A survey on multistage/multiphase statistical modeling methods for batch processes

Yuan Yao, Furong Gao *

Department of Chemical and Biomolecular Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, People’s Republic of China

ABSTRACT

In industrial manufacturing, most batch processes are inherently multistage/multiphase in nature. To ensure both quality consistency of the manufactured products and safe operation of this kind of batch process, different multivariate statistical process control (MSPC) methods have been proposed in recent years. This paper gives an overview of multistage/multiphase statistical process control methods used for process analysis, monitoring, quality prediction and online quality improvement. Different types of phase divisions and modeling strategies are introduced and the methods properties are discussed. For comparisons, a selection guide to these methods for different application purposes is provided. Finally, some promising research directions are suggested based on existing works.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Due to the high dimensionality and complexity of batch processes and the quick product-to-market time required in contemporary industrial settings, it is difficult to create batch process models based on first-principles. The applications of multivariate statistical process control (MSPC) methods, which require only process history data, have attracted research attention. Among them, multiway principal component analysis (MPCA) and multiway partial least squares (MPLS), which extend the applications of principal component analysis (PCA) and partial least squares (PLS) techniques from continuous processes to batch processes, are most widely used (Nomikos & MacCregor, 1994; Nomikos & MacCregor, 1995a; Nomikos & MacCregor, 1995b). These methods allow process variable trajectory information to be projected into low-dimensional latent variable spaces. Therefore, batch process performance can be easily analyzed and monitored in the reduced space. Quality predictions can also be made.

However, in many batch processes, there is an important multistage/multiphase characteristic that is not revealed by MPCA/MPLS. The definitions of multistage/multiphase are as follows. A batch process with a single processing unit but multiple operational regimes is called a multiphase batch process, while a batch process with multiple processing units is a multistage batch process (Undey & Cinar, 2002; Wang, Zhou, & Gao, 2008a).

The nature of the process, including variable correlations, could differ between phases (stages). Even within a single operating phase (stage), the process correlation structure might change due to process dynamics and/or time-varying factors. In conventional MPCA/MPLS methods, which take the data from the entire batch as a single object, such process features are not considered. This lack of consideration of changing process features leads to difficulties in understanding the processes and it affects the monitoring efficiency and quality prediction ability of the multivariate statistical process control (MSPC) strategy. To approximate multistage/multiphase process correlation structures, a batch process can be divided into several blocks or segments according to either changes in the operating stage/phase or changes in the process characteristics. In this paper, a block/segment is called a “modeling phase”, representing process characteristics over a certain process duration, which may be different from the physical “operating stage/phase”. Unless otherwise noted, the word “phase” refers to the “modeling phase” in this paper.

In recent years, different phase-division methods have been proposed and different modeling methods have been developed that take the phase effects into consideration. This survey paper gives an overview of the multistage/multiphase statistical batch process modeling. A brief introduction to PCA/PLS methods is first presented. Then, pre-treatment methods used on batch process data are described. Section 4 reviews different types of phase-division methods. Section 5 introduces and discusses the multistage/multiphase batch process modeling methods for process monitoring, quality prediction and online control. Comparisons of the methods are presented in Section 6, including the advantages...
as well as the disadvantages of each method. Appropriate selection criteria to solve various problems are suggested. In the last section, this paper is concluded with a summary and some thoughts on future directions in research in this area.

2. Fundamentals of PCA and PLS

2.1. Principal component analysis (PCA)

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

$$X = \sum_{j=1}^{m} t_i^p p_j^T + E = \hat{X} + E,$$

where $t_i(n \times 1)$ is the latent vector that is also called the principal component (PC) vector or the score vector, $p_j(m \times 1)$ is the loading vector, which projects data into the score space and contains variable correlation information. The first several PCs contain most of the variance information about $X$, and the last several PCs may contain only noise. Thus, by retaining the first several PCs, most important variance information is extracted and the dimension of the data is largely reduced. The original data space is divided into the score space, $\hat{X} = TP^T$, and the residual space, $E$, where $\hat{X}$ is the PCA model prediction, $T$ is the matrix of retained PCs, $P$ contains the corresponding loadings and $E$ is the residual matrix mainly consisting of noise. Several algorithms have been developed for PCA model calculation (Jackson, 1991; Jolliffe, 2002). To determine the most appropriate retained number of PCs, cross-validation and many other procedures have been introduced by researchers (Jackson, 1991; Jolliffe, 2002; Wold, 1978).

In process monitoring, Hotelling’s $T^2$ and $SPE$ statistics are calculated after building a PCA model (Jackson, 1991). The corresponding control limits can be calculated based on certain statistical assumptions (Jackson & Mudholkar, 1979; Jackson, 1991; Nomikos & MacGregor, 1995a) and then utilized in fault detection. After a fault is detected by the $T^2$ or $SPE$ control chart, the cause of the fault must be determined. Contribution plots (Miller, Swanson, & Heckler, 1998) are the most widely applied tools in PCA-based fault detection.

2.2. Partial least squares (PLS)

Different from PCA, 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 one is a product quality data matrix $Y(n \times m_y)$, where $n$ is the number of samples, $m_x$ is the number of process variables, and $m_y$ is the number of quality variables. PLS not only extracts the variation of $X$, but also predicts $Y$ as far as possible. The equations are as follows:

$$X = TP^T + E = \sum_{i=1}^{A} t_i^p p_i^T + E,$$

$$Y = UQ^T + F = \sum_{i=1}^{A} u_i q_i^T + F,$$

$$\hat{u}_i = b_i^T t_i,$$

where $b_i = t_i^T u_i / (T^T t_i t_i^T)$ is the regression coefficient between a pair of latent variables, $t_i$ and $u_i$. A PLS model can also be written in a compact way as

$$Y = \Theta F + F^*,$$

where $\Theta$ is a regression parameter matrix and $F^*$ is the residual matrix. More details about PLS can be found in the literature (Geladi & Kowalski, 1986; Hoskuldsson, 1988; Wold, Rube, Wold, & Dunn, 1984).

PLS models are often used in quality prediction. Besides this, process monitoring and fault diagnosis based on PLS models are similar to those based on PCA models (Kourtis, 2005).

2.3. Normalization

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

$$\hat{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \quad (i = 1, \ldots, n; \ j = 1, \ldots, m),$$

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

3. Data pre-treatment

3.1. Unfolding and splitting methods

Historical process data collected from a batch process are usually represented by a three-dimensional data matrix, $X(l \times J_x \times K)$, where $l$ is the number of total batches, $J_x$ is the number of process variables, and $K$ is the number of sampling time intervals in a batch. As mentioned earlier, PCA and PLS can only deal with two-dimensional data matrices. Therefore, $X$ should be unfolded to a two-dimensional data matrix or split into several two-dimensional data matrices before modeling.

Batch-wise unfolding is the most widely applied method to unfold a three-dimensional data matrix. It keeps the dimension in the batch direction and merges the variable and time dimensions. Each row of the unfolded matrix, $X(l \times J_x \times K)$, contains all data within a batch. After unfolding, normalization can be performed on the two-way matrix to highlight the variations between batches. Such normalization can be called batch-wise normalization. MPCA/MPLS was developed based on this kind of normalization (Nomikos & MacGregor, 1994; Nomikos & MacGregor, 1995a; Nomikos & MacGregor, 1995b). Since all the data in the batch are regarded as one single object, future measurement estimation is needed for online applications based on MPCA/MPLS. Another disadvantage is that the different characteristics between operating stages/phases cannot be captured.

Different from batch-wise unfolding, variable-wise unfolding maintains the dimension of the variables and merges the other two dimensions (Wold, Kettaneh, Friden, & Holmberg, 1998). Each sampling point is considered as an object. After normalization, the variable trajectory information is left in the data matrix. An advantage of variable-wise unfolding is that the uneven batch durations problem in process modeling is spontaneously solved without trajectory synchronization. In addition, future measurement estimation is unnecessary in online applications. However, since the systematic variations around normal trajectories are not emphasized, the monitoring efficiency based on this method is limited.

Besides these methods, batch dynamic unfolding is an alternative option. It can be regarded as variable-wise unfolding with lagged measurements included as additional variables. Therefore, the dynamics within each batch can be captured by PCA based on such unfolding. This type of modeling method is called batch dynamic PCA/PLS (BDPCA/BDPLS) (Chen & Liu, 2002).
which extends the applications of dynamic PCA (Ku, Storer, & Georgakis, 1995) from continuous processes to batch processes. With this method, no future measurement estimation is needed for online monitoring. However, an assumption of BDPCA/BDPLS is that the dynamic structures remain the same throughout the whole batch. This method is thus unsuitable for multiphase/multistage batch process modeling.

Splitting can also be used to transform three-dimensional data arrays into a series of two-dimensional data matrices (Ramaker, van Sprang, Westerhuis, & Smilde, 2005). Each two-dimensional matrix represents the data in a time slice of all batches. Then, time-slice local PCA models can be built based on the time-slice matrices and used to describe the local variable correlations along the operating time in the batch. However, time-slice PCA models lead to rather heavy computation and large storage. Recently, these different unfolding and splitting methods were analyzed and compared by Camacho, Picó, and Ferrer (2008b).

3.2. Batch trajectory synchronization

Most multivariate statistical modeling methods for batch process monitoring and quality prediction are based on the assumption that the batch durations are the same. However, in real industrial settings, the batch lengths of many processes are not fixed because of disturbances and changes in operating conditions. In such situations, trajectory alignment, which is also called synchronization, is required. The simplest synchronization methods are cutting the batches to the minimum length (Rothwell, Martin, & Morris, 1998) or treating the absent parts of shorter trajectories as missing data (Kourtis, 2003a). A more reasonable solution is to find a proper indicator variable that can be used to replace the time dimension. Then, process modeling, analysis and online monitoring can be performed based on the progress of the indicator variable (Nomikos & MacGregor, 1995a). Prior process knowledge is needed to select such an indicator variable. Dynamic time warping (DTW) has also been utilized to solve the uneven-length problem (Kassidas, MacGregor, & Taylor, 1998). However, DTW may distort the fault patterns and reduce the fault detection ability as well as the fault diagnosis accuracy. Undey, Ertunc, and Cinar (2003) developed an MPLS model between the variable-wise unfolded process data matrix and the local batch time, and used the model prediction to estimate the progress of an evolving cycle. They found that doing a reasonable phase division before synchronization is helpful to achieve better data alignment results. Besides the above, curve registration identifies landmarks within a trajectory (or set of trajectories) and then warps the test trajectories to the reference trajectory containing reference landmarks (Ramsay & Silverman, 1997; Undey & Cinar, 2002). Cho and Kim (2003) compared the variable trajectories in the current evolving batch with the trajectories in the reference batches and selected the most similar batch to supply the unknown future data. Similar to DTW, the last two methods may also distort the fault patterns.

4. Phase-division methods

Another important feature of batch process data is that many batch processes have multiple operating stages/phases. As mentioned in Section 1, each operating stage/phase may have its unique characteristics. This is the reason why multistage/multiphase batch processes have attracted so many research efforts. Phase division is an important step before multistage/multiphase batch process modeling. The effectiveness of a multiphase model is doubtful without a proper phase division. There are three major ways to divide batch processes into phases (Camacho, Picó, and Ferrer, 2008a).

The first way is based on expert knowledge. A process can be divided into segments according to different processing units and distinguishable operation phases inside each unit (Dong & McAvoy, 1996a; Kosanovich, Piovoso, & Dahl, 1994). Reinkainen and Hősukulsson (2007) suggested that the batch process data could be naturally divided into groups before modeling and analysis. It is another expression of phase division based on expert knowledge. Recently, Liu and Wong (2008) studied the inter-stage relation of a two-stage batch process based on a given division of the stages. The multiblock modeling technique (Smilde, Westerhuis, & de Jong, 2003; Westerhuis, Kourtis, & MacGregor, 1998) also assumes that the phase division points are known beforehand. This kind of phase division reflects the process operation status well, but its shortcoming is that the known prior knowledge may not be sufficient to divide processes into phases reasonably.

The second way to divide batch processes is through process analysis. Gollmer and Posten (1996) used the DTW technique to compare cycles with a prototype cycle whose phase division points are known. The comparison results give suggestions on the phase division of the new batches.

Muthuswamy and Srinivasan (2003) identified the division points using the process variable features described in the form of multivariate rules. Undey and Cinar (2002) used an indicator variable that contains significant landmarks to detect the completion of each phase. Facco, Olivi, Rebuscini, Bezzo, and Barolo (2007) also divided phases based on some easily detectable landmark events. Doan and Srinivasan (2008) did phase division based on singular points in some known key variables, which are also a kind of landmark. These methods work well when certain required process features are known.

Besides the above methods, Kosanovich, Dahl, and Piovoso (1996) pointed out that the changes in the process variance information explained by the principal components (PCs) can indicate the process stage division points. However, in their work, they did not give an algorithm to identify phases precisely.

To overcome the shortcomings of above methods, in recent years, data-based automatic phase division methods have been developed based on process variable correlation changes along the operating time in each batch. This idea was first proposed by Lu, Gao, and Wang (2004) as follows. In multistage/multiphase batch processes, different operating stages/phases may have different variable correlation structures. In other words, the changes in variable correlations reflect the changes in the process nature and indicate phase changes. Thus, the basic procedure is designed as follows: first, the local variable correlation information is extracted by the loading matrices or regression coefficient matrices of the time-slice PCA or PLS models; then, a clustering algorithm is performed on these matrices to detect the segment points between phases. The detailed steps will be introduced later. An alternative method was proposed by Camacho and Picó (2006a) and called the multiphase (MP) algorithm, which detects the phase division points based on the prediction abilities of the PCA model. This algorithm was further revised in their following works (Camacho & Picó, 2006b; Camacho et al., 2008a). Zhao, Wang, Mao, Lu, and Jia (2008c) recently adopted the basic idea of the MP algorithm for phase division in their adaptive monitoring method called multiphase independent component analysis (AMPICA).

The comparisons of different types of phase-division methods are summarized in Table 1. The data-based automatic phase-division methods have advantages because they do not rely on prior process knowledge and can be performed on different batch processes. In addition, a significant difference between the automatic phase-division methods and the other methods is that the phases identified with the other methods usually correspond to the operating stages/phases, while the automatic methods divide phases based on the process correlation information and may lead to divisions not exactly the same as operating phases/stages. To
analyze the process nature in each operating stage/phase, the division based on operating stage/phase information is more intuitive. On the other hand, for the purpose of online monitoring, the division of phases based on process correlation structures can provide more accurate process PCA/PLS models, since the process is nearly time invariant in each phase identified in this way.

5. Modeling methods

As mentioned earlier, the conventional MPCA/MPLS model does not consider the unique characteristics of each phase in multistage/multiphase batch processes. To model these processes more reasonably, two different groups of techniques have been developed. The first group of techniques includes various multiblock PCA/PLS modeling methods (Smilde et al., 2003; Westerhuis et al., 1998) that separate process variables in different operating stages/phases into different blocks and model the variable correlations within each block together with the correlations among blocks. Another group of techniques models each phase separately. Undey and Cinar (2002) stated that “local models were proven to be advantageous when different phases exist in process stages and when precise phase separation is crucial”. These modeling methods have been applied to several process analysis and online monitoring, quality-related analysis and online quality prediction, and online quality control.

5.1. Modeling methods for process analysis and online monitoring

5.1.1. Multiblock techniques

5.1.1.1. Multiblock PCA/PLS. Different versions of multiblock PCA and PLS models have been developed to group process variables into meaningful blocks and to consider both the intra-relationships within each block and the inter-relationships among blocks (Berglund & Wold, 1999; Casin, 2001; Slama, 1991; Wangen & Kowalski, 1988; Wold, Hellberg, Lundstedt, Sjostrom, & Wold, 1987; Wold, Kettaneh-Wold, & Tjøsem, 1996). Some of the multiblock techniques have been analyzed and compared by researchers (Smilde et al., 2003; Westerhuis et al., 1998).

These multiblock modeling methods can be utilized in multistage/multiphase batch process monitoring and operating. In applications, the process variables in each operating stage/phase are blocked, and multiblock models are then developed for process monitoring (Kourti, Nomikos, & MacGregor, 1995; Lee & Vanrolleghem, 2003; MacGregor, Jæckle, Kiparissides, & Koutoudi, 1994; Westerhuis & Coenegracht, 1997). To achieve better process monitoring and fault diagnosis, several kinds of statistics, including block and total $T^2$ and SPE together with the corresponding contribution plots, were proposed by Qin, Valle, and Piovoso (2001) and Choi and Lee (2005).

5.1.1.2. Adaptive hierarchical PCA (AHPCA). To avoid future measurement estimation, a special multiblock modeling method called adaptive hierarchical PCA (AHPCA) was proposed by Ränner, MacGregor, and Wold (1998). In AHPCA, the blocks are the time-slice data matrices. Different from conventional multiblock PCA, AHPCA looks at one time slice at a time instead of all blocks at once and builds a model for each time interval. A weighting factor is used to make the models adaptive. For multiphase batch processes, the weighting factor should be assigned a large value to make the model adapt quickly from one phase to another.

The AHPCA method with a proper selected weighting factor performs better than MPCA in online monitoring of multiphase batch processes. But there are still drawbacks. One problem is that too many models are needed since AHPCA models each time slice separately. This increases the computation and storage burdens. In addition, in some situations, a proper weight may not be found for the entire batch run. The inter-stage discontinuity further limits the application of AHPCA to multitastage batch processes.

5.1.2. Phase-separated modeling techniques

5.1.2.1. Phase MPCA and phase AHPCA. A direct method for phase-based modeling is to model each phase with a separate MPCA or AHPCA model. The benefits of building multiple phase models instead of a single batch model are shown by many examples (Dong & McAvoy, 1996a; Kosanovich et al., 1994; Undey & Cinar, 2002). In these studies, phase MPCA models lead to better fault detection results than MPCA. In phase AHPCA, different AHPCA models are developed for each stage to avoid the problem caused by inter-stage discontinuity, and different weighting factors can be assigned to different stages. However, the phase MPCA and the phase AHPCA methods share the shortcomings of the conventional MPCA and AHPCA, such as future data estimation or heavy computation burden.

5.1.2.2. Phase-based sub-PCA. In recent years, a series of phase-oriented statistical modeling strategies have been proposed. The first work in this series was proposed to introduce a phase-based
There are two levels to sub-PCA model building. Firstly, a batch process may be divided into several modeling phases corresponding to the process variable correlation changes. Secondly, within each modeling phase, the process correlation nature is similar although the process may be time varying. Therefore, a series of sub-PCA models can be built for each modeling phase.

There are several major steps in sub-PCA modeling procedures as shown in Fig. 1, namely batch process data matrix splitting, phase division, and sub-PCA model building. In a batch process data matrix, \( X(l \times j_s \times K) \), each vertical slice, \( X^k(j_s \times K) \), is a time-slice data matrix. After using the PCA algorithm on these split time-slice matrices, the variable correlation information on each time interval is contained in \( K \) number of time-slice loading matrices, \( \hat{P}^k \). In the same phase, the time-slice loading matrices are similar, while different phases have different loadings, reflecting changes in variable correlation structures and underlying process behaviors. Since each column of a PCA loading matrix contains different process variance information, the time-slice loading matrices, \( \hat{P}^k \), are transformed into a weighted form, \( \hat{P} \), with the importance of each column taken into consideration.

\[
P^k = [\hat{p}^k_1, \hat{p}^k_2, \ldots, \hat{p}^k_{i_k}],
\]

where \( \hat{p}^k_j \) is the \( j \)th column of \( \hat{P}^k \), \( g^k_j = \lambda^k_j / \sum_{i=1}^{i_k} \lambda^k_i \), and \( \lambda^k_j \) is the eigenvalue of the covariance matrix, \( \left(X^k \right)^T X^k \).

The phase division can be accomplished by partitioning the time-slice weighted loading matrices, \( \hat{P} \), into clusters with a variant \( k \)-means clustering algorithm (Jain, Murty, & Flynn, 1999). The Euclidean distance is used to assess the dissimilarity between matrices. A parameter should be specified for the clustering, which is the threshold of the minimal distance between two cluster centers. The final phase division results are achieved based on the clustering result associated with the operating time information.

After all phases are identified, sub-PCA models for each phase are then built by taking the average of the time-slice PCA models in the corresponding phase. The calculation of the phase sub-PCA loading matrix is as below:

\[
P^c_k = \frac{1}{n_c} \sum_{k=1}^{n_c} \hat{P}^c_k,
\]

where \( n_c \) is the number of time slices in phase \( c \), \( \hat{P}^c_k \) is the representative sub-PCA loading matrix for phase \( c \) and \( \hat{P}^c_k \) is the \( k \)th time-slice loading matrix in phase \( c \).

In online monitoring, before calling a phase model, the current phase is determined by checking the current time interval. Then, the sub-PCA model of the corresponding phase 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 for the fault.

The sub-PCA method has been verified in injection molding process applications, which are typical multiphase batch processes (Lu, Yang, Gao, & Wang, 2004). In such applications, the different phase characteristics that cannot be revealed by the MPCA model are significant. In these methods do not consider the multiple phase features. Therefore, a method to model and monitor multistage/multiphase batch processes without trajectory synchronization is desired. The phase-based sub-PCA method was extended to uneven-length batch processes to solve the problem (Lu, Gao, & Wang, 2004). In this work, two models are built for each phase, one for the phase division and the other for process monitoring.

5.1.2.4. Sub-PCA for uneven-length batch process monitoring. As shown in Fig. 2, in multistage/multiphase processes, both the whole batch duration and each phase duration can be different from run to run, due to disturbances in operating conditions or different process settings. Although many different batch trajectory synchronization methods have been proposed, shortcomings still exist and these methods do not consider the multiple phase features. Therefore, a method to model and monitor multistage/multiphase batch processes without trajectory synchronization is desired. The phase-based sub-PCA method was extended to uneven-length batch processes to solve the problem (Lu, Yang, Gao, & Wang, 2004). In this work, two models are built for each phase, one for the phase division and the other for process monitoring.

On account of the variations in phase durations, batch-wise normalization cannot be performed properly before the phase division. Instead, variable-wise unfolding and normalization can be applied directly to uneven-length data. However, this kind of unfolding will affect the monitoring efficiency. The variable-wise unfolding and normalization is utilized for phase-division purposes. After the phase division, the batch-wise normalization is performed for phase model building.

The shortest batch length, \( K_s \), in all reference batches is easy to determine. After variable-wise unfolding and normalization, \( K_i \) time-slice PCA models are built based on the time-slice data matrices. Then, the \( k \)-means clustering algorithm is used on the weighted loading matrices of these PCA models. Since the stable clusters are only found for the common part of each phase, such as
spans A and C in Fig. 2, the first stable cluster indicates the common part of the first phase. Therefore, a phase-division PCA model can be built for phase I based on this cluster. After selecting the number of the retained principal components, the SPE values can be computed. Since the process correlation structure should not change significantly until the process enters the next phase, all data in phase I should be explained well by this phase-division PCA model with relatively small SPE values. In contrast, the model will result in large SPE values for data not belonging to phase I. Thus, the phase-division PCA model can be used to find the actual durations of this phase for all reference batches by checking the SPE values and comparing them with a threshold. After the lengths of phase I are identified, the first phase data are removed from the reference batch data, and a new reference data set is formed with the remaining data. The same procedure is repeated to determine the durations of the following phases. After the above steps, the data in each phase are re-normalized by batches to achieve better monitoring efficiency. The phase sub-PCA models for online monitoring are then built.

A question to be answered in online monitoring is how to differentiate process abnormalities from phase changes. Suppose that the lengths of phase c vary from \( t_{\text{min}} \) to \( t_{\text{max}} \). If the data belong to \( [t_{\text{min}}, t_{\text{max}}] \), there are two possibilities if the SPE or \( T^2 \) value goes beyond the control limits, a fault has occurred or the process is entering the next phase. When such situation happens, the data are re-normalized and monitored with the modeling phase of \( c + 1 \). If the statistics are outside of the control limits again, a fault is detected. Otherwise, it indicates the beginning of a new phase.

5.1.2.4. Sub-PCA with limited reference data. Most multivariate statistical process modeling methods utilize normal historical data that cover the entire normal operating region. To collect enough normal operation reference data could be very time consuming for slow batch processes such as some bio-processes. If there were a method to model such processes with limited or even minimal batch cycles, this problem could be solved. With such a motivation, the phase-based batch process monitoring method with limited reference data was proposed by Lu, Yang, Wang, and Gao (2004) and revised by Zhao, Wang, Lu, and Jia (2007).

The major difference between this method and the previous sub-PCA method is that a moving window is used to extract the local process characteristics instead of the time-slice data matrix. In each window, data are arranged as a two-way matrix, whose rows contain values of the process variables measured at different sampling intervals along the time direction in the window.

In Lu, Yang, Wang, et al.'s work (2004), the phase division and modeling start with the data from an arbitrary normal operating batch run. The data of this batch are stored as \( X(K \times J_k) \). The moving window strategy is utilized along the time direction in this cycle. Therefore, the data in each window form a two-way matrix, \( X^d(d \times J_d) \), where \( k \) is the index of the window, and \( d \) is the length of the window. The data in each window are normalized. PCA models are then built for the data in each window and used to extract local correlation information. Then, the phase division can be conducted by clustering these local PCA loading matrices, similar to the procedure in the conventional sub-PCA method. After phase division, sub-PCA models together with SPE and \( T^2 \) control limits can be calculated to monitor each phase. The control limits can be updated to focus more on batch-to-batch variations with additional normal operation data available.

This method was revised recently (Zhao, Wang, et al., 2007). In the revised method, several reference batches are used instead of one batch in the initial batch process model to cover more process variance information in both time and batch directions. Both the phase models and the control limits are updated with new batch data. Data renormalization may be also necessary before control limit updating when the data mean and standard deviation drift with the operation evolution.

5.1.2.5. Sub-PCA based transition modeling and online monitoring. In many multistage/multiphase batch processes, there exist gradual transitions between neighboring steady phases. In transitions, the process dynamic behaviors are more complex than in steady phases, so abnormal behaviors are very likely to happen in transitions, which affect operation safety and product quality. Therefore, it is important to model and monitor the transitions properly.

The conventional phase-based sub-PCA method divides batch processes into phases with the \( k \)-means clustering algorithm, which is a kind of hard partition. This kind of partition ignores the transition characteristics from phase to phase. “Misclassification may occur at the beginning and end of each stage” and “it may lead to false alarm and missing alarm” (Lu, Gao, et al., 2004). Therefore, the sub-PCA method was revised and extended, and two kinds of transition modeling and online monitoring methods were developed to solve the problem (Yao & Gao, 2009; Zhao, Wang, Lu, & Jia, 2007). Both methods identify the gradual transitions by analyzing the changes in time-slice loading matrices, which reflect the changes in process behaviors. Based on fuzzy membership functions, each sampling interval in a transition is modelled as a weighted sum of two neighboring phase models.

In 2007, Zhao, Wang, Lu, et al. (2007) proposed the soft-transition multiple PCA (STMPCA) method. The basic idea is as follows. Each phase identified by the \( k \)-means clustering algorithm can be further divided into the steady phase and the transition range. In the steady phase, each time-slice PCA loading matrix is similar to the cluster center, while the dissimilarities between the time-slice loadings and the cluster center become larger in the transition range. Therefore, the ranges of transitions are determined based on Euclidean distances between the weighted time-slice PCA loading matrices and the cluster center of each phase, which is described as an average value of all weighted time-slice PCA loading matrices in that cluster. A threshold can be computed with some user-specified parameters. If the distance between a sample and the cluster center is larger than the threshold, the sample is identified as a transition point. Then, STMPCA calculates the PCA model of each sampling interval in transitions as a weighted sum of two weighted phase center loading matrices. The values of the weights are also calculated based on Euclidean distances. After transition modeling, the control limits of \( T^2 \) and SPE statistics are calculated for each sampling interval for online monitoring.

Recently, Yao and Gao (2009) developed another method to identify the transitions automatically and statistically without user-specified parameters, together with a transition modeling method that leads to reasonable transition models. In this research, a factor that describes the PCA similarity by checking the angles between score spaces of different PCA models was proposed. Instead of the Euclidean distance, this new PCA similarity factor was utilized for phase division based on the \( k \)-means clustering algorithm. Then, the dissimilarities between the time-slice PCA loading matrices and the cluster-center PCA loading matrices were calculated and plotted on univariate control plots. The successive outliers detected at the beginning and end of each cluster were identified as the points in transitions and removed from each cluster. The remaining part of each cluster is the range of the steady phase. The steady phase models can then be calculated based on the data of each steady phase, while the transition models are described as membership functions of two neighboring steady-phase models. The parameters in transition models are calculated by solving an optimization problem to maximize the similarities between the transition models and the corresponding time-slice
PCA models, which describe the local correlation structures most accurately. Then after the determination of the PC numbers, the control limits of $T^2$ and $SPE$ statistics for each model can be calculated accordingly for online monitoring.

5.1.2.6. Multiphase PCA (MPPCA). In 2006, Camacho and Picó (2006a) proposed a multiphase (MP) algorithm that is an alternative method for automatic phase identification. In the MP algorithm, each phase is identified as a segment of the batch that can be approximated by a linear PCA model. Let us take a two-phase batch process as an example and consider two ways to model the process. In the first modeling attempt, the whole batch process is modelled with a single PCA model and the model predictions can be calculated. In the second modeling attempt, each phase is modelled separately and the model predictions are obtained again. If the phase division is reasonable, better PCA model predictions should be got in the second case than in the first one. Therefore, it is easy to imagine that the best phase division point can lead to the best PCA model prediction. This is the basic idea of the MP algorithm. The whole algorithm consists of two major steps: a top-down recursive procedure to find an initial phase division result, and a bottom-up procedure to reduce the number of phases and get the final solution. After phase division, the MPPCA models can be built for each phase.

The initial version of MPPCA models the batch process data based on batch-wise unfolding. To avoid future data estimation, an alternative version of MPPCA was proposed with variable-wise unfolding or batch dynamic unfolding (Camacho & Picó, 2006b). The MP algorithm was redesigned in recursive steps to determine the phase number and the PC number of each PCA model simultaneously. The application results demonstrate the method's benefits. At the same time, the computation burden of this algorithm is heavier than before. In addition, a direct comparison between increasing PC numbers and increasing phase numbers may not be appropriate for some processes. Most recently, Camacho et al. (2008a) modified the MP algorithm again to make it suitable for use with any kind of unfolding method (batch-wise, variable-wise and batch dynamic unfolding) with either PCA or PLS. A merging algorithm was also proposed to reach a compromise between model complexity and modeling performance in terms of prediction ability.

5.2. Modeling methods for quality prediction

5.2.1. Multiblock techniques

5.2.1.1. Multiblock PLS. Multiblock PLS methods (Wangen & Kowalski, 1988; Westerhuis et al., 1998; Wold et al., 1987, 1996) can be used for quality prediction. Compared with the conventional PLS model, multiblock PLS methods can provide better interpretation. However, such methods do not have significant advantages in online quality prediction of batch processes. Future data estimation is needed for online applications. Besides, the phase data, which are not critical to the quality are also built into the prediction model, which may affect the prediction accuracy as described later.

5.2.2. Phase-separated modeling techniques

5.2.2.1. Multiphase PLS (MPPLS). Camacho et al. (2008a) extended the MP algorithm from PCA to PLS. MPPLS models can be built for auto-identified phases. Lagged variables are included into the MPPLS models to take the batch dynamics into consideration. Such MPPLS models are then applied in online quality prediction. In applications, the MPPLS method outperformed other methods studied by Camacho et al. (2008a), but the different phase effects and the phase accumulation effects on the quality were not revealed.

5.2.2.2. Phase-based PLS. In multistage/multiphase batch processes, different phases often have different effects on the final product quality. A particular end-quality may be determined in some particular phases and by some particular process variables. Therefore, in prediction model building, using the information on the phases that are not critical to quality may reduce the prediction accuracy. Based on these findings, a phase-based process analysis and quality prediction method has been developed (Lu & Gao, 2005).

The phase division procedure is similar to the phase division procedure in the sub-PCA method. Time-slice PLS models regress the time-slice data matrices to the quality data matrix. Then, the $k$-means clustering algorithm is adopted for phase division based on the time-slice regression coefficient matrix, $\bar{\Theta}_k$. Thus, the relationships between the process variables and the final product quality are similar and stable at all sampling intervals in each phase divided in this way. The phase PLS models can be calculated as an average of all time-slice PLS models in each phase.

Quality-related process analysis can then be conducted based on the phase PLS model. The multiple coefficient of determination, $R^2$ (Johnson & Wichern, 2002), is used to evaluate the fitness of the phase PLS models and determine the critical-to-prediction phases, while the contribution rate of each process variable to each quality prediction is used to find key process variables in each critical phase.

With the process knowledge extracted with the analysis described above, online quality prediction can be performed. There are two kinds of quality variables. One is determined by only one specific critical-to-prediction phase, and the other kind is determined by two or more phases cumulatively. The first type of quality variable can be predicted online at each sampling interval only in the critical-to-prediction phase. The predicted quality values may vary with time within the same phase, since only time-slice information is used in each prediction. An advantage of doing so is avoiding future measurement prediction. At the end of the critical-to-prediction phase, an end-of-phase prediction can be given as the average value of all predictions in the critical phase. For the second type of quality variable, since there are several phases that affect the quality cumulatively, the prediction can be calculated with the stacked modeling methods (Breiman, 1996).

To pay more attention to the time accumulation effect on the quality in each phase, Zhao, Wang, Mao, Lu, and Jia (2008a) proposed another phase-oriented quality prediction method. This algorithm was further revised in their following work (Zhao, Wang, Mao, Lu, & Jia, 2008b). The phase-division method is slightly different from that in Lu and Gao’s work (2005). After building the time-slice PLS models, the time-slice loading matrices, $\bar{P}^i$, from PLS input spaces are used in the phase clustering rather than $\bar{\Theta}_k$. Then, in each phase, the phase-specific average trajectories, $\bar{X} (I \times f_j)$, are obtained by averaging all normalized time-slice data matrices. The phase PLS models can be built by regressing each $\bar{X}$ on the quality data matrix, which provides a platform for the analysis of the critical-to-prediction phases and for quality prediction. This type of phase PLS model makes use of the values of variable trajectories throughout the whole phase, so that future measurement estimation is needed for online applications. Since only future data in one phase are estimated each time, the estimation accuracy is likely higher than estimating the future data of the whole batch.

5.2.3. Another technique

5.2.3.1. Priority PLS. Recently, the priority PLS method was utilized in multistage batch process quality prediction (Reinikainen & Höskuldsson, 2007). This method analyzes the cumulative
influence of the stage on product qualities. A reliable prediction can be provided at the end of some stage before the entire cycle is over. However, limitations still exist. A phase division algorithm is lacking in this method. The different critical-to-prediction stages for different quality variables are not revealed. Besides, in the application of this method, the process variables seem to have steady settings in each stage instead of dynamic trajectories.

5.3. Modeling methods for online quality control

5.3.1. Phase-based PLS model for online quality control

To achieve acceptable and consistent end-product qualities, an efficient online quality control strategy is desired. As discussed earlier, it is very common that quality can be affected by only some key variables in certain phases. Adjustment of manipulated variables outside these phases does not lead to any quality improvement, but only causes control difficulties or affects process stability. With the development of the phase-based PLS method, accurate online quality predictions of batch process are possible and the effects of the phase on quality variations can be revealed. Therefore, an online batch process quality control strategy based on phase-based PLS models was proposed (Lu & Gao, 2006). To the best of our knowledge, this strategy is the first attempt to take the phase-specific effects of process variables into consideration in batch process online quality control based on multivariate statistical models.

The phase division and phase-based PLS model building are the same as the procedures described by Lu and Gao (2005) with the addition of the critical-to-prediction phase identification. The critical-to-prediction phases with manipulated variables that have impacts on the quality are called critical-to-control phases. Manipulated variables, \(x_{sp}\), are then selected in each critical-to-control phase based on prior knowledge that is easy to determine in industrial settings.

Then, the no-control region of the end quality can be defined based on product specifications or normal history quality data. In online quality control, only when the quality predictions fall into the no-control region will adjustments to manipulated variable set points be performed. Thus, too frequent adjustments that may reduce the batch process operation efficiency are avoided. In the procedures of online quality control as shown in Fig. 3, the quality prediction is performed on each sampling point in the corresponding critical-to-control phases. If the quality prediction is out of control, the set points of the manipulated variables are adjusted in the remaining period of the current critical-to-control phase in order to compensate for the quality loss. The set points can be calculated by solving the following optimization problem:

\[
J = \min \left( \left( Q_{comp} - Q_{loss} \right)^2 + Q_{cost} \right)
\]

\[
\Delta x_{sp} = x_{sp,new} - x_{sp,old}
\]

\[
\Delta x_{sp,min} \leq \Delta x_{sp} \leq \Delta x_{sp,max}
\]

where \(x_{sp,old}\) is the original set point of the manipulated variables, while \(x_{sp,new}\) is the new set point, \(\Delta x_{sp,min}\) and \(\Delta x_{sp,max}\) are the hard constraints of the manipulated variable adjustments, \(Q_{loss}\) is the accumulated quality loss up to the 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 the manipulated variables. The detailed calculations of \(Q_{loss}\), \(Q_{comp}\) and \(Q_{cost}\) can be found in the original paper.

6. Comparison

Table 2 summarizes our comparison between the different multistage/multiphase statistic modeling methods. Among all the methods, the phase-based sub-PCA/PLS methods and the MP algorithms can automatically identify phases based on the reference data, while other methods rely on prior expert knowledge or process analysis. Therefore, to model a multistage/multiphase process with unknown phase division points, the phase-based sub-PCA/PLS methods and the MP algorithms are better choices.

Besides the issues of multiple stages/phases, other batch process characteristics are also considered by some methods. The phase-based sub-PCA method has been extended to uneven-length batch processes. It handles uneven-length problems without variable trajectory synchronization. The statistical modeling and monitoring issue of the transitions is studied by the extensions of sub-PCA, and not considered by other methods. Sub-PCA also can model and monitor batch processes with limited reference data. As shown in Table 2, the sub-PCA method and its extensions have can solve a series of problems in batch process modeling with beneficial outcomes.

However, sub-PCA does not build batch dynamics into the model. It only models variable cross-correlations. Multiblock techniques build batch process models with the entire batch data set. Therefore, the autocorrelations are also modelled. Similarly, the phase MPCA takes the entire phase data set as an object and models both variable cross-correlations and autocorrelations. However, future unavailable measurements should be estimated when these methods are applied online. MP algorithms make use of lagged variables and, thus, include the dynamic information in the model. At the same time, future data estimation is avoided. MP algorithms may still be improved, since the length of the lagged steps is chosen to be same for all phases in current algorithms, which they may not be the best choice since each phase has different dynamic features. Also, it may be possible to extend sub-PCA to dynamic batch process modeling by including the lagged variables in the phase sub-PCA models.

Many methods listed in Table 2 were developed for the purpose of process analysis and online monitoring. Multiblock PCA/PLS reveals the relationship between phases in offline analysis,
<table>
<thead>
<tr>
<th>Methods</th>
<th>Automatic phase division</th>
<th>Dealing with uneven-length problem</th>
<th>Modeling batch dynamics</th>
<th>Future measurement estimation</th>
<th>Application areas</th>
<th>Other comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiblock PCA/PLS</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Need</td>
<td>Process analysis, monitoring and quality prediction</td>
<td>Revealing the relationship between phases in offline process analysis</td>
</tr>
<tr>
<td>AHPCA</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Process monitoring</td>
<td>Heavy computation and storage burden</td>
</tr>
<tr>
<td>Phase MPCA</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Need</td>
<td>Process analysis and monitoring</td>
<td>Inheriting some features from MPCA</td>
</tr>
<tr>
<td>Phase AHPCA</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Process monitoring</td>
<td>Inheriting some features from AHPCA; different weight parameters in different phases</td>
</tr>
<tr>
<td>Priority PLS</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Quality-related analysis and quality prediction</td>
<td>Identification of the last critical-to-quality stage; not revealing the different critical stages for different quality variables</td>
</tr>
<tr>
<td>MP algorithm</td>
<td>MPPCA</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Process analysis and monitoring</td>
<td>Quality prediction</td>
</tr>
<tr>
<td>MP algorithm</td>
<td>MPPLS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Quality prediction</td>
<td>More complicated algorithm than the phase-based sub-PCA/PLS methods; same length of lagged steps for all phase models</td>
</tr>
<tr>
<td>Phase-based sub-PCA</td>
<td>Conventional algorithm</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Process analysis and monitoring</td>
<td>Solving a series of problems related to batch process characteristics</td>
</tr>
<tr>
<td>For uneven-length batch processes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Process analysis and monitoring</td>
<td></td>
</tr>
<tr>
<td>For limited reference data</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Process analysis and monitoring</td>
<td></td>
</tr>
<tr>
<td>For transition modeling</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Process analysis and monitoring</td>
<td></td>
</tr>
<tr>
<td>Phase-based PLS</td>
<td>Conventional algorithm</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Quality-related analysis, quality prediction and online quality control</td>
<td>Identification of critical phases and key variables for quality prediction and control</td>
</tr>
<tr>
<td>Using phase-specific average trajectory</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Need</td>
<td>Quality-related analysis and quality prediction</td>
<td>Looking at the time accumulation effects on qualities</td>
</tr>
</tbody>
</table>
together with the variable correlations within each phase. This is the only method in Table 2 that can explore inter-phase relationships. This is quite helpful in understanding a process through offline process analysis, although this type of technique does not have significant benefits on online monitoring. Phase MPCA can also be used to analyze each phase separately, although its online monitoring performance is affected by future data estimation. AHPCA was developed for higher monitoring efficiency than possible with MPCA. The cost is heavier computation and a larger storage burden. Meanwhile, the weight parameter in AHPCA is set to be same throughout the entire batch, which does not reflect different phase characteristics. Phase AHPCA overcomes this problem, but its computation and storage burdens are heavy. AHPCA and phase AHPCA models each sample intervals separately, and they are seldom used in process analysis. MPPCA and phase-based sub-PCA can be regarded as two comparable and alternative methods, which were proposed based on similar motivations. Both methods have advantages in process monitoring applications, although the calculation of MPPCA models is more complicated than that of sub-PCA. Besides monitoring, sub-PCA has been used in process analysis. MPPCA has not been used for such purpose, although process analysis can also be achieved based on MPPCA by looking at the phase score plots or loading plots.

In batch process quality prediction, multiblock PLS can work, but does not have many advantages in online quality prediction compared to other methods. Priority PLS studies accumulated phase effects on quality. This method includes all data before the end of the last critical-to-prediction stage in the prediction model, although some stages may not affect the quality. This could be a shortcoming. At the same time, the different critical-to-prediction stages for different quality variables are not revealed. MPPLS performs quality prediction based on phase models, while the accumulated phase effects on quality are not considered. The phase-based PLS method has advantages in quality-related analysis. It identifies the critical-to-quality phases and the key process variables significantly contributing to quality. Based on the gathered knowledge, better product quality can be achieved with tight process control of key variables in critical phases. At the same time, reasonable and meaningful quality prediction models can be built for each quality variables with the phase cumulative effects taken into consideration. Online quality control can be performed to improve the product quality. Another phase-based PLS method utilizes the phase-specific average trajectory in quality prediction and focuses on both the time and phase accumulation effects on quality.

7. Conclusion and prospects for future work

Multivariate statistical methods have been widely applied in batch process modeling, monitoring, quality prediction and quality control. However, most MSPC methods take the entire batch data set as a single object and do not pay attention to the multiple stage/phase characteristics of many batch processes. To ensure process operation safety and product quality, such multistage/multiphase features should be taken into consideration while building a MSPC model.

There are several ways to divide phases, including process knowledge-based phase division, process analysis-based phase division and data-based automatic phase division. Among them, the automatic division methods are preferred. Since these methods are based on data, they do not require prior process knowledge, which may not be available in many cases. They provide a basis for the analysis and modeling of different kinds of multistage/multiphase batch processes.

Different methods have been proposed to model the multistage/multiphase batch processes, which can be divided into two big groups: multiblock techniques and separate phase modeling techniques. Multiblock techniques interpret the process as a single model with the data grouped in several blocks. Separate phase modeling techniques model each phase's data with a separate MSPC model. In this paper, both types of modeling methods are reviewed and compared. The advantages and disadvantages of each method are discussed. We hope that the comparison between the methods can provide a guideline to choose the proper modeling method for specific application purposes.

The existing phase division and modeling methods carefully consider many characteristics of multistage/multiphase batch processes and yield satisfactory application results. At the same time, further research efforts are still needed.

In the literature on the multiphase (multistage) batch process modeling, phase dynamics have been largely ignored, while recent phase-based statistical modeling methods can provide a good platform for analyzing batch processes. Future studies should consider extending to include phase dynamics for more effective modeling, monitoring and quality prediction.

Most MSPC methods including the phase-based statistical modeling methods take the assumption that all process variables are sampled under the same sampling rate. However, in real-world industrial settings, different sampling rates are used for different variables. The variables with high sampling rates are usually down-sampled for modeling, which wastes information. How to model multistage/multiphase batch processes with multiple sampling rates is a topic deserving study.

Nonlinearity is another characteristic of many batch processes, while most MSPC methods, including most methods reviewed in this paper, are linear. Although most nonlinearities are removed from the data through normalization, a proper nonlinear multivariate statistical method can model the batch processes better. There have been some research works combining PCA or PLS with other techniques, such as different types of artificial neural networks (ANN), to solve the nonlinear modeling problem (Dong & McAvoy, 1996b; Lin, Qian, & Li, 2000; Qin & McAvoy, 1992), but there is still a long way to go before these methods can be widely applied to the industrial processes. Applicable nonlinear MSPC batch process modeling methods that take multistage/multiphase characteristics into consideration would be useful.

As discussed in the last section, each modeling method has advantages and disadvantages and can be applied for different purposes. In 2002, Undey and Cinar (2002) proposed a monitoring scheme to combine phase MPCA, phase AHPCA and a high-level model for better process monitoring, fault diagnosis and process analysis. After that, many other multistage/multiphase modeling methods were developed in succession for process analysis, monitoring, quality prediction and quality control, as introduced earlier. If the advantages of these methods can be further combined into an integrated scheme, process safety and product quality can be ensured. Such integration work will be meaningful and will lead to more practical applications of multistage/multiphase MSPC methods.

Process monitoring and fault diagnosis can be used to detect system failures and estimate faults, but such diagnosis does not have the ability to maintain process performance at an acceptable level in the presence of faults. Fault tolerant control (F TC) can be applied to achieve such a target. However, FTC algorithms are usually designed for continuous processes, and there are few studies on FTC for batch processes (Wang, Shi, Zhou, & Gao, 2006; Wang, Yang, Zhou, & Gao, 2007; Wang, Zhou, & Gao, 2008b). Obviously, active FTC algorithms with monitoring modules could be proposed for batch processes in the future. In particular, there is no reported study on FTC for multistage/multiphase batch processes, so the multistage/multiphase SPC methods will play an important role in this field.


Dr. Yuan Yao was born in Hangzhou, Zhejiang, China, 1978. He received the Bachelor and Master degrees of Engineering in the Department of Control Science and Engineering from Zhejiang University in China, in 2001 and 2004, respectively. Then, in 2009, he received the Ph.D. degree in the Department of Chemical and Biomolecular Engineering from the Hong Kong University of Science and Technology. He is presently doing postdoctoral research in Center for Polymer Processing and Systems (CPPS) of HKUST Fok Ying Tung Graduate School. His current research interests mainly focus on multivariate statistical process control.

Dr. Furong GAO received his BEng in Automation from China University of Petroleum in 1985, MEng and PhD degrees both in Chemical Engineering from McGill University in 1989 and 1993, respectively. Prior to his joining of Hong Kong University of Science and Technology (HKUST) in 1995, he worked as a Senior Research Engineer in Moldflow Ltd., Melbourne, in charge of the technical development of the intelligent process control project for the company. Currently, Dr. Gao is a Professor of Chemical & Biomolecular Engineering at HKUST, and the founding director of Center for Polymer Processing and Systems, in HKUST Fok Ying Tong Graduate School. He received several best paper awards, and he is an Associate Editor of *Journal of Process Control*, an editorial advisory board member of *Industrial & Engineering Chemistry Research* (I&ECR). His research interests are in process monitoring and control, and polymer processing.