Cinar A. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE J*. 1994;42: 995–1009.
23. Johannesmeyer M, Seborg DE. Abnormal situation analysis using pattern recognition techniques and historical data. *AIChE Annual Meeting*, 1995.
24. Bontempi G, Birattari M, Bersini H. Lazy learners at work: The lazy learing toolbook. EUFIT'99: The 7th European Congress on Intelligent Techniques and Soft Computing. 1999

Appendix A: CSTR Model

The CSTR model used in the case study is described.[23] The mass, energy, and component balances around the reactor and the cooling jacket are as follows:

$$\frac{dC_A}{dt} = -k_0 e^{-E/RT} C_A + \frac{Q_F C_{\mathrm{AF}} - Q C_A}{Ah} \tag{A1}$$

$$\frac{dT}{dt} = \frac{k_0 e^{-E/RT} C_A(-\Delta H)}{\rho C_p} + \frac{Q_F T_F - QT}{Ah} + \frac{UA_C(T_C - T)}{\rho C_P A H} \tag{A2}$$

$$\frac{dT_{\mathrm{C}}}{dt} = \frac{Q_{\mathrm{C}}(T_{\mathrm{CF}} - T_{\mathrm{C}})}{V_{\mathrm{C}}} + \frac{UA_{\mathrm{C}}(T - T_{\mathrm{C}})}{\rho_{\mathrm{C}} C_{\mathrm{PC}} V_{\mathrm{C}}} \tag{A3}$$

$$\frac{dh}{dt} = \frac{Q_{\mathrm{F}} - Q}{A} \tag{A4}$$

The dynamics of control valves for coolant inlet and product outlet are modeled as Eq. (A5), and PI controllers are used.

$$P_V(s) = \frac{K}{\tau s + 1} \tag{A5}$$

where $K = 1/16$, $\tau = 2$ s. The nominal operating conditions and model parameters are given in Table A1.

Table A1. Nominal Operating Conditions and Model Parameters for the CSTR Case Study

$Q = 0.1$ m^3/min	$A = 0.1666$ m^2
$Q_{\mathrm{C}} = 1.5 \times 10^{-2}$ m^3/min	$k_0 = 7.2 \times 10^{10}$ min^{-1}
$T_{\mathrm{CF}} = 300$ K	$\Delta H = -5 \times 10^4$ J/mol
$T = 402.35$ K	$\rho C_P = 2.39 \times 10^5$ J/(m^3 K)
$T_{\mathrm{C}} = 345.44$ K	$\rho_{\mathrm{C}} C_{\mathrm{PC}} = 4.175$ times 10^6 J/(m^3 K)
$T_{\mathrm{C}} = 320$ K	$E/R = 8.75 \times 10^3$ K
$C_{\mathrm{AF}} = 1.0 \times 10^3$ mol/m^3	$U A_{\mathrm{C}} = 5.0 \times 10^4$ J/(min K)
$C_{\mathrm{A}} = 37$ mol/m^3	$V_c = 1.0 \times 10^3$ m^3
$h = 0.6$ m	

Manuscript received Apr. 17, 2008, and revision received Nov. 4, 2008.