Regularization-Based Statistical Batch Process Modeling for Final Product Quality Prediction

Zhengbing Yan, Chih-Chiun Chiu, and Yuan Yao
Dept. of Chemical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

Weiwei Doing
Division of Advanced Manufacturing and Automation, Guangzhou HKUST Fok Ying Tung Research Institute, Nansha, Guangzhou 511458, China

DOI 10.1002/aic.14476
Published online April 30, 2014 in Wiley Online Library (wileyonlinelibrary.com)

Prediction accuracy and model interpretation are two important aspects with regard to regression models. In the field of statistical modeling of chemical batch processes, most research focuses on prediction accuracy, while the importance of the latter aspect is often overlooked. In multiphase batch processes, it is possible that only a few phases are relevant to certain quality indices, while different time points belonging to the same relevant phase usually have similar contribution to the quality. The regression coefficients of batch process model should reflect such process characteristics, that is, the coefficients corresponding to the irrelevant phases should be close to zero, while the coefficients of each variable within the same phase should vary smoothly. In this study, regularization techniques are introduced to statistical modeling of chemical batch processes to achieve both accurate prediction and good interpretation. The application to an injection molding process shows the feasibility of the proposed methods. © 2014 American Institute of Chemical Engineers *AIChE J*, 60: 2815–2827, 2014
Keywords: batch process, quality prediction, regularization, model interpretation, lasso

Introduction

Batch processes have been widely applied in modern chemical industry to produce high value-added products and meet the demands of the rapidly changing market. For achieving consistent and high final product quality, quality-related research of batch processes has become an active field. In most cases, measurement of final product quality is unavailable during batch processing and can only be achieved through laboratory analysis after the completion of the entire batch. Hence, great research attention has been devoted to batch process modeling and online quality prediction based on multivariate statistical methods,[1] because such type of models can be derived directly from historical process data with little prior process knowledge.

Among all these methods, multiway partial least squares (MPLS)[2] is most famous, with good applications in chemical industry, for example.[3] By treating the entire batch data as a single object in batch process modeling, MPLS utilizes all of the information contained in process variable trajectories during operation. However, many batch processes consist of multiple operation phases, implying that the process variables may make different contributions to the final product quality at different phases in a batch run. MPLS does not consider such multiphase characteristic explicitly. Furthermore, MPLS involves all process variables and sampling intervals into the model, no matter they are critical to the final product quality or not. Considering that the multiplicity of phases is an inherent nature of many batch processes, multiphase models have been developed to capture the changing process behaviors over phases.[4] Specifically, Lu and Gao[5] proposed a phase-based PLS modeling method for online prediction of final product quality. In their method, all measurements within the same phase share the same regression model, that is, the phase model. By measuring the goodness-of-fit of each phase model, the critical-to-quality phases can be identified. Only if the process measurements are in such phases, they are utilized in quality prediction. Comparing to MPLS, a major drawback of the phase-based PLS method is that all phase models are based on the relationship between the time-slice process data and the quality variable, while the within-phase and between-phase time-dependence information is missing in phase model building. There are several revised versions of phase-based PLS following the pioneering research of Lu and Gao, for example.[6,7] However, these methods share the same limitation with the method in.[5] In addition to these two types of modeling techniques, multiblock methods[8] can also be used to model batch processes. However, such methods do not provide any solutions to phase division. In the meantime, although multiblock methods can assign different weights to different blocks for achieving models better conforming to process knowledge, they do not tell how to calculate the weights. Most recently, a multiway elastic net (MEN) method was proposed,[9] which weights the predictors automatically during regression. Nevertheless, the irrelevant information from the noncritical-to-quality phase cannot be entirely removed from the MEN model.

Correspondence concerning this article should be addressed to Y. Yao at yyao@mx.nthu.edu.tw.

As discussed in,[10] the criteria for evaluating the quality of a regression model differs according to the circumstances, but typically there are two important aspects that are prediction accuracy on future data and model interpretation. A model with a good interpretation is usually more reliable than a model that is difficult to interpret. In addition, model interpretation is useful for the enhancement of process understanding, which is essential for process optimization, quality control, and so forth. However, most research on statistical batch process modeling pays more attentions to prediction accuracy, while the importance of the latter aspect is overlooked. In a chemical batch process, it is common that the final product quality is cumulatively determined by the trajectories of process variables during the entire batch run, and each process variable may contribute differently to the product quality at different time intervals. In particular, in multiphase batch processes, certain quality variable may be determined in a few phases, while other phases only contain irrelevant information that is not helpful for prediction. In the meantime, the impacts of different time points on the final product quality are usually similar in each relevant phase. Considering model interpretation, such process characteristics should be reflected by the regression coefficients of the statistical batch process model. In details, the regression coefficients corresponding to the noncritical-to-quality phases should be close to zero, and the coefficients within the same critical-to-quality phases should vary slowly and smoothly. However, due to the existence of measurement noise and the small sample size comparing to the model dimension, which are commonly observed in batch process data, the regression coefficients in conventional batch process models, for example, MPLS, seldom look like that. Such problem may be solved in the framework of statistical model selection.

In statistical model selection, the bias-variance compromise is a fundamental issue. On one hand, an unnecessarily complex model may yield highly variable estimates and predictions. On the other hand, an overly simplistic model often yields significantly biased estimates and predictions. In the both cases, prediction accuracy and model interpretation cannot be ensured. According to the well-known Occam's razor, the simplest model that adequately accommodates the data should be selected. In a regression model, the contribution of each predictor variable may be indicated by the corresponding regression coefficient.[11] Large regression coefficients could be indicative of significant contributions corresponding to predictors that need to be retained in the model.[12] However, this is not always the case. In some situations, the insignificant predictors with negligible effects on the response variable may also have large coefficients, especially when the sample size is small relative to the model dimension or there are correlations between the predictors (as in the cases of batch process modeling). In such situations, penalized regression can be utilized to shrink the coefficients of the insignificant predictor variables toward a value of zero, leading to a parsimonious model. With properly selected parameters, such penalization improves both predictability and interpretation of the model by inducing bias yet substantially reducing variability. The most representative methods in penalized regression are ridge regression[13] that adopts L_2 norm regularization and the least absolute shrinkage and selection operator (lasso) regularization[14] imposing a bound on the L_1 norm of the regression coefficients. In recent years, various forms of regularization terms have been combined with regression to penalize different types of complexities.[10,15–17]

Inspired by penalized regression, regularization techniques are introduced into statistical batch process modeling in this study. First, a lasso-type L_1 penalty term is involved into the regression model, which automatically selects relevant phases in the meantime of coefficients estimation. Then, the algorithm is further modified by adding an L_2-fusion term that smoothes the regression coefficients of the predictors within the same phase. As a result, the interpretation of the regression coefficients is further improved. The proposed penalized regression method is named as phase fused lasso, which has better prediction accuracy and model interpretation than the conventional MPLS[2] and phase-based PLS.[5] In addition, an MPLS-based phase division algorithm is designed as a pretreatment step before applying phase fused lasso. Comparing to the conventional approaches only utilizing time-slice information contained in process data, the developed division method better recognizes the changes in process dynamics.

The article is organized as following. In Section Ridge Regression and Lasso, the ridge regression and lasso approaches are introduced. Then, the regularization-based batch process modeling method are proposed in Section Regularization-Based Batch Process Modeling, including data unfolding, the phase fused lasso algorithm, the MPLS-based phase division approach, and online quality prediction based on phase fused lasso. For building the phase fused lasso model, an efficient computation algorithm and a generalized cross-validation (GCV) procedure are introduced. In Section Application Results, the proposed method is applied to an injection molding process for illustrating their good feasibility in both product quality prediction and process knowledge mining. Finally, conclusions are drawn in Section Conclusions.

Ridge Regression and Lasso

Before proposing the phase fused lasso approach, the most popular penalized regression methods, including ridge regression and lasso, are briefly reviewed in this section. A usual linear regression model with M predictors and one response is formulated as

$$\mathbf{y}=\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{X}(N\times M)$is the matrix of predictors, $\mathbf{y}(N\times 1)$ is the vector consisting of response measurements, $\boldsymbol{\beta}(M\times 1)=[\beta_1 \quad \beta_2 \quad \cdots \quad \beta_M]^T$ contains the unknown regression coefficients, $\boldsymbol{\varepsilon}$ is the residual vector, and N is the total number of samples. The ordinary lease square (OLS) estimate can be obtained by minimizing the residual sum of squares (RSS)

$$\min_{\boldsymbol{\beta}} (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \tag{2}$$

It is well known that OLS does poorly in both prediction and interpretation under certain situations, especially when the predictors in $\mathbf{X}$ are collinear or $N<M$. Many techniques, including ridge regression and lasso, have been proposed to improve OLS.

Ridge regression[13] minimizes the RSS subject to a bound on the L_2 norm of the coefficients

$$\min_{\boldsymbol{\beta}} ((\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\|\boldsymbol{\beta}\|_2^2) \tag{3}$$

where $\|\cdot\|_2$ stands for the L_2 norm, $\|\boldsymbol{\beta}\|_2^2=\sum_{i=1}^{M}\beta_i^2$, and λ is a tuning parameter. By doing so, the coefficients are shrunk toward zero continuously. Consequently, better prediction is achieved through a bias-variance trade-off. However, ridge

regression always keeps all the predictors in the model due to the nature of the L_2-norm penalty, and hence it cannot result in a parsimonious model.

Different from ridge regression, lasso[14] minimizes a penalized RSS by introducing an L_1 penalty term into the objective function

$$\min_{\boldsymbol{\beta}} ((\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta)+\lambda\|\boldsymbol{\beta}\|_1) \tag{4}$$

where $\|\cdot\|_1$ represents the L_1 norm, and $\|\boldsymbol{\beta}\|_1=\sum_{i=1}^{M}|\beta_i|$. Attributing to the mathematical property of the L_1 norm, lasso does continuous shrinkage, which leads to automatic variable selection. Therefore, lasso often results in a parsimonious model with good prediction power. Comparing to conventional model selection methods, such as stepwise regression, lasso has several major advantages.[18] First, the theoretical properties of lasso are easier to understand. Second, the estimates of regression coefficients are more stable. Third, the computational costs are much lower especially when the number of predictors is large.

Regularization-Based Batch Process Modeling

Data unfolding and normalization

Usually, batch process data are stored in a three-way matrix $\underline{\mathbf{X}}(I\times J\times K)$ that should be transformed to a two-way form before regression, where I is the number of batches, J is the number of process variables, and K is the number of sampling intervals in each batch. To perform such transformation, the batch-wise unfolding is most commonly used, where $\underline{\mathbf{X}}$ is unfolded into a two-dimensional matrix with I rows and $P=J\times K$ columns. In the unfolded matrix, each row contains all the measurements collected throughout a batch run. Then, the mean of each column is subtracted from the unfolded matrix and the variance of each column is scaled to unit through normalization. As a consequence, the between-batch variations are highlighted, while the major nonlinearities contained in variable trajectories are eliminated. The regression model built on the basis of such data structure has chance to capture the time-dependence information involved in process data and reflect the cumulative effect of variable trajectories on the product quality. Therefore, the model proposed in below is trained using the batch-wise unfolded and normalized matrix $\mathbf{X}(I\times P)$ of process variables and the centered vector $\mathbf{y}(I\times 1)$ of quality measurements.

Phase fused lasso

Even though lasso enjoys great computational advantages and excellent performance comparing to conventional model selection methods, it is not suited to batch process modeling. The reasons are of twofolds. First, in (4), lasso selects at most N predictors when $N<M$. Second, if there are groups of highly correlated predictors, lasso tends to select arbitrary one from each group and bring about difficulties in model interpretation. In the pretreated batch process data matrix $\mathbf{X}$, $N=I$ and $M=P$, where the situation of $I\ll P$ is common. Meanwhile, correlation exists among the columns in $\mathbf{X}$. Especially, in batch processes with multiple phases, the predictors, that is, the trajectory points of process variables, belonging to the same phase are usually highly correlated. Such problem may be solved using group lasso.[15]

Notice that the trajectory points of a variable within the same phase usually contribute similarly to the end-product quality. Therefore, they should be selected into or kept out of the batch process model simultaneously (as a group). Motivated by the group lasso approach,[15] phase selection can be achieved automatically by solving the following optimization problem

$$\min_{\beta}\left(\frac{1}{2}(\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta)+\lambda\sum_{s=1}^{S}\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2\right) \tag{5}$$

where $\boldsymbol{\beta}_s(s=1,...,S)$ contains the regression coefficients corresponding to the variable trajectory points in the sth phase, S is the number of total phases, and K_s is the number of sampling intervals in the sth phase, which takes the different lengths of phases into consideration. The L_2 norm in the equation encourages a grouping effect and tends to select or reject the entire phase data together, while the lasso penalty expressed by the sigma notation encourages sparsity at the phase level. By specifying a proper tuning parameter λ, all regression coefficients belonging to the noncritical-to-quality phases can be shrunk to zero simultaneously, while the variable trajectory information in the relevant phases is retained in the regression model. As a consequence, improved model interpretation can be achieved, while prediction accuracy can also be enhanced due to noise suppression by selecting meaningful phases. Bach[19] discussed the model consistency of group lasso, while more detailed theoretical analysis of group lasso can be found in the references.[15,19]

The objective function in (5) can be modified to further improve the interpretation of the coefficients. As discussed in previous, the regression coefficients within the same relevant phases should vary smoothly, because the relationships between the corresponding predictors and the final product quality are similar. Recently, Hebiri and van de Geer[16] proposed to use a L_2-fusion penalty, $\sum_{m=2}^{M}(\beta_m-\beta_{m-1})^2$, to make the coefficients smooth in the linear regression problem. Similar idea is adopted here to achieve within-phase smoothness. After modification, the penalized regression named phase fused lasso is formulated as

$$\min_{\boldsymbol{\beta}}\left(\frac{1}{2}(\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta)+\lambda_1\sum_{s=1}^{S}\sum_{j=1}^{J}\sum_{k=2}^{K_s}(\beta_{j,k}^s-\beta_{j,k-1}^s)^2+\lambda_2\sum_{s=1}^{S}\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2\right) \tag{6}$$

where $\beta_{j,k}^s=\beta_{j,(\sum_{t=1}^{s-1}K_t)+k}$ denotes the regression coefficient corresponding to the kth trajectory point of variable j in the sth phase, which also corresponding to the $((\sum_{t=1}^{s-1}K_t)+k)$-th trajectory point of variable j in the entire batch duration. It can be noted that the proposed phase fused lasso involves a group lasso penalty term in the model, and share the benefits of group lasso. Therefore, phase fused lasso can also handle variable correlation.

The solving of the optimization problem expressed by (6) is not very direct, because both of the penalty terms in phase fused lasso are nonsmooth. To facilitate problem solving, each row $\mathbf{x}_i(i=1,2,\cdots,I)$ in the training data matrix $\mathbf{X}$ is reorganized as

$$\mathbf{x}_i=\begin{bmatrix}\mathbf{x}_i^1 & \mathbf{x}_i^2 & \dots & \mathbf{x}_i^S\end{bmatrix} \tag{7}$$

$$\mathbf{x}_i^s=\begin{bmatrix}\mathbf{x}_{i,1}^s & \mathbf{x}_{i,2}^s & \dots & \mathbf{x}_{i,J}^s\end{bmatrix} \tag{8}$$

$$\mathbf{x}_{i,j}^{s}=\begin{bmatrix} x_{i,j,1}^{s} & x_{i,j,2}^{s} & \cdots & x_{i,j,K_s}^{s} \end{bmatrix} \tag{9}$$

where $\mathbf{x}_i^s$ contains all the data belonging to the sth phase in the ith batch, $\mathbf{x}_{i,j}^s$ consists of the measurements of the jth process variable in the sth phase in batch i, and $x_{i,j,k}^s$ is the measured value of variable j on the kth sampling interval in phase s and batch i. Accordingly, the structure of the regression coefficient vector becomes

$$\boldsymbol{\beta}=\begin{bmatrix} \boldsymbol{\beta}_1^T & \boldsymbol{\beta}_2^T & \cdots & \boldsymbol{\beta}_S^T \end{bmatrix}^T \tag{10}$$

$$\boldsymbol{\beta}_s=\begin{bmatrix} (\boldsymbol{\beta}_1^s)^T & (\boldsymbol{\beta}_2^s)^T & \cdots & (\boldsymbol{\beta}_J^s)^T \end{bmatrix}^T \tag{11}$$

$$\boldsymbol{\beta}_j^s=\begin{bmatrix} \beta_{j,1}^s & \beta_{j,2}^s & \cdots & \beta_{j,K_s}^s \end{bmatrix}^T \tag{12}$$

where $\boldsymbol{\beta}_j^s$ is the coefficient vector corresponding to the trajectory of variable j in the sth phase, and the other symbols have the same meaning as previously described. Based on such definition, the objective function in (6) is transformed into the following format

$$\begin{aligned} L(\boldsymbol{\beta}) &= \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \\ &\quad +\lambda_1\sum_{s=1}^{S}\sum_{j=1}^{J}\sum_{k=2}^{K_s}(\beta_{j,k}^s-\beta_{j,k-1}^s)^2+\lambda_2\sum_{s=1}^{S}\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2 \\ &= \frac{1}{2}\left(\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+2\lambda_1\|\mathbf{0}-\mathbf{B}\boldsymbol{\beta}\|_2^2\right)+\lambda_2\sum_{s=1}^{S}\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2 \\ &= \frac{1}{2}\left(\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\|\mathbf{0}-\sqrt{2\lambda_1}\mathbf{B}\boldsymbol{\beta}\|_2^2\right)+\lambda_2\sum_{s=1}^{S}\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2 \\ &= \frac{1}{2}\left(\|\tilde{\mathbf{y}}-\tilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2\right)+\lambda_2\sum_{s=1}^{S}\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2, \end{aligned} \tag{13}$$

where

$$\tilde{\mathbf{y}}=\begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \tag{14}$$

$$\tilde{\mathbf{X}}=\begin{bmatrix} \mathbf{X} \\ \sqrt{2\lambda_1}\mathbf{B} \end{bmatrix} \tag{15}$$

$\mathbf{B}$ is a block diagonal matrix whose dimensions are $P\times P$

$$\mathbf{B}=\mathrm{diag}\,(\mathbf{B}_1,\mathbf{B}_2,\cdots,\mathbf{B}_S) \tag{16}$$

$\mathbf{B}_s(JK_s\times JK_s)$ is a block diagonal matrix by repeating the matrix $\mathbf{C}_s$ for J times

$$\mathbf{B}_s=\mathrm{diag}\,(\mathbf{C}_s,\mathbf{C}_s,\cdots,\mathbf{C}_s) \tag{17}$$

and $\mathbf{C}_s$ is a square matrix with dimension $K_s\times K_s$, which has a similar structure to an upper dual-diagonal matrix

$$\mathbf{C}_s=\begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & & 0 \end{bmatrix} \tag{18}$$

Then, the optimization problem with the objective function described in (13) can be solved efficiently according to the following derivation.

Taking the derivative of $L(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ and setting the derivative to zero, (19) is achieved

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}=\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{\beta}-\tilde{\mathbf{X}}^T\tilde{\mathbf{y}}+2\lambda_2\mathbf{D}\boldsymbol{\beta}=0 \tag{19}$$

where $\mathbf{D}$ is a diagonal matrix with the ith diagonal element as

$$d_s=\frac{1}{2\sqrt{K_s}\|\boldsymbol{\beta}_s\|_2} \tag{20}$$

Hence

$$(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+2\lambda_2\mathbf{D})\boldsymbol{\beta}=\tilde{\mathbf{X}}^T\tilde{\mathbf{y}} \tag{21}$$

Denote $\mathbf{A}=\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+2\lambda_2\mathbf{D}$. Because $\mathbf{A}$ is in general positive definite, the problem is convex. Consequently, $\boldsymbol{\beta}$ calculated in (22) is a unique global optimal solution to the optimization problem

$$\boldsymbol{\beta}=\mathbf{A}^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{y}} \tag{22}$$

Note that the size of A is $P\times P$, which is usually quite large. Hence, the inverse of A is computationally expensive. To reduce the computational burden, QR decomposition is used instead to solve $\boldsymbol{\beta}$. The details are as following

$$\mathbf{A}^T=\mathbf{QR} \tag{23}$$

where $\mathbf{Q}$ is an orthogonal matrix, and $\mathbf{R}$ is an upper triangular matrix, both of which are derived from QR decomposition of $\mathbf{A}^T$. Therefore

$$(\mathbf{QR})^T\boldsymbol{\beta}=\tilde{\mathbf{X}}^T\tilde{\mathbf{y}} \tag{24}$$

$$\boldsymbol{\beta}=(\mathbf{Q}^T)^{-1}(\mathbf{R}^T)^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{y}}=\mathbf{Q}(\mathbf{R}^T)^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{y}} \tag{25}$$

Since the matrix $\mathbf{D}$ is dependent to $\boldsymbol{\beta}$, which is unknown beforehand, an iterative algorithm is proposed to obtain the solution. In each iteration, $\boldsymbol{\beta}$ is calculated with the current estimate of $\mathbf{D}$, and then $\mathbf{D}$ is updated using the calculated $\boldsymbol{\beta}$. Such procedure is repeated until the algorithm converges. The detailed steps are listed in below.

1. Initialize $\mathbf{D}$.
2. Calculate $\mathbf{A}=\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}+2\lambda_2\mathbf{D}$.
3. Conduct QR decomposition of $\mathbf{A}$ as (23).
4. Calculate $\boldsymbol{\beta}$ using (25).
5. According to (20), update the matrix $\mathbf{D}$ with $\boldsymbol{\beta}$ calculated in step 4.
6. Return to step 2, until converges.

In (6), there are two tuning parameters, λ_1 and λ_2, to be specified. As well known, cross-validation (CV) is the most popular method for parameter selection. However, for each pair of parameters, the conventional V-fold CV algorithm requires repeating the model training procedure for V times, which is computationally demanding. To efficiently select the tuning parameters, a parameter selection procedure based on the GCV criterion is utilized in this article, as suggested by Golub et al.[20] and Tibshirani[14] for building the ridge regression and lasso models. Originally proposed to reduce the computational burden, the GCV criterion that can be regarded as a rotation-invariant version of the ordinary CV has been found to have a number of favorable properties.[21] Similar to that in,[14,20] the GCV estimate of λ_1 and λ_2 in the phase fused lasso estimate (25) is the minimizer of GCV (λ_1,λ_2) given by

$$\mathrm{GCV}\,(\lambda_1,\lambda_2)=\frac{\frac{1}{I}\|(\mathbf{I}-\mathbf{F}(\lambda_1,\lambda_2)\tilde{\mathbf{y}})\|_2^2}{\left[\frac{1}{I}\mathrm{Trace}\,(\mathbf{I}-\mathbf{F}(\lambda_1,\lambda_2))\right]^2} \tag{26}$$

where I is the total number of batches contained in the training dataset, $\mathbf{I}$ is an identity matrix

$$\mathbf{F}(\lambda_1,\lambda_2)=\tilde{\mathbf{X}}\mathbf{Q}(\mathbf{R}^T)^{-1}\tilde{\mathbf{X}}^T \tag{27}$$

and $\tilde{\mathbf{X}}$, $\mathbf{Q}$, and $\mathbf{R}$ are derived using based on (15) and (23) based on λ_1 and λ_2. Accordingly, the optimal values of λ_1 and λ_2 are achieved using the grid search algorithm

1. Set a grid in the solution space of (λ_1, λ_2).
2. For each pair of (λ_1, λ_2) on the grid nodes, calculate a phase fused lasso model using the iterative steps described in previous.
3. Compute GCV (λ_1,λ_2) for each model derived in step 2.
4. Choose the λ_1 and λ_2 minimizing GCV (λ_1,λ_2) on all the nodes.

Figure 1 shows a flow chart of the grid search.

Phase division

The performance of the proposed phase fused lasso model highly depends on the results of phase division. In the past decade, a number of phase division methods have been developed,[4] most of which are based on the time-slice correlation information contained in process data and ignore the changes in process dynamics. To solve such problem, a MPLS-based phase division algorithm is, therefore, introduced in this work.

As the first step, an MPLS model is trained as below, by applying the PLS algorithm to the pretreated training dataset $\{\mathbf{X},\mathbf{y}\}$

$$\mathbf{X}=\mathbf{T}\mathbf{P}^T+\mathbf{E} \tag{28}$$

$$\mathbf{y}=\mathbf{U}\mathbf{Q}^T+\mathbf{F} \tag{29}$$

Here, $\mathbf{X}$ consists of a series of time-slice data matrices as

$$\mathbf{X}=[\mathbf{X}_1 \quad \mathbf{X}_2 \quad \ldots \quad \mathbf{X}_K] \tag{30}$$

where $\mathbf{X}_k$ is the time-slice data matrix corresponding to the kth sampling interval. The above MPLS model can be written in a compact form as

$$\mathbf{y}=\mathbf{X}\boldsymbol{\theta}+\mathbf{F}^* \tag{31}$$

where $\mathbf{T}$ and $\mathbf{U}$ are the scores matrices, $\mathbf{P}$ and $\mathbf{Q}$ are the loading matrices, $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{F}^*$ are the residual matrices, and $\boldsymbol{\theta}$ is a vector containing MPLS regression coefficients. In the regression coefficient vector $\boldsymbol{\theta}(P\times 1)$, every J elements indicate the partial correlation between the process measurements at a certain sampling time interval and the final product quality. Accordingly, $\boldsymbol{\theta}$ can be cut to pieces along the time axis, resulting in K subvectors each of which has dimensions of $(J\times 1)$, that is

$$\boldsymbol{\theta}=\left[\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \cdots \quad \boldsymbol{\theta}_K^T\right]^T \tag{32}$$

where $\boldsymbol{\theta}_k$ contains the regression coefficients corresponding to $\mathbf{X}_k$.

In a multiphase batch process, the process data collected in the same phase usually contribute similarly to the final product quality, while different phases have different statistical features. Thus, it can be inferred that $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{k+1}$ should have a high degree of similarity, if the time intervals k and $k+1$ belong to the same phase; otherwise, the similarity between these two subvectors is relatively low. Therefore, the range of each phase can be identified by clustering $\boldsymbol{\theta}_k$, $k=1,2,\cdots,K$. At this point, many different clustering algorithms can be applied. The popular k-means clustering algorithm[22] is adopted in this article for illustration. The detailed steps are as following.

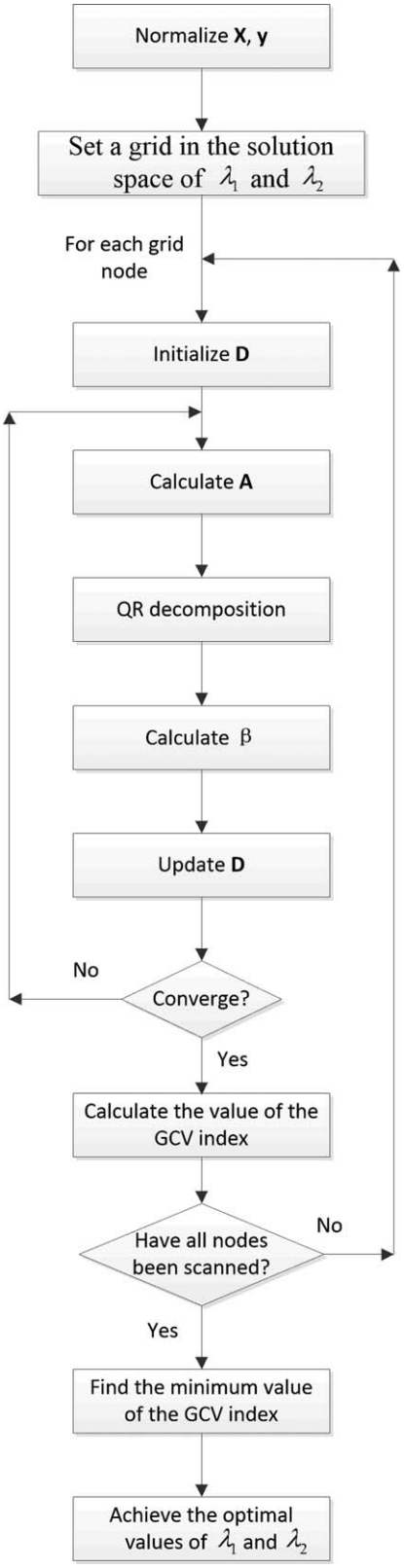

Figure 1. Flow chart of parameters selection in phase fused lasso.

1. Set $k=1$, $m=1$. Choose $\boldsymbol{\theta}_k$ as the initial cluster center $\mathbf{w}_m$.
2. Calculate the Euclidean distance between $\mathbf{w}_m$ and $\boldsymbol{\theta}_{k+1}$.

3. If the distance is less than a predefined threshold δ, assign $\boldsymbol{\theta}_{k+1}$ to class m, and update $\mathbf{w}_m$ by making $\mathbf{w}_m$ equal to the average of all subvectors belonging to class m. Otherwise, add a new cluster center $\mathbf{w}_{m+1} = \boldsymbol{\theta}_{m+1}$, and let $m=m+1$.

4. Let $k=k+1$, and go back to step 2 until $k > K$.

As in any clustering algorithm, the threshold δ makes a trade-off between the model accuracy and complexity. In particular, a large threshold results in few classes, that is, modeling phases, but less accurate modeling, and vice versa. Note that the number of the modeling phases may not be exactly same as the number of the operation phases defined by process engineers. The modeling phases are identified based on their statistical characteristics. Hence, an operation phase with changing statistical features may be divided into several modeling phases, while several successive operation phases with similar statistical features may be combined into one single modeling phase.

Online quality prediction

After phase fused lasso modeling, the elements in the regression coefficient vector $\boldsymbol{\beta}$ are rearranged as

$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1^T & \boldsymbol{\beta}_2^T & \cdots & \boldsymbol{\beta}_K^T\end{bmatrix}^T \tag{33}$$

$$\boldsymbol{\beta}_k=\begin{bmatrix}\beta_{1,k} & \beta_{2,k} & \cdots & \beta_{J,k}\end{bmatrix}^T \tag{34}$$

Then, after the process measurements of a new batch, are collected, the final product quality can be predicted as

$$y_{\text{new}}=\mathbf{x}_{\text{new}}^T\boldsymbol{\beta} \tag{35}$$

where y_{new} is the prediction value of the final product quality, and $\mathbf{x}_{\text{new}}^T(1\times P)$ is a vector containing all the normalized process measurements in a new batch, which is prepared in the following way. After collecting the data matrix of the new batch $\mathbf{X}_{\text{new}}(J\times K)$, it is unfolded into a vector with dimensions $(1\times P)$ and then normalized to $\mathbf{x}_{\text{new}}^T$ utilizing the means and variances predetermined in the model training procedure.

However, such procedure cannot be utilized directly in online prediction, since $\mathbf{x}_{\text{new}}^T$ contains all process data in an entire batch, which are unavailable until the end of the batch. Therefore, future data estimation is necessary for online application of the prediction model. In the previous research, several different types of estimation approaches have been proposed, including the "zero deviation" approach, the "current deviation" approach, and the "PLS projection" approach.[2,23] However, the performances of the former two approaches highly rely on the strong assumptions of the future operation that may not be satisfied in real industry, while the third approach can only be used in MPLS.

Here, a feature-selective k-nearest neighbor (FS-kNN) method is developed to fill the unavailable future data with the information from the historical data belonging to the most similar A number of batches to the current batch. The basic idea is as following. At each sampling interval, the known part of the data in the current batch is compared with the corresponding part of the historical data in the training dataset. By doing so, a number of nearest neighbors of the current batch are identified. The unavailable future measurements are then estimated from the average of these nearest neighbors. Different from the conventional k-nearest neighbor (kNN) method, the batch process data are weighted with a feature-selective vector for the neighbor finding. As a result, the estimate of the future data is not affected by the variable trajectories irrelevant to the final product quality.

The online future data estimation and final quality prediction procedure is described in below.

1. Compute the dissimilarity index D_i between the new batch and the training batches

$$D_i=\|\mathbf{x}_i^{k^-}\odot\boldsymbol{\xi}^{k^-}-\mathbf{x}_{\text{new}}^{k^-}\odot\boldsymbol{\xi}^{k^-}\|_2, \tag{36}$$

where $\boldsymbol{\xi}^{k-}$ is a feature-selective vector

$$\boldsymbol{\xi}^{k-}=\begin{bmatrix}\boldsymbol{\xi}_1^T & \boldsymbol{\xi}_2^T & \cdots & \boldsymbol{\xi}_k^T\end{bmatrix}^T \tag{37}$$

$$\boldsymbol{\xi}_h=\begin{bmatrix}\xi_{1,h} & \xi_{2,h} & \cdots & \xi_{J,h}\end{bmatrix}^T \tag{38}$$

$\xi_{j,h}=0$ when $\beta_{j,h}=0$, otherwise $\xi_{j,h}=1$ ($j=1,\cdots,J$ and $h=1,\cdots,K$), i is the batch index in the training set ($i=1,\cdots,I$), $\mathbf{x}_{\text{new}}^{k^-}$ is a vector containing the normalized data in the new batch up to the current time interval k, $\mathbf{x}_i^{k^-}$ consists of the normalized data from the first sampling interval to the kth sampling interval in the ith training batch, and $\odot$ is a multiplication operator between the corresponding elements of two vectors.

2. Find A number of smallest D_i values. The corresponding batch runs are selected as the members of the nearest neighbor set U.

3. Estimate the unknown part of the new batch by the average of the nearest neighbors as

$$\hat{\mathbf{x}}_{\text{new}}^{k^+}=\frac{1}{A}\sum\nolimits_{\mathbf{x}_m\in U}\mathbf{x}_m^{k^+} \tag{39}$$

where $\hat{\mathbf{x}}_{\text{new}}^{k+}$ contains the approximated future data of the new batch from time interval $(k+1)$ to K, and $\mathbf{x}_m^{k+}$ consists of the process measurements of the mth batch run in training dataset from time interval $(k+1)$ to K.

4. The complemented vector of $\mathbf{x}_{\text{new}}$ can be written as $\hat{\mathbf{x}}_{\text{new}}^k=\left[\left(\mathbf{x}_{\text{new}}^{k-}\right)^T,\left(\hat{\mathbf{x}}_{\text{new}}^{k+}\right)^T\right]^T$.

5. The online quality prediction at the kth sampling interval can then be achieved by calculating the following equation

$$y_{\text{new}}^k=\left(\hat{\mathbf{x}}_{\text{new}}^k\right)^T\boldsymbol{\beta}+\bar{y} \tag{40}$$

where $\bar{y}$ is the mean value of all quality measurements contained in the training set.

It can be noted that an underlying assumption of the FS-kNN approach is that the correlation between collected measurements and future measurements in the new batch is the same in the training data. Such assumption is consistent with the common supposition of any statistical modeling methods, that is, the training data should cover the entire area of interest where the model is expected to operate.

Application Results

Injection molding process

Injection molding, a typical batch process, is utilized in this section to illustrate the effectiveness of the proposed phase fused lasso method. There are several sequential operation phases in each cycle of such process, which are filling, packing-holding, plastication, and cooling. Each operation phase has different dynamic characteristics and contributes to the final product quality in different ways. In the course of filling, the screw moves forward, injecting the melt into

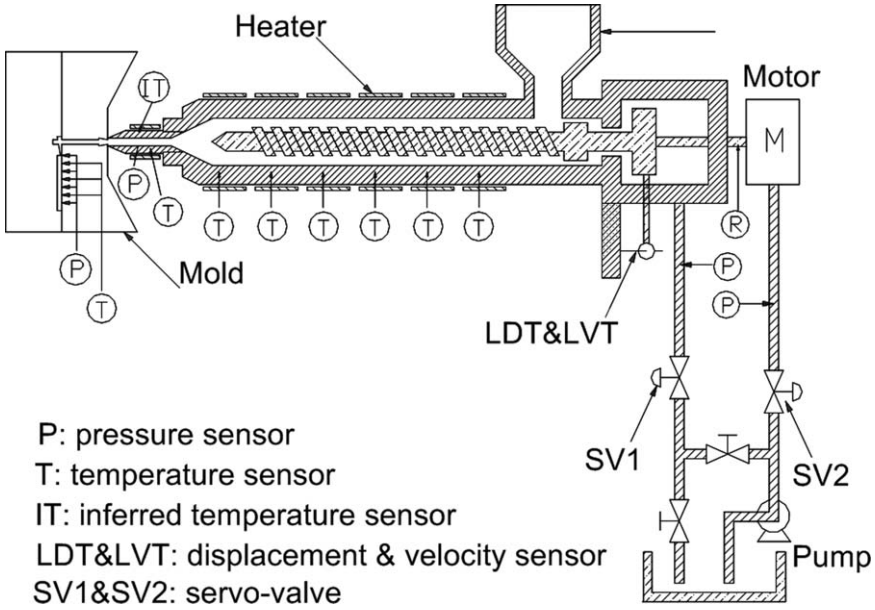

Figure 2. Simplified illustration of an injection molding machine.

the mold cavity. Then, the process switches to the packing-holding phase, during which additional melt is packed into the cavity at a high pressure and then such pressure is held for a while. Consequently, the shrinkage caused by material cooling and solidification is compensated by doing so. The packing-holding phase lasts for several seconds, followed by the plastication phase. During plastication, the screw rotates and moves back, shearing and melting the polymer material in the barrel and conveying the melt to the front of the screw. In the meantime, the material in the mold cavity is cooled down. Hence, the product can be completely solidified and easily ejected from the mold. In comparison with plastication, cooling usually takes longer time. Figure 2 illustrates a typical reciprocating-screw injection molding machine with instrumentations.[5]

High-density polyethylene (HDPE) is utilized as the feed material in the experiments. As listed in Table 1, there are eight key process variables measured in real time, including valve opening, stroke displacement, velocity, pressure, and temperatures. Among them, the trajectory of the valve opening 2 is exactly the same as that of the valve opening 1, which means these two process variables are highly correlated. In addition, at the end of each batch, the product weight is measured as a final quality variable. The operating conditions are as follows. The set point of the temperature in the third barrel zone is 160°C. The packing-holding time is fixed to be 5 s. The sampling time interval is 0.05 s. To exciting the process and generating necessary regression information for predictive modeling, a 3^3 full factorial design of experiments (DOE) is conducted, where the three factors to be designed are the injection velocity, the packing pressure and the barrel temperatures in the first two zones. The three levels of the injection velocity are 0.025, 0.035, and 0.045 m/s, respectively. Regarding the packing pressure, the three levels are set to be 25×10^5, 30×10^5, and 35×10^5 Pa. The barrel temperatures in the first two zones are chosen to be the third factor in DOE, which are set to the same values in each level. The three levels are 190, 200, and 210°C, respectively.

The DOE generates 27 different treatment combinations, each of which has 10 repetitive experiments. In each combination, five batches are randomly chosen as the training batches, while the other five are reserved as the test batches. Hence, process and quality data from totally 270 batches are recorded, among which 135 batches are served as training data for process modeling, while the other 135 batches are utilized as test data to check the effectiveness of the proposed method. Noted that different injection velocities lead to variations in the filling time, data alignment is, therefore, necessary to be conducted in the filling phase. Here, the process variables are resampled in this phase with respect to an indicator variable, the displacement of the screw stroke. After data alignment, there are totally 503 measurements for each variable in each batch.

Phase division

Before process modeling, phase division is conducted using the proposed MPLS-based approach and the time-slice PLS-based approach as introduced in,[5] respectively. Figure 3 shows the division results of the proposed method, while the division results and the variable trajectories of valve opening 1 from all 135 batches in the training set are plotted together in Figure 4. It can be observed that the entire process with 4 operation phases is divided into seven modeling phases. Phase 1 only contains several sampling intervals, which corresponds to the machine start-up at the beginning of each

Table 1. Process Variables

No.	Variable	Units
1	Valve opening 1	%
2	Valve opening 2	%
3	Screw stroke	m
4	Injection velocity	m/s
5	Injection pressure/ back pressure	Pa
6	Barrel temperature (zone 1)	^{o}C
7	Barrel temperature (zone 2)	^{o}C
8	Barrel temperature (zone 3)	^{o}C

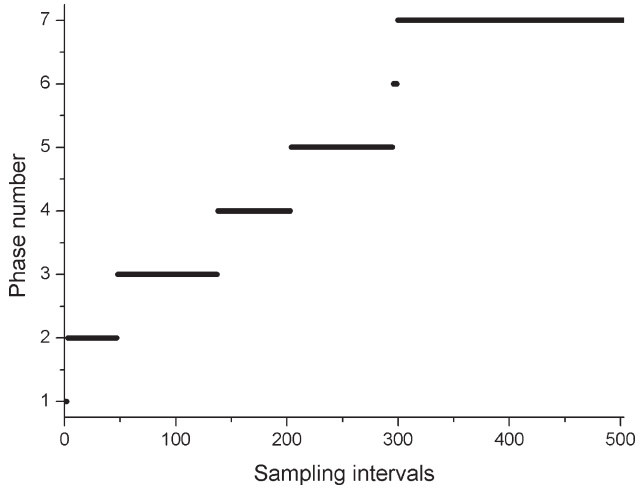

Figure 3. Phase division results based on the proposed approach.

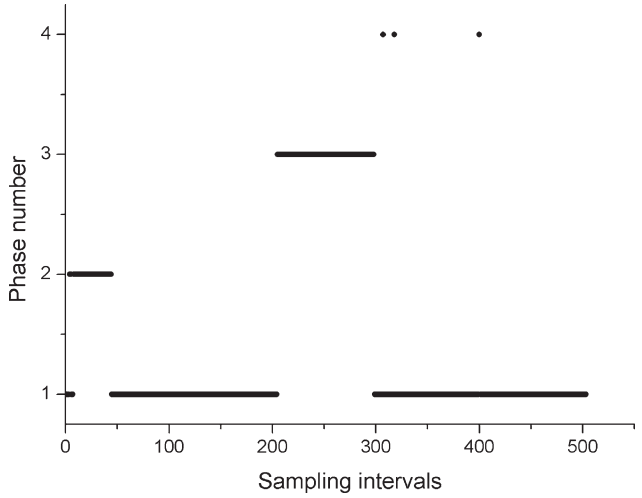

Figure 5. Phase division results based on time-slice PLS.

batch. Then, the filling behavior is represented by Phase 2. The packing-holding period is partitioned into two modeling phases at sampling interval 137, implying that the process characteristics change at this time point even though the time points within this period belong to the same operation phase. Engineering knowledge verifies the reasonability of such division. To be exactly, Phase 3 corresponds to packing, while Phase 4 is about holding. More detailed discussion about the changing characteristics between these two modeling phases will be provided later. Phase 5 agrees with the operation phase of plastication, during which the screw moves backward from a position near the nozzle to a predetermined position. Because the screw moves forward by different distances during filling, the durations of plastication are also different. Therefore, the time intervals from 296 to 299 are clustered as Phase 6, indicating the uneven tails of plastication in different batches. Phase 7 represents the cooling period after plastication. In this modeling phase, no more manipulation is applied to the process except the circulation of the cooling water.

For comparison, the results from the time-slice PLS-based phase division are plotted in Figure 5. It is clearly that the time intervals in machine start-up, packing-holding and cooling phase are clustered into the same class, although they can be discriminated using the clustering results associated with process operation time. In addition, the changing process characteristics during packing-holding are not revealed, because such method ignores the time-dependent information contained in process data.

Process modeling and quality prediction

In this section, the training data are modeled with different types of batch process modeling methods, including MPLS, phase-based PLS, and phase fused lasso, where the latter two models are built based on the phase division results shown previously. Based on the GCV criterion introduced in Section Application Results, the parameters of the phase fused lasso model are selected. The selection of the grid nodes, that is, the candidate values of λ_1 and λ_2, is an open problem in grid search. Here, the candidate values of both λ_1 and λ_2 are chosen as $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$, which forms a grid with 81 nodes. Following the steps of grid search, the minimum value of GCV(λ_1, λ_2) is achieved on the grid node of (1e-4, 1). Hence, the

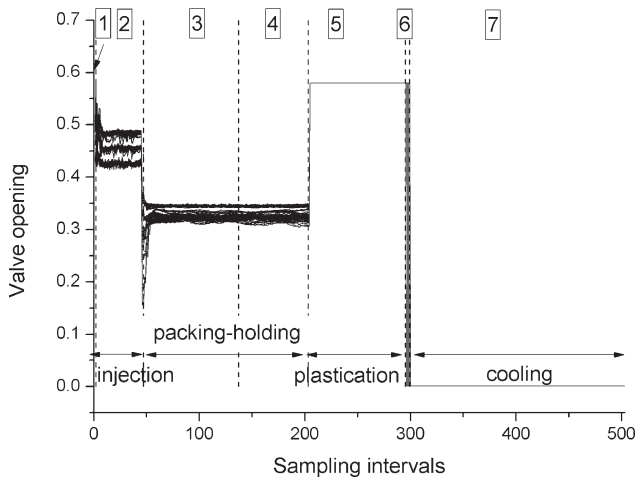

Figure 4. Trajectories of valve opening together with the phase division results based on the proposed approach.

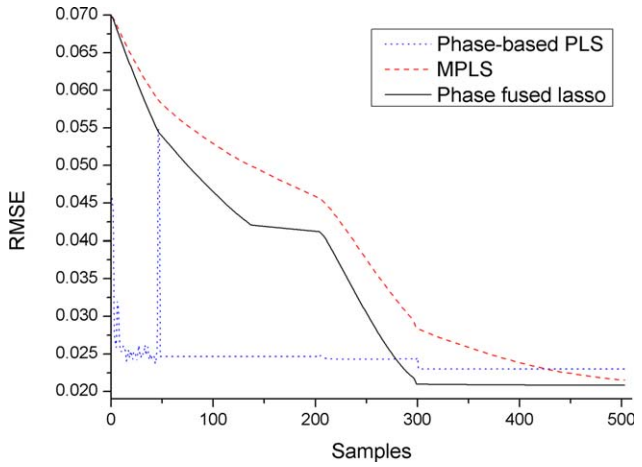

Figure 6. Online prediction results using "zero deviation" future data estimation.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 2. Offline Prediction Results

	RMSE	
Modeling methods	Training dataset	Test dataset
MPLS	0.0193	0.0215
Phase-based PLS	0.0238	0.0230
Phase fused lasso	0.0205	0.0208

parameters of the phase fused lasso model are selected as $\lambda_1 = 1\times10^{-4}$ and $\lambda_2 = 1$. The numbers of latent variables in the other two models are determined with CV.

The prediction accuracy of each model can be measured with the index of root mean squared error (RMSE). Specifically, the offline prediction accuracy is reflected by

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N}} \tag{41}$$

where N is the total number of batches to predict (when the model is applied to the training dataset, $N = I$), y_n and $\hat{y}_n$ are the measured value and the predicted value of the final product quality of batch n, respectively. Similarly, the measure of the online prediction accuracy at each time interval during batch processing is formulated as

$$\mathrm{RMSE}_k = \sqrt{\frac{\sum_{n=1}^{N}(y_n - \hat{y}_{n,k})^2}{N}} \tag{42}$$

where $\hat{y}_{n,k}$ is the quality prediction of batch n at the kth sampling interval, and $k = 1, \cdots, K$. Smaller RMSE values indicate more accurate predictions. The values of RMSE_k can be plotted in a figure to show their changes along the operation time.

Based on RMSE, a comparison of the offline prediction accuracy is conducted for MPLS, phase-based PLS, and phase fused lasso as shown in Table 2. In the discussed case, it is clear that the phase-based PLS model performs worst in the predictions of both the training data and the test data. Such results are not surprising, because phase-based PLS discards both within-phase and between-phase time-dependence information and cannot reflect the cumulative contributions of the process variable trajectories to the final product quality. MPLS provides best prediction accuracy of the training dataset. However, its generalization to the test dataset does not show equally good results. In comparison, the performance of phase fused lasso is more balanced as observed in Table 2. Meanwhile, phase fused lasso provides best predictions of the test data among all three methods, attributing to its ability in selecting the relevant phases and smoothing the regression coefficients in each phase. Such ability makes the phase fused lasso model better conforming to the process than other two types of methods.

For online quality prediction, three different types of future data estimation approaches are tested, which are based on "zero deviation," "current deviation," and FS-kNN, respectively. In the FS-kNN algorithm, the number of neighbors is selected to be 5. The online prediction results are plotted in Figures 6–8. Please note that there is no need for phase-based PLS to estimate the future available data in online application, since such method only utilizes time-slice information. However, for facilitating comparison, the results based on phase-based PLS are also plotted in Figures 6–8. It is interesting to observe that, in Figures 6 and 7, the phase-based PLS model performs even better than the other two methods in the early period of online prediction, although the final prediction accuracy of such method is worse. As discussed in,[5] if the online estimation of future measurements is not accurate, the online quality prediction will be distorted. Therefore, more reliable future data estimation is desired. Figure 9 indicates that the proposed FS-kNN is a good substitute for the existing approaches, since it provides more accurate future data estimation and improves the online prediction accuracy very significantly. Using FS-kNN, the performances of both phase fused lasso and MPLS are improved significantly and become much better than that of phase-based PLS, as shown in Figure 8. The comparison between the results of phase fused lasso and MPLS shows that, in two of the three figures the prediction accuracy of MPLS is higher than that of phase fused lasso in the early period of operation, but phase fused lasso always outperforms MPLS in the latter half of the batches. Such

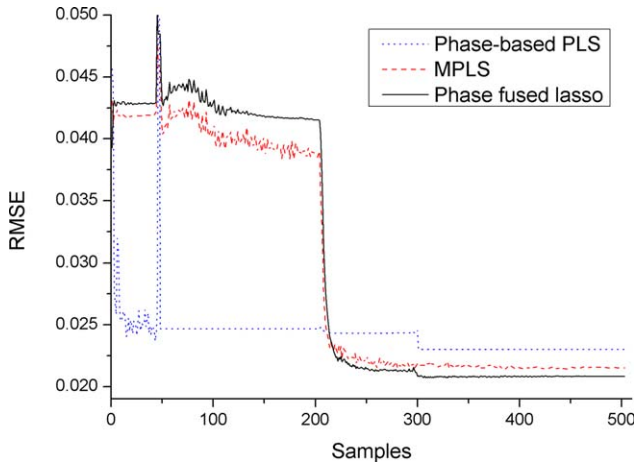

Figure 7. Online prediction results using "current deviation" future data estimation.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

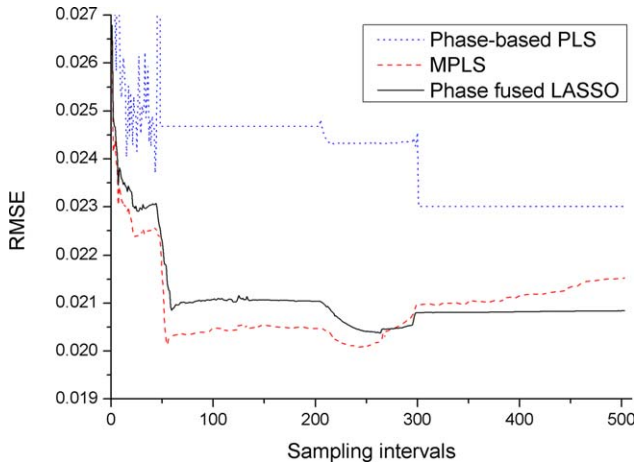

Figure 8. Online prediction results using FS-kNN future data estimation.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

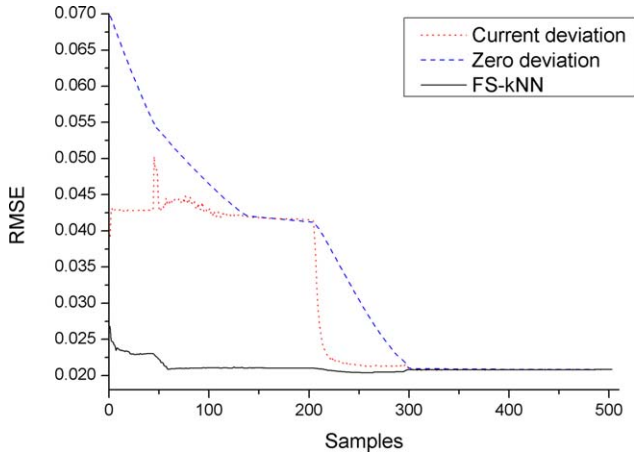

Figure 9. Online prediction results of phase fused lasso using different future data estimation approaches.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

phenomenon can be explained by the mathematical properties of different types of regression models and the engineering insight of the process data. MPLS is an unbiased regression model, while phase fused lasso provides a biased estimation method by introducing two penalty terms. Therefore, it is not surprised that MPLS may perform better when the data do not contain much noise. Conversely, when the data are quite noisy, phase fused lasso provides a more reliable estimate of the prediction model by involving the regularization technique. During filling and packing-holding, the signal-to-noise ratio of the data is relatively high. As a result, the regularization conducted by phase fused lasso may lead to the loss of some detailed information in the data. However, after the process switches to the plastication phase, the noise contained in the normalized data, which is not relevant to quality prediction, become significant. Under such situation, MPLS is affected more seriously than phase fused lasso. This is the reason why phase fused lasso has a better overall performance than MPLS. In addition, the RMSE values of phase fused lasso stabilizes after the 298th sampling interval, indicating that the final prediction of the product quality can be made at that moment instead of the last time point of the entire batch run. More detailed analysis will be provided in the next section.

Process understanding via model interpretation

For model interpretation, the regression coefficients of the phase fused lasso model are plotted in Figure 10. In these figures, the horizontal coordinate corresponds to the sampling intervals, while the vertical coordinate represents the regression coefficients. Four of the totally eight process variables are plotted. Since the trajectory of the valve opening 2 synchronizes with that of the valve opening 1, the corresponding coefficients of these two variables are also same. Therefore, only one of them is plotted. Meanwhile, the barrel temperature variables are closed-loop controlled throughout the entire batch operation. As a result, the values of these variables are almost constant. The coefficients corresponding to these variables are not plotted either. Figure 10 well

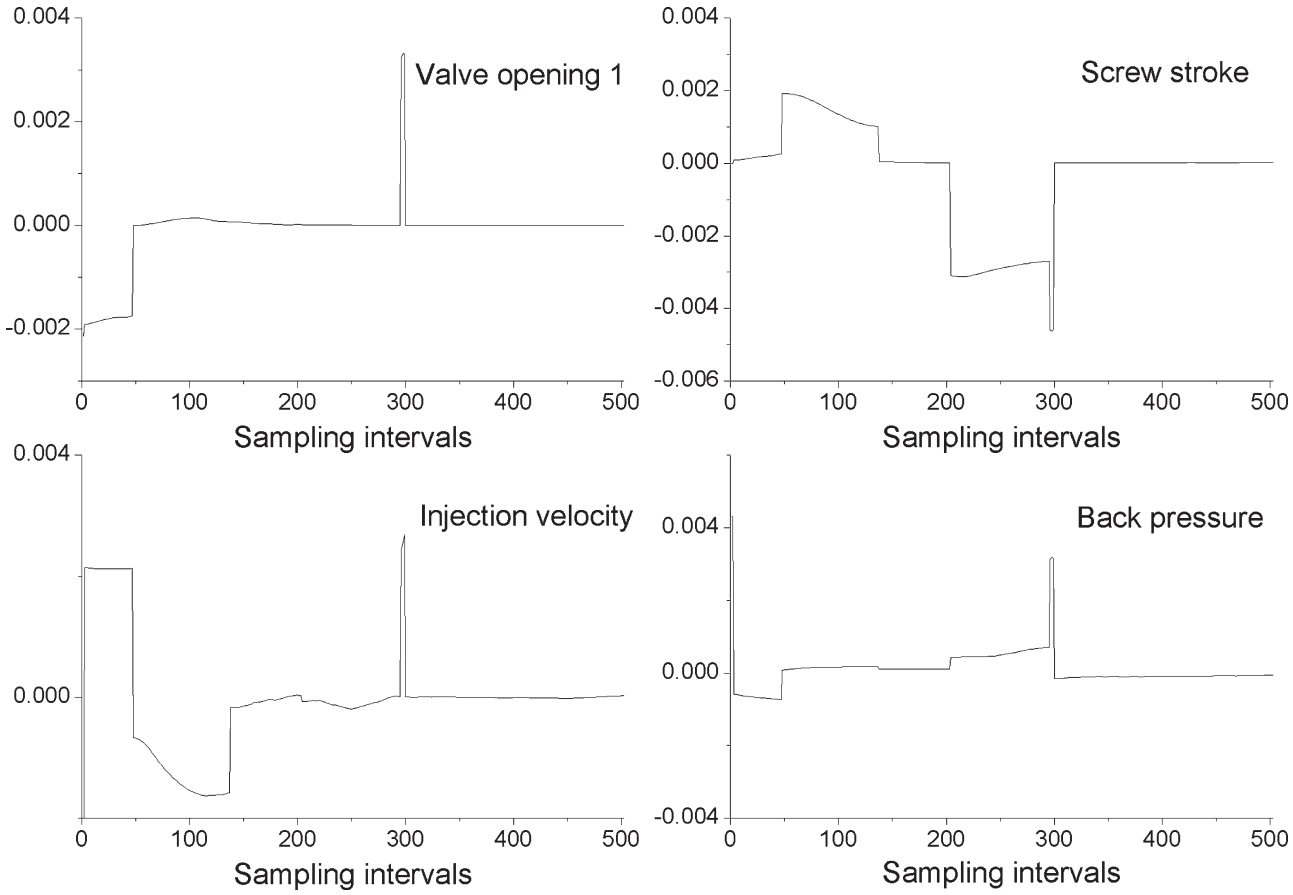

Figure 10. Regression coefficients of the phase fused lasso model.

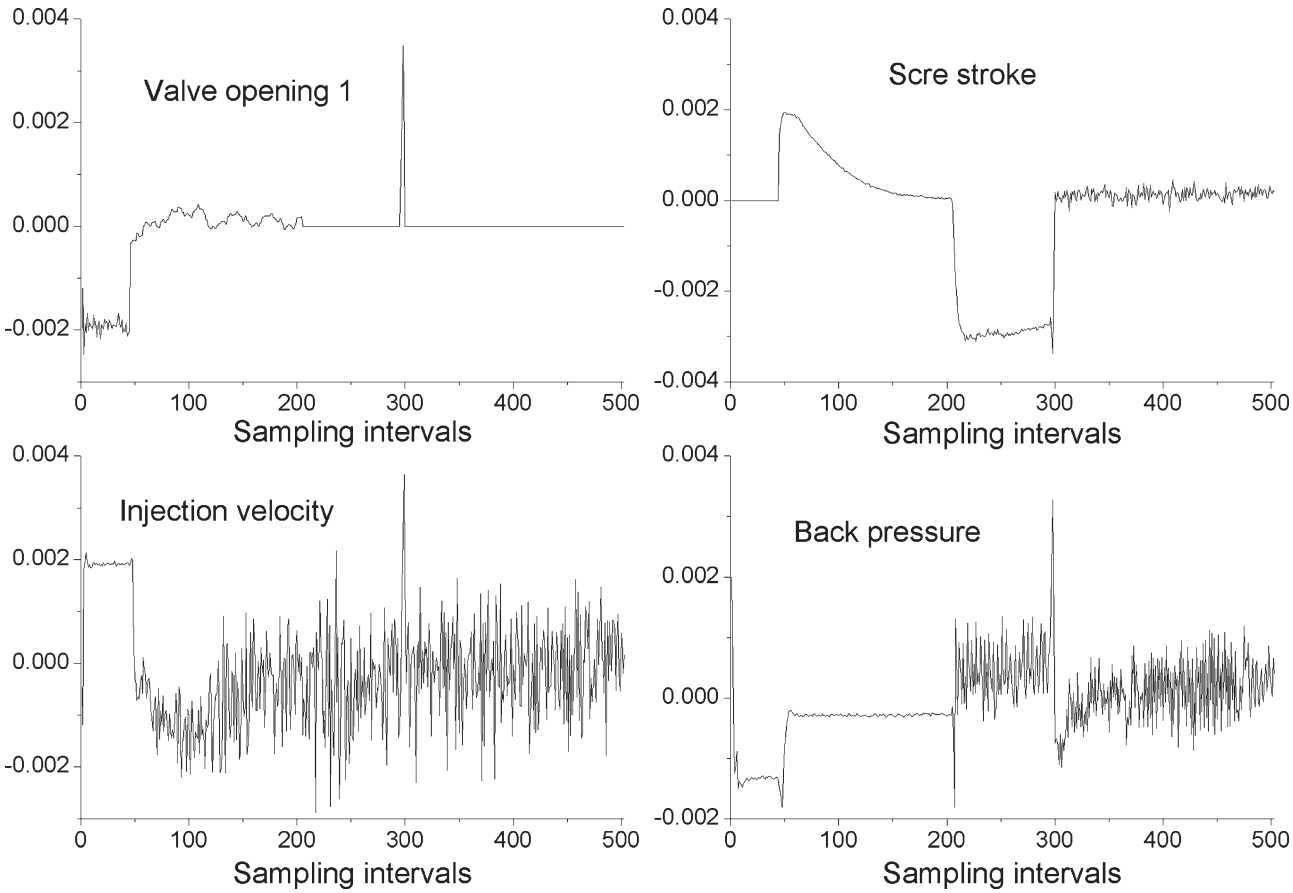

Figure 11. Regression coefficients of the MPLS model.

reflects the changing characteristics of the injection molding process and are beneficial to process understanding. Please be noted that the direction toward the nozzle is defined as the negative direction of both injection velocity and screw stroke in the measurement system of this process.

During filling (sampling intervals 1–47), injection velocity contributes most to the product weight, while the valve opening and the back pressure also show significant impacts. Considering the coordinate of the measurement system, the positive regression coefficients of the injection velocity indicates that faster filling corresponds to lighter product weight. Usually, people think that faster filling can inject more melt into the mold cavity in a unit time span and should lead to heavier product. However, the model coefficients tell us that this is a misunderstanding. First, the variations in injection velocity cause different lengths of the filling phase. Faster injection leads to a shorter filling time, and thus may not convey more melt during the entire filling phase. Second, due to the material property of HDPE, the gate between the runner and the cavity cannot entirely freeze at the end of the packing-holding phase. Therefore, it is possible for the melt to reflux after packing-holding, which lead to a weight loss. If the injection velocity is smaller, more time is spent on filling, which provides more opportunities for the melt in the cavity to cool down before plastication. In a consequence, the probability of the gate to freeze is increased, leading to less weight loss in the final product. Since the injection velocity is manipulated by the valve opening and the back pressure, the significant correlation between these two variables and the product weight is easy to understand.

As introduced in previous, the operation phase of packing-holding (sampling intervals 48–204) has been divided into two modeling phases. Figure 10 shows that these two modeling phases have different process characteristics. During the former phase, the screw stroke positively correlates to the product weight. Taking the coordinate of the measurement system into consideration, this means that larger displacement toward the nozzle indicates lighter final product. At a first glance, such results seem strange, because people usually think that, if the screw moves forward longer, more melt can be conveyed into the mold cavity and the final product should become heavier. In fact, such inference is only half correct. Large values of the screw stroke also indicate small melt density. Because weight is equal to volume multiplied by density, more melt in volume does not necessary mean larger weight. The back pressure shows little contribution to the product weight, which means that the main driving force for packing the melt into the mold cavity is the shrinkage of the material in the cavity instead of the magnitude of back pressure. In the latter half period of packing-holding, all regression coefficients are near zero. The fundamental behind such phenomenon is that there is no more space in the cavity for packing. In other words, packing has been completed, and the process switches to holding. This is the reason why this operation phase is divided into two modeling phases. The above model interpretation shows that such division well conforms to the process knowledge.

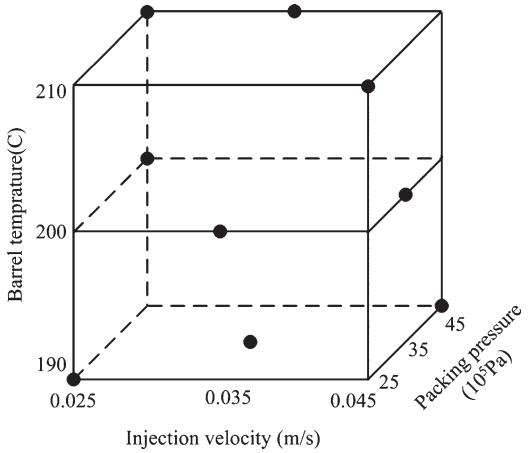

Figure 12. Operating conditions involved in training dataset.

During plastication (sampling intervals 205–300), the valve opening is kept constant, whose regression coefficients are hence zero. The difference in the back pressure values between different batches is mainly caused by the variations in melt densities. Higher back pressure indicates larger density. Therefore, the back pressure positively correlates to the product weight. The screw stroke is the most important variable in such phase, since its coefficients have the largest absolute values. The reason is as following. In the plastication phase, the screw moves backward from a position near the nozzle to a predetermined position. Therefore, the measured values of the screw stroke highly depend on the position of the screw at the beginning of this phase, which correlate to the distance that the screw goes in the filling and packing-holding phases and are good indicators of the amount of the melt injected into the mold cavity.

In the studied injection molding process, the gate freezes during plastication. As a result, no melt can flow into or out of the mold cavity after the 300th sampling interval, and the product weight cannot be changed any longer. In the phase fused lasso model, all regression coefficients are penalized toward zero, indicating this phase is noncritical to quality. Accordingly, one can inferred that it is impossible to implement real-time quality control by online set point adjustment in the cooling phase.

For comparison, the MPLS model coefficients are plotted in Figure 11. It is noted that the coefficients corresponding to the injection velocity and the back pressure look very "noisy" especially in the plastication and cooling phases. This is the reason why the online prediction accuracy of MPLS keeps decreasing during these two phases. In addition, most MPLS coefficients in cooling are not zero, creating a wrong impression that the process variables still have chance to influence the product weight during such phase. Obviously, the model interpretation of MPLS is worse than that of phase fused lasso. If the back pressure shifts in the cooling phase, there is a large chance for the MPLS model to indicate a change in the product weight, due to the nonzero regression coefficients. In this situation, the operators may think that the product quality is below grade, although it is not the case. Meanwhile, the nonzero coefficients of the MPLS model indicate that quality control is still possible in such phase. However, such conclusion is incorrect. Any attempt of quality adjustment in the cooling phase is impossible to succeed.

Prediction for new operating conditions

In this section, the predictive ability of the phase fused lasso method is verified with data from new operating conditions which are not used in model training. In this case study, 9 out of 27 operating conditions described previously are involved in the training dataset, which are formed according to a fractional factorial design[24] and indicated by black points in Figure 12. For each of the nine selected operating conditions, five batches are included in the training dataset, while the other five serve as test data. For each operating condition not included in model training, five batches are selected randomly as test data. Consequently, there are 45 batches contained in the training dataset, while the test dataset includes totally 135 batches.

Based on the GCV criterion, the parameters in the phase fused lasso model are specified as $\lambda_1 = 1 \times 10^{-3}$ and $\lambda_2 = 2$. Again, the model prediction error is characterized with RMSE. For the training data used in process modeling, the value of RMSE equals to 0.0184. In comparison, the RMSE for the entire test dataset is 0.0217. Taking the constitution of the test data into account, such results are quite satisfactory. In order to investigate further, the test data are analyzed as two groups: the 45 batches collected under the same operating conditions as the training data versus the 90 batches collected under new operating conditions. The RMSE values for these two groups are 0.0200 and 0.0226, respectively, which are comparable.

It should be pointed out that the proposed phase fused lasso belongs to the family of statistical regression. A common supposition of any statistical modeling methods is that the training data should cover the entire area of interest, where the model is expected to operate. In other words, the success of the statistical models depends on the coverage of training data. Therefore, no statistical regression method can guarantee its performance in the situation of significant extrapolation. Nevertheless, the above case study shows that, if the training data have a reasonable distribution, the phase fused lasso model works well for interpolation and slight extrapolation.

Conclusions

For the final product quality prediction of batch processes, MPLS and phase-based PLS are two typical types of modeling methods. However, each method has certain shortcomings limiting their performance in quality prediction and model interpretation. In this article, a phase fused lasso approach is proposed to solve the problems of the existing methods. Both the phase characteristics and time-dependent information are utilized in phase fused lasso modeling. Especially, by involving a lasso-type L_1 penalization term, phase fused lasso is able to select critical-to-quality phases automatically, while the regression coefficients corresponding to the irrelevant phases are shrunk to zero. In the meantime, the L_2-fusion term in the objective function smoothes the regression coefficients within each relevant phase. By doing so, the model interpretation is further improved. As components of the entire phase fused lasso modeling scheme, an efficient computation algorithm together with a GCV parameter selection procedure is developed. Besides, an MPLS-based phase division method is introduced, and an FS-kNN future data estimation approach is proposed for online quality prediction. The case study on an injection molding

process verifies the feasibility and effectiveness of the phase fused lasso method. The application results show that phase fused lasso not only provides better prediction in the situation that the process measurements are noisy, but also enhances process understanding which is very helpful for online quality control. Such method is practically useful and can be used in other chemical engineering problems.

Acknowledgments

This work was supported in part by the Ministry of Economic Affairs, R.O.C. under Grant No. 103-EC-17-A-09-S1-198 and in part by the Ministry of Science and Technology, R.O.C. under Grant No. NSC 102–2221-E-007–130-.

Literature Cited

1. Kourti T. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *J Chemom*. 2003;17:93–109.
2. Nomikos P, MacGregor J. Multi-way partial least squares in monitoring batch processes. *Chemom Intel Lab Syst*. 1995;30:97–108.
3. Gunther JC, Conner JS, Seborg DE. Process monitoring and quality variable prediction utilizing PLS in industrial fed-batch cell culture. *J Process Control*. 2009;19:914–921.
4. Yao Y, Gao F. A survey on multistage/multiphase statistical modeling methods for batch processes. *Annu Rev Control*. 2009;33:172–183.
5. Lu N, Gao F. Stage-based process analysis and quality prediction for batch processes. *Ind Eng Chem Res*. 2005;44:3547–3555.
6. Ge Z, Zhao L, Yao Y, Song Z, Gao F. Utilizing transition information in online quality prediction of multiphase batch processes. *J Process Control*. 2012;22:599–611.
7. Zhao C, Gao F, Sun Y. Between-phase calibration modeling and transition analysis for phase-based quality interpretation and prediction. *AIChE J*. 2013;59:108–119.
8. Westerhuis J, Kourti T, MacGregor J. Analysis of multiblock and hierarchical PCA and PLS models. *J Chemom*. 1998;12:301–321.
9. Chiu C-C, Yao Y. Multiway elastic net (MEN) for final product quality prediction and quality-related analysis of batch processes. *Chemom Intell Lab Syst*. 2013;125:153–165.
10. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc: Series B Stat Methodol*. 2005;67:301–320.
11. Boggia R, Forina M, Fossa P, Mosti L. Chemometric study and validation strategies in the structure-activity relationships of new cardiotonic agents. *Quant Struct-Act Relat*. 1997;16:201–213.
12. Forina M, Casolino C, Pizarro Millan C. Iterative predictor weighting (IPW) PLS: a technique for the elimination of useless predictors in regression problems. *J Chemom*. 1999;13:165–184.
13. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*. 1970;12:55–67.
14. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Series B Methodol*. 1996:267–288.
15. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *J R Stat Soc Series B Stat Methodol*. 2006;68:49–67.
16. Hebiri M, van de Geer S. The smooth-lasso and other $\ell 1+ \ell 2$-penalized methods. *Electron J Stat*. 2011;5:1184–1226.
17. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. *J R Stat Soc Series B Stat Methodol*. 2005;67:91–108.
18. Wang X, Park T, Carriere K. Variable selection via combined penalization for high-dimensional data analysis. *Comput Stat Data Anal*. 2010;54:2230–2243.
19. Bach FR. Consistency of the group lasso and multiple kernel learning. *J Mach Learn Res*. 2008;9:1179–1225.
20. Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*. 1979; 21:215–223.
21. Li K-C. Asymptotic optimality for C_p, C_L, cross-validation and generalized cross-validation: discrete index set. *Ann Stat*. 1987;15:958–975.
22. Jain A, Murty M, Flynn P. Data clustering: a review. *ACM Comput Surv*. 1999;31:264–323.
23. Nomikos P, MacGregor J. Multivariate SPC charts for monitoring batch processes. *Technometrics*. 1995;37:41–59.
24. Montgomery D. *Design and Analysis of Experiments*, 6th ed. New York: Wiley, 2005.

Manuscript received Nov. 8, 2013, and revision received Apr. 1, 2014.