Chinese J. Chem. Eng., **14**(6) 754—758 (2006)

On-line Fault Detection Using SVM-based Dynamic MPLS for Batch Processes*

LI Yunfeng(李运锋)[a], **WANG Zhifeng**(汪志锋)[b] **and YUAN Jingqi**(袁景淇)[a, c,**]
[a] Department of Automation, Shanghai Jiao Tong University, Shanghai 200030, China
[b] Department of Automation, Shanghai Second Polytechnic University, Shanghai 201209, China
[c] State Key Laboratory of Bioreactor Engineering, East China University of Science and Technology, Shanghai 200237, China

Abstract In this article, a nonlinear dynamic multiway partial least squares (MPLS) based on support vector machines (SVM) is developed for on-line fault detection in batch processes. The approach, referred to as SVM-based DMPLS, integrates the SVM with the MPLS model. Process data from normal historical batches are used to develop the MPLS model, and a series of single-input-single-output SVM networks are adopted to approximate nonlinear inner relationship between input and output variables. In addition, the application of a time-lagged window technique not only makes the complementarities of unmeasured data of the monitored batch unnecessary, but also significantly reduces the computation and storage requirements in comparison with the traditional MPLS. The proposed approach is validated by a simulation study of on-line fault detection for a fed-batch penicillin production.
Keywords fault detection, multiway partial least squares, support vector machines, time-lagged window

1 INTRODUCTION

In batch or fed-batch processes, raw materials are converted to products within a finite duration. In practical production, the process commonly exhibits large variations from batch to batch due to such influencing factors as the quality fluctuation of raw materials, defect of equipments, contaminations, and other unpredicted disturbances. These variations may have an adverse effect on the final product quantity and quality. But it is generally difficult to discern these variations for the operators. In this case, early detection of fault condition appears especially important. In fact, the process monitoring and diagnosis has been playing an important role in high-quality, safe, and efficient production.

During the last decade, several data-driven techniques have been developed for batch process monitoring and diagnosis[1,2]. Multivariate statistic techniques such as partial least squares (PLS) are particularly focused[3—6]. For standard linear PLS, the inner relationship between the process variables and the final quality variables is modeled using a linear function. For highly nonlinear batch processes, however, this linear approximation is sometimes unsatisfactory. Nonlinear partial least squares (NPLS) models are more appropriate than linear models to study the behavior of the process. Wold *et al*. proposed nonlinear PLS using the quadratic or spline functions to model the inner relation[7]. Unfortunately, the algorithm of Wold's NPLS is available only for pseudo-linear functions, such as quadratic and spline functions. As far as other nonlinear relationships, such as logarithmic, exponential, reciprocal functions, are concerned, the above NPLS cannot be carried out. Another approach to nonlinear PLS modeling is to use artificial neural networks instead of the linear inner relationship (NNPLS)[8]. In theory, ANN is able to approximate any nonlinear function. In practical application, however, there exist some disadvantages such as the local bad minima, the topological structure of ANN, and over-training phenomenon. Moreover, for multiway partial least squares (MPLS) model, which is an extension of PLS to handle three-way data matrix in batch processes, the number of training samples of ANN is the number of reference batches, which is commonly limited (the number of 20—60). It is difficult for ANN to exhibit good performance with such small training samples.

The support vector machines (SVM) is a novel machine learning method based on statistical learning theory (SLT)—a small-sample statistical theory, and thereby it is suitable for solving the problems arising due to small samples, nonlinearity, high dimension, and local minima[9]. Due to high accuracy and good generalization, SVM has been applied in machine learning, regression and prediction, classification task, and pattern recognition[10,11].

In this article, a dynamic MPLS based on SVM (SVM-based DMPLS) is developed for on-line fault detection in batch processes. The SVM-based DMPLS enables the MPLS linear outer projection to calculate the latent variables and builds nonlinear inner models

Received 2005-08-09, accepted 2006-06-09.
* Supported by the National Natural Science Foundation of China (No.60574038) and the Open Project Program of the State Key Laboratory of Bioreactor Engineering/ECUST.
** To whom correspondence should be addressed. E-mail: jqyuan@sjtu.edu.cn

using SVM. The inner models are a series of single-input-single-output SVM predictor, which are trained on the latent variables generated from raw data. Therefore, SVM-based DMPLS not only handles correlated variables, but also builds a nonlinear relationship through inner modeling. In addition, the application of the time-lagged window technique not only makes the complementarities of the unmeasured data of monitored batch unnecessary, but also is helpful for tracking the dynamic process data. Besides, the computation and storage requirements are significantly reduced in comparison with the traditional MPLS, because the SVM-based DMPLS only requires the previous partial measured data.

2 METHODS

2.1 Support vector machines nonlinear regression

For a given data set $S=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i$ is the input vector, y_i the desired value, and l the total number of data pair, support vector machines' nonlinear regression maps the original data x onto a high-dimensional feature space via a nonlinear mapping function $\varphi(\cdot)$ and then performs a linear regression in the feature space[9]. The linear regression function in the feature space is defined as:

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x})+b \tag{1}$$

where $\varphi(\boldsymbol{x})$ is the mapped data in the feature space; both $\boldsymbol{w}$ and b are coefficients that will be estimated by the training data. The linear ε-insensitive loss function is chosen:

$$L^{\varepsilon}\left(\boldsymbol{x}_i,y_i,f\right)=\begin{cases}0, & \text{for}\left|f\left(\boldsymbol{x}_i\right)-y_i\right|<\varepsilon\\ \left|f\left(\boldsymbol{x}_i\right)-y_i\right|-\varepsilon, & \text{otherwise}\end{cases} \tag{2}$$

The optimal regression function can be obtained by minimizing the following objective function:

$$\boldsymbol{Q}=\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}L^{\varepsilon}(\boldsymbol{x}_i,y_i,f)$$

$$\text{subject to}\begin{cases}y_i-\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}_i)-b\leqslant\varepsilon+\xi_i\\ \boldsymbol{w}^{T}\varphi(\boldsymbol{x}_i)+b-y_i\leqslant\varepsilon+\xi_i^{*}\\ \xi,\xi_i^{*}\geqslant 0\end{cases} \tag{3}$$

The first term in Eq.(3) is a weight decay, which is used to regularize weight sizes and penalize large weights. The second term is the empirical error (risk), which is scaled by the ε-insensitive loss function $L^{\varepsilon}(\cdot)$ in Eq.(2). ε is called the tube size of the SVM. Errors greater than $\pm\varepsilon$ are denoted with the slack variables ξ (above ε) and ξ^{*} (below ε), respectively. Parameter C is the regularization constant determining the compromise between the empirical error and the regularized term. Both C and ε need to be chosen empirically.

By introducing Lagrange multipliers and kernel function, the optimal regression function [see Eq.(1)] is obtained in the following explicit form:

$$f(\boldsymbol{x})=\sum_{i=1}^{l}\left(\alpha_i-\alpha_i^{*}\right)K\left(x_i,x\right)+b \tag{4}$$

where α_i and α_i^{*} are the Lagrange multipliers satisfying the expression $\alpha_i\alpha_i^{*}=0$, $\alpha_i\geqslant 0$, $\alpha_i^{*}\geqslant 0$. Based on the Karush-Kuhn-Tucker (KKT) conditions[9], only a certain number of coefficients $\left(\alpha_i-\alpha_i^{*}\right)$ are non-zero. The data pairs corresponding to these non-zero coefficients are named support vectors. $K(\boldsymbol{x}_i,\boldsymbol{x})$ is the kernel function used to avoid the computation of the nonlinear mapping.

2.2 Unfolding of three-way data matrix and time-lagged window technique

The historical data collected from the batch and fed-batch processes are organized into two matrices: the process data matrix $\underline{\boldsymbol{X}}$ ($I\times J\times K$) and the final-product quality data matrix $\boldsymbol{Y}$($I\times M$), where I, J, K and M are the numbers of reference batches, process variables, samples, and final-product quality variables respectively. Before performing PLS, the three-way matrix $\underline{\boldsymbol{X}}$ ($I\times J\times K$) needs be unfolded into a two-way matrix $\boldsymbol{X}$ ($I\times KJ$) by preserving the batch direction. Each of the vertical slices of $\underline{\boldsymbol{X}}$ is a ($I\times J$) matrix representing the values of J process variables for I reference batches at a sampling time. Then, the unfolding matrix $\boldsymbol{X}$ is represented in the following form:

$$\boldsymbol{X}^{1\sim I}=\begin{bmatrix}\overbrace{x_1\ x_2\ \cdots\ x_J}^{i=1,k=1} & \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=1,k=2} & \cdots\\ \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=2,k=1} & \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=2,k=2} & \cdots\\ \vdots & \vdots & \ddots\\ \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=I,k=1} & \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=I,k=2} & \cdots\end{bmatrix}\begin{matrix}\overbrace{x_1\ x_2\ \cdots\ x_J}^{i=1,k=K}\\ \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=2,k=K}\\ \vdots\\ \overbrace{x_1\ x_2\ \cdots\ x_J}^{i=I,k=K}\end{matrix}=\begin{bmatrix}f_1^1 & f_2^1 & \cdots & f_K^1\\ f_1^2 & f_2^2 & \cdots & f_K^2\\ \vdots & \vdots & \ddots & \vdots\\ f_1^I & f_2^I & \cdots & f_K^I\end{bmatrix} \tag{5}$$

For on-line process monitoring of a new batch, traditional MPLS requires complete history of the new batch run. The unmeasured data from current time to the end of batch have to be filled with the estimated data. Although several different methods are proposed to estimate the missing data[3,12,13], the estimation values may not exactly follow the actual dynamic process behavior, and this may lead to false detection. This especially easily occurs during the initial phase of cul-

tivation due to large amounts of missing data. Moreover, the size of the unfolding matrix $\boldsymbol{X}$ in Eq.(5) is usually rather large, as a result of which considerable effort is required in computation. In this article, only partial measured process data of new batch are used in SVM-based DMPLS modeling by applying the time-lagged window technique[14,15]. Suppose the current sampling time of new batch (the monitored batch, numbered I+1) is k, then its time-lagged matrix $\boldsymbol{X}_k^{I+1}$ and the corresponding $\boldsymbol{X}_k^{1\sim I}$ of reference batches are:

$$\boldsymbol{X}_k^{I+1}=\begin{bmatrix} f_{k-d+1}^{I+1} & f_{k-d+2}^{I+1} & \cdots & f_k^{I+1}\end{bmatrix} \tag{6}$$

$$\boldsymbol{X}_k^{1\sim I}=\begin{bmatrix} f_{k-d+1}^{1} & f_{k-d+2}^{1} & \cdots & f_k^{1} \\ f_{k-d+1}^{2} & f_{k-d+2}^{2} & \cdots & f_k^{2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{k-d+1}^{I} & f_{k-d+2}^{I} & \cdots & f_k^{I}\end{bmatrix} \tag{7}$$

where superscripts I+1 and 1~I represent the (I+1)th batch and I reference batches in total, and d is the number of time lags.

2.3 Development of SVM-based DMPLS

Standard PLS model decomposes the matrices $\boldsymbol{X}_k^{1\sim I}$ and $\boldsymbol{Y}_k^{1\sim I}$ at time k into two linear combination of scores matrices $\boldsymbol{T}(I\times R)$ and $\boldsymbol{U}(I\times R)$, loading matrices $\boldsymbol{P}(dJ\times R)$ and $\boldsymbol{Q}(M\times R)$, and residual matrices $\boldsymbol{E}(I\times dJ)$ and $\boldsymbol{F}(I\times M)$.

$$\boldsymbol{X}_k^{1\sim I}=\sum_{r=1}^{R}\boldsymbol{t}_r\boldsymbol{p}_r+\boldsymbol{E}=\boldsymbol{T}\boldsymbol{P}^{\mathrm{T}}+\boldsymbol{E} \tag{8}$$

$$\boldsymbol{Y}_k^{1\sim I}=\sum_{r=1}^{R}\boldsymbol{u}_r\boldsymbol{q}_r+\boldsymbol{F}=\boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}}+\boldsymbol{F} \tag{9}$$

where vectors $\boldsymbol{t}_r$ and $\boldsymbol{u}_r$ represent the rth latent variables retained in the model, R is the number of latent variables, the final-product quality data matrix $\boldsymbol{Y}_k^{1\sim I}$ at any time k is invariable, namely, $\boldsymbol{Y}_k^{1\sim I}=\boldsymbol{Y}^{1\sim I}$. For linear PLS, the inner relationship between latent variable matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ is modeled using a linear function, *i.e.* $\boldsymbol{U}=\boldsymbol{T}\boldsymbol{B}$, where matrix $\boldsymbol{B}(R\times R)$ is a diagonal matrix. The SVM-based DMPLS proposed in this article is to replace the linear inner relationship with SVM networks, while maintaining the linear outer relationship in Eqs.(8) and (9) unchanged. A series of single-input–single-output SVM networks are used to model the relationship between latent variables $\boldsymbol{T}$ and $\boldsymbol{U}$. That is,

$$\boldsymbol{u}_r=S_r(\boldsymbol{t}_r)+\boldsymbol{h}_r \tag{10}$$

where $S_r(\cdot)$ stands for the rth inner relationship presented by a SVM network, and $\boldsymbol{h}_r$ is the residual. Standard SVM regression function in Eq.(4) is rewritten as follows:

$$\hat{u}_r=\sum_{i=1}^{I}\left(\alpha_i-\alpha_i^*\right)K\left(t_{r,i},\boldsymbol{t}_r\right)+b \tag{11}$$

where $K(\cdot)$ is the kernel function, I is the number of reference batches in total, and the regression coefficients $(\alpha_i-\alpha_i^*)$ and the constant b are determined by SVM regression algorithm[9]. Establishment of SVM-based DMPLS model is based on NIPALS algorithm proposed by Wold[16]. The detailed procedure is given in Appendix A.

During the progress of the monitored batch, a time-lagged matrix $\boldsymbol{X}_k^{I+1}$ in Eq.(6) becomes available at each sampling time k. After Scaling each column of $\boldsymbol{X}_k^{I+1}$ to zero-mean and unit-variance, the predicted scores matrix $\boldsymbol{t}_k^{I+1}(1\times R)$ and the residuals matrix $\boldsymbol{e}_k^{I+1}(1\times dJ)$ are given by

$$\boldsymbol{t}_k^{I+1}=\boldsymbol{X}_k^{I+1}\boldsymbol{W}\left(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}\right)^{-1} \tag{12}$$

$$\boldsymbol{e}_k^{I+1}=\boldsymbol{X}_k^{I+1}-\boldsymbol{t}_k^{I+1}\boldsymbol{P}^{\mathrm{T}} \tag{13}$$

where $\boldsymbol{W}(dJ\times R)$ is a weight matrix, which is composed of the weight vector $\boldsymbol{w}_r$ produced by NIPALS algorithm.

3 CASE STUDY

Fed-batch penicillin fermentation process is taken as a case study. Process data are generated from a simulator named Pensim, which is based on a detailed mathematical model describing penicillin fed-batch cultivation. This simulator can be used to develop statistical monitoring/control approaches as a tool[17,18]. A total of 14 variables including five input variables (Nos.1—4 and 14) and nine process variables (Nos.5—13) consist of the process data matrix $\boldsymbol{X}$, while two final quality variables construct the matrix $\boldsymbol{Y}$, as shown in Table 1. The duration of each batch run is 250h, and the sampling interval is chosen to be 1h. Moreover, measurement noises are also added to all the variables used in monitoring. Thirty normal batches are selected as reference batches to construct MPLS model. The size of matrices $\overline{\boldsymbol{X}}$ and $\boldsymbol{Y}$ are (30×14×250) and (30×2), respectively.

Table 1 Variables used in the monitoring

No.	Variables
1	aeration rate, h^{-1}
2	agitator power, W
3	substrate feed flow rate, h^{-1}
4	substrate feed temperature, K
5	substrate concentration, $g\cdot L^{-1}$
6	dissolved oxygen concentration, $mmol\cdot L^{-1}$
7	biomass concentration, $g\cdot L^{-1}$
8	penicillin concentration, $g\cdot L^{-1}$
9	culture volume, L
10	carbon dioxide concentration, $mmol\cdot L^{-1}$
11	pH
12	bioreactor temperature, K
13	generated heat, J
14	cooling water flow rate, h^{-1}
y_1	final penicillin concentration, $g\cdot L^{-1}$
y_2	amount of penicillin produced (computed), g

For on-line monitoring, the number of time lags d is set to 50 by trial and error. The size of unfolding matrix $\boldsymbol{X}_k^{1\sim I}$ is (30×700) when $k \geqslant 50$. When $k < 50$, the unfolding matrix $\boldsymbol{X}_k^{1\sim I}$ is not constructed due to the time lags. In this case, all the data up to time k are used to construct MPLS model, namely, $\boldsymbol{X}_k^{1\sim I}(30\times 14k)$. For SVM networks, sigmoid function is selected as the kernel. By cross-validation, six latent variables are found to be sufficient for describing the relationship between $\boldsymbol{X}_k^{1\sim I}$ and $\boldsymbol{Y}_k^{1\sim I}$, which is able to explain about 80% of the variation of $\boldsymbol{Y}_k^{1\sim I}$.

Two testing batches are used to evaluate the monitoring performance. Batch 31 is a normal batch. For Batch 32, the substrate feed rate is abnormal from 150 due to the fault of a feeding pump. It is linearly decreased from 0.0431h^{-1} at 150h to 0.0318h^{-1} at the end of batch operation.

Figures 1 and 2 show T^2-chart and *SPE*-chart of the monitored batches, respectively. For normal Batch 31, the T^2 and *SPE* values, respectively, remain below their control limits during the whole batch run. For abnormal Batch 32, the fault can be detected successfully and quickly, where *SPE* values of Batch 32 exceed the 99% control limits at about 178h.

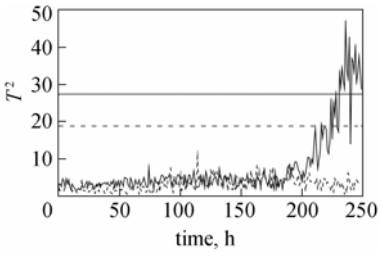

Figure 1 T^2-chart of two testing batches
– – – Batch 31; —— Batch 32;
– – – 95% control limit; —— 99% control limit

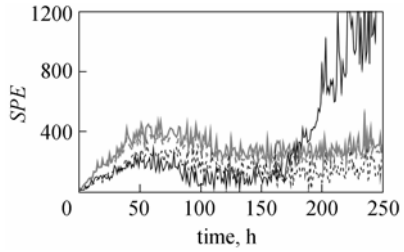

Figure 2 *SPE*-chart of two testing batches
– – – Batch 31; —— Batch 32;
– – – 95% control limit; —— 99% control limit

4 CONCLUSIONS

This article proposed a new nonlinear dynamic MPLS model based on SVM (SVM-based DMPLS) for on-line batch fault detection. By introducing SVM into traditional PLS model, highly nonlinear dynamic batch processes are modeled effectively. In addition, the time-lagged window technique is adopted to avoid complementing unmeasured data of the monitored batch. At the same time, the computation and storage requirements are reduced significantly in comparison with traditional MPLS. The application of SVM-based DMPLS to penicillin production demonstrates that it provides an effective approach for fault detection in batch or fed-batch processes.

NOMENCLATURE

$\boldsymbol{B}$	diagonal matrix representing the inner relation between $\boldsymbol{T}$ and $\boldsymbol{U}$
b	scalar threshold value
C	penalty factor
d	number of time lags
$\boldsymbol{E}$, $\boldsymbol{F}$	residual matrices
I	number of reference batches
J	number of process variables
K	number of samples
$K(x_i, x)$	kernel function
k	current sampling time of present batch
$L^{\varepsilon}(\cdot)$	ε-insensitive loss function in SVM
M	number of final-product quality variables
$\boldsymbol{P}$, $\boldsymbol{Q}$	loading matrices
$\boldsymbol{p}_r$, $\boldsymbol{q}_r$	the rth loading vector
R	number of latent variables
$S_r(\cdot)$	standing for the inner relationship between $\boldsymbol{t}_r$ and $\boldsymbol{u}_r$ using a SVM network
$\boldsymbol{T}$, $\boldsymbol{U}$	scores matrices
$\boldsymbol{t}_r$, $\boldsymbol{u}_r$	the rth score vector
$\boldsymbol{W}$	weight matrix consisting of weight vector
$\boldsymbol{w}$	weight vector in SVM
$\boldsymbol{w}_r$	produced by NIPALS algorithm
$\boldsymbol{X}$	two-way process data matrix
$\bar{\boldsymbol{X}}$	three-way process data matrix
$\boldsymbol{X}_k^{I+1}$	time-lagged matrix of the (I+1)th batch at time k
$\boldsymbol{X}_k^{1\sim I}$	$\boldsymbol{X}$ time-lagged matrix for I reference batches at time k
$\boldsymbol{Y}$	two-way product quality data matrix
$\boldsymbol{Y}_k^{1\sim I}$	$\boldsymbol{Y}$ time-lagged matrix for I reference batches at time k
α_i , α_i^*	Lagrange multipliers
ε	regression precision
ξ , ξ^*	slack variables
$\varphi(\cdot)$	nonlinear mapping function in SVM

REFERENCES

1 Golobic, I., Gjerkes, H., Bajsic, I., Malensek, J., "Software sensor for biomass concentration monitoring during industrial fermentation", *Instrum. Sci. Technol.*, **28**(4), 323—334(2000).
2 Li, Y., Zhou, D.H., Xie, Z., Qin, S.J., "Faulty sensor detection and reconstruction for a PVC making process", *Chin. J. Chem. Eng.*, **12**(2), 227—233(2004).
3 Nomikos, P., MacGregor, J. F., "Monitoring batch processes using multiway principal component analysis", *AIChE J.*, **40**, 1361—1375(1994).
4 Song, K., Wang, H.Q., Li, P., "VPLS based quality and cost control for Tennessee Eastman process", *Chin. J. Chem. Eng.*, **13**(1), 62—67(2005).
5 Gregersen, L., Jorgensen, S.B., "Supervision of fed-batch

fermentations", *Chem. Eng. J.*, **75**(1), 69—76(1999).
6 Undey, C., Tatara, E., Cinar, A., "Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations", *J. Biotechnol.*, **108**(1), 61—77(2004).
7 Wold, S., Kettaneh-Wold, N., Skagerberg, B., "Nonlinear PLS modeling", *Chemom. Intell. Lab. Syst.*, **7**, 53—65(1989).
8 Qin, S.J., McAvoy, T.J., "Nonlinear PLS modeling using neural networks", *Comput. Chem. Eng.*, **16**(4), 379—391 (1992).
9 Vapnik, V.N., The Nature of Statistical Learning Theory, Springer, Berlin (1995).
10 Cai, C.Z., Wang, W.L., Sun, L.Z., Chen, Y.Z., "Protein function classification via support vector machine approach", *Math. Biosci.*, **185**(2), 111—122(2003).
11 Li, Y.F., Yuan, J.Q., "Prediction of key state variables using support vector machines in bioprocesses", *Chem. Eng. Technol.*, **29**(3), 313—319(2006).
12 Cho, H., Kim, K., "A method for predicting future observations in the monitoring of a batch process", *J. Qual. Technol.*, **57**, 63—75(2003).
13 Meng, X., Morris, A. J., Martin, E. B. "On-line monitoring of batch processes using a PARAFAC representation", *J. Chemometr.*, **17**, 65—81(2003).
14 Chen, J.H., Liu, K.C., "On-line batch process monitoring using dynamic PCA and dynamic PLS models", *Chem. Eng. Sci.*, **57**(1), 63—75(2002).
15 Ku, W.F., Storer, R.H., Gerorgakis, C., "Disturbance detection and isolation by dynamic principal component analysis", *Chemometr. Intell. Lab.*, **30**(1), 179—196(1995).
16 Wold, H., "Nonlinear estimation by iterative least squares procedures", In: Research Papers in Statistics, David, F., Ed., Wiley, New York (1996).
17 Birol, G., Cerk, U., Ali, C., "A modular simulation package for fed-batch fermentation penicillin production", *Comput. Chem. Eng.*, **26**, 1553—1565(2002).
18 Cenk, Ü., Eric, T., Ali, C., "Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations", *J. Biotechnol.*, **108**(1), 61—77(2004).

APPENDIX

SVM-based DMPLS modeling procedure

1 Scale the data to zero-mean and unit-variance.
2 Let $\boldsymbol{E}_0=\boldsymbol{X}$, $\boldsymbol{F}_0=\boldsymbol{Y}$ and $r = 1$.
3 For each latent variable r, take $\boldsymbol{u}_r$ = some column of $\boldsymbol{Y}$.
4 PLS outer transform:

$$\boldsymbol{w}_r = \boldsymbol{E}_{r-1}^{\mathrm{T}}\boldsymbol{u}_r \big/ \left(\boldsymbol{u}_r^{\mathrm{T}}\boldsymbol{u}_r\right)$$

normalize $\boldsymbol{w}_r$ to norm 1: $\boldsymbol{w}_r = \boldsymbol{w}_r \big/ \left\|\boldsymbol{w}_r\right\|$

$$\boldsymbol{t}_r = \boldsymbol{E}_{r-1}\boldsymbol{w}_r$$

$$\boldsymbol{q}_r^{\mathrm{T}} = \boldsymbol{t}_r^{\mathrm{T}}\boldsymbol{F}_{r-1} \big/ \left(\boldsymbol{t}_r^{\mathrm{T}}\boldsymbol{t}_r\right)$$

normalize $\boldsymbol{q}_r$ to norm 1: $\boldsymbol{q}_r = \boldsymbol{q}_r \big/ \left\|\boldsymbol{q}_r\right\|$

$$\boldsymbol{u}_r = \boldsymbol{F}_{r-1}\boldsymbol{q}_r$$

Iterate this step until it converges.
5 Calculate the $\boldsymbol{X}$ loadings and rescale $\boldsymbol{t}_r$ and $\boldsymbol{w}_r$:

$$\boldsymbol{p}_r^{\mathrm{T}} = \boldsymbol{t}_r^{\mathrm{T}}\,\boldsymbol{E}_{r-1} \big/ \left(\boldsymbol{t}_r^{\mathrm{T}}\boldsymbol{t}_r\right)$$

$$\boldsymbol{t}_r = \boldsymbol{t}_r \left\|\boldsymbol{p}_r\right\|$$

$$\boldsymbol{w}_r = \boldsymbol{w}_r \left\|\boldsymbol{p}_r\right\|$$

normalize $\boldsymbol{p}_r$ to norm 1: $\boldsymbol{p}_r = \boldsymbol{p}_r \big/ \left\|\boldsymbol{p}_r\right\|$

6 Inner SVM training
Build a single-input–single-output SVM network to learn the inner relationship between $\boldsymbol{u}_r$ and $\boldsymbol{t}_r$:

$$\boldsymbol{u}_r = S_r(\boldsymbol{t}_r) + \boldsymbol{h}_r$$

7 Calculate the residuals:
for matrix $\boldsymbol{X}$, $\boldsymbol{E}_r = \boldsymbol{E}_{r-1} - \boldsymbol{t}_r\boldsymbol{p}_r^{\mathrm{T}}$
for matrix $\boldsymbol{Y}$, $\boldsymbol{F}_r = \boldsymbol{F}_{r-1} - \hat{\boldsymbol{u}}_r\boldsymbol{q}_r^{\mathrm{T}}$
where $\hat{\boldsymbol{u}}_r = S_r\left(\boldsymbol{t}_r\right)$
8 Determine the number of latent variables:
If $r<R$, then $r=r+1$ and return to Step 2 to calculate the next latent variables, where R is the number of latent variables and determined by cross-validation. Otherwise, terminate the modeling algorithm.