Developing Soft Sensors Based on Orthogonal Projections to Latent Structures with Kernel Algorithm

Jialin Liu*, David Shan Hill Wong**

*Department of Information Management, Fortune Institute of Technology, 1-10, Nwongchang Rd., Neighborhood 28, Lyouciyo Village, Daliao Township, Kaohsiung, Taiwan, Republic of China
(Tel: 886-7-7889888 ext. 6122; e-mail: jialin@center.foitech.edu.tw).

**Department of Chemical Engineering, National Tsing Hua University, 101, Section 2, Guang Fu Road, Hsinchu 30043, Taiwan, Republic of China (e-mail: dshwong@che.nthu.edu.tw)

Abstract: Considering the time-varying nature of an industrial process, a soft sensor based on fast moving window algorithm was developed. The proposed approach adapted the parameters of the inferential model with the dissimilarities between the new and oldest data and incorporated them into the proposed kernel algorithm for the orthogonal projections to latent structures (OPLS). The computational loading of the model adaptation was therefore independent on the window size. Since the non-correlated systematic variation in the predictor variables is removed by the OPLS method, it reduces the complexity of the prediction model, furthermore, clarifying the correlation between the predictor and the response variables. A simulated example of a continuous stirred tank reactor (CSTR) with feedback control systems illustrated that the process characteristics captured by the OPLS could be adapted to accommodate a nonlinear process.

Keywords: Soft sensor, orthogonal projections to latent structures, kernel algorithm for PLS, fast moving window algorithm.

1. INTRODUCTION

In industrial processes, operators adjust manipulated variables to maintain product qualities or exhaust gases within the specifications of the product or government regulations, according to online analyzers and laboratory tests. Due to malfunctions of the online analyzers or significant delays during laboratory testing, soft sensors inferring the primary output from other process variables provide useful information for regulating the process operation. Soft sensor applications have attracted significant attention in the process industry (Kadlec et al., 2009). The projections to latent structures or partial least squares (PLS) algorithm is a popular multivariate statistical tool to model input/output data. It has been proven that the maximal covariance between two datasets can be captured by PLS (Höskuldsson, 1988). Since industrial processes are time-varying, the PLS models need to be adapted to accommodate this behavior. Qin (1998) proposed a block-wise recursive PLS (RPLS) for adapting the inferential model. Although RPLS accounts for the time-varying nature of processes by updating models with the newest data, it leads to a reduction in the speed of adaptation as the data size increases. Dayal and MacGregor (1997) incorporated a variable forgetting factor into the recursive exponentially weighted PLS to discount the old data; however, the factor is difficult to determine without the process knowledge. The moving window algorithm is an alternative approach to exclude the oldest data when new data are available. Qin (1998) reported the computational loading of the moving window PLS is proportional to the window size. Wang et al. (2005) proposed a fast moving window algorithm to adapt the PCA model for monitoring processes with a time-varying nature. In their approach, the computational loading is independent on the window size, which is more practicable for online updating models. In this paper, the fast moving window algorithm was enhanced and applied to the OPLS method for modeling the time variant process.

Since the standard PLS requires many components or latent variables (LVs), which contain data variation of the predictor variables uncorrelated to the response variables, it leads the complexity of the prediction models and the difficulty of interpreting the models. Wold et al. (1998) proposed the orthogonal signal correction (OSC) method to remove systematic variation of the predictor variables that is uncorrelated to the response variables. However, the OSC method contains an internal time-consuming iteration to find orthogonal components. In addition, the OSC method cannot give a unique solution, which depends on the initial vector. In order to conquer the drawbacks of the OSC method, Trygg and Wold (2002) proposed the orthogonal projections to latent structures (OPLS) algorithm to achieve the same objective of the OSC method but with the more robust properties. They modified the non-linear iterative partial least squares (NIPALS) algorithm to remove non-correlated variation from the predictor variables. After removing all OPLS components, the PLS component is extracted based on the retained information of the predictor variables and the variation of the response variables. Therefore, it is easier to interpret the correlation between input and output variables. The NIPALS method is the common algorithm to estimate the parameters of the PLS model. When the datasets contain a massive amount of observations, it becomes inefficient due...
to iterating the score vectors and deflating the data matrices. Lindgren et al. (1993) proposed a kernel algorithm for PLS using the covariance of the input and cross covariance of the input and output data to estimate the parameters of PLS. In their approach, the number of data affects the computational loading only when calculating the covariance and cross covariance, which are calculated once at the beginning of the algorithm. It is suitable to derive a fast moving window algorithm for OPLS using the kernel algorithm because the computational loading is independent on the window size. In this paper, a one-step fast moving window algorithm was provided to update the OPLS model with the dissimilarities of the new and the oldest data. It is more efficient than the work of Wang et al. (2005). In their approach, when the new data are available, the mean of each variable and the covariance matrix were adapted twice using the recursive adaptation, which removes the oldest data from the model and then adds the new data into it. The remainder of this paper is organized as follows. Section 2 gives a preliminary of OPLS and the kernel algorithm for PLS. The proposed approach of the fast moving window OPLS (FMWOPLS) algorithm is detailed in section 3. In section 4, a simulated example is given. The simulated example, which is a nonisothermal continuous stirred tank reactor (CSTR) with feedback control systems, illustrates how the OPLS model has the capability to capture process behaviors coming from the short-term effect of disturbances, whereas the model is adapted to accommodate process variations due to the long-term effect of disturbances by the proposed approach. Finally, conclusions are given.

2. BASIC THEORY

2.1 Orthogonal projections to latent structures

Trygg and Wold (2002) modified the original NIPALS algorithm to remove the systematic variation of the predictor variables in the dataset that is uncorrelated with the response variable, named orthogonal projections to latent structures (OPLS). Assuming a linear correlation between a single output and multiple inputs in a dataset, it is always possible to find a one-component PLS model to depict the linear relation after preprocessing the dataset properly. Verron et al. (2004) proved that the number of PLS components in the OPLS-pretreated PLS model can be reduced to a single PLS component model without degrading the prediction performance. Consider the data matrix $X \in R^{m \times n}$ with $m$ rows of observations and $n$ columns of predictor variables and a response vector $y$. Each variable is normalized to zero mean and unit variance. The OPLS algorithm is listed in Table 1 (Trygg and Wold, 2002).

In Table 1, the steps 1-3 and 8 are the original NIPALS algorithm for PLS, and that the steps 4-6 remove the orthogonal components. The orthogonal components ($t_{ortho}$) are uncorrelated to the response vector $y$.

$$t_{ortho}^T y = w_{ortho}^T X y = \left[ p^T - (w^T p) w^T \right] X^T y$$

Since $X^T y = w (y^T y)$, above equation can be rewritten as:

$$t_{ortho}^T y = \left[ p^T w - (w^T p) w^T \right] (y^T y) = 0$$

in which $w^T w = 1$ and $w^T p = p^T w$ are used. Therefore, the removed components are orthogonal to the response vector $y$. Once new inputs ($x_{new}$) are available, each orthogonal component is extracted repeatedly as follows:

$$t_{ortho} = x_{new}^T w_{ortho} \cdot x_{new}^T = x_{new}^T - t_{ortho} p_{ortho}$$

After removing all orthogonal components, the response value ($\hat{y}$) can be predicted using $\hat{y} = x_{new}^T b$.

**Table 1. OPLS algorithm for single output model**

1. $w^T = y^T X / (y^T y)$, $w = w / \|w\|$. $t = Xw$, $c = t^T y / (t^T t)$, $u = yc / c^2$.
2. $p^T = t^T X / (t^T t)$. $w_{ortho} = p - (w^T p) w$, $w_{ortho} = w_{ortho} / \|w_{ortho}\|$. $t_{ortho} = Xw_{ortho}$, $p_{ortho} = t_{ortho}^T X / (t_{ortho}^T t_{ortho})$.
3. Save found parameters $T = [t_{ortho}^T]$ , $W_{ortho} = [W_{ortho}^T]$.
4. For additional orthogonal components, return to step 2 and set $X = E_{PLS}$.
5. Perform steps 1-3, after extracting orthogonal components.
6. $q = y^T u / u^T u$, $b = qw$.

2.2 Kernel algorithm for partial least squares

PLS regression is a popular statistical tool for modeling the predictor and response datasets. A set of latent variables is extracted iteratively to describe the predictor ($X$) and response ($Y$) data matrices.

$$X = T_k P_k + E, Y = T_k Q_k + F$$

where $T_k$ is the first $k$ terms of the latent variables or the score vectors, $P_k$ and $Q_k$ respectively are the loading vectors of the data matrices $X$ and $Y$, $E$ and $F$ are the residual terms of PLS. In general, each score vector is extracted through deflating $X$ and $Y$ by the NIPALS algorithm until all variance in the data structure is explained. It is a time-consuming procedure when the data matrices contain a massive amount of data. However, the score vectors are not necessary for a regression model as the following equation:

$$Y = X B_{PLS} + F$$

where $B_{PLS}$ is the matrix of the regression coefficients. Lindgren et al. (1993) proposed a kernel algorithm for PLS.

They used the covariance of $X$, $\Sigma_X = (X^T X) / (m - 1)$, and the cross covariance of $X$ and $Y$, $\Sigma_{XY} = (X^T Y) / (m - 1)$, to evaluate the regression coefficients of PLS. The algorithm is listed as follows:

1. Set $a = 1$, $(\Sigma_{XY})_a = \Sigma_{XY}$, $(\Sigma_X)_a = \Sigma_X$ and $H_a = I$.
2. Perform singular value decomposition (SVD) on the $(\Sigma_X \Sigma_{XY}^T)_a$. The eigenvector of the largest eigenvalue is the weighting vector $w_a$ and an auxiliary vector is defined as $r_a = H_a w_a$.
The loading vectors of the \( p_a \) and \( q_a \) can be obtained using
\[
\begin{align*}
\Sigma_X Xp_a &= w_a, \\
\Sigma_X Xq_a &= w_a, \\
\end{align*}
\]
and
\[
\begin{align*}
p_a^T &= \left( \Sigma_X Xp_a \right)^T, \\
q_a^T &= \left( \Sigma_X Xq_a \right)^T, \\
\end{align*}
\]
4. The deflation procedure is conducted as:
\[
\left( \Sigma_X X \right)_{a+1} = \left( I - w_a w_a^T \right)^T \left( \Sigma_X X \right)_a \\
\left( \Sigma_X X \right)_{a+1} = \left( I - w_a w_a^T \right)^T \left( \Sigma_X X \right)_a \left( I - w_a w_a^T \right). \\
\begin{align*}
H_{a+1} &= H_a - w_a p_a^T, \quad \text{for the next iteration.}
\end{align*}
\]
5. Set \( a = a + 1 \) go to step 2, until all data structures of covariance and cross covariance are extracted.
6. The matrix of the regression coefficients can be obtained using \( B_{PLS} = RQ^T \), where \( R = [r_1 \ r_2 \ \ldots \ r_q] \) and \( Q = [q_1 \ q_2 \ \ldots \ q_q] \).

For a dataset with massive amount of observations, the most time-consuming parts of the NIPALS algorithm respectively are to iterate the score vector, which projects the data matrix onto the weighting vector \( w \), and to conduct the deflation procedure, which multiplies the score vector by the loading vectors. In the kernel algorithm, the score vectors are not necessary and the deflation procedure is conducted on the matrices of covariance and cross covariance. Therefore, the most time-consuming part is to estimate the covariance and the cross covariance of datasets, which is only performed once in the algorithm. When there is a single output variable, the SVD procedure of step 2 can be replaced by the following equation.
\[
w_a^T = (\Sigma_X X)^T / \Sigma_X X \\
\]
Therefore, the algorithm is further simplified for this special case.

3. PROPOSED APPROACH

In this paper, the kernel algorithm for PLS has been extended to OPLS. Since the original OPLS algorithm is based on the NIPALS algorithm, it suffers the difficulty that the computational loading is proportional to the amount of observations in the dataset. The kernel algorithm for OPLS is listed in Table 2. In the table, the steps 1-2 and 7 are the original kernel algorithm, and the steps 3-5 are the proposed algorithm to remove the orthogonal components. The kernel algorithm for multiple output OPLS is given in Appendix. Given a dataset with \( m \) measurements, in which the numbers of the predictor and response variables respectively are \( n \) and \( l \), the input and output matrices are \( W \in R^{m \times n} \) and \( Z \in R^{m \times l} \). The mean and the standard deviation of each variable are as follows.
\[
\begin{align*}
\bar{w} &= \frac{1}{m} \sum_{i=1}^{m} w_i, \\
\bar{z} &= \frac{1}{m} \sum_{i=1}^{m} z_i, \\
\Sigma_X &= \text{diag} \left( s_{x_1} \ldots s_{x_n} \right), \Sigma_Y &= \text{diag} \left( s_{y_1} \ldots s_{y_l} \right) \\
\end{align*}
\]
where \( w_i \) and \( z_i \) are the \( i \)th element of the mean of the variable; \( s_{x_i} \) and \( s_{y_i} \) are the \( i \)th diagonal element for the standard deviation of the variable. The covariance of the input matrix \( \Sigma_X \) and the cross covariance of the input and output matrices \( \Sigma_{XY} \) can be derived from the above equations.
\[
\begin{align*}
\Sigma_X &= \frac{1}{m-1} \sum_{i=1}^{m} \left( W^T W - m \bar{w} \bar{w}^T \right) S_X^{-1} \\
\Sigma_{XY} &= \frac{1}{m-1} \sum_{i=1}^{m} \left( W^T Z - m \bar{w} \bar{z}^T \right) S_Y^{-1} \\
\end{align*}
\]
Once the new observations are available and the oldest ones are discarded, the adaptive means and standard deviations can be written as:
\[
\begin{align*}
\bar{w} &= \frac{1}{m} \sum_{i=1}^{m} (w_{m+1} - w_1), \\
\bar{z} &= \frac{1}{m} \sum_{i=1}^{m} (z_{m+1} - z_1), \\
\end{align*}
\]
\[
\begin{align*}
\Sigma_{XX} &= \frac{1}{m-1} \sum_{i=1}^{m} \left( (w_{m+1} - w_1)^2 - m (w_i - \bar{w})^2 \right) \\
\Sigma_{XY} &= \frac{1}{m-1} \sum_{i=1}^{m} \left( (z_{m+1} - z_1)^2 - m (z_i - \bar{z})^2 \right) \\
\end{align*}
\]
where the notations with a superscript asterisk are the adaptive quantities and the subscripts \( m+1 \) and \( 1 \) stand for the new and the oldest data. It can be observed that the means and standard deviations are adapted based on the original quantities and the dissimilarities between the new and the oldest data. Similarly, the adaptive covariance and cross covariance can be derived based on the same concept.
\[
\begin{align*}
\Sigma_{XX} &= \left( \frac{1}{m-1} \sum_{i=1}^{m} \left( (w_{m+1} - w_1)^2 - m (w_i - \bar{w})^2 \right) \right)^{-1} \\
\Sigma_{XY} &= \left( \frac{1}{m-1} \sum_{i=1}^{m} \left( (z_{m+1} - z_1)^2 - m (z_i - \bar{z})^2 \right) \right)^{-1} \\
\end{align*}
\]
The OPLS model is updated by the kernel algorithm and the adaptive covariance and cross covariance. Since the computational loading for model updating is independent on the window size, the fast moving window OPLS (FMWOPLS) algorithm is given.

Table 2. Kernel algorithm for single output OPLS model

1. Prepare the covariance and cross covariance, \( \Sigma_X = \left( X^T X \right) / (m-1) \) and \( \Sigma_{XY} = \left( X^T Y \right) / (m-1) \).
2. \( w^T = \Sigma_{XY} / \Sigma_{XX} \), \( p^T = w^T \Sigma_X / (\Sigma_{XY} w) \).
3. \( w_{ortho} = p - (w p^T) w, \quad w_{ortho} = w_{ortho} / \| w_{ortho} \| \).
4. \( p_{ortho}^T = w_{ortho}^T \Sigma_X / (w_{ortho}^T \Sigma_X w_{ortho}) \).
5. Save found parameters \( P_{ortho} = [P_{ortho} \ p_{ortho}] \).
6. Perform step 2, after extracting orthogonal components.
7. \( q = w^T \Sigma_X w / (w^T \Sigma_X w) \), \( b = q w \).

4. ILLUSTRATIVE EXAMPLE
A nonisothermal continuous stirred tank reactor (CSTR) with feedback control systems was simulated, in which a first-order irreversible reaction, \(A \rightarrow B\), was assumed. A schematic diagram of the CSTR and feedback control system is shown in Fig. 1. The CSTR model can be derived as follows:

\[
\frac{dC_i}{dt} = -k_0 e^{-\frac{E}{RT}} C_i + \frac{Q_f}{V} C_{af} - QC_i
\]

(16)

\[
\frac{dT}{dt} = \frac{k_0 e^{-\frac{E}{RT}} (-\Delta H)}{\rho C_p} C_i + \frac{Q_f}{\rho C_p Ah} (T_i - T) + \frac{UA_c (T_f - T)}{\rho C_p Ah}
\]

(17)

\[
\frac{dT_f}{dt} = \frac{Q_e (T_f - T_c)}{V_c C_p} + \frac{UA_c (T - T_f)}{\rho C_p V_c}
\]

(18)

\[
\frac{dh}{dt} = \frac{Q_e - Q}{A}
\]

(19)

where the nominal operating conditions and model parameters are given in Table 3 and the details of the simulated CSTR can be found in the work of Singhal and Seborg (2002). The data were generated according to the problem settings of Fujiwara et al. (2009). The setpoint of the reactor temperature was changed between \(\pm 2\) K every day. For the catalyst deactivation, the frequency factor \((k_0)\) was considered to be linearly decreasing from 7.2x10^18 to 5.4x10^15 over 180 days. It can be expected that there were two types of disturbances to affect the simulated process. One was a short-term effect due to the setpoint changes of the reactor temperature, and another was a long-term effect coming from the catalyst deactivation. The process variables shown in Figure 1 were used as input variables to predict the measured reactant concentration \((C_i)\). The input variables were the temperatures of the reactor, coolant, feed and coolant feed \((T, T_c, T_f, T_{cf})\), the reactor level \((h)\), the flow rates of reactor exit, coolant and feed \((Q, Q_c, Q_e)\). The data were collected every 15 minutes to simulate the sampling rate of an online analyzer. The white noises were added with 1% of the nominal values listed in Table 3 as the standard deviations of Gaussian distributions.

The root mean square error of prediction (RMSEP) was used to assess the prediction performance of the inferential model.

\[
RMSEP = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2
\]

(20)

where \(y_i\) and \(\hat{y}_i\) respectively are the measured and predicted values of the online analyzer and \(m\) is the number of observations in the test dataset. In this example, the test dataset contained 480 observations. For a PLS-based model, the number of PLS components affects the model prediction performance, however, for a single output model, it can always be reduced to a single-component PLS model by removing a sufficient number of OPLS components. The prediction performance of the OPLS model depends on the number of removing components that are uncorrelated to the response variable. Figure 2 compares the RMSEP of the test dataset with different modeling window size and the number of removing OPLS components. It shows the OPLS model with 2-day window size and removing 4 components that are orthogonal to the output variable had the best prediction performance.

From the physical perspective of an irreversible exothermic reaction, the measured reactant concentration \((C_i)\) was inversely proportional to the reactor temperature \((T)\). From the control strategy of the reactor temperature, the coolant flow rate \((Q_c)\) was also inversely proportional to the reactor temperature. Therefore, the measured reactant concentration was proportional to the coolant flow rate. Since the cooling jacket temperature was proportional to the reactor temperature, the reactant concentration was inversely proportional to the coolant temperature \((T_c)\). It can be observed in Fig. 3, which displays the regression coefficients of the OPLS model with one PLS component. The OPLS model removes systematic variation from the predictor.
variables that is unrelated to the response variable, making the interpretation of the prediction model easier.

Fig. 2. The RMSEP of the test dataset, (a) comparing with different window sizes and numbers of removing OPLS components, (b) the window size is two day.

Fig. 3. The regression coefficients of the OPLS.

The prediction results of the OPLS, RPLS and FMWOPLS are compared in Fig. 4. In Fig. 4(a), three different models had a comparable prediction performance, since the initial OPLS model had captured the process variations due to the short-term effects of disturbance. However, from Fig. 4(b) to 4(d), as the long-term effect evolved, the prediction performance also declined as well for an OPLS model without model updating. On the other hand, although the RPLS model had been adapted with the new data, the prediction results gradually fluctuated as the catalyst deactivated. It is obvious that a global linear model cannot depict this simulated example with nonlinear relationships between inputs and output. Since the proposed approach discarded the oldest data once the new data were available, the prediction performance was maintained in a consistent way, as Table 4 lists. Table 4 lists the RMSEP using the different models. It shows the prediction performance of OPLS and RPLS declined as the long-term effect of disturbances becoming significant, on the other hand, the RMSEP by the proposed approach maintains with a consistent range, as the long-term disturbance evolved.
5. CONCLUSIONS
In this paper, a method was presented to develop a soft sensor that can cope with colinearity among predictor variables as well as nonlinearity between the predictor and response variables. The OPLS method effectively removes the non-correlated variation from the predictor variables, therefore, improves the interpretation of the prediction model. The FMWOPLS algorithm inherits the characteristics of the OPLS model, capturing the linear relationships between inputs and outputs, and adapts the model with new data, gradually correcting the captured relationships, to describe a nonlinear process behavior. In the simulated example, it was demonstrated that the OPLS precisely captured the process behavior stimulated by the short-term effect of disturbances, and correctly performed a one-step-ahead prediction. The inferential models were adapted to accommodate the long-term effect of disturbances where the captured relationships using the FMWOPLS model were consistent. The results show that the proposed approach has potential for the implementation of soft sensors in the process industry.

ACKNOWLEDGEMENTS
This work was supported by the National Science Council, Republic of China, under Grant NSC-99-2221-E-268-MY2.

REFERENCES


APPENDIX. KERNEL ALGORITHM FOR MULTIPLE OUTPUT OPLS MODEL
1. $w^T = y^T \Sigma^{-1} \Sigma^{-1} y$. For each column in Y, estimate the corresponding w and create a matrix $W = [W \ w]$. 
2. Estimate the principal components of $W$, i.e., $W = T_r P_r + E_r$. 
3. Perform the kernel algorithm to find the $w_r$ and $p_r$ for the $r^{th}$ PLS component. 
4. Orthogonalize $p_r$ to each column in $T_r$, i.e., $p_r = p_r - \left( t_r^T p_r / (t_r^T t_r) \right) t_r$, then set $w_{ortho} = p_r$. 
5. $w_{ortho} = w_{ortho} \bigg/ \| w_{ortho} \|$. 
6. $p_{ortho} = w_{ortho} \Sigma X / \left( w_{ortho}^T \Sigma X w_{ortho} \right)$. 
7. Save found parameters $P_{ortho} = [P_{ortho} \ p_{ortho}]$, $W_{ortho} = [W_{ortho} \ w_{ortho}]$. For additional orthogonal components, return to step 3 and set $\Sigma_r = \left( I - w_{ortho} p_{ortho}^T \right) \Sigma X \left( I - w_{ortho} p_{ortho}^T \right) $. 
8. Perform the kernel algorithm to find the $w_r$, $p_r$, $r_r$ and $q_r$ for the $r^{th}$ PLS component, after extracting orthogonal components. 
9. Save found parameters $W_{PLS} = [W_{PLS} \ w_r]$, $P_{PLS} = [P_{PLS} \ p_r]$, $R_{PLS} = [R_{PLS} \ r_r]$ and $Q_{PLS} = [Q_{PLS} \ q_r]$. 
10. Deflate $\Sigma_{xy}$ and $\Sigma_{yx}$ by $\Sigma_{xy} = \left( I - w_r p_r^T \right)^T \Sigma_{xy}$ and $\Sigma_{yx} = \left( I - w_r p_r^T \right)^T \Sigma_{yx} \left( I - w_r p_r^T \right)$, and set $a = a + 1$ return to step 1 for next PLS component.