Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

NONLINEAR PROCESS MODELING AND OPTIMIZATION BASED ON MULTIWAY KERNEL PARTIAL LEAST SQUARES MODEL

Liqing Di
Zhihua Xiong
Xianhui Yang

Dept. of Automation
Tsinghua University
Beijing 100084, CHINA

ABSTRACT

MKPLS (Multiway Kernel Partial Least Squares) methods are used to model the batch processes from process operational data. To improve the optimization performance, a batch-to-batch optimization strategy is proposed based on the idea of the similarity between the iterations during numerical optimization and successive batch runs. SQP (Sequential Quadratic Programming) coupling with MKPLS model is used to solve the optimization problem, and the plant data, instead of the MKPLS model predictions, are used in gradient calculation. The proposed strategy is illustrated on a simulated bulk polymerization of styrene. The results demonstrate that the optimization performance has been improved in spite of the model-plant mismatches.

1 INTRODUCTION

Batch processes play an important role in the production of low-volume and high value-added products, such as polymers, pharmaceuticals and biochemical products. There have been growing interests in batch process optimization in recent years (Bonvin 1998). To achieve good optimal control performance, it is very important to obtain an accurate model. Comparing with the first principle models, data-based empirical models can be developed easily (Xiong and Zhang 2005).

Recently, regression models from experimental or historical data are also used, such as partial least squares (PLS) (Wold et al. 1989). However, the high dimensionality and collinearity of such data makes it difficult to build a accurate model to predict the final product qualities of batch processes. PLS has been shown to be a powerful technique for process modeling and calibration in systems where the predictor variables are collinear, measurement data contain noise, variables have high dimensionality, and where there are fewer observations than predictor variables (Kim et al. 2005).

Technically, PLS supposes that process data are linear. As complex physical systems always exhibit significant nonlinear characteristics, such linear PLS is inappropriate for describing the underlying data structure of nonlinear processes. To address the problem of data nonlinearity, a number of approaches have been proposed that incorporate nonlinear features into the linear PLS framework, such as polynomial nonlinear mapping, neural networks, and so on (Qin and McAvoy 1992, Dong et al. 1996). Recently, a new non-linear PLS technique, called kernel PLS (KPLS), was developed (Rosipal and Trejo 2001). KPLS is quite different from other nonlinear PLS algorithms in that the original input data are nonlinearly transformed into a feature space of arbitrary dimensionality, and then a linear PLS model is built in the feature space. KPLS avoids nonlinear optimization by utilizing the kernel function corresponding to the inner product in the feature space. As a result, KPLS essentially requires only linear algebra, making it as simple as standard PLS (Kim et al. 2005). Furthermore, KPLS may use many different kinds of kernel functions, thus it can handle a wide range of nonlinearities. It has been shown that KPLS performs better than linear PLS in regressing and classifying data in nonlinear systems (Rosipal et al. 2003). However, the number of latent variables selected for KPLS may be larger than that for linear PLS because KPLS extracts the major latent variables from the infinite high-dimensional feature space, whereas linear PLS extracts the major latent variables from the finite input-dimensional space.

In general, PLS and KPLS regression models tend to require a large number of latent variables to achieve an adequate model. However, such models are undesirable in terms of interpretability and robustness. Wold et al. (1998) developed orthogonal signal correction (OSC) to remove systematic variation that is unrelated, or orthogonal, to the response matrix (Y) from the predictor matrix (X). The

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

main advantage of OSC lies not in improved prediction, but rather in an improvement of the interpretability of the model by reducing the number of model latent variables required (Kim et al. 2005).

On the other hand, model-plant mismatches often exist due to low-quality or limited operational data or variations in process conditions. Thus the optimal control profile calculated from the previous mentioned regression models may not be optimal when applied to the actual processes.

Utilizing the repetitive nature of batch process operations, it is possible to improve the optimization performance. Many strategies (Lee et al. 1999, Flores-Cerrillo and MacGregor 2005) were proposed based on iterative learning control for batch to batch optimization. Another idea was introduced by Zafiriou et al. (1990) based on an analogy between the iterations during numerical optimization and successive batch runs, in which the plant data were used in the gradient calculation. However, their approach was based on a first principle model which was very difficult to obtain. Dong et al. (1996) provided the similar strategy using neural network based multiway partial least squares (NNMPLS) model in place of the first principle model.

In this paper, a nonlinear model based on multiway kernel partial least squares (MKPLS) is proposed to model the batch process. Unlike neural network model, kernel function based methods are only required to solve eigenvalue problems, not nonlinear optimization problem. Furthermore, combining the iterations during numerical optimization with successive batch runs, sequential quadratic programming (SQP) is used for batch-to-batch optimization by coupling with MKPLS model. The plant data are also used in gradient calculation to correct the control profile until there is no further change in the objective function.

2 PROCESS MODELING BASED ON MKPLS

Operational data of batch processes can be formed as a three-way matrix ($\underline{X}(I \times J \times K)$) (Kouti et al. 1995), as shown in Figure 1, where I is the number of batch runs, J is the number of variables, and K is the number of sample intervals in each batch, respectively. Figure 1 also shows the dataset structure used for modeling which includes a response matrix $Y(I \times M)$ and a batch-wise unfolded regressor matrix $X(I \times JK)$, where M is the number of end-point quality variables. MKPLS is equivalent to performing ordinary KPLS methods on the unfolded regressor matrix X and the response matrix Y (Rosipal and Trejo 2001).

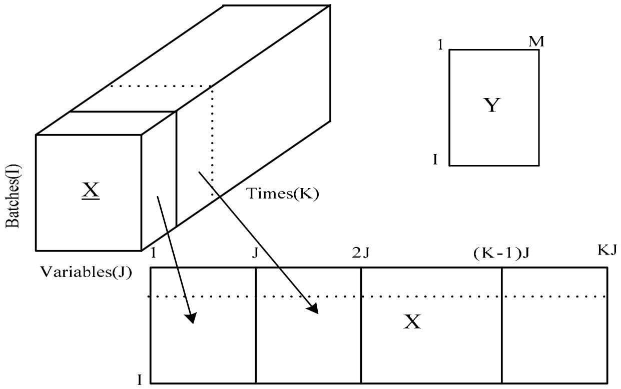

Figure 1: Data set structure for modeling a batch process

2.1 Multiway Partial Least Squares (MPLS)

MPLS can model a linear relationship between matrix $X(I \times JK)$ and $Y(I \times n_y)$ which are both scaled and centered. The non-linear iterative partial least squares (NIPALS) algorithm (Kim et al. 2005) is applied to the PLS regression in order to extract the latent vectors t, u and the weight vectors w, q from the matrices X and Y. As a result, PLS decomposes X and Y matrices as follows:

$$X = \sum_{i=1}^{A} t_i p_i^T + E = TP^T + E \qquad (1)$$

$$Y = \sum_{i=1}^{A} u_i q_i^T + F = UQ^T + F \qquad (2)$$

where $T(I \times A)$ and $U(I \times A)$ are the score matrixes, $P(JK \times A)$ and $Q(n_y \times A)$ are the loading matrixes, A represents the number of extracted principle components, B is a diagonal matrix, $B=diag\{b_1,b_2,...,b_A\}$, which represents the regression coefficient between T and U, E and F represent the residuals matrixes, respectively. The vectors t and u are normalized here.

The PLS model can be also expressed with regression coefficient C^{PLS} and the residual matrix F as follow:

$$Y = XC^{PLS} + F \qquad (3)$$

where regression coefficient C^{PLS} can be calculated as follow (Manne 1987):

$$C^{PLS} = W(P^T W)^{-1} BQ^T \qquad (4)$$

where W and Q consist the weight vectors w and q from the matrices X and Y, respectively.

Furthermore, it holds

$$p_i^T w_j = 0, \qquad i>j \tag{5}$$

$$p_i^T w_j \neq 0, \qquad i<j \tag{6}$$

So the matrix $P^T W$ is upper triangular and thus invertible. Moreover, using the fact that

$$t_i^T t_j = 0, \qquad i \neq j \tag{7}$$

$$t_i^T u_j = 0, \qquad j>i \tag{8}$$

Then the following equalities can be derived:

$$W = X^T U \tag{9}$$

$$P = X^T T(T^T T)^{-1} \tag{10}$$

$$BQ^T = Y^T T(T^T T)^{-1} \tag{11}$$

Substituting Eq(9)~(11) into Eq(4), and utilizing the orthogonality of all columns of the matrix *T*, the linear regression of MPLS model can be expressed further by coefficient C^{PLS} (Kim et al. 2005):

$$C^{PLS} = X^T U(T^T X X^T U)^{-1} T^T Y \tag{12}$$

2.2 Multiway Kernel Partial Least Squares (MKPLS)

MPLS can only effectively extract linear feature from the input and output matrix. However, most batch processes have the characteristics of nonlinearity. Thus MPLS may be inappropriate for describing the underlying data structure of batch processes. KPLS has been proposed by Rosipal and Trejo (2001) to tackle the problem of data nonliearity. In KPLS algorithm, the original input data are nonlinearly transformed into a feature space of arbitrary dimensionality, and then a linear PLS model is built in the feature space. KPLS can efficiently compute regression coefficients in high-dimensional feature spaces using nonlinear kernel functions, and it avoids nonlinear optimization by utilizing the kernel function corresponding to the inner product in the feature space. As a result, KPLS essentially requires only linear algebra, making it as simple as standard PLS (Rosipal et al. 2003).

According to Cover's theorem (Kim et al 2005), the nonlinear data structure in the input space is more likely to be linear after high-dimensional nonlinear mapping. The high-dimensional linear space is referred to as the feature space (*F*), and MKPLS is formulated in this feature space to extend linear MPLS to its nonlinear kernel form.

The nonlinear mapping $\Phi(\cdot)$ is expressed as

$$x_i \in R^{JK} \to \Phi(x_i) \in F \tag{13}$$

where $i = 1, 2, \ldots I$.

Denote $\mathbf{\Phi}$ as the *(I×S)* matrix whose *i*-th row is the vector $\Phi^T(x_i)$ in an *S*-dimensional feature space. MKPLS is equivalent to performing linear PLS between $\mathbf{\Phi}$ in high-dimensional feature space and *Y* in original output space.

In the same way, the regression coefficient C^{KPLS} in MKPLS algorithm has the form:

$$C^{KPLS} = \mathbf{\Phi}^T U(T^T K U)^{-1} T^T Y = \mathbf{\Phi}^T G \tag{14}$$

where

$$G = U(T^T K U)^{-1} T^T Y \tag{15}$$

In order to avoid performing explicit nonlinear mappings and computing dot products, the different kinds of kernel functions which satisfy Mercer's theorem are introduced. In this paper, the kernel function is selected as

$$k(x, y) = \exp(-\|x - y\|^2 / \sigma^2) \tag{16}$$

In the feature space, the kernel trick is used as follows.

$$\Phi(x_i)^T \Phi(x_j) = k(x_i, x_j) \tag{17}$$

Note that $\mathbf{\Phi\Phi}^T$ represents the *(I×I) kernel Gram matrix K* of the cross dot products between all mapped input data points $\Phi(x_i)$ (Kim et al. 2005).

When new batch run data $x_{new}^T \in R^{1\times JK}$ are obtained, the prediction of MKPLS can be made as follows:

$$\hat{y}_{new}^T = \Phi^T(x_{new}) C^{KPLS} = K_{new} G \tag{18}$$

where $K_{new} = \Phi^T(x_{new}) \mathbf{\Phi}^T \in R^{1\times I}$.

It should be noted that the prediction of MKPLS has not contained the explicit expression about the mapping function. Before applying MKPLS, mean centering of *K* in the high-dimensional space should be performed:

$$\tilde{K} = (\mathbf{I} - (1_I 1_I^T)/I) K (\mathbf{I} - (1_I 1_I^T)/I) \tag{19}$$

where $\mathbf{I}$ indicates an *I*-dimensional identity matrix and 1_I represents an *(I×1)* vector whose all elements are ones, *I* indicates the number of batch runs, respectively. Then the mean centering of K_{new} can be calculated by

$$\tilde{K}_{new} = (K_{new} - (1_I^T K)/I)(\mathbf{I} - (1_I 1_I^T)/I) \qquad (20)$$

3 BATCH TO BATCH OPTIMIZATION

The model-based optimization may not be optimal in the actual process due to the model-plant mismatches or unknown disturbances. The batch-to-batch optimization methods are proposed based on the analogy between the iterations during numerical optimization and the successive batches. The objective function of batch process optimization can be written as:

$$\min_x J = f(y) \qquad (21)$$

In general, the gradient information $g = \partial J / \partial x$ is needed to solve the optimization problem. For the MKPLS model, an analytical gradient can be calculated as follows:

$$g = \frac{\partial J}{\partial y}\frac{\partial y}{\partial K_{new}}\frac{\partial K_{new}}{\partial x} \qquad (22)$$

From Eq(18), it can be found that $\partial y / \partial K_{new} = G^T$, and $\partial K_{new} / \partial x$ can be obtained:

$$\partial K_{new} / \partial x = [\partial k_1 / \partial x, \partial k_2 / \partial x, ..., \partial k_I / \partial x] \qquad (23)$$

where *i*=1,2,…, *I*, and

$$\partial k_i / \partial x = -2(x_{new} - x_i)^T / \sigma^2 \exp(-\| x_{new} - x_i \|^2 / \sigma^2) \qquad (24)$$

The nonlinear programming technique SQP is used for optimization, in which the Hessian matrix needs to be estimated and can be updated recursively as follows:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T (s_k^T s_k) H_k}{s_k^T H_k s_k} \qquad (25)$$

where

$$s_k = x_{k+1} - x_k \qquad (26)$$

$$q_k^T = g_{k+1} - g_k = \partial J / \partial x_{k+1} - \partial J / \partial x_k \qquad (27)$$

and *k* is the number of iteration and also the index of successive batch runs.

At the starting point, H_0 can be set to any symmetric positive definite matrix, for example, the identity matrix. At each iteration *k*, the control variables trajectories x_{k+1} can be updated by:

$$x_{k+1} = x_k - H_k^{-1} g_k^T \qquad (28)$$

For model-based optimization techniques, the values of *y* in the gradient calculation are the predictions of model. So if the model is not very accurate, the optimization result may be not optimal in the actual process because of the model-plant mismatches. In order to deal with model-plant mismatches, the plant data *y* of the previous batch run are used in the gradient calculation $\partial J / \partial y$ during the batch to batch optimization. The plant data can be used to correct the gradient information as a feedback during the batch-to-batch optimization until there is no further change in the objective function.

4 APPLICATION TO A POLYMERIZATION REACTOR

In this section, the proposed method is validated through simulations on the thermally initiated bulk polymerization of styrene in a batch reactor. The differential equations describing the polymerization process are given by Kwon and Evans (1975) through reaction mechanism analysis and laboratory testing. Gattu and Zafiriou (1999)report the parameter values of the first principle model. Dong et al. (1996) also use it to demonstrate batch-to-batch optimization. The differential equations describing the polymerization process are given as follows:

$$\dot{y}_1(t) = \frac{(r_1 + r_2 T_c)^2}{M_m}(1 - y_1)^2 \exp(2y_1 + 2\chi y_1^2) \\ (\frac{1 - y_1}{r_1 + r_2 T_c} + \frac{y_1}{r_3 + r_4 T_c}) A_m \exp(-\frac{E_m}{T}) \qquad (29)$$

$$\dot{y}_2(t) = \frac{\dot{y}_1(t) y_2}{1 + y_1}(1 - \frac{1400 y_2}{A_w \exp(B/T)}) \qquad (30)$$

$$\dot{y}_3(t) = \frac{\dot{y}_1(t)}{1 + y_1}(\frac{A_w \exp(B/T)}{1500} - y_3) \qquad (31)$$

where y_1, y_2 and y_3 are, respectively, the conversion, the dimensionless number-average (NACL) and dimensionless weight-average chain lengths (WACL), the outputs are the end-point quality variables $y_1(t_f)$, $y_2(t_f)$ and $y_3(t_f)$, t_f is the final time. $u=T/T_{ref}$ is the dimensionless reactor temperature as the control variable. A_w and B are coefficients in the relation between WACL and temperature obtained from the experiments. The parameter values used in above equations are given in (Gattu and Zafiriou 1999).

The duration of a batch run is set to 300 minutes, and the whole control profile is divided into 30 equal intervals. During each interval, the temperature is kept constant. The objective is to derive an optimal temperature profile so that the conversion is maximized and values of dimensionless NACL and WCAL are close to 1.0 at the end of a batch, i.e.

$$\min_{u} J = (y_1(t_f)-1)^2 + (y_2(t_f)-1)^2 + (y_3(t_f)-1)^2 \qquad (32)$$

In this study, the above mechanical model, Eq(29)~ Eq(31), is assumed to be not available for optimization and an MKPLS model is used to build the nonlinear process. In order to generate the training datasets for building the MKPLS model, the random changes with uniform distribution and magnitude of ±15% are added to the nominal trajectory, and 50 batch runs are generated Data set of 40 batch runs are used to develop the MKPLS model, and 10 batch runs to validate the model. Then the dimension of unfolded input matrix is 40×(1×30), and the dimension of output matrix is 40×3.

After cross-validation, 5 principle components are selected, and σ is set to 10 (Kim et al. 2005). The prediction results on the validation batch runs are shown in Figure 2. It can be noted that the model is accurate but the model errors still exist.

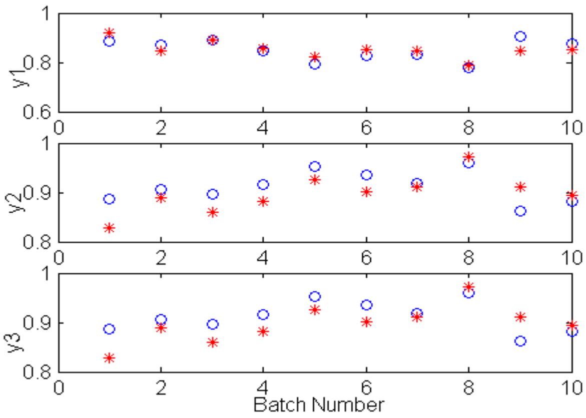

Figure 2: The prediction results of 10 test batch runs ('o' represents the plant data and '*' the predictions)

In order to further simulate the model-plant mismatches, the coefficients A_w and B are deliberately changed while the MKPLS model is kept unchanged. As shown in Table 1, there are 5 cases that the coefficients A_w and B change in different values. When there are kinetic parameter variations, the optimal control profile based on MKPLS model is not optimal when applied to the process. Batch to batch optimization coupling with the plant data in gradient calculation may achieve different control profiles with the change of actual plant parameters. In 5 cases, the end-point quality and objective index based on MKPLS model are compared with those of batch-to-batch correction based plant data, as shown in Table 1. It can be seen that the batch to batch optimization can obtain better optimization performance despite the model-plant mismatches.

Figure 3 shows the changes of objective index in batch-to-batch correction via batch run in Case 1. It can also be found that the optimization under batch-to-batch correction converges after 5 iterations.

Figure 4 shows the optimal control profiles of both the MKPLS model and batch-to-batch correction coupling with plant data in Case 1.

Table 1. Comparison of the optimal results

Case no.	Mis-match	MKPLS model		Couple with the plant data	
		End-point quality	Objective index	End-point quality	Objective index
Case 1:	A_w=A_w ; B=B	y1=0.9995 y2=0.8459 y3=0.8442	0.04802	y1=0.9999 y2=1.0108 y3=0.9794	0.00054
Case 2:	A_w= $1.1A_w$; B=B	y1=0.9995 y2=1.2344 y3=1.2571	0.12104	y1=0.9998 y2=1.0107 y3=0.9791	0.00055
Case 3:	A_w= $1.5A_w$; B=B	y1=0.9995 y2=1.1889 y3=1.1883	0.07114	y1=0.9997 y2=1.0107 y3=0.9789	0.00056
Case 4:	A_w= $0.5A_w$; B=B	y1=0.9995 y2=1.6302 y3=2.5742	2.87526	y1=0.9987 y2=0.9832 y3=1.0337	0.00142
Case 5:	A_w=1; B= $1.1B$	y1=0.9995 y2=1.3746 y3=1.5323	0.42367	y1=1.0000 y2=1.0115 y3=0.9790	0.00057

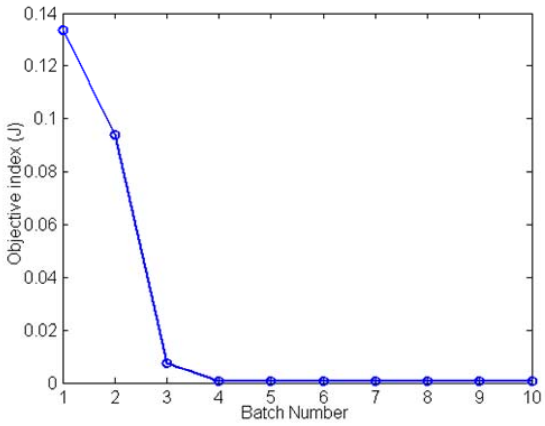

Figure 3: Objective index in batch to batch correction versus batch number in Case 1

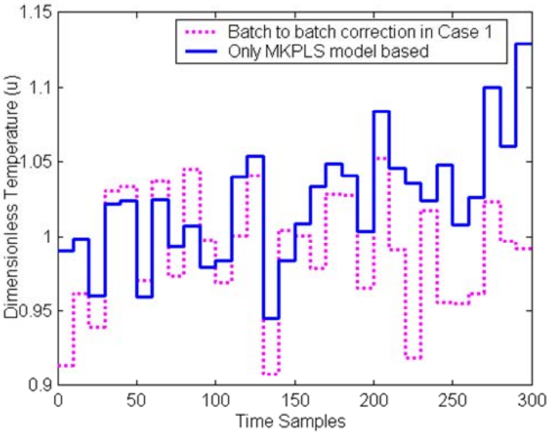

Figure 4: The optimal control profiles for both the MKPLS model based and batch to batch correction coupling with plant data in Case 1

5 CONCLUSIONS

A nonlinear MKPLS model is proposed to model the batch process. Due to the model-plant mismatches, the model-based optimal control policy may not be optimal when used in actual process. A batch to batch optimization strategy is provided to correct the control profile. The framework of SQP is used for batch to batch optimization coupling with MKPLS model, in which the plant data in place of MKPLS model predictions are used in the gradient calculation. Application to a simulated batch polymerization process demonstrates that the proposed method can obtain good optimal results despite the model-plant mismatches.

ACKNOWLEDGMENTS

The authors greatly appreciate the financial support of National Natural Science Foundation of China (No. 60404012), National High-tech 863 Program of China (No. 2007AA04Z193), and New Star of Science and Technology of Beijing City, China.

REFERENCES

Bonvin, D. 1998. Optimal operation of batch reactors - A personal view. *Journal of Process Control* 8: 355-368

Dong, D., T. J. McAvoy, E. Zafiriou. 1996. Batch-to-batch optimization using neural network models. *Ind. Eng. Chem. Res.* 35: 2269-2276

Flores-Cerrillo, J., J. H. MacGregor. 2005. Iterative learning control for final batch product quality using partial least squares models. *Ind. Eng. Chem. Res.* 44: 9146- 9155

Gattu, G., E. Zafiriou. 1999. A methodology for on-line setpoint modification for batch reactor control in the presence of modeling error. *Chemical Engineering Journal* 75: 21-29

Kim, K., J. Lee, I. Lee. 2005. A novel multivariate regression approach based on kernel partial least squares with orthogonal signal correction. *Chemometrics and Intelligent laboratory Systems*. 79: 22-30

Kouti, T., P. Nomikos, J. H. MacGregor. 1995. Analysis, Monitoring and diagnosis of batch process using multiblock and multiway partial least square. *Journal of Process Control*. 5: 277-284

Kwon, Y. D., Evans, L. B. 1975. A coordinate transformation method for the numerical solution of nonlinear minimum-time control problems. *AIChE J.* 21: 1158-1164.

Lee, K. S.; S. I. Chin, J. H. Lee. 1999. Model Predictive control technique combined with iterative learning for batch processes. *AIChE Journal*. 45: 2175-2187

Manne, R. 1987. Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration. *Chemometrics and Intelligent Laboratory Systems*. 2: 187-197

Qin, S. J., T.J. McAvoy, 1992. Nonlinear PLS modeling using neural networks, *Computers and Chemical Engineering*, 16: 379-391.

Rosipal, R.; L. J. Trejo. 2001. Kernel partial least squares regression in reproducing kernel Hilbert space. *Journal of Machine Learning Research*. 2: 97-123

Rosipal, R., L.J. Trejo, B. Matthews. 2003. Kernel PLS-SVC for linear and nonlinear classification, *Proceedings of the Twentieth International Conference on Machine Learning* (ICML-2003), Washington DC, 640-647.

Wold, S., N. Kettaneh-Wold, B. Skagerberg. 1989. Nonlinear PLS modeling, *Chemometrics and Intelligent Laboratory Systems*, 7: 53-65.

Wold, S., H. Antti, F. Lindgren, J. Ohman. 1998. Orthogonal signal correction of near-infrared spectra, *Chemometrics and Intelligent Laboratory Systems*, 44: 175-185.

Xiong, Z. H., J. Zhang. 2005. A batch-to-batch iterative optimal control strategy based on recurrent neural network models. *Journal of Process Control*. 15: 11-21

Zafiriou, E., J. M. Zhu. 1990. Optimal control of semibatch processes in the presence of modeling error. *Proceedings of 1990 American Control Conference*, San Diego, CA, 1644-1649.

AUTHOR BIOGRAPHIES

LIQING DI received her PhD from Tsinghua University, Beijing, China in 2008. She currently is an senior engineer in the Research and Development of Hollysys Co. Ltd., Beijing, China. Her research interests include process modeling and optimization, statistical process monitoring.

ZHIHUA XIONG is an Associate Professor in the Department of Automation at Tsinghua University, Beijing, China. He received his PhD from Tsinghua University in 2000. Dr. Xiong has published over 50 refereed journal articles and conference papers. His research and teaching interests include complex systems modeling, control, and optimization. He is a member of IEEE.

XIANHUI YANG is a Professor in the Department of Automation at Tsinghua University, Beijing, China. Her current research interests are in statistical process monitoring, functional safety modeling and analysis.