Preprints of the
9th International Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
June 7-10, 2015, Whistler, British Columbia, Canada

Dynamic-Inner Partial Least Squares for Dynamic Data Modeling

Yining Dong * **S. Joe Qin** **

* *Ming Hsieh Department of Electrical Engineering,University of Southern California, Los Angeles, CA 90089 USA (e-mail: yiningdo@usc.edu).*
** *Mork Family Department of Chemical Engineering and Material Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: sqin@usc.edu)*

Abstract: Partial least squares(PLS) regression has been widely used to capture the relationship between inputs and outputs in static system modeling. Several dynamic PLS algorithms were proposed to capture the characteristic of dynamic systems. However, none of these algorithms provides an explicit description for dynamic inner model and outer model. In this paper, a dynamic inner PLS is proposed for dynamic system modelling. The proposed algorithm gives explicit dynamic inner model and makes inner model and outer model consistent at the same time. Several examples are given to show the effectiveness of the proposed algorithm.

Keywords: dynamic partial least squares, data-driven modeling

1. INTRODUCTION

Since the work of Wold et al. (1984), partial least squares(PLS) has been widely applied in prediction(Kaspar and Ray (1993), Dayal and MacGregor (1997), Pan et al. (2013)) and process monitoring(Wise and Gallagher (1996), MacGregor et al. (1994), Zhou et al. (2010)) as an effective dimension reduction method. The idea of PLS is to extract latent variables from inputs and outputs such that the covariance between latent variables is maximized. However, only static relationships are considered in traditional PLS. In the case that dynamic relationships exist between inputs and outputs, traditional PLS will leave a large amount of covariance unmodeled. This reduces the effectiveness of PLS and makes it even unsuitable for dynamic system modeling.

Several modified PLS algorithms have been proposed to deal with dynamic systems. A straightforward method was proposed by Qin and McAvoy (1996), where a certain number of lagged inputs and outputs are included in the augmented input matrix. Nonlinear finite impulse response(NFIR) inner models are built between scores of inputs and outputs after outer models converge. The disadvantage of this method is that it doesn't give explicit representation of the dynamic relationship. The augmented loading matrix makes it difficult to interpret the model and explore the underlying data structures. Also, the dimension of the loading vector increases with the number of lags, which will increase the computational complexity.

Kaspar and Ray (1993) proposed a modified PLS modeling algorithm. It provides a compact representation: no lagged variables are included in the outer model. Prior dynamic knowledge was used to design filters such that dynamic components in the inputs are removed. A dynamic inner model is built between input scores and output scores after a static model is built between filtered inputs and outputs. Lakshminarayanan et al. (1997) proposed a similar method by building dynamic inner relationship between inputs scores and output scores. The disadvantage of both methods is that dynamic inner model is inconsistent with static outer model.

Li et al. (2011) proposed a dynamic PLS method by utilizing a weighted combination of lagged input data as the input to the algorithm. An inner model is built between output scores and a weighted combination of lagged input scores. This gives a compact inner and outer model. However, the inner model is not explicit and difficult to interpret.

In this paper, a dynamic inner PLS (DiPLS) algorithm is proposed. The proposed algorithm provides explicit inner model and outer model. In addition, the inner model and outer model this algorithm gives are consistent. The explicit and consistent representation makes it easy to interpret the results.

The remaining sections of the paper is organized as follows. Section 2 reviews the traditional algorithm. Section 3 presents the proposed DiPLS algorithm. Section 4 discusses several examples to show the effectiveness of the algorithm. Section 5 gives conclusions and discussions.

2. PARTIAL LEAST SQUARES

PLS is first proposed by Wold et al. (1984) to perform regression with interrelated input variables, which is common for routine operation data, to provide a way to trade off between the model prediction variance and bias. PLS extracts latent variables from inputs and outputs such that the covariance between a pair of latent variables is maximized. First, loading vectors for inputs and outputs are used to generate the latent variables and are calculated to maximize the covariance of latent variables. Then a linear inner model is built between input scores and output scores. The input scores are used to deflate the input matrix, while the estimated output scores calculated from the inner linear model are used to deflate the output matrix.

Consider the input matrix $\mathbf{X}$ and output matrix $\mathbf{Y}$, the objective of PLS is

$$\begin{aligned} \max \quad & \mathbf{q}^T\mathbf{Y}^T\mathbf{X}\mathbf{w} \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{1}$$

where $\mathbf{w}$ and $\mathbf{q}$ are input and output weights, respectively. The solution to this optimization problem is

$$\begin{aligned} \mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w} = \lambda_w \mathbf{w} \\ \mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{q} = \lambda_q \mathbf{q} \end{aligned} \tag{2}$$

which indicates λ_w and λ_q are eigenvalues and $\mathbf{w}$ and $\mathbf{q}$ are eigenvectors of corresponding matrices. After the weighting vectors $\mathbf{w}$ and $\mathbf{q}$ are obtained, the input latent score vector $\mathbf{t}$ and the output latent score vector $\mathbf{u}$ can be calculated as $\mathbf{t} = \mathbf{X}\mathbf{w}$, and $\mathbf{u} = \mathbf{Y}\mathbf{q}$. The process that builds the relationship between the latent score vectors and the corresponding observations is called outer modeling. After the outer model is obtained, the inner model can be built between the latent scores $\mathbf{u}$ and $\mathbf{t}$ as follows

$$\mathbf{u} = b\mathbf{t} + \mathbf{r}. \tag{3}$$

Deflate $\mathbf{X}$ and $\mathbf{Y}$

$$\mathbf{X} := \mathbf{X} - \mathbf{t}\mathbf{p}^T. \tag{4}$$

$$\mathbf{Y} := \mathbf{Y} - b\mathbf{t}\mathbf{q}^T. \tag{5}$$

Iterate this process until enough factors are extracted. Detail algorithm for PLS can be found in Höskuldsson (1988) and Geladi and Kowalski (1986).

It is clear from the objective and procedure of PLS that only static relations in the input and output are extracted by PLS. In the case that dynamic relationships exist between the input and output data, traditional PLS will leave the dynamics unmodeled. This restricts the applicability of PLS and makes it unsuitable for dynamic data modeling. To build dynamic PLS(DPLS) models, a straightforward approach is to extend the input matrix with time-lagged inputs, as proposed in Qin and McAvoy (1996) in a nonlinear dynamic PLS scheme. While this DPLS approach is reasonable, it is difficult to interpret the extracted latent factors and the dynamic relationship tends to have excessive parameters. An alternative approach proposed by Kaspar and Ray (1993) and Lakshminarayanan et al. (1997) keeps the outer model the same as in static PLS, but builds a dynamic inner model between $\mathbf{u}$ and $\mathbf{t}$. This approach is inconsistent between the outer model treatment and inner model treatment, as the statically extracted latent scores $\mathbf{u}$ and $\mathbf{t}$ are forced to have a dynamic relation in the inner model. In the extreme case that $\mathbf{u}$ and $\mathbf{t}$ are statically uncorrelated but dynamically correlated, this approach fails.

To build a dynamic PLS model with consistent inner and outer relation, both the inner model and outer model should aim to extract a dynamic inner relation such as

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k$$

with the latent variables related to data as follows

$$\begin{aligned} u_k &= \mathbf{y}_k^T\mathbf{q} \\ t_k &= \mathbf{x}_k^T\mathbf{w} \end{aligned}$$

where $\mathbf{x}_k$ and $\mathbf{y}_k$ are the input and output vectors at time k. For each factor, the inner model prediction should be

$$\begin{aligned} \hat{u}_k &= \mathbf{x}_k^T\mathbf{w}\beta_0 + \mathbf{x}_{k-1}^T\mathbf{w}\beta_1 + \cdots + \mathbf{x}_{k-s}^T\mathbf{w}\beta_s \\ &= [\mathbf{x}_k^T \quad \mathbf{x}_{k-1}^T \cdots \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w}) \end{aligned}$$

where $\boldsymbol{\beta} = (\beta_0 \quad \beta_1 \cdots \beta_s)^T$ and $\boldsymbol{\beta} \otimes \mathbf{w}$ is the kronecker product. The outer model that is consistent with the above inner dynamic model should maximize the covariance between u_k and $\hat{u}_k$, that is, to maximize

$$\frac{1}{N}\sum_{k=s}^{N+s} \mathbf{q}^T\mathbf{y}_k[\mathbf{x}_k^T \quad \mathbf{x}_{k-1}^T \cdots \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w}) \tag{6}$$

This objective leads to the dynamic inner PLS (DiPLS) algorighm to be derived in the next section.

3. DYNAMIC INNER PLS ALGORITHM

3.1 Objective

Let $\mathbf{x}_k$ and $\mathbf{y}_k$ be the input and output vectors at time $k, k = 0, 1, ...N + s$, and $N + s + 1$ samples of input and output are collected in the following matrices

$$\mathbf{X} = [\mathbf{x}_0 \quad \mathbf{x}_1 \quad \cdots \quad \mathbf{x}_{s+N}]^T$$

$$\mathbf{Y} = [\mathbf{y}_0 \quad \mathbf{y}_1 \quad \cdots \quad \mathbf{y}_{s+N}]^T$$

Define

$$\mathbf{X}_i = [\mathbf{x}_i \quad \mathbf{x}_{i+1} \quad \cdots \quad \mathbf{x}_{i+N}]^T$$

$$\mathbf{Z}_s = [\mathbf{X}_s \quad \mathbf{X}_{s-1} \quad \cdots \mathbf{X}_0]$$

$$\mathbf{Y}_s = [\mathbf{y}_s \quad \mathbf{y}_{s+1} \quad \cdots \quad \mathbf{y}_{s+N}]^T$$

The objective of DiPLS that is consistent with (6) should be formulated as

$$\begin{aligned} \max \quad & \mathbf{q}^T\mathbf{Y}_s^T\mathbf{Z}_s(\boldsymbol{\beta} \otimes \mathbf{w}) \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1, \|\boldsymbol{\beta}\| = 1 \end{aligned} \tag{7}$$

where s is the dynamic order of the model. The dimension of $\mathbf{w}$ is the same as the number of input variables, which gives an explicit outer model. If $s = 0$, $\mathbf{Y}_s$ is related to $\mathbf{X}_s$ only, and DiPLS is reduced to traditional PLS.

3.2 Outer modeling

Lagrange multipliers are used to solve this optimization in (7). Define

$$\begin{aligned} J \; =& \mathbf{q}^T\mathbf{Y}_s^T\mathbf{Z}_s(\boldsymbol{\beta} \otimes \mathbf{w}) + \frac{1}{2}\lambda_q(1 - \mathbf{q}^T\mathbf{q}) \\ &+ \frac{1}{2}\lambda_\beta(1 - \boldsymbol{\beta}^T\boldsymbol{\beta}) + \frac{1}{2}\lambda_w(1 - \mathbf{w}^T\mathbf{w}) \end{aligned} \tag{8}$$

where

$$(\boldsymbol{\beta} \otimes \mathbf{w}) = (\boldsymbol{\beta} \otimes \mathbf{I})\mathbf{w} = (\mathbf{I} \otimes \mathbf{w})\boldsymbol{\beta}$$

Taking derivatives with respective to $\mathbf{q}, \mathbf{w}, \boldsymbol{\beta}$ and setting the results to zero lead to:

$$\frac{\partial J}{\partial \mathbf{q}} = \mathbf{Y}_s^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{w}) - \lambda_q \mathbf{q} = 0$$

$$\frac{\partial J}{\partial \mathbf{w}} = (\boldsymbol{\beta} \otimes \mathbf{I})^T \mathbf{Z}_s^T \mathbf{Y}_s \mathbf{q} - \lambda_w \mathbf{w} = 0 \tag{9}$$

$$\frac{\partial J}{\partial \boldsymbol{\beta}} = (\mathbf{I} \otimes \mathbf{w})^T \mathbf{Z}_s^T \mathbf{Y}_s \mathbf{q} - \lambda_\beta \boldsymbol{\beta} = 0$$

Therefore

$$\begin{aligned}\lambda_q &= \mathbf{q}^T \mathbf{Y}_s^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{w}) = \mathbf{q}^T \mathbf{Y}_s^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{I}) \mathbf{w} \\ &= \lambda_w = \lambda_\beta\end{aligned}$$

This implies that $\lambda_q, \lambda_w, \lambda_\beta$ are equal to the maximum value of J. However, from (9) we can see that the vectors $\mathbf{q}, \mathbf{w}, \boldsymbol{\beta}$ cannot be solved for explicitly. Therefore, the following iterative method can be used to solve the problem.

(1) Initialize $\mathbf{w}, \mathbf{q}, \boldsymbol{\beta}$ to unit vectors

(2) Calculate $\mathbf{w}, \mathbf{q}, \boldsymbol{\beta}$ by iterating the following three relations until convergence.

$$\mathbf{q} = \mathbf{Y}_s^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{w}); \quad \mathbf{q} := \mathbf{q} / \|\mathbf{q}\|$$

$$\mathbf{w} = (\boldsymbol{\beta} \otimes \mathbf{I})^T \mathbf{Z}_s^T \mathbf{Y}_s \mathbf{q}; \quad \mathbf{w} := \mathbf{w} / \|\mathbf{w}\|$$

$$\boldsymbol{\beta} = (\mathbf{I} \otimes \mathbf{w})^T \mathbf{Z}_s^T \mathbf{Y}_s \mathbf{q}; \quad \boldsymbol{\beta} := \boldsymbol{\beta} / \|\boldsymbol{\beta}\|$$

(3) Calculate $V = \mathbf{q}^T \mathbf{Y}_s^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{w})$

where V is the value of the objective function from (7). Sometimes, the iteration converges to a local optimum. This will lead to a smaller V. To avoid local optimum, initialize the vectors $\mathbf{w}, \mathbf{q}, \boldsymbol{\beta}$ in randomly Step (1) and perform multiple trials. Take the results $\mathbf{w}, \mathbf{q}, \boldsymbol{\beta}$ that maximize value of V.

3.3 Inner modeling

After the outer model is obtained, the scores of input and output can be calculated as

$$\begin{aligned}\mathbf{t} &= [t_0 \quad t_1 \cdots t_{s+N}]^T = \mathbf{X}\mathbf{w} \\ \mathbf{u} &= [u_0 \quad u_1 \cdots u_{s+N}]^T = \mathbf{Y}\mathbf{q}\end{aligned} \tag{10}$$

Similar to $\mathbf{X}_i$ and $\mathbf{Y}_s$ define $\mathbf{t}_i$ and $\mathbf{u}_s$ as follows

$$\begin{aligned}\mathbf{t}_i &= [t_i \quad t_{i+1} \quad \cdots \quad t_{i+N}]^T \\ \mathbf{u}_s &= [u_s \quad u_{s+1} \quad \cdots \quad u_{s+N}]^T\end{aligned}$$

It is obvious that

$$\begin{aligned}\mathbf{t}_i &= \mathbf{X}_i \mathbf{w}; \quad i = 0, 1, \cdots, s \\ \mathbf{u}_s &= \mathbf{Y}_s \mathbf{q};\end{aligned} \tag{11}$$

In dynamic PLS, a dynamic relationship is built between inputs and outputs. Therefore, input scores and output scores calculated from the outer model are maximally dynamically related. Dynamic inner models should be built to capture these dynamic variations.

The inner model describing the dynamic relationship between input scores and output scores should be built between $\mathbf{u}_s$ and $\mathbf{t}_s, \mathbf{t}_{s-1}, \cdots \mathbf{t}_0$ as

$$\mathbf{u}_s = \alpha_0 \mathbf{t}_s + \alpha_1 \mathbf{t}_{s-1} + \cdots + \alpha_s \mathbf{t}_0 + \mathbf{r}_s \tag{12}$$

where $\mathbf{r}_s$ is the residual of the regression. Ordinary least squares can be applied to solve $\alpha_0, \alpha_1, \cdots, \alpha_s$. After α's are solved, (12) can be used to predict $\mathbf{u}_s$ from $\mathbf{t}_s, \mathbf{t}_{s-1}, \cdots, \mathbf{t}_0$, where

$$\hat{\mathbf{u}}_s = \alpha_0 \mathbf{t}_s + \alpha_1 \mathbf{t}_{s-1} + \cdots + \alpha_s \mathbf{t}_0 \tag{13}$$

$\hat{\mathbf{u}}_s$ can be used further to deflate $\mathbf{Y}_s$.

Remark 1. An AR model can be built for $\mathbf{t}_s$ as long as $\mathbf{t}_s$ is auto-correlated.

$$\mathbf{t}_s = \varphi_1 \mathbf{t}_{s-1} + \varphi_2 \mathbf{t}_{s-2} + \cdots + \varphi_s \mathbf{t}_0 + \mathbf{e}_t$$

This AR model can be used to predict $\mathbf{t}_s$ from past samples, and subsequently be used to predict $\mathbf{X}$.

3.4 Deflation

After $\mathbf{t}$, $\hat{\mathbf{u}}$ are calculated, the loading vector $\mathbf{q}$ for $\mathbf{X}$ can be derived as

$$\mathbf{p} = \mathbf{X}^T \mathbf{t} / \mathbf{t}^T \mathbf{t}$$

Delation can be performed as

$$\begin{aligned}\mathbf{X} &:= \mathbf{X} - \mathbf{t}\mathbf{p}^T \\ \mathbf{Y}_s &:= \mathbf{Y}_s - \hat{\mathbf{u}}_s \mathbf{q}^T\end{aligned} \tag{14}$$

After the residuals of $\mathbf{X}$ and $\mathbf{Y}$ are calculated, subsequent factors can be obtained from these residuals by repeating the same procedure.

3.5 Procedure of DiPLS modeling

In the DiPLS outer modeling, the kronecker expressions can be simplified using (11) as follows.

$$\mathbf{w} = (\boldsymbol{\beta} \otimes \mathbf{I})^T \mathbf{Z}_s \mathbf{Y}_s \mathbf{q} = \sum_{i=0}^{s} \boldsymbol{\beta}_i \mathbf{X}_{s-i}^T \mathbf{Y}_s \mathbf{q} = \sum_{i=0}^{s} \boldsymbol{\beta}_i \mathbf{X}_{s-i}^T \mathbf{u}_s$$

$$\mathbf{q} = \mathbf{Y}_s^T \mathbf{Z}_s (\boldsymbol{\beta} \otimes \mathbf{w}) = \mathbf{Y}_s^T \sum_{i=0}^{s} \boldsymbol{\beta}_i \mathbf{X}_{s-i} \mathbf{w} = \mathbf{Y}_s^T \sum_{i=0}^{s} \boldsymbol{\beta}_i \mathbf{t}_{s-i}$$

$$\boldsymbol{\beta} = [\mathbf{t}_s \quad \mathbf{t}_{s-1} \quad \cdots \quad \mathbf{t}_0]^T \mathbf{u}_s$$

The procedure of DiPLS modeling can be summarized in Table 1.

Since the procedure of proposed DiPLS is similar to the procedure of PLS, the structure of DiPLS is similar to PLS as well. Let $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \cdots \mathbf{w}_l]$, $\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \cdots \mathbf{p}_l]$, $\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \cdots \mathbf{q}_l]$, $\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \cdots \mathbf{q}_l]$, $\mathbf{T} = [\mathbf{X}\mathbf{w}_1 \quad \mathbf{X}\mathbf{w}_2 \cdots \mathbf{X}\mathbf{w}_l]$, where l is the number of latent variables. Let $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$, then we have

$$\mathbf{T} = \mathbf{X}\mathbf{R} \tag{15}$$

$\hat{\mathbf{U}}_s$ can be calculated from $\mathbf{T}$ such that

$$\hat{\mathbf{Y}} = \hat{\mathbf{U}}_s \mathbf{Q}^T \tag{16}$$

where $\hat{\mathbf{Y}}$ is the output predicted from $\mathbf{X}$.

Therefore, once samples $\mathbf{x}_k$ up to time k are measured, $\mathbf{t}_k$ can be calculated, and $\mathbf{y}_k$ can be predicted.

Remark 2. In the case that it is desirable to include auto-regression of the output, the objective of DiPLS can be modified as

Table 1. DiPLS Algorithm

1. Scale $\mathbf{X}$ and $\mathbf{Y}$ to zero-mean and unit-variance. Initialize $\boldsymbol{\beta}$ with $[1, 0, \cdots, 0]'$, and $\mathbf{u}_s$ as some column of $\mathbf{Y}_s$.
2. Outer modeling. Iterate the following relations until convergence achieved.

$$\mathbf{w} = \sum_{i=0}^{s} \boldsymbol{\beta}_i \mathbf{X}_{s-i}^T \mathbf{u}_s; \mathbf{w} := \mathbf{w} / \|\mathbf{w}\|$$

$$\mathbf{t} = \mathbf{X}\mathbf{w} \quad \text{and form } \mathbf{t}_{s-i} \text{ from } \mathbf{t} \text{ for } i = 0, 1, \cdots, s$$

$$\mathbf{q} = \mathbf{Y}_s^T \sum_{i=0}^{s} \boldsymbol{\beta}_i \mathbf{t}_{s-i}; \mathbf{q} := \mathbf{q} / \|\mathbf{q}\|$$

$$\mathbf{u}_s = \mathbf{Y}_s \mathbf{q}$$

$$\boldsymbol{\beta} = [\mathbf{t}_s \quad \mathbf{t}_{s-1} \cdots \mathbf{t}_0]^T \mathbf{u}_s; \boldsymbol{\beta} := \boldsymbol{\beta} / \|\boldsymbol{\beta}\|$$

3. Inner model building. Build a linear model between $\mathbf{t}_s, \mathbf{t}_{s-1}, \cdots, \mathbf{t}_0$ and $\mathbf{u}_s$

$$\mathbf{u}_s = \alpha_0 \mathbf{t}_s + \alpha_1 \mathbf{t}_{s-1} + \cdots + \alpha_s \mathbf{t}_0 + \mathbf{r}_s$$

Calculate predicted $\hat{\mathbf{u}}_s$

$$\hat{\mathbf{u}}_s = \alpha_0 \mathbf{t}_s + \alpha_1 \mathbf{t}_{s-1} + \cdots + \alpha_s \mathbf{t}_0$$

4. Deflation. Deflate $\mathbf{X}$ and $\mathbf{Y}$ as

$$\mathbf{X} := \mathbf{X} - \mathbf{t}\mathbf{p}^T$$

$$\mathbf{Y}_s := \mathbf{Y}_s - \hat{\mathbf{u}}\mathbf{q}^T$$

5. Repeat to Step 2 until enough latent variables are extracted.

$$\begin{aligned} \max \quad & \mathbf{q}^T(\gamma_0 \mathbf{Y}_s^T + \gamma_1 \mathbf{Y}_{s-1}^T + \cdots + \gamma_f \mathbf{Y}_{s-f}^T) \\ & *(\beta_0 \mathbf{X}_s + \beta_1 \mathbf{X}_{s-1} + \cdots + \beta_s \mathbf{X}_0)\mathbf{w} \qquad (17) \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1, \|\boldsymbol{\beta}\| = 1, \|\boldsymbol{\gamma}\| = 1 \end{aligned}$$

where γ's are the weights of different lag of $\mathbf{Y}$, f is the order of the output. After (17) is solved, inner model can be built as an ARX model of $\mathbf{u}_s$

$$\begin{aligned} \mathbf{u}_s = \; & \varphi_0 \mathbf{u}_{s-1} + \cdots + \varphi_f \mathbf{u}_{s-f} \\ & + \alpha_0 \mathbf{t}_s + \cdots + \alpha_s \mathbf{t}_0 + \mathbf{r}_s \end{aligned} \qquad (18)$$

Then the prediction of output score $\hat{\mathbf{u}}_s$ can be calculated and be used to predict the output.

3.6 Determination of model parameters

In DiPLS modeling, there are two parameters to be determined. One is the lag number s, the other one is the number of latent variables. The lag number s is also the impulse response order of the data. From the simulation results in Section 4, we can see the prediction is not sensitive to s. When more lags than the dynamic order of the system are included in augmented input, DiPLS tends to diminish the coefficients corresponding to the excess lags. Compared to the number of lags, the number of latent variables usually has more impact on the prediction results. Therefore, the number of latent variables need to be determined carefully. Cross-validation is a popular method used in PLS to select the number of latent variables, aiming to minimize the prediction error. Similar procedures can be applied to DiPLS to obtain best prediction results.

4. CASE STUDIES ON SIMULATION DATA

In this section, three sets of data are simulated, each corresponds to a scenario. In Scenario 1, both input $\mathbf{X}$ and $\mathbf{Y}$ are generated from a static model. In Scenario 2, input $\mathbf{X}$ is generated from a dynamic model, and output $\mathbf{Y}$ is generated from a static model. In Scenario 3, input $\mathbf{X}$ is generated from a static model, and output $\mathbf{Y}$ is generated from a dynamic model. The advantages of DiPLS over traditional PLS is demonstrated in these three basic examples.

4.1 Scenario 1

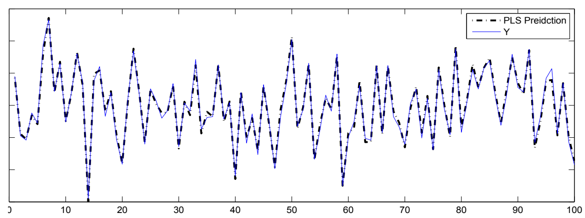
$\mathbf{X}$ and $\mathbf{Y}$ are generated from static process.

$$\mathbf{x}_k = \mathbf{P}\mathbf{t}_k + \mathbf{e}_k$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

$$\mathbf{P} = \begin{pmatrix} 0.5765 & 0.2856 & 0.1614 \\ 0.3660 & 0.0458 & 0.9060 \\ 0.5889 & 0.4645 & 0.9942 \\ 0.3572 & 0.3450 & 0.7396 \\ 0.4036 & 0.6851 & 0.2262 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0.7451 \\ 0.4928 \\ 0.7320 \\ 0.4738 \\ 0.5652 \end{pmatrix}^T$$

where $\mathbf{e}_k \in \mathbb{R}^5 \sim N([0, 0.5^2])$, and $\mathbf{v}_k \in \mathbb{R} \sim N([0, 0.5^2])$, $\mathbf{t}_k \in \mathbb{R}^3 \sim N([0, 2^2])$.

1000 data points are generated. First 500 data points are used as training dataset to train the model, the next 400 data points are used as development dataset to select the parameter, the last 100 data points are used as test dataset to evaluate the prediction result. The optimal parameters determined by cross validation is $s = 0$ and the number of components is 3. $s = 0$ indicates that DiPLS reduces to traditional PLS, which is consistent with the static model of inputs and outputs.



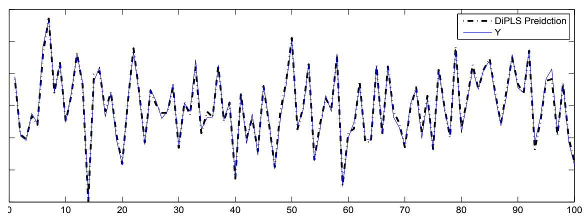


Fig. 1. Prediction result of DiPLS and PLS for Scenario1

Fig. 1 shows the prediction results of DiPLS and PLS, from which we can see DiPLS gives the same result as PLS. This is consistent with our analysis. If we increase s to 1, the values of β_0, β_1, which are the weights of time lag 0 and 1, are listed in Table 2. We can tell from the table that $\beta_0^2 >> \beta_1^2$ for each iteration, which implies the input data with lag 1 has little impact in the outer model building.

Table 2. β_0, β_1 for each factor in Scenario1

Value	factor 1	factor 2	factor 3
β_0	0.9998	-0.9590	-0.9978
β_1	-0.217	0.2833	0.0670

DiPLS reduces to PLS even though excess time lags are included.

4.2 Scenario 2

$\mathbf{X}$ is generated from dynamic process, $\mathbf{Y}$ is generated from static process.

$$\begin{aligned} \mathbf{t}_k &= \mathbf{A}_1\mathbf{t}_{k-1} + \mathbf{A}_2\mathbf{t}_{k-2} + \mathbf{f}_k \\ \mathbf{x}_k &= \mathbf{P}\mathbf{t}_k + \mathbf{e}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

where $\mathbf{f}_k \in \mathbb{R}^3 \sim N([0, 0.5^2])$, $\mathbf{P}$ and $\mathbf{C}$ are the same as Scenario 1.

$$\mathbf{A}_1 = \begin{pmatrix} 0.6767 & 0.5809 & 0.9315 \\ 1.2812 & -0.5343 & -1.6000 \\ -1.5083 & 0.9991 & 0.7529 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 0.7155 & -0.0652 & 1.1192 \\ 1.1132 & -0.5371 & -0.1691 \\ -0.5571 & -1.0748 & 0.2330 \end{pmatrix}$$

Same number of data points are used as training dataset, development dataset, testing dataset. The optimal parameters determined by cross validation is $s = 0$ and the number of factors is 3. The reason that $s = 0$ is that DiPLS considers covariance between the input and output, not covariance of input or output. Therefore, s is only determined by the relationship between input and output. In this scenario, there's a static model between $\mathbf{X}$ and $\mathbf{Y}$. Therefore, $s = 0$. The prediction results of DiPLS and PLS are shown in Fig.2 From the figure, we can see DiPLS

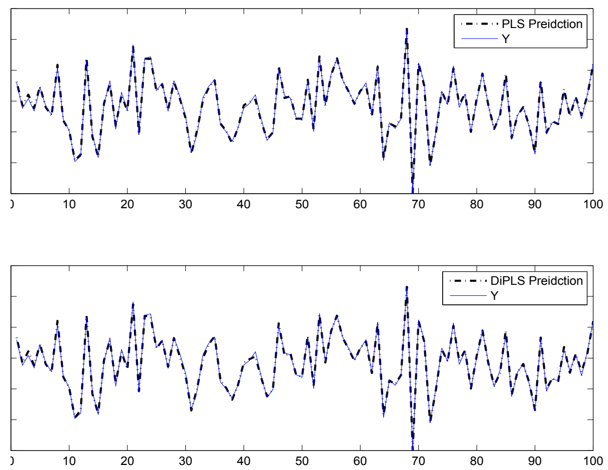


Fig. 2. Prediction result of DiPLS and PLS for Scenario2

gives the same result as PLS, which is consistent with the analysis. If s is increased to 4, the values of $\beta_0, \beta_1, \beta_2, \beta_3$ are listed in Table 3. We can tell from the table that β_0 is much larger than the square of other β's, therefore, the input data with no lag dominates the DiPLS result. DiPLS performs like a PLS model even though excess lagged input data are included.

Table 3. β_0, β_1,β_3,β_4 for each factor in Scenario2

Value	factor 1	factor 2	factor 3
β_0	0.9940	0.9156	-0.9060
β_1	0.0821	0.3233	-0.1166
β_2	0.0691	0.0690	0.3554
β_3	0.0197	-0.2290	0.1980

Table 4. β_0, β_1,β_3,β_4 for each factor in Scenario3

Value	factor 1	factor 2	factor 3	factor 4
β_0	0.4251	0.1102	0.4528	0.9514
β_1	0.9009	0.-0.9850	0.8808	0.2832
β_2	-0.0707	0.1069	-0.1062	-0.0694
β_3	-0.0517	-0.0789	-0.0888	-0.0996

4.3 Scenario 3

$\mathbf{X}$ is generated from static process, $\mathbf{Y}$ is generated from dynamic process.

$$\begin{aligned} \mathbf{x}_k &= \mathbf{P}\mathbf{t}_k + \mathbf{e}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{C}_2\mathbf{x}_{k-1} + \mathbf{v}_k \end{aligned}$$

where $\mathbf{P}, \mathbf{C}$ are the same as Scenario 1.

$$\mathbf{C}_2 = \begin{pmatrix} 1.9939 & 0.7728 & 1.0146 & 1.1563 & 1.2307 \end{pmatrix}$$

Same number of data points are used as training dataset, development dataset, testing dataset. The optimal parameters determined by cross validation is $s = 1$ and the number of component is 4. Since $\mathbf{y}_k$ is related to both $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$, $s = 1$ is consistent with the dynamic structure between $\mathbf{X}$ and $\mathbf{Y}$. The prediction results of DiPLS and PLS are shown in Fig.3. From Fig.3, we can see when

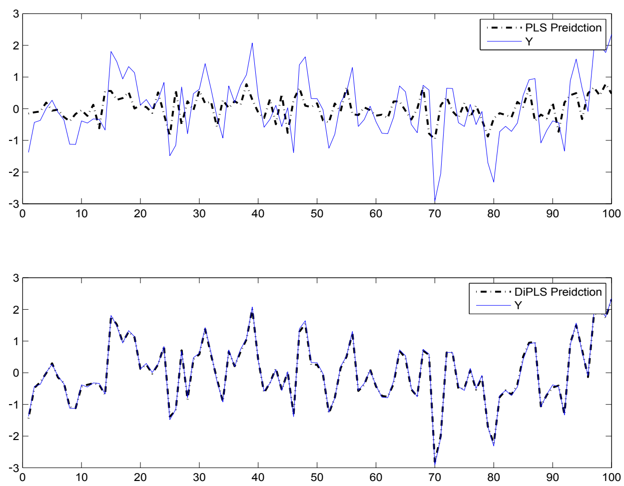


Fig. 3. Prediction result of DiPLS and PLS for Scenario3

there are dynamics between inputs and outputs, DiPLS gives much better results than PLS. If s is increased to 4, the values of $\beta_0, \beta_1, \beta_2, \beta_3$ are listed in Table 4

We can see the squares of β_2, β_3 are much smaller than β_0, β_1 in general. Therefore, the input data with a lag of 2 and a lag of 3 will have little impact on the result. The DiPLS models built with $s = 1$ and $s = 3$ are similar.

Table 5. MSE of DiPLS and PLS for TEP XMEAS(38)

	DiPLS	PLS
MSE	0.5503	0.8817

5. CASE STUDY ON THE TENNESSEE EASTMAN PROCESS

The Tennessee Eastman Process(TEP) was developed to provide a realistic simulation of an industrial process for the evaluation of monitoring methods (Downs and Vogel (1993)). The process contains 12 manipulated variables and 41 measured variables. The measured variables contain 22 process variables sampled every 3 minutes, and 19 quality variables sampled with dead time and time delays. In this case study, 22 process variables XMEAS(1-22) and 11 manipulated variables XMV(1-11) are used as input, XMEAS(38), which is the calculated data from the analyzer, is used as output. Sampling frequency for XMEAS(38) is 0.25h and dead time is 0.25h. 300 data points are used as the training dataset, 100 data points are used as the development dataset, and 960 data points are used as the testing dataset. Data are pre-shifted to compensate the 0.25h delay. The optimal parameters determined by cross validation is $s = 5$ and the number of component is 3. The prediction results of DiPLS and PLS is shown in Fig.4

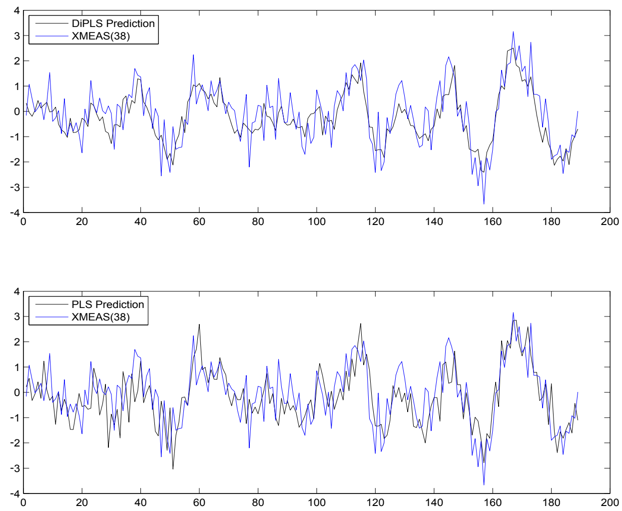


Fig. 4. Prediction result of DiPLS and PLS for TEP

Note that for XMEAS(38), there is only one measured value every five data points due to the lower sampling rate, and the subsequent four values are artificial. Therefore, only 1/5 of the 960 data points are compared. The mean squared error(MSE) for DiPLS and PLS are listed in Table 5 From Table 5, we can see the prediction of DiPLS is closer to the true value than PLS. Fig.4 also show that DiPLS captures trends in the data better than PLS. Therefore, this case study shows that DiPLS performs better than PLS in dynamic process modeling.

6. CONCLUTION

In this article, a dynamic inner PLS(DiPLS) was proposed for dynamic process modeling. The proposed method gives an explicit representation of the structures, and provides consistent inner model and outer models at the same time. For the data from static processes, DiPLS reduces to traditional PLS, and provides the same results as PLS. Cross-validation can be used to determine the optimal number of lags and the number of components. Case studies on simulation data and Tennessee Eastman Process show the effectiveness of the proposed method.

REFERENCES

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

Downs, J.J. and Vogel, E.F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3), 245–255.

Geladi, P. and Kowalski, B.R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185, 1–17.

Höskuldsson, A. (1988). PLS regression methods. *Journal of chemometrics*, 2(3), 211–228.

Kaspar, M.H. and Ray, W.H. (1993). Dynamic pls modelling for process control. *Chemical Engineering Science*, 48(20), 3447–3461.

Lakshminarayanan, S., Shah, S.L., and Nandakumar, K. (1997). Modeling and control of multivariable processes: dynamic pls approach. *AIChE Journal*, 43(9), 2307–2322.

Li, G., Liu, B., Qin, S.J., and Zhou, D. (2011). Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic T-PLS approach. *Neural Networks, IEEE Transactions on*, 22(12), 2262–2271.

MacGregor, J.F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock pls methods. *AIChE Journal*, 40(5), 826–838.

Pan, Y.C., Qin, S.J., Nguyen, P., and Barham, M. (2013). Hybrid inferential modeling for vapor pressure of hydrocarbon mixtures in oil production. *Industrial & Engineering Chemistry Research*, 52(35), 12420–12425.

Qin, S.J. and McAvoy, T. (1996). Nonlinear FIR modeling via a neural net pls approach. *Computers & chemical engineering*, 20(2), 147–159.

Wise, B.M. and Gallagher, N.B. (1996). The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6), 329–348.

Wold, S., Ruhe, A., Wold, H., and Dunn, III, W. (1984). The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3), 735–743.

Zhou, D., Li, G., and Qin, S.J. (2010). Total projection to latent structures for process monitoring. *AIChE Journal*, 56(1), 168–178.

