Augmented dynamic PCA approach for online monitoring of multi–phase batch processes

Xuan-Tien Doan [a,*], R. Srinivasan [a,b]

[a] *Institute of Chemical and Engineering Sciences, 1 Pesek road, Jurong Island, Singapore 627833*

[b] *Department of Chemical and Biomolecular Engineering, National University of Singapore, 10 Kent Ridge crescent, Singapore 119260*

Abstract

Online monitoring of batch processes using multivariate statistical process control (MSPC) techniques, PCA in particular, has been a challenging problem. The key issues include the 3-D nature of batch data, unequal batch lengths or variation in the timing for key dynamic events in reference database, and incomplete online data for evolving batch as first outlined in Nomikos and MacGregor (1995a,b,c). In addition, complex dynamics of batch processes (ie., highly nonlinear, time–varying, multi–stage/multi–phase) presents the extra challenges to their monitoring (Ündey and Çinar, 2002). To deal with each of these issues, many solutions have been proposed. However, we observe that no single method can handle all of the identified issues. Each of the methods was designed to specifically and particularly deal with one or two of the issues but not all and hence a combination of different methods is necessary. We propose a framework for such a combination by integrating *dynamic feature synchronization* and *dynamic time warping* (DTW) with *Dynamic Principal Component Analysis* (DPCA) proposed in Chen and Liu (2002). The strategy here is firstly identifying the singular points marking different process stages that are then aligned optimally by DTW and later analyzed by DPCA. We use the concept of singular points (SP) as defined in (Srinivasan and Qian, 2005) and observe that a SP breaks normal correlation of residuals from the best fit of recent moving window. In addition, we propose and implement an algorithm for online DTW application by removing the end–point constraint and considering the optimization of all possible end–point matching. Even though this could lead to sub-optimal warping, the modified DTW algorithm can be implemented online in a computationally efficient fashion. For DPCA monitoring, scaling against batch mean trajectory is selected because the goal is to detect deviations from the desired operation.

The proposed method, which is called augmented DPCA, is implemented on PenSim simulation – a dynamic simulation of fed–batch penicillin production (Undey and Cinar, 2002). Original DPCA as proposed in Chen and Liu (2002) is also implemented. Comparison between augmented DPCA and original DPCA shows that the augmented DPCA outperforms the original one in monitoring PenSim. The superi-

ority of augmented DPCA demonstrates the need for integrating different methods for online monitoring of multi-stage batch processes.

Key words:
online monitoring, batch process, multi–phase process, feature synchronization, dynamic time warping, dynamic PCA

1 Introduction

Recently chemical industry and research community's focus has been shifted from large–volume continuous processing to low volume but high value-added batch operations. Manufacturing chemicals through batch processing is increasingly common in most of the major industries such as pharmaceuticals, fine and specialty, semiconductor, polymers etc. Hence supervision of batch processes has attracted extensive industrial as well as academic research attention. With recent advances in online data acquisition, database of hundreds of measured variables recorded on a frequent basis for tens to hundreds of batches has become available to process operators. The ability to utilize effectively the information contained in the database for batch process supervision has been the driving force in using multivariate statistical process control (MSPC) methods for online monitoring of batch processes.

J. F. MacGregor and his co–workers pioneered the work in monitoring of batch processes using MSPC methods. Their papers not only identified the key issues in batch process monitoring using MSPC approaches but also proposed some of the most widely used solutions to the issues. Important issues identified in Nomikos and MacGregor (1995a,b,c) include the 3–D nature of batch data, variation in batch duration, and incomplete online data for evolving batch. In addition, other studies (Ündey and Çinar, 2002; Chen and Liu, 2002; Lee et al., 2004a,b) suggest that complex dynamics (including nonlinear, unsteady state, time–varying, multistage/multiphase) in batch processing presents further challenges to the batch monitoring problem. These issues are now discussed

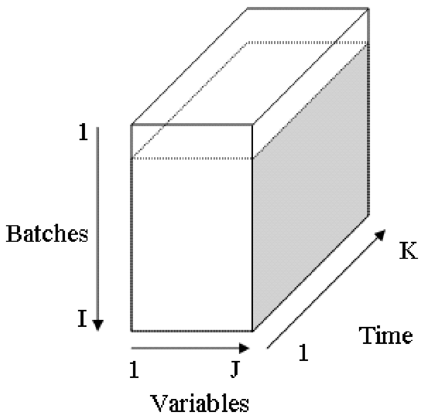

Fig. 1. Historical batch data are three dimensional matrix: batch dimension $(\overline{1,I})$, variable dimension $(\overline{1,J})$, and time dimension $(\overline{1,K})$

1.1 3-D nature of batch data

Fig. 1 illustrates the 3–D nature of historical batch data. The additional dimension comes from batch order (I) in the historical data set. Various MSPC methods exist for dealing with 3–D historical data set including unfolding technique combined with MSPC methods (such as multiway principal component analysis), and three–way data modelling techniques (such as Tucker3, parallel factor analysis – PARAFAC).

Of these, the most common methods are multiway principal component analysis (MPCA)/multiway partial least square (MPLS), introduced by J. F. Macgregor and his co–workers (Nomikos and MacGregor, 1995a,b,c). The principal idea behind these multiway methods is unfolding 3–D historical data set into 2–D matrix and then applying principal component analysis (PCA)/partial least square (PLS) techniques. There are three possible ways to accomplish the unfolding: time–wise unfolding for analyzing sample variation; variable–wise for variation across the batch variables, and batch–wise for variability among batches. In literature, batch–wise and variable–wise unfolding (Fig. 2 and 3 respectively) have been applied in batch process monitoring. Nomikos and MacGregor (1995c) recommended batch-wise unfolding is most appropriate for monitoring batch processes. Wold et al. (1998) studied variable-wise unfolding for batch process monitoring. Recently, Lee et al. (2004a) combined the two approaches by scaling in batch–wise mode and then rearranging into variable–wise mode.

* Corresponding author. Tel.: +65 6796 3945; fax: +65 6316 6185
Email addresses: doan_xuan_tien@ices.a-star.edu.sg (Xuan-Tien Doan), chergs@nus.edu.sg (R. Srinivasan).

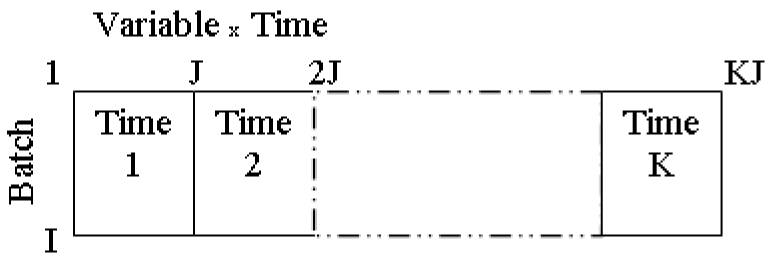

Fig. 2. Batch–wise unfolding

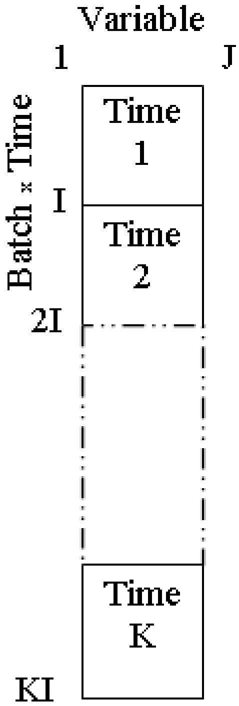

Fig. 3. Variable–wise unfolding

Alternative methods for dealing with 3–D batch data set such as PARAFAC and Tucker3 have also been studied (Nomikos and MacGregor, 1989; Louwerse and Smilde, 2000). However, these methods are not as popular might be because the generalization of MSPC research results for 2–D continuous processes to 3–D batch processes is not always a straightforward matter.

1.2 Variation in batch duration

Regardless of which method is used for the 3–D problem, all batches and/or their individual phases/stages in a single batch process might take invariantly different duration of time to complete. Fig. 4 illustrates the timing variation problem in analyzing batch process data. Two trajectories from the same batch process have corresponding end–points (A, C and A', C'), and dynamic

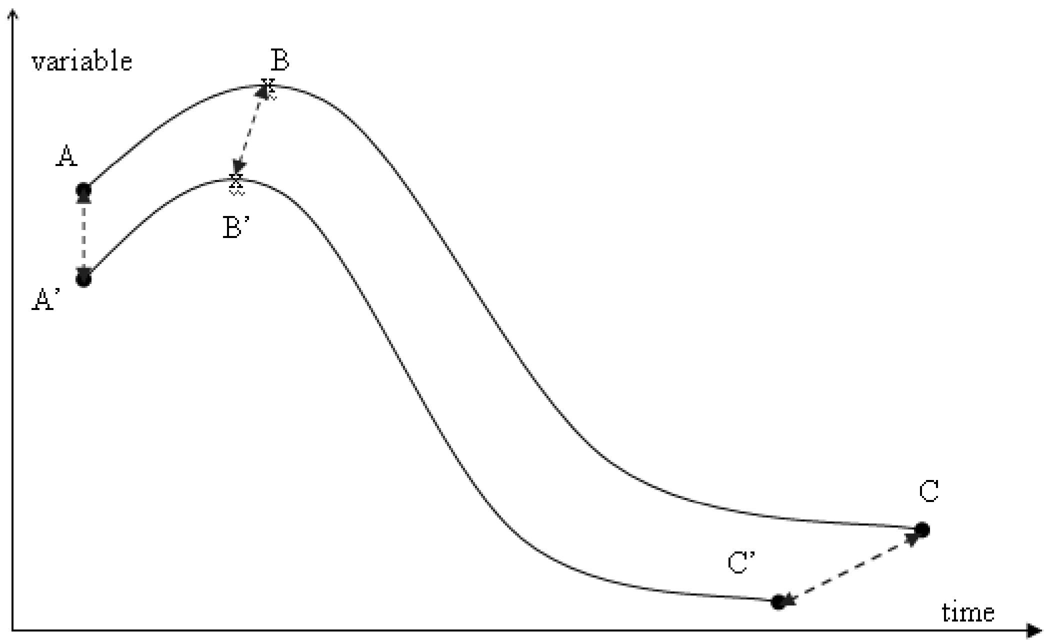

Fig. 4. Batch trajectory timing varies from batch to batch, which causes mismatch in trajectory end-points as well as other dynamic features: A, B, C and A', B', C' are not (vertically) aligned in time respectively

features (B and B'). Even though both trajectories start at the same time (A and A' are aligned vertically), they do not reach maximum (B and B') nor complete (C and C') at the same instances. The reasons might be due to variation in process operating condition, in raw materials, and in process operator's (manual) decision. As these factors are inherent to process operation, variation in batch duration is not avoidable when analyzing its historical data set.

A number of solutions to this practical problem have been discussed and reviewed in various literature (Nomikos and MacGregor, 1995c; Ündey and Çinar, 2002; Ündey et al., 2003). The easiest solution would be cutting bach data to an equal length. Although it has been commonly used in practice (Gregersen and Jorensen, 1999; Lennox et al., 2000; L. Lu. Zheng and Chen, 2001), this technique is not recommended because significant information would be lost (due to discarding the cut–off data) (Ündey and Çinar, 2002), and even so it does not deal with the mismatch in timing of other dynamic features (B and B' in Fig. 4).

Another simple but widely used technique is the *indicator variable technique* (*IVT*). The principal idea is to select a process variable and use it for aligning the batch trajectory. The selected variable must be smooth, monotonically progress in time, and have the same characteristics (starting/ending values and other dynamic features) for all batches (Nomikos and MacGregor, 1995c). Examples of possible indicator variables are reaction extent (Neogi and Schlags, 2001), percent of reactor volume decrease and percent of substrate addition (Ündey et al., 2004) etc. IVT is simple, easy to implement but its applicability relies on the existence of a suitable indicator variable which may not always exist for some batch process.

Alternatively, dynamic time warping (DTW) has also been used to overcome the problem of variation in batch length. DTW, originated from speech recognition research, is a flexible, distance–based pattern matching method. It can locally translate, compress and expand a pair of patterns in such a way that similar features are aligned and minimum distance is obtained (Ündey and Çinar, 2002). Application of DTW was first reported in Gollmer and Posten (1996) with a strong focus in supervision of bioprocesses. Kassidas et al. (1998) applied DTW to trajectory synchronization for batch process monitoring. More recently, its application in synchronizing spectroscopic batch data was presented in Ramaker et al. (2003).

By comparison, DTW requires relatively higher computational resource (Ündey and Çinar, 2002) and extra efforts in its implementation (for being more complicated than IVT). In addition, since it is a distance–based technique and does not in anyway account for process dynamic behaviors, care must be taken to achieve optimal pairings between two trajectories. On the other hand, it is a versatile technique in the sense that it can be applied to all batch processes as opposed to IVT's limited applicability.

1.3 Incomplete online data

Application of MPCA/MPLS requires data for a fully finished batch. MPCA/MPLS, which use batch wise unfolding (cf. Fig. 2), only work on measurement vector of KJ length (where K is the batch time length; J is the number of batch variables). However, for online monitoring, the ongoing batch has its evolving time length $k < K$. This leads to a dimension mismatch as illustrated in Fig. 5 and the online batch data is not applicable to any MPCA/MPLS model that has been developed from corresponding off–line batch data.

Many different methods have been proposed to overcome the problem for online MPCA/MPLS application. Nomikos and MacGregor (1995c) suggested three different ways:

zero deviation Assuming that the future batch data equals to its mean trajectories evaluated from off–line training data sets.

current deviation Assuming that the deviation of the future data from its mean trajectories remains constant.

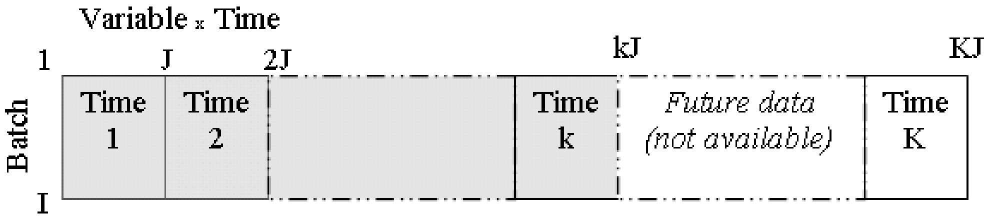

Fig. 5. A dimension mismatch: unfolded measurement vector of an evolving batch is shorter than would be required by MPCA/MPLS model

PCA projection Treating the future data as missing data and then using MPCA/MPLS to predict the missing data.

Although the three methods are attractively simple and have been used in industrial applications (Lennox et al., 2000), their limitations have also been well identified and discussed. For example, *zero deviation* and *PCA projection* might not perform well at the start of a new batch. In addition, *current deviation* method seems to outperform the other two, provided that some knowledge of process disturbance is available (Nomikos and MacGregor, 1995c).

Another method for anticipating future online batch data is proposed by Cho and Kim (2003). The principal idea is to choose a batch from historical batch data sets, that is the most similar to the evolving one based on the sum of squared error. The potential disadvantage lies in the requirement for an extensive high–quality data sets of historical batch which directly affects the prediction accuracy (Cho and Kim, 2003).

1.4 Time–varying, unsteady state, multi–phase characteristics

Batch processes are inherently time–varying, unsteady state and more often than not multi–phase. These characteristics have further complicated the online monitoring of batch processes (Lee et al., 2004a). The same dynamic problems encountered in monitoring continuous processes have been studied and reported extensively. Wold (1994) coupled exponentially weighted moving average into PCA for monitoring time–varying continuous processes. Ku et al. (1995) used the "time–lag shift" method to account for dynamic behavior using a PCA model (that is otherwise static). The same concept was applied for predictive monitoring of continuous process using MPCA (Chen and McAvoy, 1998). However, the author also observed that few attempts had made in adapting the results to batch processes and then presented the application of dynamic PCA (DPCA) to batch process monitoring (Chen and Liu, 2002). Their proposed method is based on averaging the covariance matrices of the time–lag augmented data for each batch. It deals with the unsteady state transient behaviors as well as the 3–D characteristic of historical batch data (cf. Section 1.1) and the incomplete online data issue (cf. Section 1.3). However, as no particular scaling method was stated and if scaling against batch trajectory is chosen (as suggested in Nomikos and MacGregor (1995c)), other problem such as mismatch in dynamic feature alignment (cf. Section 1.2) would be expected.

Multi–phase is another phenomenon that makes batch process monitoring challenging. Essentially, individual phase has (significantly) different dynamics and should deserve its own modelling. The advantage of phase–based monitoring was first demonstrated in Kosanovich et al. (1994); Dong et al. (1996). However, both of the studies only considered stages with fixed time length. Ündey and Çinar (2002) looked further on online monitoring of multi–stage/multi–phase batch processes using PCA coupled with indicator variable technique (IVT) for identifying and aligning corresponding phases/stages. Their approach requires some process knowledge about phases/stages in the batch process and suffers from IVT's disadvantage (of being limited applicability). Taking a different approach, Lu and Gao (2004) reasoned that process correlation remains more or less the same within an "operation" stage and hence used K–means clustering to identify process stages. However, the problem of variation in time length of corresponding stages was not addressed. In a continuing paper (Lu et al., 2004), they proposed to evaluate the mean trajectory from what is available. Effectively, this means using the time span for matching and aligning the data, which clearly does not solve the problem of mismatch in process dynamic features (ie. phases). More recently, Lu and Gao (2005) presented the application of their method to an injection molding process, in which IVT was used for data aligning.

It is observed that even though there are numerous proposed solutions to the key issues in monitoring of batch processes using MSPC, no particular approach is able to handle effectively and efficiently all of the identified issues. For example, most of the time, the problem of variation in time span of batch data and/or its phases was dealt with by making assumptions or using IVT, the applicability of which is not always guaranteed.

In this study, we apply DPCA approach coupled with DTW to each "phase" of a multi–phase batch process for online monitoring. As we define, a "phase" might not necessarily correspond to but otherwise will always include operational or phenomenological phases/stages. The proposed framework includes *dynamic feature synchronization* plus *dynamic time warping* (DTW) plus *Dynamic Principal Component Analysis* (DPCA) (Chen and Liu, 2002). Our approach identifies and aligns the dynamic features which mark the begins and ends of process phases. A pseudo–online DTW algorithm is then implemented for warping and aligning the corresponding phases. Subsequently, DPCA will be used for monitoring each phase of the multi–phase batch process. PenSim simulation – a dynamic simulation of fed–batch penicillin production, is used to verify the superiority of the proposed framework over alone DPCA approach. Similar analysis is also carried out on injection molding data which was very kindly provided by Dr. Lu and Prof. Gao (Lu and Gao, 2004).

2 Augmented DPCA

2.1 DPCA

DPCA introduced in Chen and Liu (2002) is principally the dynamic PCA (DPCA) concept being applied to batch processes. Even though its concept was dated back in Chen and McAvoy (1998), it was not until Chen and Liu (2002) who first applied it to monitoring of batch processes.

Mathematically, DPCA starts with forming a time–lagged window for each of the batches in reference database

$$\mathbf{X}_d^i = \begin{pmatrix} (\mathbf{x}^i(d+1))^T & (\mathbf{x}^i(d))^T & \ldots & (\mathbf{x}^i(1))^T \\ (\mathbf{x}^i(d+2))^T & (\mathbf{x}^i(d+1))^T & \ldots & (\mathbf{x}^i(2))^T \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ (\mathbf{x}^i(K))^T & (\mathbf{x}^i(K-1))^T & \ldots & (\mathbf{x}^i(K-d))^T \end{pmatrix} \tag{1}$$

Where: $\mathbf{x}(k) = [x_{1,k}\ x_{2,k}\ \ldots\ x_{J,k}]^T$ is the J–dimensional observation vector at time k. K is the batch length. d is the time lag. i indicates i^{th} batch.

The corresponding covariance matrix $\mathbf{S}^i$ for the time–lagged data is

$$\mathbf{S}^i = \frac{(\mathbf{X}_d)^T(\mathbf{X}_d)}{K-d-1} \tag{2}$$

And the average covariance matrix $\mathbf{S}^{avg}$ for I batches

$$\mathbf{S}^{avg} = \frac{(K-d-1)\Sigma_{i=1}^{I}\mathbf{S}^i}{I(K-d)} \tag{3}$$

Solving the eigen–decomposition of the covariance matrix $\mathbf{S}^{avg}$ and retaining a principal components results in the DPCA model of the batch process.

T^2 statistic

When PCA is used in monitoring, Hotelling's T^2 is usually employed as a monitoring index. T^2 statistic is a scaled squared 2–norm of an observation vector from its mean.

$$(T_k)^2 = (\mathbf{X}(k))^T \mathbf{P}_a \mathbf{\Lambda}_a^{-1} \mathbf{P}_a^T \mathbf{X}(k) \tag{4}$$

Where: $\mathbf{X}(k) = [(\mathbf{x}(k))^T \quad (\mathbf{x}(k-1))^T \ldots (\mathbf{x}(k-d))^T]$ is the time–lagged vector of the current measurement $\mathbf{x}(k)$. $\mathbf{P}_a$ is the loading matrix containing a loading vectors. $\mathbf{\Lambda}$ is a diagonal matrix containing the a principal components.

The control limit T_α^2 can be approximated by means of the F–distribution

$$T_\alpha^2 = \frac{a(K-d-1)(K-d+1)}{(K-d)(K-d-a)} \mathbf{F}_{a,K-d-a,\alpha} \tag{5}$$

Where: α is the confidence limit, which is set at 95% in this study.

Procedure to implement DPCA for batch process monitoring is briefly described below

2.1.1 Off–line modelling

(1) Equalize all batch lengths in the training set by cutting off to the shortest.
(2) Perform time–lagged operation for each batch in reference database with $d = 2$ (Equation 1)
(3) Obtain the corresponding batch mean and standard deviation matrices and use it to scale all reference batches.
(4) Evaluate covariance matrix S^i (Equation 2) for each reference batch and then the average covariance S^{avg} (Equation 3)
(5) Obtain DPCA model by eigen–decomposing S^{avg} and retaining $a = 3$ principal components.
(6) Evaluate T^2 control limit at 95% confidence limit (Equation 5).

2.1.2 Online monitoring

(1) Obtain the new measurement $\mathbf{x}$.
(2) Perform time–lagged operation on $\mathbf{x}$ with $d = 2$.
(3) Retrieve the reference and batch mean trajectories. Identify the match of $\mathbf{x}$ in the batch mean trajectory, which has the same time index as that of $\mathbf{x}$.
(4) Scale $\mathbf{x}$ using its corresponding points in the batch mean trajectory .

(5) Evaluate T^2 statistic using the scaled measurement and the DPCA model. Compare the obtained T^2 statistic with its control limit. If exceeded, announce a fault. Otherwise go back to Step 1 for the next measurement.

The attractiveness that DPCA has to offer are two folds: its capability to capture dynamic behaviors of a batch process and its simplicity. The dynamic capability would help to improve PCA performance in detecting fault occurrence from serially correlated data (Chen and Liu, 2002). More importantly, DPCA model development and especially online implementation are more or less the same as ordinary PCA applied in a continuous process. The only difference is in evaluating the average covariance matrix. In online monitoring, DPCA processes measurement vector $\mathbf{x}$ $(1 \times J)$ one at a time, as opposed to a whole batch data in MPCA approach. This means that the problem of incomplete online data (cf. Section 1.3) is avoided in DPCA.

DPCA as presented in Chen and Liu (2002) implicitly assumed that all batches in reference database have the same batch length (K). In addition, Chen and Liu (2002) did not explicitly state any scaling procedure that was used. As scaling is a critical operation for any PCA–based monitoring techniques, this issue must be properly addressed. For batch process monitoring, scaling against batch mean trajectory would be most appropriate because the goal is to detect deviations from the desired operation. As a result, batches from the reference training set must be of the same length and their dynamic features must be aligned so that batch–wise unfolding of the 3–D database could be carried out and thereby obtain the batch mean trajectory. This raises the fact that DPCA needs a technique to ensure that all batches in reference database are aligned in terms of dynamic behaviors and hence have equal lengths. Chen and Liu (2002) implicitly assumed all batches have the same length and hence did not experience the issue. In this study, we do not make the same assumption and instead select *Dynamic Time Warping* (DTW) for warping and aligning the data.

2.2 DTW

DTW originates from speech recognition and is capable of translating, compressing, and expanding a pair of trajectories in such a way that similar events are aligned and a minimum distance between them is obtained Ündey and Çinar (2002). Generally, there are two classes of DTW methods including *symmetric* DTW and *asymmetric* DTW. While the former one treats the two trajectories equally ie., both time axes are transformed onto a newly defined common axis, the latter maps the time axis of the *sample* trajectory onto that of the *reference* trajectory. In this paper, asymmetric DTW is considered.

Let $\mathbf{S}(M \times J)$ and $\mathbf{R}(K \times J)$ denote two multivariate trajectories which correspond to *sample* and *reference* signals respectively. DTW warps $\mathbf{S}$ onto $\mathbf{R}$ by searching for an optimal sequence $\mathbf{F}^*$ of P points on an i–j plane such that a minimum distance measure between $\mathbf{S}$ and $\mathbf{R}$ is obtained. Mathematically,

$$\mathbf{F} = \{c(1), c(2), \ldots, c(p), \ldots, c(P)\} \tag{6}$$

$$c(p) = [i(p) \quad j(p)] \tag{7}$$

$$d(c(p)) = \|\mathbf{R}(i(p)) - \mathbf{S}(j(p))\| \tag{8}$$

$$D(\mathbf{S}, \mathbf{R}) = \frac{\sum_{p=1}^{P} d(c(p)) \cdot w(p)}{\sum_{p=1}^{P} w(p)} \tag{9}$$

and

$$\mathbf{F}^* = argmin_{\mathbf{F}}[D(\mathbf{S}, \mathbf{R})] \tag{10}$$

Where: i and j are the time index of the $\mathbf{R}$ and $\mathbf{S}$ trajectories respectively. $w(p)$ are weighting coefficients, which are set at 1 in this study.

Fig. 6 illustrates schematically the DTW's concept.

In addition, DTW needs a set of endpoint, global and local constraints. Usually endpoint constraints which require that the endpoints of $\mathbf{S}$ and $\mathbf{R}$ must match, are used in off-line analysis. On the other hand, global constraints define and restrict the search space while the local ones define the localized feature of the optimal path Kassidas et al. (1998). It is the set of these constraints that determines the details of DTW algorithm. In this paper, asymmetric DTW with slope 1/2 and band global constraint $B = 30$ is used. Detailed DTW algorithm can be found in Sakoe and Chiba (1978).

For online application of DTW, there is an additional challenge. As a batch evolves, its online data is not complete and hence its endpoints are unknown. Consequently, endpoint constraints can not be imposed as the corresponding (in the reference $\mathbf{R}$ trajectory) of the current online measurement is unknown. The DTW algorithm in this paper replaces that constraint by minimizing the distance between the current trajectory with a possible match in the reference trajectory.

$$i^* = argmin_{\mathbf{i}}[D(\mathbf{S}, \mathbf{R}_i)] \tag{11}$$

Where: $\mathbf{R}_i$ is the reference trajectory from time $t = 1$ to time $t = i$.

In other words, the idea is to choose the best pairing among those that are available and satisfied the constraints. The strategy is illustrated in Fig. 7. Similar idea for online DTW implementation was also proposed (but not yet

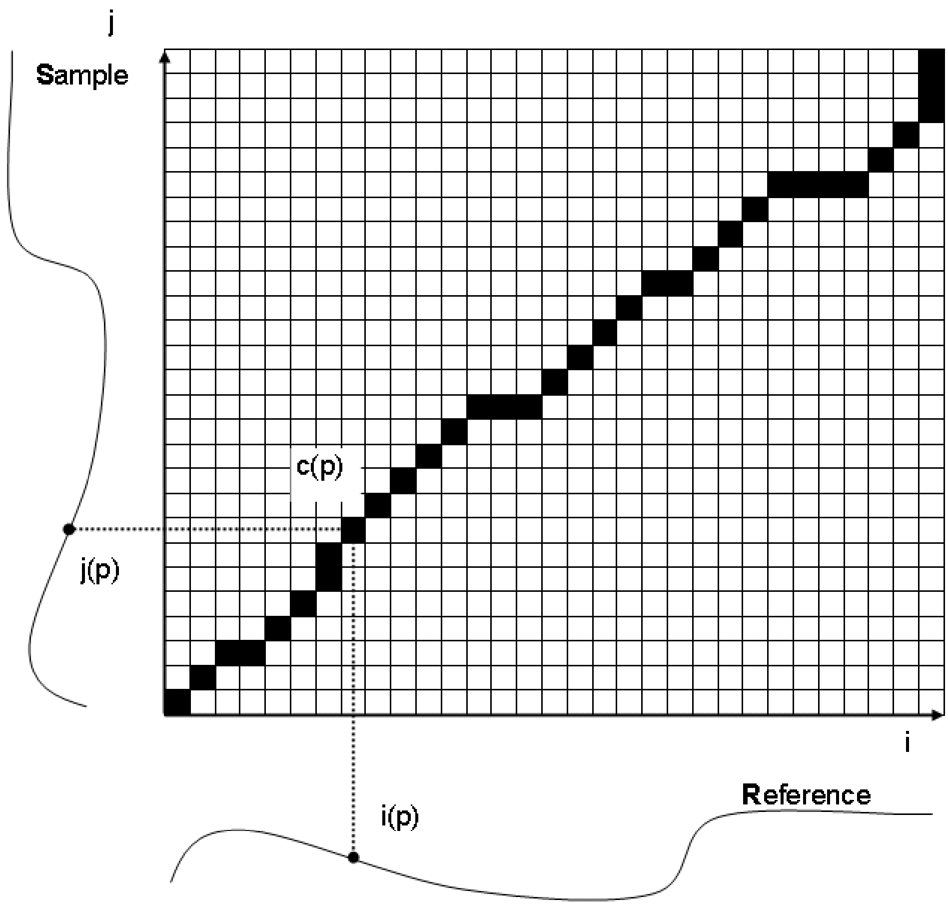

Fig. 6. DTW concept: warping **S** onto **R** by searching for an optimal sequence $\mathbf{F}^*$ of $c(p)$ points.

verified) in Kassidas et al. (1998). The difference is that online DTW algorithm in this paper stops at obtaining i^* because DPCA only needs it for scaling purpose.

DTW is a powerful technique in trajectory alignment and it has been successfully applied to batch process monitoring Kassidas et al. (1998); Ramaker et al. (2003). However, since DTW is a distance–based technique and does not account for the trajectory's dynamic behaviors, it may fail to identify the correct correspondence between two trajectories. To overcome this limitation, Srinivasan and Qian (2005) proposed augmenting DTW algorithm with feature synchronization by restricting the search for corresponding points in the trajectory pair to within corresponding *landmarks*.

2.3 Dynamic Feature Synchronization

Srinivasan and Qian (2005) noted that *"Information content is not homoge-*

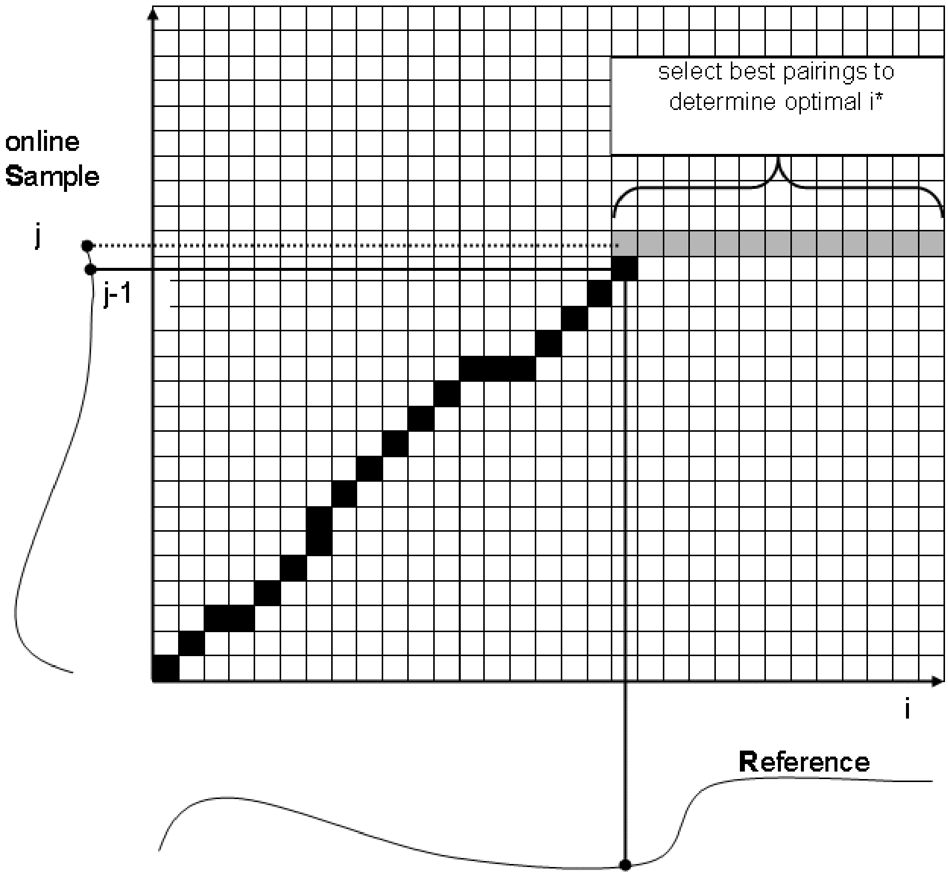

Fig. 7. online DTW strategy: choose the best pairing among those that are available.

nously distributed throughout a signal". In other words, some landmark points, termed *singular points*, in a trajectory contain more information about process dynamic behaviors than other points. Examples of singular points (SP) include points of discontinuities, trend changes, and extrema. For their high information content, SP could be used for signal synchronization and comparison as described in Srinivasan and Qian (2005). For this paper's purpose, SP are used to decompose batch trajectories into multiple "phases" on which DTW are applied.

For SP identification, it is observed that a SP breaks normal correlation of residuals from the best fit of a recent moving window. In other words, the procedure is followed:

(1) Maintain a recent moving window of size τ.
(2) Obtain the residuals from fitting the moving window with a straight line.
(3) Evaluate the standard deviation of the residuals.
(4) Compare current standard deviation with the corresponding past value. If there is significant difference (ie. exceeding a predefined threshold), a SP has been identified. Otherwise, shift the moving window forward.

Two parameters are involved in the proposed procedure. The first one is the moving window size τ. In this study, we have selected τ empirically by inspecting and experimenting with the training data. Although this approach is recognized as of an "ad-hoc" basis, we think it is sufficiently effective and a more analytical approach for evaluating τ would be unnecessary. However, for further work, some relationship between τ and process time constant could be sought as this would provide rough estimate of τ for other study. The second parameter here is the threshold limit for the standard deviation of the residuals. Again, empirical approach was used in determining the threshold. The procedure is applied to the training data (with known SP locations) and the threshold is set so as to correctly detect the known SPs. This threshold limit is then used for online implementation.

Selecting a key variable for SP identification is an important issue. The sole criteria is obviously that the variable's dynamic behavior has to characterize that of the process. In other words, the variable trajectory must reflect all the phases that the process possesses. In some cases, 2 or more key variables could be necessary for the SP identification purpose and process prior knowledge plays a significant role in tuning up the algorithm as well as handling the switching between the many key variables. In our analysis, only a single variable is used for SP identification.

2.4 Augmented DPCA – The proposed framework

As discussed above, DPCA needs additional measure to achieve equal batch length as well as alignment of dynamic features. DTW, selected for this task, can deal with the former issue but may not guarantee the latter, which justifies the need for a dynamic feature synchronization technique. This paper proposes a scheme which integrates *dynamic feature synchronization*, *DTW*, and *DPCA* which is now termed *augmented DPCA*.

2.4.1 Off–line modelling

The following procedure is carried for off–line implementation of the proposed augmented DPCA.

(1) SP identification for all batches in the reference database. Synchronize the identified SP and decompose batch trajectories into corresponding episodes.
(2) Group all corresponding episodes across all the reference batches to obtain 3–D episodes with different lengths.
(3) Apply DTW to the episodes to obtain 3–D episodes with same lengths.

(4) Apply DPCA algorithm to obtain a DPCA model for each of the 3-D episodes.

2.4.2 Online monitoring

Procedure for online monitoring by augmented DPCA approach is outlined in Fig. 8 and summarized below

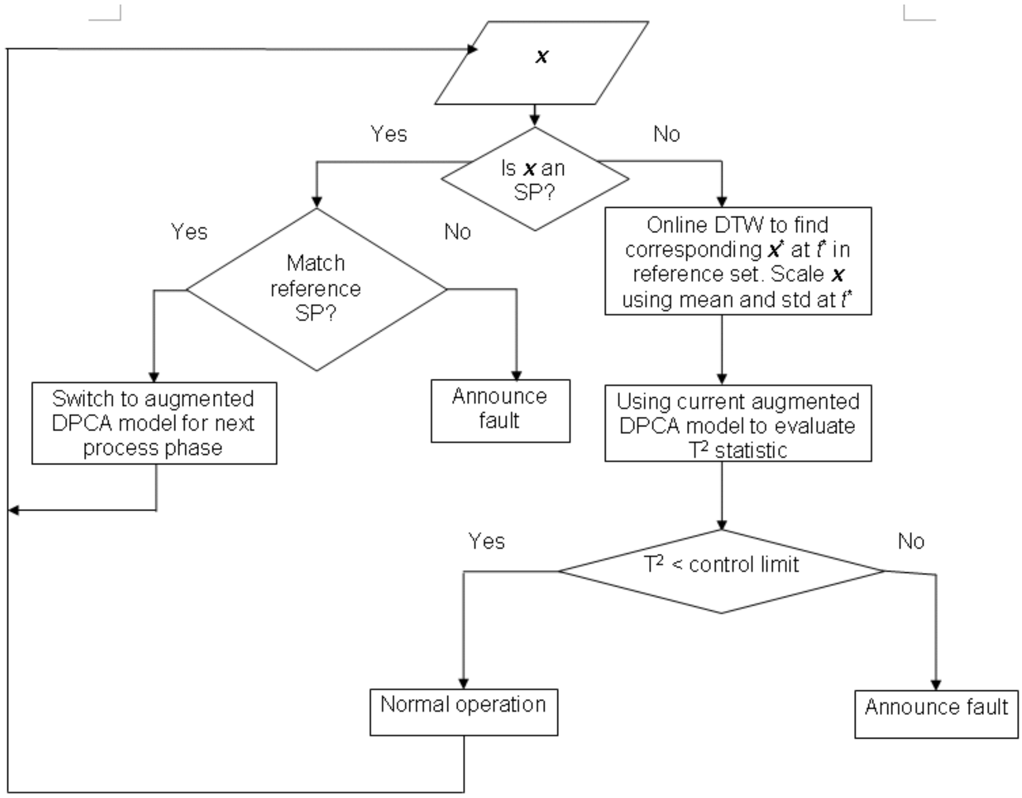

Fig. 8. Online monitoring by augmented DPCA.

(1) Obtain the new measurement **x**. Check if it is a SP. If it is a SP which match the reference SP, retrieve the PCA model that corresponds to the next episode and go back to Step 1. If a SP is detected but does not match the reference, announce a fault. Otherwise, perform time–lagged operation and proceed to Step 2
(2) Retrieve the reference and batch mean trajectories from database of the current episode. Identify the match of **x** in the reference trajectory using online DTW.
(3) Identify the corresponding point in the batch mean trajectory and apply appropriate scaling.
(4) Evaluate T^2 statistic using the scaled measurement and the current DPCA model. Compare the obtained T^2 statistic with its control limit. If exceeded, announce a fault. Otherwise go back to Step 1 for the next measurement.

3 PenSim case study

3.1 PenSim

A fed–batch penicillin cultivation process simulation, known as PenSim, is used as a case study. PenSim, available for download from *http://www.chee.iit.edu/˜cinar*, was developed by the monitoring and control group of the Illinois Institute of Technology to provide a test bed for several applications including batch process monitoring methods (Birol et al., 2002). It has been used in several studies such as Lee et al. (2004a,b); Ündey et al. (2004). The variables generated from PenSim simulation which form the training and test data set are listed in Table 1.

Table 1
Variables used in PenSim case–study

No.	Variables
1	Aeration rate (L/h)
2	Agitator power (W)
3	Substrate feed temperature (K)
4	Dissolved oxygen concentration (% saturation)
5	Culture volume (L)
6	Carbon dioxide concentration (mmol/L)
7	pH

The range of initial conditions and set points are listed in Table 2, which are taken from Lee et al. (2004a). In addition, PenSim can also simulate a number of faults as tabulated in Table 3

3.2 Training and test data

In this study, 41 batches of data are generated using PenSim with sampling time of 0.5 hour; initial conditions and set points randomly chosen in the range specified in Table 2. 31 of the batches have duration of 400 hours. These batches are used to form the training set for DPCA model. The 32^{nd}–41^{st} batches have their duration specified randomly between 380 and 420 hours. These 10 batches together with three other batches (29^{th}–31^{st}) forms training set for the proposed augmented DPCA approach.

Table 2
Initial conditions and set points in PenSim case–study

No.	Variables
Initial conditions	
Substrate concentration (g/L)	14–18
Dissolved oxygen concentration (% sat.)	1–1.2
Biomass concentration (g/L)	0
Penicillin concentration (g/L)	0
Culture volume (L)	100–104
Carbon dioxide concentration (mmol/L)	0.5–1
pH	4.5–5.5
Bioreactor temperature (K)	295–301
Generated heat (kcal)	0
Set points	
Aeration rate (L/h)	8–9
Agitator power (W)	29–31
Substrate feed flow rate (L/h)	0.039–0.045
Substrate feed temperature (K)	295–296
Bioreactor temperature (K)	297–298
pH	4.95–5.05

Table 3
Fault scenarios in PenSim

Fault	**Corresponding variable**	**Type**
1	Aeration rate	15% step
2	Agitator power	15% step
3	Substrate feed temperature	30% step

For test data, Faults 1, 2, and 3 were introduced at the 10^{th}, 10^{th} and 60^{th} and terminated at the 20^{th}, 20^{th} and 80^{th} hour, respectively.

Figures 9, 10 and 11 show the results from implementing DPCA and augmented DPCA approaches on PenSim for Faults 1, 2 and 3 respectively. Summary of the results is presented in Table 4. As the results show, augmented DPCA approach generally outperforms DPCA in detecting the three simulated faults. While DPCA can only detect Fault 2, augmented DPCA detects all of the faults being simulated. Detection delay is one hour for the first two faults (ie., Faults 1 and 2 are detected at 22^{th} sample) due to forming time–lagged data for analysis. However, augmented DPCA takes 11 hours to detect Fault 3, while DPCA completely misses it. The reason is believed due to the insignificant effect of the fault. Fault 3 corresponds to a 30 % step increase in the substrate feed temperature. Under normal conditions, this temperature is around 295–296 (K). In Fault 3 scenario, it is increased to 383–384 (K). However, the substrate feed flowrate is only 0.0412 (L/h) while the culture volume is more than 100 L. Obviously, the 30% jump in substrate feed temperature can not result in any significant effect to the evolving process in the fermentation reactor and hence the fault is more difficult to detect.

Table 4
Monitoring results for PenSim case–study

Fault	**Introduction time** (hour)	**Termination time** (hour)	**Detection delay**(hour)	
			DPCA	Augmented DPCA
1	10	20	not detected	1
2	10	20	1	1
3	60	80	not detected	11

4 Conclusion

We proposed a framework for combining different methods to deal effectively with the most critical issues in batch process monitoring. Our approach, which is termed *augmented DPCA*, integrates *dynamic feature synchronization* with *dynamic time warping* (DTW) and *Dynamic Principal Component Analysis* (DPCA). An algorithm for online DTW was also proposed. We implemented both *DPCA* (Chen and Liu, 2002) and our *augmented DPCA* on PenSim simulation. Comparison between augmented DPCA and original DPCA shows that the augmented DPCA outperforms the original one in monitoring PenSim. The superiority of augmented DPCA demonstrates the need for integrating different methods for online monitoring of multi–phase batch processes.

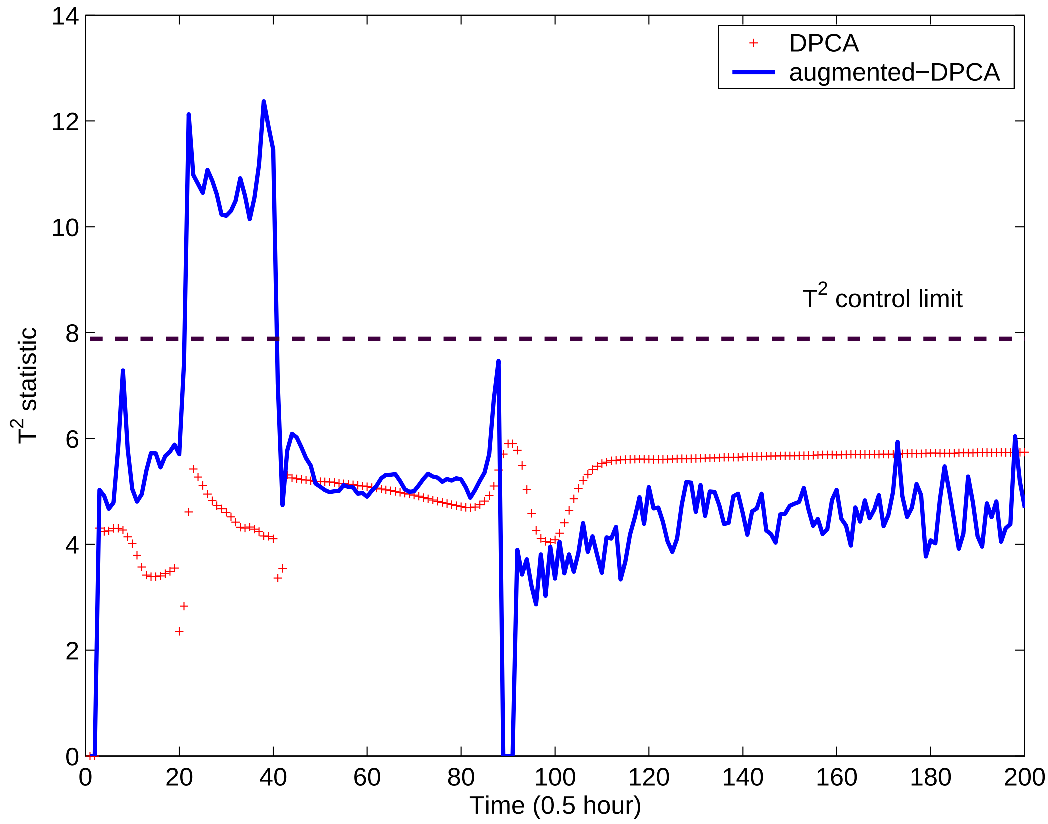

Fig. 9. Fault 1 (aeration rate) simulation – PenSim case study

For future work, we are implementing augmented DPCA on data from a crystallization experiment and injection molding process. Initial results look promising and we will publish the results in a journal soon.

References

Birol, G., Ündey, C., Çinar, A., 2002. A modular simulation package for fed–batch fermentation: penicillin production. Computers and Chemical Engineering 26, 1553–1565.

Chen, G., McAvoy, T. J., 1998. Predictive on–line monitoring of continuous processes. Journal of Process and Control 8, 409–420.

Chen, J., Liu, K. C., 2002. On–line batch process monitoring using dynamic PCA and dynamic PLS. Chemical Engineering Science 57, 63–75.

Cho, H.-W., Kim, K.-J., 2003. A method for predicting future observations in the monitoring of a batch process. Journal of Quality Technology 35, 59–69.

Dong, D., McAvoy, T. J., Wang, F., 1996. Batch tracking via nonlinear principal component analysis. AIChE J. 42, 2199–2208.

Gollmer, K., Posten, C., 1996. Supervision of bioprocesses using a dynamic time warping algorithm. Control Eng. Practice 4, 1287–1295.

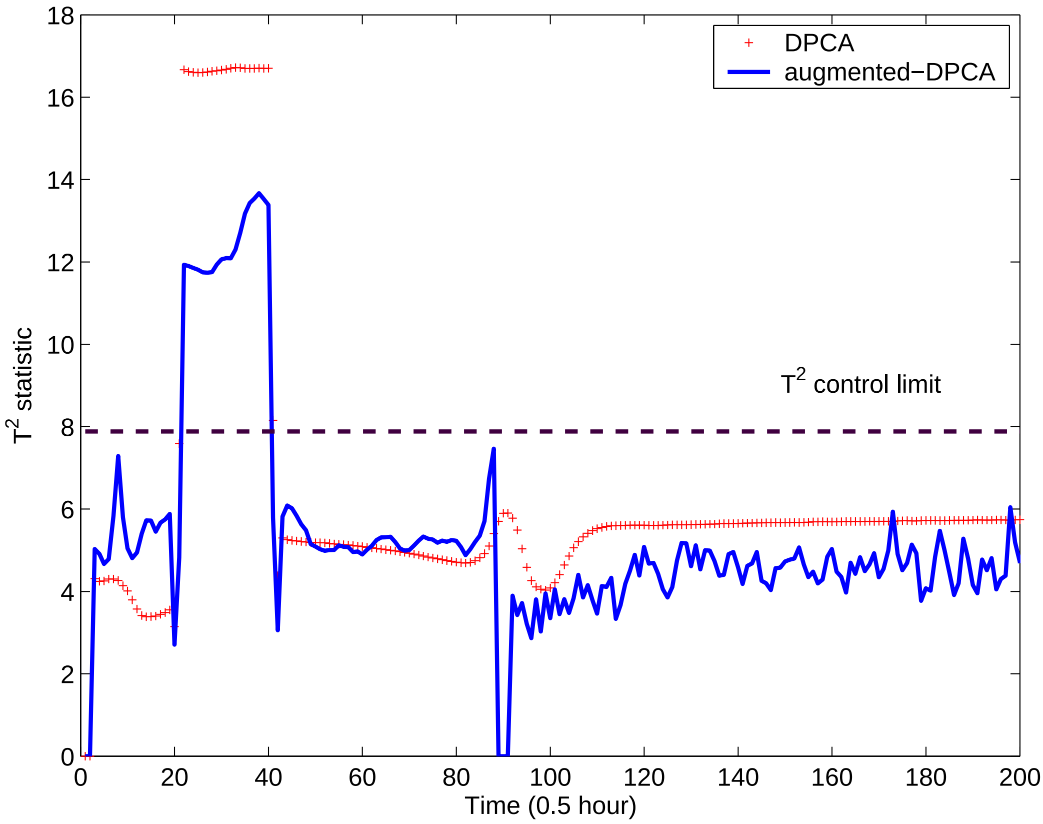

Fig. 10. Fault 2 (agitator power) simulation – PenSim case study

Gregersen, L., Jorensen, S. B., 1999. Supervision of fed–batch fermentations. Chemical Engineering Journal 75, 69–76.

Kassidas, A., MacGregor, J. F., Taylor, P. A., 1998. Synchronization of batch trajectories using dynamic time warping. AIChE J. 44, 864–875.

Kosanovich, K. A., Piovoso, M. J., Dahl, K. S., MacGregor, J. F., Nomikos, P., June 1994. Multi–way PCA applied to an industrial batch process. In: Proceedings of the American Control Conference. pp. 1294–1298.

Ku, W., Storer, R. H., Georgakis, C., 1995. Disturbance detection and isolation by dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems 30, 179–196.

L. Lu. Zheng, Thomas J. McAvoy, Y. H., Chen, G., 2001. Application of multivariate statistical analysis in batch processes. Industrial & Engineering Chemistry Research 40, 1641–1649.

Lee, J. M., Yoo, C. K., Lee, I. B., 2004a. Enhanced process monitoring of fed–batch penicillin cultivation using time–varying and multivariate statistical analysis. Journal of Biotechnology 110, 119–136.

Lee, J. M., Yoo, C. K., Lee, I. B., 2004b. Fault detection of batch processes using multiway kernel principal component analysis. Computers and Chemical Engineering 28, 1837–1847.

Lennox, B., Hiden, H. G., Montague, . A., Kornfeld, G., Goulding, P. R., 2000. Application of multivariate statistical process control to batch operations.

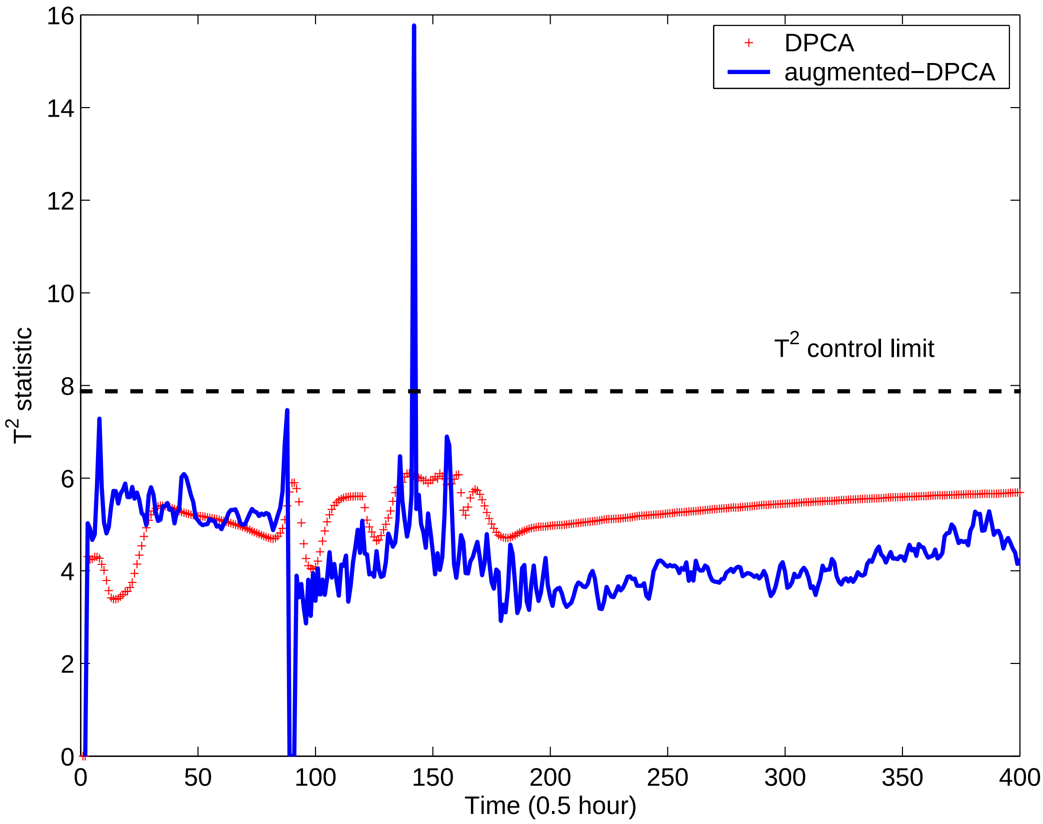

Fig. 11. Fault 3 (substrate feed temperature) simulation – PenSim case study

Computers and Chemical Engineering 24, 291–296.

Louwerse, D. J., Smilde, A. K., 2000. Multivariate statistical process control of batch processes based on three–way models. Chemical Engineering Science 55, 1225–1235.

Lu, N., Gao, F., 2004. Sub–PCA modeling and on–line monitoring strategy for batch processes. AIChE J. 50, 255–259.

Lu, N., Gao, F., 2005. Stage–based process analysis and quality prediction for batch processes. Industrial & Engineering Chemistry Research 44, 3547–3555.

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

Neogi, D., Schlags, C. E., 2001. Applications of multivariate statistical techniques for monitoring emulsion batch processes. Proceedings of the American Control Conference, 1177–1181.

Nomikos, P., MacGregor, J. F., 1989. Analysis of multi–way (multi–mode) data. Chemometrics and Intelligent Laboratory Systems 7, 11–30.

Nomikos, P., MacGregor, J. F., 1995a. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. Journal of Process Control 5, 277–284.

Nomikos, P., MacGregor, J. F., 1995b. Multi–way partial least squares in monitoring batch processes. Chemometrics and Intelligent Laboratory Systems 30, 97–108.

Nomikos, P., MacGregor, J. F., 1995c. Multivariate SPC charts for monitoring batch processes. Technometrics 37, 41–59.

Ramaker, H. J., van Sprang, E. N. M., Westerhuis, J. A., Smilde, A. K., 2003. Dynamic time warping of spectroscopic BATCH data. Analytica Chimica Acta 498, 133–153.

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-26, 43–49.

Srinivasan, R., Qian, M. S., 2005. Offline temporal signal comparison using singular points augmented time warping. Industrial & Engineering Chemistry Research In press.

Ündey, C., Çinar, A., 2002. Statistical monitoring of multistage, multiphase batch processes. IEEE Control Systems Magazine 10, 40–52.

Ündey, C., Ertunç, S., Çinar, A., 2003. Online batch/fed–batch process performance monitoring, quality prediction, and variable–contribution analysis for diagnosis. Industrial & Engineering Chemistry Research 42, 4645–4658.

Ündey, C., Tatara, E., Çinar, A., 2004. Intelligent real–time performance monitoring and quality prediction for batch/fed–batch cultivations. Journal of Biotechnology 108, 61–77.

Wold, S., 1994. Exponentially weighted moving principal components analysis and projections to latent structures. Chemometrics and Intelligent Laboratory Systems 23, 149–161.

Wold, S., Kettanhe, N., Friden, H., Holmberg, A., 1998. Modelling and diagnostics of batch processes and analogous kinetic experiments. Chemom. Intell. Lab 44, 331–340.