Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Stage-Oriented Statistical Batch Processes Monitoring, Quality Prediction and Improvement

Yuan Yao* and Furong Gao**

** Dept. of Chemical Engineering, Hong Kong University of Science and Technology, Clear water bay, Kowloon, Hong Kong SAR, P. R. China,*
*** Dept. of Chemical Engineering, Hong Kong University of Science and Technology, Clear water bay, Kowloon, Hong Kong SAR, P. R. China*
(Tel: +852-2358-7139, Fax:+ 852-2358-0054, e-mail: kefgao@ ust.hk).

Abstract: Multistage/multiphase is an inherent characteristic of many batch processes, which should be taken into consideration to ensure better batch process monitoring, analysis, quality prediction and improvement. In this paper, a series of stage-oriented multivariate statistical methods on such topics are reviewed. Then, the modeling problem of transition between stage and stage is discussed. A new proposed stage and transition identification and modeling method is introduced as a necessary complementary of the existing work.

1. INTRODUCTION

With more and more wide applications of batch processes in today's industrial manufacture, there are increasing requirements of effective batch process monitoring, quality prediction and improvement. Due to the process high dimensionality, complexity and limited product-to-market time, it is difficult to build batch process models based on first-principle or process knowledge. Therefore, the applications of multivariate statistical modeling methods, which only require process history data, have attracted many research interests. Among them, multiway principal component analysis (MPCA) and multiway partial least square (MPLS) are most widely used (Nomikos and MacGregor, 1994, 1995a, 1995b).

For many batch processes, multistage/multiphase is an inherent characteristic, which means the batch processes are carried out in a sequence of steps. Steps occurring in a single processing unit are called phases; on the other hand, steps occurring in different processing units are called stages (Undey and Cinar, 2002; Wang, et al., 2007). Since MPCA and MPLS methods take the entire batch data as a single object, the unique process correlation information of different stages is not reflected. This not only makes difficulties on understanding of process characteristics, but also affects monitoring efficiency and quality prediction ability.

In batch process monitoring, Undey and Cinar (2002) demonstrated that "local models were proven to be advantageous when different phases existing in process stages and when precise phase separation is crucial". However, in many research works, the separation of stages is based on prior process knowledge which may not be available in many batch processes. At the same time, the stage models of MPCA type inherit the common weakness of MPCA that the unavailable future measurements should be estimated for online monitoring of a batch process. Using adaptive hierarchical PCA (AHPCA) (Rannar et al., 1998; Westerhuis et al., 1998) can avoid such estimation. But at the same time, the computation burden is increased significantly.

For quality prediction of multistage/multiphase batch processes, multiblock PLS can be applied to include the stage information (MacGregor et al., 1994; Kourti et al., 1995). However, the multiblock strategies still use the entire batch data together in modeling, and relate all of them to end qualities. The inherent different characteristics and different behaviours of each stage can not be reflected clearly. Facco et al. (2007) developed multi-phase PLS model for a batch polycondesation reaction producing a resin. However, the phase division is still based on the process knowledge.

To model multistage/multiphase batch processes more reasonably, a series of research works have been published. The first work was published in 2004 and a sub-PCA model was developed for batch process monitoring (Lu et al., 2004a). In this work, a data-based method was proposed to identify the stages and monitor each stage with a sub-PCA model which does not need future data estimation. This method was applied to the analysis and online monitoring of an injection molding process (Lu et al., 2004b). Then, since uneven batch duration and uneven stage duration are very common situations in real industries, the stage-based sub-PCA method was extended into uneven-length batch process monitoring (Lu et al., 2004c). In the same year, another method was proposed to build stage-based PCA model with limited batch data (Lu et al., 2004d). Similar to sub-PCA, stage-based PLS model was then developed to interpret the batch processes better and get more accurate and efficient online quality prediction results (Lu and Gao, 2005). With the online quality prediction, online quality control can be performed (Lu and Gao, 2006).

There are also some other related works about this topic. Camacho and Picó (2006a, 2006b) proposed multi-phase

PCA (MPPCA) method which detect phases based on the comparison of prediction errors of PCA models on the measurements at each sampling time interval. Most recently, Zhao et al. (2007) model transitions from stage to stage by a soft-transition multiple PCA (STMPCA) method.

This article is organized as following. In section 2, a brief introduction of the PCA/PLS methods is given. Then, two common methods for batch process data matrix unfolding are reviewed in section 3. A review of the series of stage-oriented statistical methods for batch process modeling, monitoring, quality prediction and control is done in the following section. In section 5, the issue of transition modeling in multistage/multiphase processes is discussed, and a new method for stage division, transition identification and modeling is proposed. Different from the existing method, the new proposed one is based on the angels between the latent spaces of time-slice PCA models. In the last section, a conclusion is made.

2. INTRODUCTION OF PCA AND PLS

2.1 Principal component analysis (PCA)

PCA is a method performed on a data matrix like $X(m\times n)$, where n is the number of samples and m is the number of variables. PCA decomposes X as

$$X = TP^T = \sum_{j=1}^{m} \mathbf{t}_j \mathbf{p}_j^T = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \cdots + \mathbf{t}_m\mathbf{p}_m^T, \quad (1)$$

where $\mathbf{t}_j(n\times 1)$ is principal component vector which is also named score vector or latent vector, $\mathbf{p}_j(n\times 1)$ is loading vector which projects data into score space and contains variable correlation information, and T and P are score matrix and loading matrix respectively. Scores are orthogonal to each other and loadings are orthonormal.

Algebraically, $\|\mathbf{t}_j\|$ is equal to the jth largest eigenvalue of the covariance matrix $\Sigma=X^TX$, and $\mathbf{p}_j$ is the corresponding eigenvector. The first PC t_1 has the maximum variance subject to $\|\mathbf{p}_1\|=1$, the second PC t_2 has the maximum variance subject to $\|\mathbf{p}_2\|=1$, and other PCs are defined in similar way. So it is easy to understand that the first several PCs contain most variance information of X when variables are correlated to each other, and the last several PCs only contain measurement noise. Thus, most variance information is able to be extracted with only first a few PCs, and the dimensions of variables are largely reduced.

By retaining the first A PCs, X can be approximated as

$$\hat{X} = TP^T = \sum_{j=1}^{A} \mathbf{t}_j \mathbf{p}_j^T . \quad (2)$$

Then

$$X = TP^T = \sum_{j=1}^{A} \mathbf{t}_j \mathbf{p}_j^T + \sum_{j=A+1}^{m} \mathbf{t}_j \mathbf{p}_j^T = \hat{X} + E, \quad (3)$$

where E is the residual matrix.

Several algorithms were developed for loading matrix calculation (Jackson, 1991; Jolliffe, 2002). To determine the proper retained number of PCs, many methods were proposed (Wold, 1978; Jackson, 1991; Jolliffe, 2002).

For process monitoring, two statistics are calculated after performing PCA. T^2 summarizes variation information contained in scores and *SPE* summarizes residual information (Jackson, 1991). The way to calculate the control limits of T^2 and *SPE* can be found in some books and articles (Jackson and Mudholkar, 1979; Jackson, 1991). In online and offline process monitoring, the values of these two statistics are compared with corresponding control limits to check whether the process is in control or not.

2.2 Partial least square (PLS)

PLS works on two data matrices. In multivariate process analysis, one of them is usually a process variable data matrix $X(n\times m_x)$, and the other is a product quality data matrix $Y(n\times m_y)$, where n is number of samples, m_x is number of process variables, and m_y is number of quality variables. Different from PCA, PLS not only extracts the variation of X, but also gives as much prediction to Y as possible.

PLS model includes descriptions of outer relations and inner relation. The equations are like below.

Outer relations:

$$\begin{aligned} X &= TP^T + E = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T + E \\ Y &= UQ^T + F = \sum_{a=1}^{A} \mathbf{u}_a \mathbf{q}_a^T + F \end{aligned} . \quad (4)$$

Inner relation:

$$\hat{\mathbf{u}}_a = b_a \mathbf{t}_a , \quad (5)$$

where $b_a=\mathbf{t}_a^T\mathbf{u}_a/(\mathbf{t}_a^T\mathbf{t}_a)$ is the regression coefficient between two groups of latent variables $\mathbf{t}_a$ and $\mathbf{u}_a$.

A PLS model also can be written in a compact way as

$$Y = X\Theta^T + F^*, \quad (6)$$

where Θ is a regression parameter matrix.

Wold et al. (1984), Geladi et al. (1986) and Hoskuldsson (1988) introduced the properties, calculation methods

(NIPALS) of PLS and the method to choose the number of latent variables (cross-validation, jackknife and so on). Dayal and MacGregor (1997) developed another algorithm called Kernel PLS to calculate the model.

2.3 Normalization

Before performing PCA or PLS, normalization is a necessary step to eliminate the effects of variable units and measuring ranges (Jackson, 1991). The most common way of normalization includes removing means and equalizing variances. The formula is shown below.

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \overline{x}_j}{s_j} \quad (i = 1, \cdots, I;\ j = 1, \cdots, J)\,, \tag{7}$$

where i is the sample index, j is the variable index, $\overline{x}_j$ is the mean value of variable x_j, and s_j is the standard deviation of variable x_j.

3. BATCH PROCESS DARA MATRIX UNFOLDING

Batch process data are often stored as a three-way matrix $\underline{X}(I \times J \times K)$, where I is the number of total batches, J is the number of variables, and K is the number of total sampling time intervals in a batch. In order to apply multivariate process monitoring methods, such as PCA and PLS, this three-way matrix is required to be expanded into a two-way one. There are two common ways to expand the matrix.

The most widely used method keeps the dimension of batches, and merges variables and time dimensions. Each row of the unfolded matrix $X(I \times KJ)$ contains all data within a batch. After unfolding, normalization can be performed on this two-way matrix and the mean trajectories of all variables are removed from the data. By doing so, the differences between batches are highlighted. This method can be called batch-wise unfolding. MPCA and MPLS perform PCA and PLS algorithm on such expanded matrix (Nomikos and MacGregor, 1994, 1995a, 1995b).

Variable-wise unfolding keeps the dimension of variables, and merges the other two dimensions (Wold et al., 1998). Each sampling point of each batch is considered as an object. The process data in unfolded matrix $X(KI \times J)$ can be normalized to zero mean and unit variance. After normalization, the grand mean of each variable over all time and all batches are removed, and the trajectories are left in the data matrix.

4. REVIEW OF STAGE-ORIENTED STATISTICAL MODELING METHODS

4.1 Stage- based sub-PCA method

The major motivation of developing a stage-based sub-PCA modeling method (Lu et al., 2004a) is like following. In multistage/multiphase batch processes, different stage/phase can have different variable correlations. The changes in correlation structure reflect the changes in process nature, and indicate stage changes. Using separate stage models can describe such characteristics better and lead to better monitoring.

The basis of sub-PCA model building is two levels. Firstly, a batch process may be divided into several stages corresponding to the process variable correlation changes. Secondly, within each stage, the process correlation nature is similar although the process may be time varying. Therefore, stage-based sub-PCA models can be built. One thing to emphasize is that, the stages identified based on correlation information may not be corresponding to certain operation stages/phases exactly.

There are several major steps in sub-PCA modeling procedures, which are batch process data matrix unfolding, stage division, and stage sub-PCA models building.

As introduced above, different stages can be indicated by the correlation structure changes along time direction in a certain batch. It is noticed that, in PCA analysis, the loading matrix extracts variable correlation information. For a batch process data matrix $\underline{X}(I \times J \times K)$, each vertical slice $\tilde{X}^k(I \times J)$ is a time-slice data matrix. Suppose the data have been normalized in batch-wise way, by performing PCA algorithm on these time-slice matrices, the variable correlation information on each time interval is contained in loading matrices $\tilde{P}^k$. This means, in the same stage, the time-slice loading matrices are similar, while different stages have different loadings. The formula of time-slice PCA models is like (1),

$$\tilde{X}^k = \tilde{T}^k (\tilde{P}^k)^T \qquad (k = 1, 2, ..., K)\,. \tag{8}$$

Since each column of a loading matrix contains different amount of process variance information, the time-slice loading matrices $\tilde{P}^k$ are transformed into a weighted form with the importance of each column taken into consideration.

$$\breve{P}^k = [\mathbf{p}_1^k \cdot g_1^k, \mathbf{p}_2^k \cdot g_2^k, ..., \mathbf{p}_J^k \cdot g_J^k]\,, \tag{9}$$

where $\mathbf{p}_j^k$ is the jth column of $\tilde{P}^k$, $g_j^k = \lambda_j^k / \sum_{i=1}^{J} \lambda_i^k$, and λ_j^k is the eigenvalue of the covariance matrix $(\tilde{X}^k)^T \tilde{X}^k$.

Then the stage division result can be achieved by comparing all the time-slice weighted loading matrices. Such compassion is carried on with k-means clustering method (Jain et al., 1999). The Euclidean distance between two weighted loading matrices is used to assess the dissimilarity. An important parameter in the clustering is the threshold of the minimal distance between two clusters' centers. Larger threshold leads to fewer clusters. This algorithm clusters the time slices with similar correlation structures together. A

stage can be identified with a series of successive samples in the same cluster. Therefore, the stage division results are achieved based on the clustering result associated with operation time information.

After stages are separated, a sub-PCA model for each stage is then built by taking the average of time-slice PCA loadings in the corresponding stage.

$$P_c^* = \frac{1}{n_c}\sum_{k=1}^{n_c} \tilde{P}_c^k \qquad (c = 1, 2, ..., C), \tag{10}$$

where C is the total number of stages identified, n_c is the number of time slices in stage c, P_c^* is the stage sub-PCA loading matrix for stage c and $\tilde{P}_c^k$ is the kth time-slice loading matrix in stage c. The singular-value diagonal matrix S_c^* can be defined similarly.

$$S_c^* = \frac{1}{n_c}\sum_{k=1}^{n_c} \tilde{S}_c^k = \text{diag}(\lambda_1^{*c}, \lambda_2^{*c}, ..., \lambda_j^{*c}) \quad (c = 1, 2, ..., C), \tag{11}$$

where $\tilde{S}_c^k = \text{diag}(\lambda_1^k, \lambda_2^k, ..., \lambda_j^k)$ is the kth time-slice singular-value diagonal matrix in stage c. The number of retained principal components A of each sub-PCA model can be calculated based on the cumulative explained variance rate as

$$A = \min_A \left(\sum_{i=1}^{A} \lambda_i^{*c} / \text{trace}(S_c^*) \geq 90\% \right). \tag{12}$$

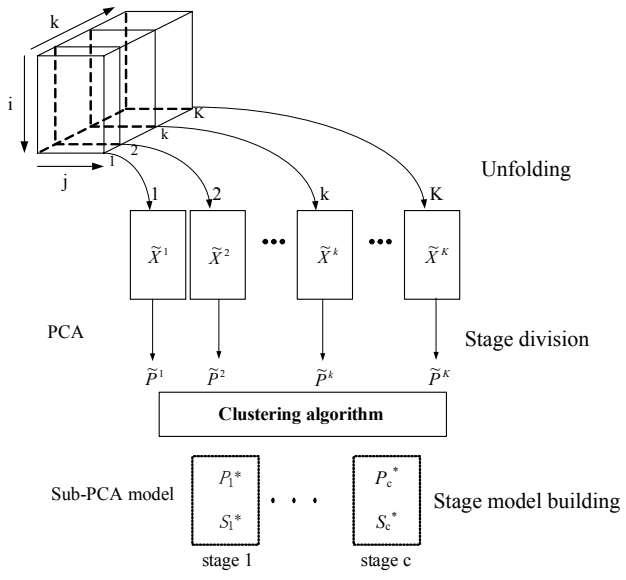


Fig. 1. Major steps of stage-based sub-PCA modeling

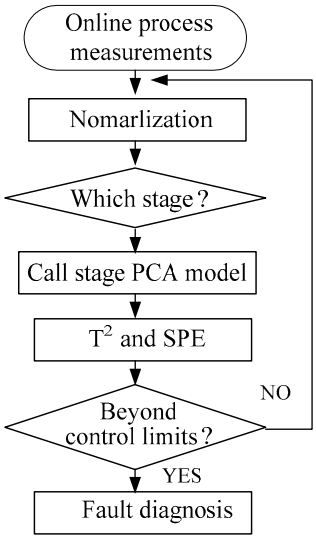


Fig. 2. Sub-PCA based online batch process monitoring

With the stage sub-PCA models, the control limits of T^2 and *SPE* on each sampling interval are calculated for online monitoring.

For online monitoring, before calling a stage model, the current stage should be determined first. Since the stages have been associated with particular time span in the stage division step, the stage of new data could be easily found be checking the current time interval. Then, the sub-PCA model of corresponding stage is used to monitor the online process data. If there is a fault detected by T^2 or *SPE*, contribution plots are used to find the reason of the fault.

The major steps of sub-PCA modeling are shown in Fig. 1 and the procedure of sub-PCA based online batch process monitoring is shown in Fig. 2.

4.2 Extension of sub-PCA to uneven-length batch processes

Most multivariate statistical modeling methods for batch process monitoring and quality prediction are based on the assumption that the batch durations are same (Nomikos and MacGregor, 1994, 1995a, 1995b). The sub-PCA method mentioned in the last section also makes such an assumption. However, many industrial processes have different batch durations from run to run because of disturbances in operating conditions. Often, even the stage durations are not fixed either. The data of a typical two-stage uneven length batch process can be shown in Fig. 3.

In this figure, X_i means the data of the ith batch. K_s is the number of sampling intervals in the shortest batch. Durations A and C are the common part of stage I and II. In duration B, some batches are in stage I and some are in stage II. Duration D have incomplete data structure of stage II. From this figure, we can see the complexity of the data structure of uneven length multistage/multiphase batch processes. To solve the uneven length and multi-stage issues simultaneously, the

stage-based sub-PCA method is extended to uneven length batch processes (Lu et al., 2004c).

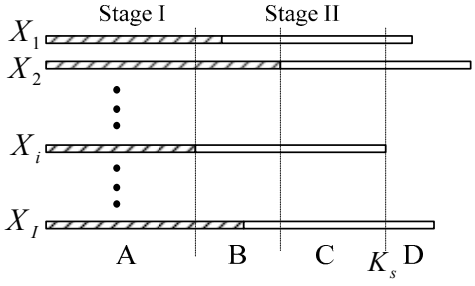


Fig. 3. An illustration of uneven-length batch process

The first important question to answer is how to normalize the data. The mean trajectories and standard deviations can not be calculated reasonably because of the uneven durations in each stage of different batches as shown in Fig. 3. So the batch-wise unfolding and normalization can not be performed directly before proper stage division. The variable-wise unfolding and normalization method can be used. However, this kind of unfolding will affect the monitoring efficiency because it focuses on the variations in variable trajectories along the time direction instead of the variations around the normal trajectories. So the variable-wise unfolding is utilized for stage-division purpose. Then, after stage division, the batch-wise unfolding is performed for stage model building.

The shortest batch length K_s is supposed to be known. After variable-wised unfolding and normalization, K_s time-slice PCA models are built. Then, k-means clustering algorithm is performed on the loading matrices of these time-slice PCA models. Since the stable clusters only could be found for the common part of stages, such as duration A and C in Fig. 3, the first stable cluster indicate the shortest duration of the first stage. Then a stage-division PCA model for stage I can be built. With this model, the duration of stage I in each batch can be known by checking the *SPE* value when all data go through this model. After that, the data of stage I can be re-normalized in the batch-wise way. The stage sub-PCA model for online monitoring is then built and the control limits are calculated. Then, the data of this stage are removed from the data set, and the above steps are repeated in an iterative way. So that, all stages can be identified and the sub-PCA models for these stages can be built.

Suppose the lengths of stage *c* are varied from $L^c_{\min}$ to $L^c_{\max}$. For the data belong to $[1, L^c_{\min}]$, the online monitoring can be performed just as the description in the original sub-PCA monitoring procedure. If the data belong to $(L^c_{\min}, L^c_{\max}]$, there are two possibilities if the *SPE* or T^2 values are outside of the control limits, a fault or a new stage. So the data is normalized and monitored with the stage model of stage *c*; if the *SPE* or T^2 values are outside the control limits, then the data are re-normalized and monitored with the model of stage *c*+1. If the *SPE* or T^2 values are within the control limits of stage *c*+1, the process enters the new stage. And if the statistics are outside the control limits again, an occurred fault is detected.

4.3 Stage- based batch process monitoring with minimal reference data

Multivariate statistical process modeling methods utilize normal history data which cover the whole normal operating region. To collect enough normal history data may be quite time consuming for some slow batch processes, such as some bio-processes. If there is a method to model such processes with limited batch cycle, for example, with the data from only one normal operating batch, this problem could be solved. With such motivation, the stage-based batch process monitoring method with minimal reference data is proposed (Lu et al., 2004d). The major difference between this method and sub-PCA is that a moving window of a batch is used to extract the local variable correlation structure information.

In this method, the modeling starts with the data from an arbitrary normal operating batch run. The data of this batch is stored as $X(K \times J)$. The moving window strategy is utilized. The data in each window form a two-way matrix $\tilde{X}^k(n \times J)$, where *k* is the index of window, *n* is the length of a window. Moving step can be set to 1. Therefore, there are (*K*-*n*) number of windows totally.

Then, the data in each window are normalized as following,

$$\begin{cases} \dfrac{x_j - \overline{x}_j}{s_j} & \text{if } s_j \geq \varepsilon \\ (x_j - \overline{x}_j)s_j & \text{if } s_j < \varepsilon \end{cases}, \qquad (13)$$

where *j* is a variable index, $\overline{x}_j$ is the mean value of x_j in that window, and ε is a small value smaller than 1. The purpose of doing so is to reduce the influences of the variables of little variation in the window. Therefore, the effects of random noise variations will not be amplified.

A PCA model is built for the data in each window and the local correlation information is extracted with the loading matrix. Then, the stage division can be conducted with clustering of these window-based PCA loading matrices, similar to the procedure in sub-PCA method.

The stage PCA models can be calculated based on the window-based PCA models in each stage, similar to (10) and (11). And the number of PCs is determined with (12). Then, the control limits of T^2 and *SPE* can be calculated accordingly. With more normal data coming, the data in each window is gradually filled with the new data, and the model focus more and more on the batch-to-batch variations. So the model precision is improved further with the model updating.

The online monitoring procedure is similar to the one used in sub-PCA based online monitoring.

4.4 Stage- based PLS modeling for batch process analysis and quality prediction

Online batch process quality measurement is difficult to conduct for the complicated process behaviors. Online quality prediction with multivariate statistical modeling is a very preferred way to solve this problem. However, the existing methods consider few of the effects of multistage/multiphase on quality (Nomikos and MacGregor, 1995b; Chen and Liu, 2002). In such batch processes, different stages/phases often have quite different effects on the final product qualities. Particular end-qualities may be determined in some particular stages/phases and by some particular process variables. Based on these findings, a stage-based process analysis and quality prediction method is developed (Lu and Gao, 2005).

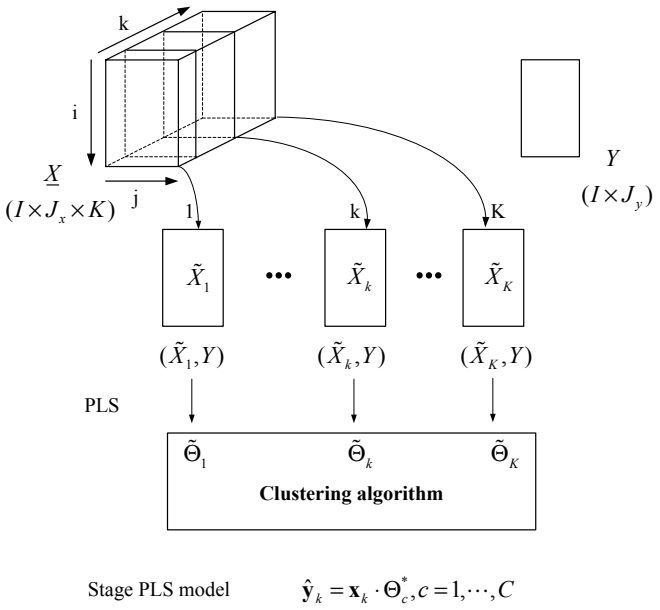


Fig. 4. Stage-based PLS batch process modeling

The first step in stage-based quality prediction is stage division. Similar to stage division based on sub-PCA method, the batch process data $\underline{X}(I\times J_x\times K)$ are normalized and divided into K time-slice matrices $\tilde{X}_k(I\times J_x)$. PLS algorithm (4) is applied to $\{\tilde{X}_k(I\times J_x)\quad Y(I\times J_y)\}$ at each time interval, where $Y(I\times J_y)$ is the data matrix storing the quality variables. A time-slice PLS model has the form as

$$Y=\hat{Y}+F^*=\tilde{X}_k\tilde{\Theta}_k^T+F^*, \tag{14}$$

where $\hat{Y}$ is the quality prediction, and the regression coefficient matrix $\tilde{\Theta}_k(J_x\times J_y)$ contain the correlation structure information between process variables and quality variables at the sampling interval k. The correlation structures are similar within the same stage, and quite different between different stages. Again, the k-means clustering algorithm is utilized for stage division based on $\tilde{\Theta}_k$.

Then the stage PLS models are calculated with the time-slice PLS models in each stage by

$$\Theta_c^*=\frac{1}{n_c}\sum\nolimits_{k=1}^{nc}\tilde{\Theta}_c^k \qquad (c=1,2,...,C), \tag{15}$$

where n_c is the number of sampling intervals in stage c. The quality prediction at each time interval is

$$\hat{\mathbf{y}}_k=\mathbf{x}_k\Theta_c^* \qquad (c=1,2,...,C;\ k=1,2,...,n_c). \tag{16}$$

The multiple coefficient of determination R^2 (Johnson and Wichern, 2002) is used to evaluate the fitness of the stage PLS models and determine the critical-to-prediction stages.

$$R_{jy,k}^2=1-\frac{\sum_{i=1}^{I}(y_{i,jy}-\hat{y}_{i,jy,k})^2}{\sum_{i=1}^{I}(y_{i,jy}-\overline{y}_{jy})^2} \qquad (j_y=1,...,J_y), \tag{17}$$

where $y_{i,jy}$ it the measurement of the j_yth quality in batch i, $\overline{y}_{jy}$ is the average of $y_{i,jy}$ of all batches, $\hat{y}_{i,jy,k}$ is the quality prediction at sampling interval k based on stage PLS model. The range of R^2 is from 0 to 1. Then the F-test is adopted to test the significance of R^2 (Globerg and Cho, 2004),

$$F=\frac{n-m-1}{m}\left(\frac{R^2}{1-R^2}\right). \tag{18}$$

If the average value of $R_{jy,k}^2$ in stage c is tested to be significant, stage c is the critical-to-prediction stage of y_{jy}.

The contribution rate of process variable x_{jx} to prediction $\hat{y}_{jy}$ can be calculated as

$$C_{\mathbf{y}_{jy},\mathbf{x}_{jx},k}=1-\frac{\|\mathbf{y}_{jy}-\hat{\mathbf{y}}_{jy,k}\mid\mathbf{x}_{jx,k}\|}{\|\mathbf{y}_{jy}\|}, \tag{19}$$

$$\hat{\mathbf{y}}_{jy,k}\mid\mathbf{x}_{jx,k}=[0,...,0,\mathbf{x}_{jx,k},0,...,0]\Theta_c^*(j_x,j_y), \tag{20}$$

where $\hat{\mathbf{y}}_{jy,k}\mid\mathbf{x}_{jx,k}$ is the contribution of x_{jx} to $\hat{y}_{jy}$. The value of $C_{\mathbf{y}_{jy},\mathbf{x}_{jx},k}$ indicates the significance of x_{jx} contribute to the

variation of y_{jy}. So large $C_{\mathbf{y}_{jy},\mathbf{x}_{jx},k}$ means the variable x_{jx} is the key process variable to quality y_{jy} in stage *c*.

Then the online quality prediction can be performed. There are two kinds of quality variables. One kind of qualities is determined by only one critical-to-prediction stage, and the other kind is determined by two or more stages.

For the first type, the quality can be predicted online as

$$\hat{y}_{jy,k} = \mathbf{x}_k \Theta_c^*(:, j_y) \quad \text{if stage } c \text{ is critical to } y_{jy}. \qquad (21)$$

If stage c is not critical to y_{jy}, then no prediction of y_{jy} is given in that stage. For the second type, since there are several stages affect the quality cumulatively, the prediction can be calculated with stacked modeling methods (Breiman, 1996). Without losing generality, suppose there are two stages c_1 and c_2 that are critical to y_{jy}, then the quality prediction in stage c_2 is calculated as

$$\hat{y}_{jy,k} = w_1 \hat{y}^*_{jy,c1} + w_2 \mathbf{x}_k \Theta^*_{c2}(:, j_y), \qquad (22)$$

where $\hat{y}_{jy,k}$ is the quality prediction at the end of stage c_1, Θ^*_{c2} is the regression coefficient matrix in stage c_2, w_1 and w_2 are stage weights and can be calculated with regression methods from

$$\hat{y}_{jy} = \sum_c w_c \hat{y}^*_{jy,c}. \qquad (23)$$

4.5 Stage- based online quality control for batch processes

Batch process online quality control can be conducted with a reliable online quality prediction. With the stage-PLS method, the stage/phase effects on quality variations are discovered. Therefore, an online batch process quality control method based on stage-based PLS models is proposed (Lu and Gao, 2006). In this method, the quality predictor and controller only work in some certain critical stages, since certain batch process quality may be only affected by some certain variables in certain stages.

The stage division and stage-based PLS model building steps are same with the one introduced in section 4.4, followed by the steps of critical-to-prediction stages identification. The critical-to-prediction stages with manipulated variables to affect the qualities are called critical-to-control stages. The determination of critical-to-control stages need some process knowledge which is known in the industry. Then, manipulated variables are selected in each critical-to-control stages with the help of some available process knowledge.

The no-control region of end qualities can be defined based on product specifications or normal history quality data. Then, regression models are built between the set points of manipulated variables and stage-concerned qualities in each critical-to-control stage.

In the procedures of online quality control, the quality prediction is performed on each sampling points in corresponding critical-to-control stages. And the control actions are conducted at decision points if quality prediction is found to be not in-control. The quality-control interval T_q defining the time of decision points is selected as the multiple times of sampling rate of online process measurement T_s. If there is only one stage affect certain qualities, the prediction can be done in that stage based on (21). If some qualities are determined by more than one stage, without losing generality, assuming that they are affected by two stages. Then, in the first stage, the prediction is like following:

$$\hat{y}_{jy,k} = w_1 \mathbf{x}_k \Theta^*_{c1}(:, j_y) + w_2 \hat{y}^*_{jy,c2}, \qquad (24)$$

where $\hat{y}^*_{jy,c2}$ is the contribution of future stage which can be estimated by assuming the future stage will be kept at the nominal operation conditions. In the second stage, the prediction can be calculated with (22).

The end-quality predictions are checked whether they fall into the on-control region based on (25),

$$(\hat{\mathbf{y}}_{kc} - \mathbf{y}_{sp})^T \mathbf{W}_1 (\hat{\mathbf{y}}_{kc} - \mathbf{y}_{sp}) > \delta \mathbf{y}^T_{jy,sp} \mathbf{W}_1 \mathbf{y}_{jy,sp}, \qquad (25)$$

where $\hat{\mathbf{y}}_{kc}$ is the average of prediction values of quality in a quality-control interval, $\mathbf{y}_{sp}$ is the quality set points, $\mathbf{W}_1$ is a diagonal weighting matrix indicating the importance of each quality variables, and δ is a small number specified based on the customer's need. If the quality prediction is out of control, the new set points of manipulated variables are changed to compensate for the quality loss. The set points can be calculated by solving the optimization problem,

$$\begin{aligned} J &= \min_{\Delta \mathbf{x}^*_{sp}} \{(Q_{comp} - Q_{loss})^2 + Q_{\cos t}\} \\ \Delta \mathbf{x}^*_{sp} &= \mathbf{x}^*_{sp,new} - \mathbf{x}^*_{sp,old} \\ \Delta \mathbf{x}^*_{sp,\min} &\le \Delta \mathbf{x}^*_{sp} \le \Delta \mathbf{x}^*_{sp,\max} \end{aligned}, \qquad (26)$$

where $\mathbf{x}^*_{sp,old}$ is old set point of manipulated variables while $\mathbf{x}^*_{sp,new}$ is the new one, $\Delta \mathbf{x}^*_{sp,\min}$ and $\Delta \mathbf{x}^*_{sp,\max}$ are hard constraints of manipulated variable adjustment, Q_{loss} is the cumulated quality loss up to current decision point, Q_{comp} is the desired quality compensation in the remaining period in

the batch run, and Q_{cost} is the cost of changing the set points of manipulated variables. Q_{loss}, Q_{comp} and Q_{cost} can be calculated as

$$Q_{loss} = \sum_{k=1}^{k_{c-q}T_q} (\hat{\mathbf{y}}_k - \mathbf{y}_{sp})^T \mathbf{W}_1 (\hat{\mathbf{y}}_k - \mathbf{y}_{sp}), \quad (27)$$

$$Q_{comp} = \sum_{k=1}^{k_{c-q+1}T_{q+1}} (\hat{\mathbf{y}}_k - \mathbf{y}_{sp})^T \mathbf{W}_1 (\hat{\mathbf{y}}_k - \mathbf{y}_{sp}), \quad (28)$$

$$Q_{cost} = (\Delta \mathbf{x}_{sp}^*)^T \mathbf{W}_2 (\Delta \mathbf{x}_{sp}^*). \quad (29)$$

When there is more than one stage affecting qualities, the $\hat{\mathbf{y}}_k$ in (27) can be calculated based on (24) and (22). However, the $\hat{\mathbf{y}}_k$ in (28) can not be calculated in this way, since the future measurements are not available at that moment. So the regression models between manipulated variables' set points and the qualities are used to calculate $\hat{\mathbf{y}}_k$ instead. $\mathbf{W}_2$ is a diagonal matrix associate with the manipulated variables' adjustment costs.

4.6 Other related method: multi-phase PCA (MPPCA)

MPPCA (Camacho and Picó, 2006a and 2006b) dose the phase division in another way. It is based on the prediction power of the PCA model. A division performance index β^t is proposed based on the prediction power,

$$\beta^t = 100(1 - \frac{SE_c^t}{SE_c}), \quad (30)$$

$$SE_c = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} (x_{ijk} - \hat{x}_{ijk})^2, \quad (31)$$

$$SE_c^t = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{t} (x_{ijk} - \hat{x}_{ijk}^1)^2 + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=t+1}^{K} (x_{ijk} - \hat{x}_{ijk}^2)^2, \quad (32)$$

where $\hat{x}_{ijk}$ is the predication of variable j at sample time k in batch i based on a PCA model with c PCs, SE_c is the squared prediction error of this PCA model, SE_c^t is the squared prediction error obtained if the batch data is divided into two phases at time t and modelled by two sub-models, $\hat{x}_{ijk}^1$ is the predication of the first sub-model and $\hat{x}_{ijk}^2$ is the predication of the second sub-model.

The major steps of MPPCA are like following. The batch process data matrix is unfolded firstly. Then the data is normalized and PCA model is built for them. Find the sampling time interval at which the division performance index β^t has the largest value, which indicates the best division point. Build a PCA model for each phase. If predictions are not improved much after phase division, then stop the division procedure. Otherwise, accept the division, and repeat the steps of PCA model building and phase division in each subdivision. After the above procedure is finished, if the number of phases is larger than a specified maximum number, some subdivisions with least prediction improvement are deleted.

5. TRANSITION IDENTIFICATION AND MODELING

5.1 Shortcoming of sub-PCA method

As discussed before, the sub-PCA method and its extensions divide batch processes into stages with k-means clustering algorithm which is a kind of hard partition. This kind of partition ignores the transition characteristics from stage to stage, which could be gradual changes and very common in multistage/multiphase batch processes. As pointed out by the authors (Lu et al., 2004a), "misclassification may occur at the beginning and end of each stage" and "it may lead to false alarm and missing alarm".

5.2 Existing method for transition identification and modeling: soft-transition multiple PCA (STMPCA)

To model and monitor the transitions better, Zhao et al. (2007) developed STMPCA method recently, which is a complement of sub-PCA.

After stage division as described in sub-PCA method, the Euclidean distances between the stage cluster centers and each time-slice PCA model are calculated as

$$d_{k,c} = \| \breve{P}^k - \breve{P}_c \|, \quad (33)$$

$$\breve{P}_c = \frac{1}{n_c} \sum_{k=1}^{nc} \breve{P}^k, \quad (34)$$

where k is the index of sampling interval, c is the index of stage, $\breve{P}^k$ is weighted loading matrix as defined in (9), $\breve{P}_c$ is the cluster center of the weighted loading matrices in stage c, and n_c is the number of sampling intervals in stage c.

Class radius $r_{c,i}$ and kernel radius $r'_{c,i}$ are utilized to define the range of transitions in stage c identified before,

$$r_{c,i} = \gamma_{c,i} \cdot \left\| \breve{P}_i - \breve{P}_c \right\|, \ 0.5 < \gamma_{c,i} < 1, \quad (35)$$

$$r'_{c,i} = \gamma'_{c,i} \cdot \left\| \breve{P}_i - \breve{P}_c \right\|, \ 0 < \gamma'_{c,i} < 0.5, \quad (36)$$

where i=c-1 or c+1. $\breve{P}_c$ is the cluster centre of current stage, $\breve{P}_i$ is the center of the neighbouring stage, γ_c and γ'_c are adjustable parameters defined by users. Therefore, the status of each sampling interval is defined as,

$$\mu_{k,c}=1, \mu_{k,j}=0 \ \ if\ d_{k,c}<r_c' \ or\ (d_{k,c}<r_c \ and\ d_{k,j}>r_j)$$
$$j=1,2,...,C, j\neq c, \quad (37)$$

$$\text{Otherwise, } \mu_{k,c}=d_{k,c\text{-}1}/(d_{k,c}+d_{k,c\text{-}1}),\ \mu_{k,c\text{-}1}=1-\mu_{k,c}, \mu_{k,j}=0,$$
$$j=1,2,...,C, j\neq c, j\neq c\text{-}1, \quad (38)$$

where $\mu_{k,c}$ is the membership grade of sampling interval k belonging to cluster c. So (37) defines that the points totally belong to some stage, and (38) describes the membership grades of points in transitions. Then, the transition models at every time interval are calculated as the weighted sum of stage models, and the weights are equal to the membership grades calculated in (38). The number of retained PCs is select to be equal to the larger one of its two neighbouring sub-PCA models.

5.3 Motivations of developing angle-based transition PCA model

Although the STMPCA proposed a way for transition modeling, some shortcomings still exist. The first problem is there are two parameters need to be defined in the calculation of class radius and kernel radius. However, there is no clear guideline for the parameters selection. If the parameters are not selected properly, the transition range identified could be incorrect. Besides this, the STMPCA calculates the membership grades based on the Euclidean distances between weighted loading matrices of time-slice PCA models and cluster centers. A question is whether the membership calculated in this way is optimal, which means whether the transition models calculated based on such membership grades have the best performance. We need a method to identify the transitions automatically without user specified parameters, and a transition modeling method which leads to optimal transition models. Here, an angle-based factor describing the similarities between PCA models is proposed to divide the stages, identify the ranges of transitions and build the transition models.

5.4 Similarity between PCA models

The similarity between PCA models is one of the important bases of the stage-orientated batch process monitoring, quality prediction and control methods. How to compare the similarity between PCA models? In the works mentioned in section 4, the Euclidean distances between weighted PCA loading matrices are used. However, since each loading matrix is weighted differently as shown in (9), the Euclidean distances calculated are not normalized values. This makes trouble in determination of the threshold of the minimal distance between two clusters' centers.

A very widely used index in PCA similarity comparison is the PCA similarity factor (Krzanowski, 1979), which is defined as

$$S_{PCA}(P_1,P_2)=\frac{\text{trace}(P_1^TP_2P_2^TP_1)}{k}=\frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k}\cos^2\theta_{ij} \quad (39)$$

where P_1 and P_2 are the loading matrices of two PCA models, k is the number of PCs, θ_{ij} means the angle between the direction of the ith PC in model 1 and the jth PCA in model 2. The range of S_{PCA} is from 0 to 1. However, there are two problems with this index. First, it dose not consider the different importance of each PC. Second, if two PCA models cover the same space, the value of S_{PCA} is always be one, no matter the directions of PC pairs of two PCA models are really same or not. A simple example can illustrate this. Suppose there are two loading matrix,

$$P_1=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \text{ and } P_2=\begin{bmatrix}\sqrt{2}/2 & -\sqrt{2}/2\\ \sqrt{2}/2 & \sqrt{2}/2\end{bmatrix}. \quad (40)$$

Then $S_{PCA}=1$, indicating two same models, although these two PCA models are very different.

To solve the first problem, a modified PCA similarity factor is proposed (Singhal and Seborg, 2002), which is

$$S_{PCA}^{\lambda}(P_1,P_2)=\frac{\text{trace}(Q_1^TQ_2Q_2^TQ_1)}{\sum_{i=1}^{k}\lambda_i^1\lambda_i^2}=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}(\lambda_i^1\lambda_j^2)\cos^2\theta_{ij}}{\sum_{i=1}^{k}\lambda_i^1\lambda_i^2}, \quad (41)$$

where $Q_j=P_j\Lambda_j$, $\Lambda_j=\text{diag}(\sqrt{\lambda_1^j},\sqrt{\lambda_2^j},...,\sqrt{\lambda_k^j})$, and λ_i^j is the ith biggest eigenvalue in the jth PCA model. This modified version takes the importance of each PC into consideration. However, the second problem is not solved well. Especially, if the values of eigenvalues are close to each other, it affects the clustering results quite significantly.

In this paper, a new PCA similarity factor is proposed to solve both problems. The formula is as below:

$$S_{PCA}^{new}(P_1,P_2)=\frac{\sum_{i=1}^{k}(\lambda_i^1\lambda_i^2)\cos^2\theta_{ii}}{\sum_{i=1}^{k}\lambda_i^1\lambda_i^2}. \quad (42)$$

The major difference between S_{PCA}^{new} and S_{PCA}^{λ} is that S_{PCA}^{new} only measures the angles between corresponding PC pairs in two models. This form can solve both problems. Since this factor always has a normalized value from 0-1, the shortcoming of using Euclidean distance in stage division is also overcome. It is much easier to set a threshold for it than the Euclidean distance. Since S_{PCA}^{new} is calculated based on the angles, it can be easily transformed to a degree in angle from 0 to 90°. Therefore, the threshold for clustering could also be given in a degree value of an angle between two PCA models which has clearer geometry meaning.

Because of these benefits, the new PCA similarity factor S_{PCA}^{new} is used in this work for stage division and transition modeling.

5.5 Transition model assessment

Before a new transition modeling method is proposed, a criteria need to be set to indicate how good a transition model is. Good transition PCA models should describe the process variable correlation structures accurately during the transition period, which means the transition PCA models should be as similar as possible to time-slice PCA models of the samples in transitions. Such similarity can be quantified with the new PCA similarity factor S_{PCA}^{new}. Therefore, different transition PCA modeling methods can be compared based on S_{PCA}^{new}.

5.6 Angle-based stage division and transition identification

Here, the new PCA similarity factor based stage division and transition identification method is proposed. The procedure is shown below.

1. Unfold and normalize batch process data matrix $\underline{X}(I \times J \times K)$ in batch-wise. Divide it into K time-slice matrices $\tilde{X}^k(I \times J)$.
2. Perform PCA on each time-slice data matrix.
3. Cluster the K loading matrices $\tilde{P}^k$ into clusters with k-means clustering algorithm. The values of new PCA similarity factor between loading matrices are used in clustering instead of distance. Associate the cluster information with operation time, and get the division of C stages. The number C is determined by choosing a threshold whose value is between 0 and 1. The threshold indicates the maximum similarity between two clusters.
4. In each stage, the stage-center PCA model is built by performing PCA on stage data matrix $X^c(In_c \times J)$, where n_c is the number of samples in stage c.
5. The similarity between each time-slice PCA model and the stage-center PCA model is calculated with the new PCA similarity factor. All these values are plot on a univariate control plot. The robust statistical method can be used in the calculation of control limits and the detection of outliers (Daszykowski et al., 2007).
6. The outliers occur at the beginning and end of each stage in sequences are identified as the points in transitions and removed from each stage. Then, the remained part of each stage is the range of a steady stage.

From the procedure, we can see the method of stage division is similar to the one used in sub-PCA. The only difference is that inputs of k-means clustering are the values of the new PCA similarity factor instead of distances. The step 5 makes use of univariate statistical monitoring method to identify the transitions. The basic idea is that in each steady stage, the values of similarity between time-slice PCA models and stage-center model should be similar, and their distribution can be approximated as normal distribution. In contrary, the similarities between the time-slice models in transitions and stage-center model could be quite different. Thus, the transition samples can be detected as outliers with control plots. Since the control limits are calculated statistically, the determination of the ranges of transitions is more objective.

5.7 Stage modeling and angle-based transitions modeling

After the identification of the ranges of steady stages and transitions, the procedure of modeling is given as following.

1. Build stage model P_c^* for each steady stage by performing PCA on the stage data matrix. Note that the transition points have been removed.
2. The PCA model of each time interval in transitions could be described as the weighted sum of the stage models before and after the transition. The weights could be calculated as,

$$(w_1, w_2) = \max_{\substack{w_1+w_2=1 \\ 0<=w_1,w_2<=1}} (S_{PCA}^{new}(\tilde{P}^k, w_1 P_c^* + w_2 P_{c+1}^*)), \qquad (43)$$

where $\tilde{P}^k$ is a time-slice model for a point in transition, P_c^* is the stage model before the transitions, P_{c+1}^* is the stage model after the transitions, $w_1$and w_2 are the weights calculated. Then the transition model of this time interval is formulated as,

$$P_{trans}^k = w_1 P_c^* + w_2 P_{c+1}^*. \qquad (44)$$

In the transition modeling, the weights are calculated by solving an optimization problem to ensure the transition models have high similarities with the time-slice PCA models. The benefit of using transition models instead of time-slice models is that each transition model only records two parameters $w_1$and w_2, which requires much smaller storage than using time-slice models.

5.8 Application results

The proposed method is applied to a closed-loop controlled three-tank system. There are five time varying process variables, including two flow rate Q1, Q2 and three levels h1, h2, h3. Two levels are controlled to the set points h1=300 mm and h2=200 mm from their initial conditions, while the other level h3 is left to float to reflect the process interactions. In each cycle, 120 samples are collected under 1 second sampling interval. For more detailed information of three-tank system, please refer to (Lu et al., 2004a).

The stage division result together with typical process variable trajectories is shown in Fig. 5. The threshold of maximum similarity between cluster centers is chosen as 0.12 which is 20degree in angle. The result is similar to the one based on Euclidean distance. However, since the value of the new PCA similarity factor is normalized, the choosing of the threshold is easier and more guided. Then the ranges of transitions are identified by outlier detection in each stage as introduced in section 5.6. After that, the stage and transition models are built. The similarities between the proposed transition models and time-slice models are calculated and compared with the similarities between the STMPCA models and time-slice PCA models. For comparison, the identified

ranges of transitions are assumed to be same when using different transition modeling methods.

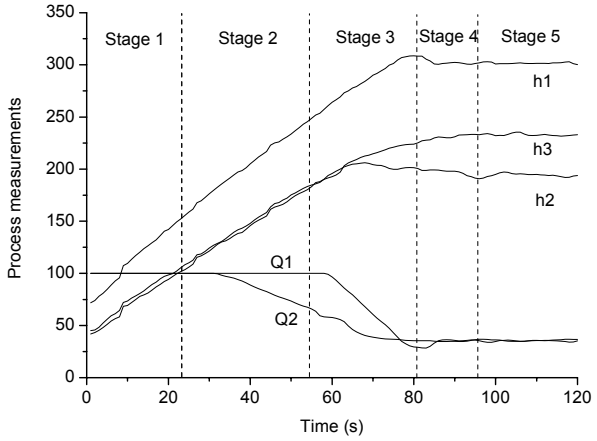


Fig. 5. Stage division based on S_{PCA}^{new}

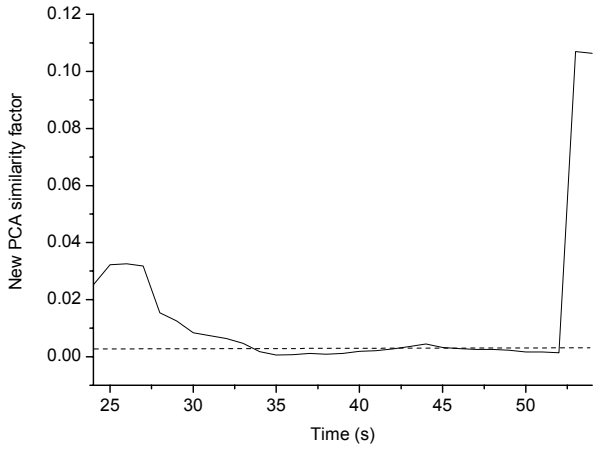
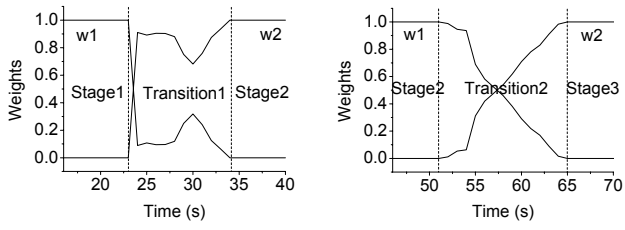
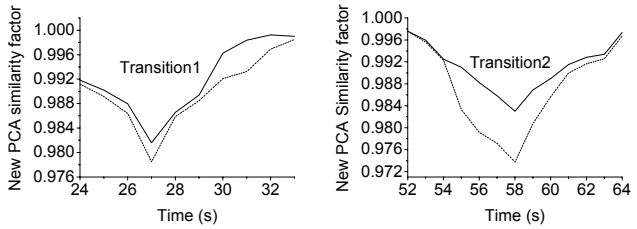


Fig. 6. Transition identification in stage 2



(a) (b)

Fig. 7. a. Modeling results of transitions from stage 1 to 2; b. modeling results of transitions from stage 2 to 3



(a) (b)

Fig. 8. a. Similarities between transition models and time-slice models in transitions from stage 1 to 2; b. similarities between transition models and time-slice models in transitions from stage 2 to 3 (solid line: angle-based transition modeling; dash line: STMPCA)

Because of the limitation on paper length, only the transition identification result in stage 2, the modeling results (weights in (43)) of the first two ranges of transitions which are the transitions from stage 1 to stage 2 and from stage 2 to stage 3, and the comparison results in these two transition ranges are shown in figures as illustrations. Fig. 6 shows the transition identification result in stage 2 from 24s to 54s. The outliers at the beginning and end of the stage are transition points. The modeling results are plotted in Fig. 7. From Fig. 8, we can see the proposed transition modeling method always leads to more similar transition models to time-slice models than STMPCA, which means the proposed transition modeling method gets more accurate models than STMPCA.

6. CONCLUSION

In this paper, stage-orientated multivariate statistical batch process modeling, monitoring, quality prediction and quality control methods were reviewed. By taking the characteristics of multistage/multiphase batch process into consideration, the stage-orientated methods reflect the variable correlation changing from stage to stage clearly and correctly. Thus, they perform better than traditional multivariate statistical methods in this research area. The issues of transition modeling were then discussed. A new method based on angles between different PCA models were proposed for stage division, transition identification and modeling. It gives a clearer guideline for stage division and identifies transition ranges more objectively. Application result shows the transition models built with this method have high accuracy.

REFERENCES

Breiman, L. (1996). Stacked regression. *Machine Learning,* **24,** 49-64.

Camacho, J. and J. Picó (2006a). Multi-phase principal component analysis for batch processes modelling. *Chemometrics and Intelligent Laboratory Systems,* **81,** 127-136.

Camacho, J. and J. Picó (2006b). Online monitoring of batch processes using multi-phase principal component

analysis. *Journal of Process Control,* **16,** 1021-1035.

Chen, J. and K. C. Liu (2002). On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chemical Engineering Science,* **13,** 63-75.

Conlin, A. K., E. B. Martin and A. J. Morris (2000). Confidence limits for contribution plots. *Journal of Chemometrics,* **14,** 725-736.

Daszykowski, M., K. Kaczmarek, Y. Vander Heyden and B. Walczak (2007). Robust statistics in data analysis – a review basic concepts. *Chemometrics and Intelligent Laboratory Systems,* **85,** 203-219.

Dayal, B. S. and J. F. MacGregor (1997). Improved PLS algorithms. *Journal of Chemometrics,* **11,** 73-85.

Facco, P., M. Olivi, C. Rebuscini, F. Bezzo and M. Barolo (2007). Multivariate statistical estimation of product quality in the industrial batch production of a resin. *Dycops 2007, Cancun,* **2,** 93-98.

Geladi, P. and B. R. Kowalski (1986). Partial least-squares regression: A tutorial. *Analytica Chimica Acta,* **185,** 1-17.

Golberg, M. A. and H. Cho (2004). *Introduction to regression analysis,* WIT Press, Boston.

Hoskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics,* **2,** 211-228.

Jackson, J. E. (1991). *A user's guide to principal components,* John Wiley & Sons, Inc, New York.

Jackson, J. E. and G. S. Mudholkar (1979). Control procedures for residuals associated with principal component analysis. *Technometrics,* **21,** 341-349.

Jain, A. K., M. N. Murty and P. J. Flynn (1999). Data clustering: a review. *ACM Computing Surveys,* **31,** 264-323.

Johnson, R. and D. Wichern (2002). *Applied multivariate statistical analysis,* Prentice Hall, New York.

Jolliffe, I. T. (2002). *Principal component analysis (second edition),* Springer-Verlag, New York.

Kourti, T., P. Nomikos and J. F. MacGregor (1995). Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *Journal of Process Control,* **5,** 277-284.

Krzanowski, W. J. (1979). Between-groups comparison of principal components. *Journal of the American Statistical Association,* **74,** 703-707.

Lu, N., F. Gao and F. Wang (2004a). Sub-PCA modeling and on-line monitoring strategy for batch processes. *AIChE Journal,* **50,** 255-259.

Lu, N., Y. Yang, F. Gao and F. Wang (2004b). Stage-based multivariate statistical analysis for injection molding. *ADCHEM 2004, Hong Kong,* 471-476.

Lu, N., F. Gao, Y. Yang and F. Wang (2004c). PCA-based modeling and on-line monitoring strategy for uneven-length batch processes. *Industrial and Engineering Chemistry Research,* **43,** 3343-3352.

Lu, N., Y. Yang, F. Wang and F. Gao (2004d). A stage-based monitoring method for batch processes with limited reference data. *Dycops 2004, Boston*.

Lu, N. and F. Gao (2005). Stage-Based Process Analysis and Quality Prediction for Batch Processes. *Industrial and Engineering Chemistry Research,* **44,** 3547-3555.

Lu, N. and F. Gao (2006). Stage-based online quality control for batch processes. *Industrial and Engineering Chemistry Research,* **45,** 2272-2280.

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

Nomikos, P. and J. F. MacCregor (1994). Monitoring batch processes using multiway principal component analysis. *AIChE Journal,* **40,** 1361-1375.

Nomikos, P. and J. F. MacCregor (1995a). Multivariate SPC charts for monitoring batch processes. *Technometrics,* **37,** 41-59.

Nomikos, P. and J. F. MacCregor (1995b). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems,* **30,** 97-108.

Rannar, S., J. F. MacGregor and S. Wold (1998). Adaptive batch monitoring using hierarchical PCA. *Chemometrics and Intelligent Laboratory Systems,* **41,** 73-81.

Singhal, A. and D. E. Seborg (2002). Pattern matching in historical batch data using PCA. *IEEE control systems magazine,* **22,** 53-63.

Undey, C. and A. Cinar (2002). Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Systems Magazine,* **22,** 40-52.

Wang, Y., Y. Yang, F. Gao and D. Zhou (2007). Control of multi-phase batch processes: formulation and challenge. *Dycops 2007, Cancun,* **2,** 339-344.

Westerhuis, J. A., T. Kourti and J. F. MacGregor (1998). Analysis of multiblock and hierarchical PCA and PLS models. *Journal of Chemometrics,* **12,** 301-321.

Wold, S. (1978). Cross validatory estimation of the number of components in factor and principal component analysis. *Technometrics,* **20,** 397-405.

Wold, S., A. Rube, H. Wold and W. J. Dunn (1984). The collinearity problem in linear regression, the partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing,* **15,** 735-743.

Wold, s., N. Kettanhe, H. Friden and A. Holmberg (1988). Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemomtrics Intelligence Laboratory,* **44,** 331-340.

Zhao, C., F. Wang, N. Lu and M. Jia (2007). Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes. *Submitted to AIChE Journal.*