González Martínez, JM.; De Noord, O.; Ferrer, A. (2014). Multi-synchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms. Journal of Chemometrics. 28(5):462-475. doi:10.1002/cem.2620.

Multisynchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms

J.M. González-Martínez[a,b,*], O. E. de Noord[b], A. Ferrer[a]

[a] *Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, Spain*

[b] *Shell Global Solutions International B.V., Shell Technology Centre Amsterdam, PO Box 38000, 1030 BN Amsterdam, The Netherlands*

Abstract

Batch synchronization has been widely misunderstood as being only needed when variable trajectories have uneven length. Batch data are actually considered not synchronized when the key process events do not occur at the same point of process evolution, irrespective of whether the batch duration is the same for all batches or not. Additionally, a single synchronization procedure is usually applied to all batches without taking into account the nature of asynchronism of each batch, and the presence of abnormalities. This strategy may distort the original trajectories and decrease the signal-to-noise ratio, affecting the subsequent multivariate analyses. The approach proposed in this paper, named Multisynchro, overcomes these pitfalls in scenarios of multiple asynchronisms. The different types of asynchronisms are effectively detected by using the warping information derived from synchronization. Each set of batch trajectories is synchronized by appropriate synchronization procedures, which are automatically selected based on the nature of asynchronisms present in data. The novel approach also includes a procedure that performs abnormality detection and batch synchronization in an iterative manner. Data from realistic simulations of a fermentation process of the *Saccharomyces cerevisiae* cultivation are used to illustrate the performance of the proposed approach in a context of multiple asynchronisms.

Keywords: Batch synchronization, warping information, asynchronism, dynamic time warping, relaxed greedy time warping.

1. Introduction

In current batch processes, on-line measurements of process variables are usually collected at different sampling points for process understanding, optimization and monitoring [1, 2]. Complex physiological behavior, operational changes, intrinsic biological variability in microorganisms or seasonal effects cause batches to have different duration. In addition, time points at which the biochemical reactions and physical activities take place (usually coinciding with process landmarks, such as peaks and valleys) may be shifted across batches. Hence, the collected batch trajectories may not only have different lengths but also the key process events do not overlap at the same time in all batches [3].

In this context of asynchronous batches, the application of multivariate projection methods, such as Principal Component Analysis (PCA) and Partial Least Squares (PLS), is not feasible. In order to ensure all batch trajectories have the same duration and the key process events happen at the same state of evolution, the synchronization of batch trajectories need to be always carried out prior to modeling. A number of proposals for dealing with the most complex synchronization problems can be found in the literature. These approaches can be roughly classified into three categories: i) methods based on compressing/expanding the

*Corresponding author

Email address: jogonmar@gmail.com (J.M. González-Martínez)

raw trajectories using linear interpolation either in the batch time dimension or in an indicator variable dimension; ii) methods based on feature extraction; and iii) methods based on stretching, compressing and translating pieces of trajectories.

Within the first category, some authors dealt with the batch alignment issue using simple ideas, such as truncating the trajectories of all batches to the shortest batch length, or compressing/expanding the trajectories using linear time adjustments by dividing each time point along the trajectory by the time at a certain percentage of the end-point [4, 5, 6]. These ideas, although simple, are often inadequate for aligning batch trajectories [7]. Nomikos and MacGregor [8] proposed the use of an indicator variable: "*One way to handle varying batch times in on-line monitoring is to replace time by another measured variable that progresses monotonically in time and has the same starting and ending value for each batch*". Some applications of this synchronization approach can be found in [5, 9, 10, 11, 12, 13, 14]. When an indicator variable is not available throughout the batch run, but some process variables can be used as an indicator at different process stages, the batch synchronization can be performed stage-by-stage [15]. If a suitable indicator variable is not available for a given batch processes this type of synchronization cannot be carried out and other approaches are required. PLS models between the variable-wise unfolded batch data matrix and the local batch time were also suggested to predict the batch 'maturity' and align accordingly [9, 16].

Procedures based on features extraction were also proposed for batch process synchronization. Kaitsha and Moore proposed a mathematical matched filter to extract key events in batch trajectories in cases where they are not known beforehand [17]. More sophisticated approaches are curve registration [18, 19, 20] and dynamic locus analysis [21, 22, 23], which identify landmarks or special points that characterize process stages and changes (the so-called singular points) in a set of batch trajectories corresponding to process variables, and then, the test trajectories are warped based on the reference landmarks. In [24], raw batch trajectories are decomposed into approximations and details at different scales using wavelets. Contributions from each scale are collected in separate matrices, and data are synchronized at each level using an algorithm based on stretching, expanding and translating pieces of trajectories. Then synchronized separate matrices are reconstructed to form new synchronized trajectories.

Other methodologies based on warping techniques, such as Dynamic Time Warping (DTW) and Correlation Optimization Warping, have been proposed as methods of pattern matching in speech recognition [25] and methods to correct peak shifts in chromatographic profiles [26, 27, 28]. In recent years, these methods have received much attention in process chemometrics to align and synchronize batch trajectories corresponding to process variables [29, 30, 31, 32]. In [29], an end-of-batch version of DTW for batch synchronization was proposed and some guidelines to carry out the real-time synchronization were presented. Nonetheless, this real-time version was proven to be inappropriate for Batch Multivariate Statistical Process Control (BMSPC) due to the high false alarm rate [33]. The Relaxed Greedy Time Warping (RGTW) is a solution to overcome this problem [33]. A Derivative DTW algorithm (DDTW) was proposed to capture the underlying process behavior fingerprinted in the batch trajectories using derivatives. Nonetheless, noisy data can severely affect the computation of numerical derivatives [34]. A robust DTW algorithm was proposed in [35] that combines a moving window least squares procedure with derivative DTW to avoid singularity points and reduce the dependency of the results on the reference trajectory. To deal with the derivatives computation problem in noisy data, the Hybrid Derivative Dynamic Time Warping algorithm was suggested [34], which combines piecewise-linear approximations of the unsynchronized trajectories and DDTW.

Much effort has been devoted to overcome the synchronization problem. Nonetheless, none of the proposals found in the literature takes into consideration abnormalities or the nature of asynchronism that may be present in batch data. The existence of faulty batches in the calibration data set may affect the accuracy of synchronization parameter estimates and, hence, the synchronization quality. The presence of complex asynchronisms producing lack of temporal concurrences also poses a threat to bilinear modeling. Four different types of asynchronism can be found: i) batches with equal duration but key process events not overlapping at the same time point in all batches (class I asynchronism), ii) batches with different duration and process pace (class II asynchronism), iii) batches with different duration due to incompletion of some batches and key process events overlapping (class III asynchronism); and iv) batches with different duration due to delay in the start but batch trajectories showing the same evolution pace after (class IV asynchronism). In this context of multiple asynchronisms, applying the same synchronization procedure

may harmfully affect the original correlations of the process variables over time, jeopardizing subsequent multivariate analysis and the accuracy of monitoring schemes for fault detection.

In this paper, a novel synchronization approach named Multisynchro is proposed to deal with scenarios of multiple asynchronisms in batch processes. The new approach uses the valuable information on the process pace of each batch derived from DTW/RGTW-based synchronization (the so-called warping information) for two purposes: i) detecting the type of asynchronism of each particular batch, and ii) implementing the appropriate synchronization procedure based on the nature of asynchronisms. The new approach also includes a procedure that performs abnormality detection and batch synchronization in an iterative way.

The outline of the paper is as follows. In Section 2, the fundamentals of the DTW and RGTW synchronization algorithms are explained. An optimization of these strategies that deals with the presence of abnormalities to enhance the synchronization quality is proposed. These methods are the core of the novel Multisynchro approach that synchronizes batches considering the nature of asynchronisms, which will be explained in Section 3. Section 4 presents the material of the research work. Section 5 illustrates i) the performance of the novel Multisynchro approach for batch synchronization in scenarios of multiple asynchronisms and ii) the effect of inappropriate synchronization on the batch trajectories. Finally, some conclusions are provided in Section 6.

2. Batch synchronization

In this section, the fundamentals of the DTW and RGTW synchronization algorithms are explained. An optimization of these strategies that deals with the presence of abnormalities to enhance the synchronization quality is proposed. These methods are the core of the novel Multisynchro approach that synchronizes batches considering the nature of asynchronisms.

Let $\mathbf{X}_n$ $(K_n \times J)$ and $\mathbf{X}_{ref}$ $(K_{ref} \times J)$ be the matrices containing the data from a n-th and reference batch in which the J process variables were collected at K_n and K_{ref} sampling points, respectively. Note that all the N calibration batches can be arranged into a three-way array $\underline{\mathbf{X}}$ $(N \times J \times K_n)$. The objective of batch synchronization is to synchronize each $\mathbf{X}_n$ with $\mathbf{X}_{ref}$ guaranteeing the overlap of the key process events.

2.1. DTW/RGTW-based synchronization

The essence of DTW is to match two multivariate batch trajectories $\mathbf{X}_n$ and $\mathbf{X}_{ref}$ by finding a minimum cost function (or warping path) $\mathbf{f}_n^T = \{w(1), w(k), ..., w(K_{w_n})\}$, where $\max(K_{ref}, K_n) \leqslant K_{w_n} \leqslant K_{ref} + K_n$. Here each $w(k)$ is an ordered pair $[i(k), j(k)]$ indicating that the i-th and j-th sampling points that belong to $\mathbf{X}_n$ and $\mathbf{X}_{ref}$, respectively, are synchronized. The synchronization is assessed with respect to a local cost function $d(i, j)$ weighted by the nonnegative diagonal matrix $\mathbf{W}$ $(J \times J)$. This matrix reflects the relative importance of each process variable in the batch synchronization. The resulting values are represented as a local distance $K_n \times K_{ref}$ matrix or grid, which assigns a matching cost for synchronizing each possible pair of sampling points from the test and reference batches. Several constraints are defined to restrict the search of the warping path, namely a band fit to the batch variability that limits the search space of such path, and local constraints (or predecessors), which restricts the warping function as monotonic and continuous. In addition, a cumulative weighted distance matrix $\mathbf{D}(f_n)$ is assessed by estimating the cumulative matching costs of each of the allowed warping paths $\mathbf{f}_n$ (also called warping profile). The optimal warping path $\mathbf{f}_n^*$ is assessed by obtaining the path that minimizes the cumulative distance from a start point, which can be fixed (the initial ordered pair $[1, 1]$) or relaxed (*e.g.* the best matching between the first point of the reference batch and the test batch, s^*) to an end point, which can be likewise fixed (the final ordered pair $[K_{ref}, K_n]$) or relaxed (*e.g.* the best matching between the last point of the test batch and the reference batch, e^*). As a result of this synchronization procedure, a data matrix $\tilde{\mathbf{X}}_n$ $(K_{ref} \times J)$ containing the synchronized batch trajectories is obtained.

The RGTW algorithm [33] builds up a piecewise solution following a greedy optimization approach, so that each time the best local synchronization improvement is incorporated to the global synchronization solution. This synchronization procedure is based on the proposal of Kassidas *et al.* [29] but synchronization

is carried out within a moving window ζ of defined width, which is optimized by cross-validation. Further details can be found [33].

The warping profiles obtained from synchronization are composed of a set of different transitions at each sampling point, *i.e.* vertical, horizontal and diagonal steps. Based on the number of the different transitions the warping path contains, conclusions regarding the performance of the different process stages can be drawn. Let us assume the test and reference batches are located on the x-axis and y-axis, respectively. An excessive number of vertical transitions in the warping profile means that the test batch needed less time than the reference batch to completion. In contrast, an excessive number of horizontal transitions is related to a slow process pace of the test batch in comparison to the reference batch. To correct these differences in the process pace, the DTW/RGTW algorithm expands and compresses the pieces of trajectories in such a way that the key process events are synchronized across batches. Note that the warping profiles contain valuable information about the duration of the process substages, which may be associated with abnormalities occurring in the process and/or the quality of the final product. Hence, the use of the warping information for process monitoring is highly recommended [15, 33, 36].

2.2. Iterative batch synchronization/abnormalities detection procedure

Batch synchronization needs to be implemented taking into account the possible presence of abnormalities in batch data. The existence of faulty batches in the calibration data set may yield inappropriate synchronizations since possible artefacts may be introduced due to abnormalities. For instance, batch trajectories that break the correlation structure usually contain different shapes in comparison to batch trajectories run under Normal Operating Conditions (NOC). It may affect the estimation of the weight matrix $\mathbf{W}$ and the synchronization quality, leading to synchronized batch trajectories with artificial shapes at different time periods. To overcome this problem, an iterative synchronization/abnormalities detection procedure is presented. The aim of this new procedure is to synchronize each batch against a reference batch in such a way that possible abnormalities present in batch data do not affect the synchronization quality. The main steps of the algorithm are (see Figure 1):

i. Synchronize all the batches contained in the starting three-way matrix $\underline{\mathbf{X}}$ using the DTW algorithm. For this purpose, select a reference batch $\mathbf{X}_{ref}$ and a criteria to weight the process variables. The algorithm returns the synchronized three-way batch data array $\underline{\tilde{\mathbf{X}}}$ ($N \times J \times K_{ref}$) and the weigth matrix $\mathbf{W}$.

ii. Preprocess batch data by trajectory centering and scaling using the estimated matrices of averages $\mathbf{\Xi}$ ($K_{ref} \times J$) and standard deviations $\mathbf{\Omega}$ ($K_{ref} \times J$)[1].

iii. Fit a PCA model on the batch-wise unfolded and preprocessed data matrix satisfying the following equation: $\tilde{\mathbf{X}}' = \mathbf{T}_A \cdot \mathbf{P}_A^T + \mathbf{E}$, where A is the number of PCs extracted[2].

iv. Design a control chart based on the Squared Prediction Error (SPE) statistic. Its control limit $SPE_{lim,\alpha}$ is estimated from the synchronized calibration batch data at (1-α) confidence limit.

v. Off-line post-batch monitor all the synchronized calibration batches for fault detection.

v.1 Compute the SPE statistic for each batch and sort out the corresponding values in ascending order.

v.2 Calculate the acceptable number of batches R that can exceed the control limits at $(1-\alpha)$ confidence level by chance as α times the number of calibration batches.

[1] This preprocessing approach is selected due to suitability for batch process modelling and monitoring [37, 38].

[2] The interest of building a PCA in this work is to design a monitoring scheme for fault detection. In process monitoring, the interest is in the distributions in latent variables and residuals, which are those used to estimate the control limits for incoming data. This should be taken into consideration to select A [**?**].

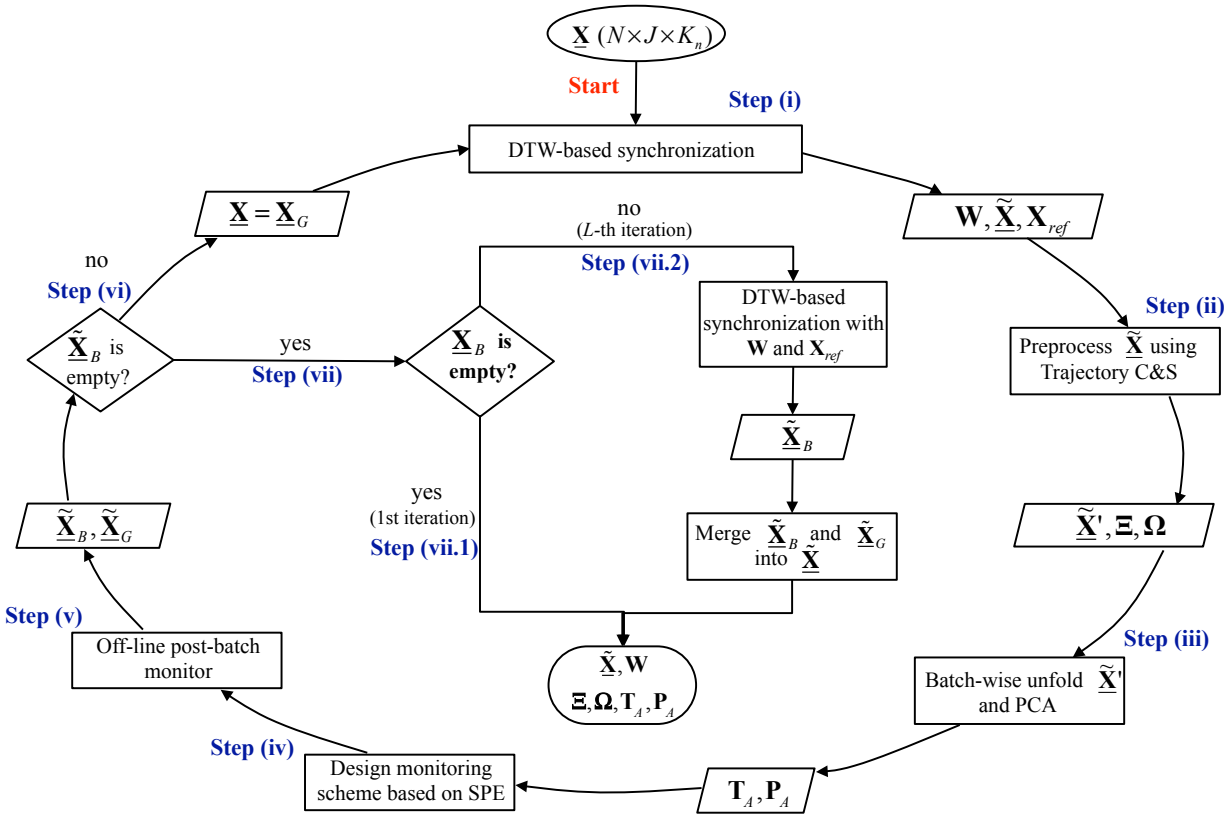


Figure 1: Flow diagram of the iterative batch synchronization/abnormalities detection procedure. Note that $\underline{\tilde{\mathbf{X}}}_B$ is the three-way array containing the synchronized faulty batches isolated at the l-th iteration whereas $\underline{\mathbf{X}}_B$ is the three-way containing all the raw faulty batches isolated in the L iterations of the iterative procedure.

v.3 If the number of batches exceeding $SPE_{lim,\alpha}$ N_f is greater than R, the first $B_l = N_f - R$ synchronized batches with the highest SPE values are treated as faulty batches. In any case those batches whose SPE values are beyond λ times $SPE_{lim,\alpha}$ are also considered as faulty. To isolate these faulty batches for subsequent synchronization different from that performed on NOC batches, arrange them into the three-way array $\underline{\tilde{\mathbf{X}}}_B$ $(B_l \times J \times K_{ref})$, recover their raw trajectories and add them to the three-way array $\underline{\mathbf{X}}_B$ $(B_L \times J \times K_b)$, which contains the rest of raw faulty batches isolated in previous iterations.

v.4 The remaining batches are considered as NOC and their trajectories are arranged into the three-way array $\underline{\tilde{\mathbf{X}}}_G$ $(G \times J \times K_{ref})$.

vi. If one or more batches were detected as abnormal in the off-line post-batch monitoring at the l-th iteration, compute the repeat loop (i)-(v) with the new calibration batch data array $\underline{\mathbf{X}} = \underline{\mathbf{X}}_G$, where $\underline{\mathbf{X}}_G$ is a $(G \times J \times K_g)$ three-way array containing the raw batch trajectories of G NOC batches. .

vii. If no batch was detected as abnormal in the off-line post-batch monitoring at the l-th iteration, synchronize the faulty batches and merge the data sets.

vii.1 If no batch was detected as abnormal in the first iteration, the iterative procedure ends up.

vii.2 If some batches were detected as abnormal in the off-line post-batch monitoring after L iterations, synchronize each faulty batch $\mathbf{X}_b$ from the three-way faulty batch array $\underline{\mathbf{X}}_B$. For this purpose, the

DTW algorithm is applied using the reference batch $\mathbf{X}_{ref}$ and the weighting matrix $\mathbf{W}$ that were assessed in the NOC batch synchronization in Step (i) at the last iteration. Once the synchronized three-way faulty batch array $\underline{\tilde{\mathbf{X}}}_B$ is available, merge it with the three-way array of NOC batches $\underline{\tilde{\mathbf{X}}}_G$ into the three-way array $\underline{\tilde{\mathbf{X}}}$.

As output, the iterative batch synchronization/abnormalities detection procedure returns: the three-way synchronized batch data array $\underline{\tilde{\mathbf{X}}}$, the reference batch $\mathbf{X}_{ref}$; the matrices of average trajectories $\boldsymbol{\Xi}$ and standard deviation trajectories $\boldsymbol{\Omega}$; the weighting matrix $\mathbf{W}$ assessed in the synchronization procedure; and, the score $\mathbf{T}_A$ and loading $\mathbf{P}_A$ matrices obtained from the PCA model on the batch-wise unfolded preprocessed matrix at the last iteration.

3. Multisynchro approach for batch synchronization

The multisynchro approach is devoted to synchronize the key process events ensuring the same evolution across batches, no matter the type of asynchronism present in batch data. The algorithm takes as inputs the three-way array arranging the calibration batches, the technique to weight the process variables and the strategy to select the reference batch. The procedure returns the synchronized batch data array and the warping time profiles that indicate how to warp the batch trajectories to make them synchronized.

The multisynchro algorithm is composed of a high-level and low-level routine (see Figure 2). The high-level routine is aimed at recognizing the different types of asynchronous trajectories for the subsequent batch classification as function of the nature of asynchronism (see Figure 2(a)). The low-level routine is in charge of synchronizing the variable trajectories of each one of the batches with a specific procedure based on the type of asynchronism (see Figure 2(b)). In the following, the algorithm is described.

3.1. Asynchronism detection

The high-level routine is divided into two steps (see Figure 2(a)). The first step is devoted to recognize the different types of asynchronous trajectories, which is carried out by using the warping time profiles derived from a preliminary synchronization as follows:

i. Select a reference batch $\mathbf{X}_{ref}$ from the three-way batch data array $\underline{\mathbf{X}}$.

ii. Synchronize all batches using the DTW algorithm giving the same weight to the process variables that contain valuable information for synchronization (*e.g.* maximum and minimum values that define the features of the multivariate trajectories and process stages). Those variables that are either showing constant values in most of production time or discarded beforehand by prior knowledge are constrained in the synchronization with a null weight. The reason why certain process variables are given the same importance is to mitigate the distortion of the warping profiles in the presence of different types of asynchronisms. The algorithm returns a three-way synchronized batch data array $\underline{\tilde{\mathbf{X}}}$ and a three-way array $\underline{\mathbf{F}}$ ($N \times 2 \times K_{w_n}$) containing the the warping paths for the N batches.

iii. For each warping time profile $\mathbf{f_n}$ from the three-way array $\underline{\mathbf{F}}$:

iii.1 Count the number of consecutive horizontal transitions denoting the number of compressions carried out by the synchronization algorithm at the first time period of the n-th batch as follows:

$$h_n = \sum_{k=1}^{Kw_n} (j(k) = 1)$$

iii.2 Count the number of consecutive vertical transitions denoting the number of expansions carried out by the synchronization algorithm at the last time period of the n-th batch as follows:

$$v_n = \sum_{k=1}^{Kw_n} (i(k) = K_n)$$

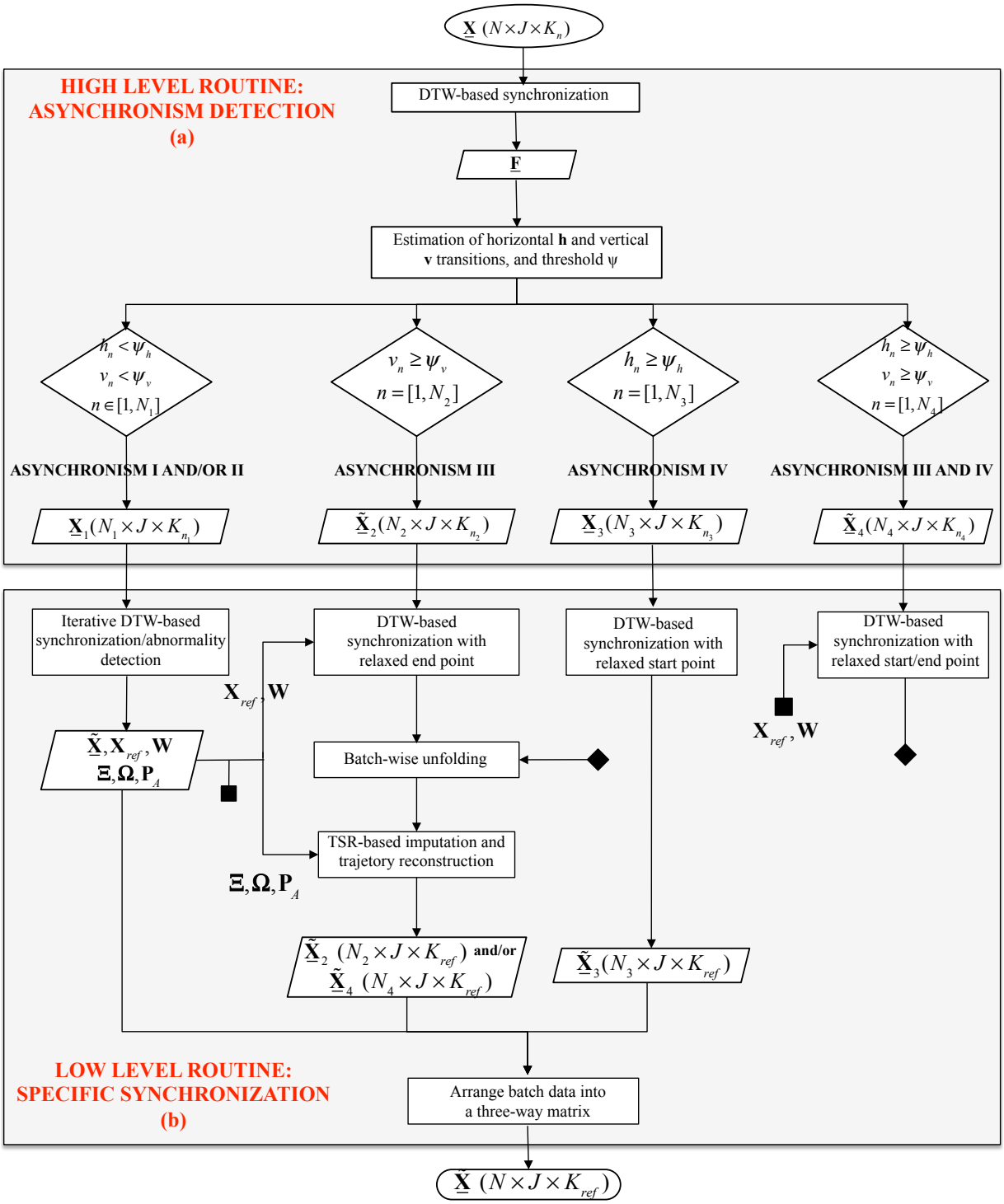


Figure 2: Flow diagram of the Multisynchro approach composed of the high-level (a) and low-level (b) routines for batch synchronization in scenarios of multiple asynchronisms.

These features of the warping time profiles are used to detect the different types of asynchronisms presented in data. In the case of class III asynchronism, incomplete batches are associated with warping profiles showing an excessive number of vertical transitions at the last time period of the runs. These transitions are related to expansions that the DTW algorithm should carry out for synchronization. Batches with a shift at the start of the run are associated with warping profiles that contain a high number of horizontal transitions at the same time period (asynchronism IV). These transitions are related to compressions that the DTW algorithm should carry out for synchronization. Finally, in class I and class II asynchronism, the resulting warping profiles show a reasonable combination of horizontal and vertical transitions throughout the batch run.

The second step of the high-level routine (see Figure 2(b)) is aimed at classifying each batch by the type of asynchronism and arranging them into different data sets. For this purpose, the features of the warping time profiles are used to distinguish between asynchronisms that can be dealt with conventional synchronization techniques and those that need more sophisticated procedures. The formers are class I and class II asynchronisms, which require a combination of vertical and horizontal transitions over time to align the key process events. In contrast, trajectories affected by the latters, *i.e.* class III and class IV asynchronisms, produces a larger number of vertical transitions at the last process stage and larger number of horizontal transitions at the start of the batch than normal, respectively. In order to identify the type of asynchronism of each batch, thresholds ψ_v and ψ_h are calculated as a fraction κ of the interquartile range of both the vertical transitions at the last time period $\mathbf{v}$ and the horizontal transitions at the first time period $\mathbf{h}$ estimated for all the synchronized batches, respectively[3].

i Repeat for all batches:

ii.1 If the number of compressions h_n and expansions v_n are less than their respective thresholds, arrange the n-th batch into the three-way array $\underline{\mathbf{X}}_1$, which contain batches affected by class I and II asynchronisms.

ii.2 If only the number of expansions at the end of the batch v_n is greater than or equal to the threshold ψ_v, arrange the n-th raw batch into the three-way array $\underline{\mathbf{X}}_2$, which contain batches affected by class III asynchronism.

ii.3 If only the number of compressions at the start of the batch h_n is greater than or equal to threshold ψ_h arrange the n-th raw batch into the three-way array $\underline{\mathbf{X}}_3$ by class IV asynchronism.

ii.4 If the number of compressions h_n and expansions v_n are greater than or equal to their respective thresholds, arrange the n-th raw batch into the data matrix $\underline{\mathbf{X}}_4$ by class III and IV asynchronisms.

3.2. Specific batch synchronization

The multisynchro approach continues the execution synchronizing the different data sets with different types of asynchronism:

i Synchronize the three-way batch data array $\underline{\mathbf{X}}_1$ using the iterative synchronization based on the DTW algorithm explained in Section 2.2. This procedure consists of synchronizing batch trajectories in such a way that possible abnormalities present in batch data do not affect the synchronization quality. The procedure returns the matrices of average trajectories $\boldsymbol{\Xi}$ and standard deviation trajectories $\boldsymbol{\Omega}$, the weighting matrix $\mathbf{W}$, the loading vector $\mathbf{P}_A$ obtained from the PCA-based modelling, and the three-way array $\underline{\tilde{\mathbf{X}}}$ that arranges the synchronized NOC and faulty batches.

ii Synchronize the three-way batch data array $\underline{\mathbf{X}}_2$ using the DTW algorithm with the relaxed end point constraint using those parameters estimated in the iterative synchronization. This version of the DTW algorithm synchronizes batches against a segment of the reference batch limited by the first point and

[3]This measure of statistical dispersion is used due to its robustness to outliers and extreme values. Even though κ is a heuristic value dependent on the distribution of the transitions, it is recommended that κ does not exceed 0.5.

the best matching end point e^* instead of the reference as a whole. The algorithm returns the batch trajectories synchronized till the best end point of each batch $\underline{\tilde{\mathbf{X}}}_2$. The missing part of each of batches are imputed using the Trimmed Score Regression method [40]. The procedure returns the three-way array $\underline{\tilde{\mathbf{X}}}_2$ containing the synchronized batch trajectories.

iii Synchronize the three-way batch data array $\underline{\mathbf{X}}_3$ using the DTW algorithm with the relaxed start point constraint and the parameters calculated in the iterative synchronization. This version of the DTW algorithm synchronizes segments of batches against a reference batch. The segments are limited by the best matching start point s^* of each batch with the first point of the reference, and their last point. The procedure returns the three-way array $\underline{\tilde{\mathbf{X}}}_3$ containing the synchronized batch trajectories.

iv Synchronize the three-way batch data array $\underline{\mathbf{X}}_4$ using the DTW algorithm with the relaxed start and end point constraint using those parameters estimated in the iterative synchronization. The procedure returns a three-way array $\underline{\tilde{\mathbf{X}}}_4$ containing the synchronized batch trajectories.

At this point, it is worth emphasizing that the iterative synchronization/abnormalities detection procedure is only performed when batch data are affected by class I and class II asynchronisms. In this case, occurrences at unrelated times are caused by external and/or internal process factors that can be straightforwardly coped with synchronization methods such as DTW or RGTW. The detection of abnormalities in these type of batches is crucial to obtain the correct parameters both for the synchronization of the remaining batches (reference batch $\mathbf{X}_{ref}$ and the weighting matrix $\mathbf{W}$) and for the missing trajectory imputation (matrices of average trajectories $\mathbf{\Xi}$ and standard deviation trajectories $\mathbf{\Omega}$, and the loading matrix $\mathbf{P}_A$). The detection of disturbances in the rest of batches is carried out jointly with those already signaled as NOC in the subsequent multivariate analysis.

After synchronizing batch data using Multisynchro, all the resulting submatrices need to be merged into a three-way array $\underline{\tilde{\mathbf{X}}}$ ($N \times J \times K_{ref}$) for subsequent bilinear process modeling. Even though some batches may have been detected as abnormal in the iterative synchronization procedure, they are not discarded for modeling. The reason is that these batches are a valuable source of information. In addition, the warping profiles obtained in each one of the specific synchronizations are added as a new variable into the synchronized three-way array $\underline{\tilde{\mathbf{X}}}$ for monitoring purpose.

The real-time application of the Multisynchro approach is straightforwardly done by using the RGTW algorithm instead of the DTW algorithm. For off-line applications, DTW is preferred since it provides us with the optimum global solution. However, if the main goal is to design a monitoring scheme for real-time application, the RGTW algorithm is required. For further details on its implementation, readers are referred to [33].

4. Material and methods

Two data sets are generated based on the biological model of the aerobic growth of *S. cerevisiae* on glucose limited medium [41]. For this purpose, the simulation scheme designed using Simulink for Matlab release 2010a (The MathWorks, Inc), available in the MP toolbox [42]) is used. In particular, data for 40 batches and 10 batches run under normal operating conditions -processed with the nominal values of the internal kinetic constants [41]- are simulated for data set #1 and #2, respectively. Measurements belonging to ten process variables are collected every sampling time over all batches: concentrations (glucose, pyruvate, acetaldehyde, acetate, ethanol and biomass), active cell material, acetaldehyde dehydrogenase (proportional to the measured activity), specific oxygen uptake rate and specific carbon dioxide evolution rate. The original time of processing from simulation is also added to the batch data matrix. In order to make the simulation realistic, gaussian noise of low magnitude in the initial conditions (10%) and measurements (5%) are introduced. In addition, the intrinsic biological variability of a population of the microorganism is taken into account in the simulation. As a result, batches with different duration and evolution pace are obtained.

At the end of the simulation, the three-way arrays $\underline{\mathbf{X}}_1$ ($N_1 \times J \times K_{n_1}$) and $\underline{\mathbf{X}}_2$ ($N_2 \times J \times K_{n_2}$) are returned, where K_{n_1} and K_{n_2} are the different sampling points at which the measurement of $J = 11$ process variables

were measured in $N_1 = 40$ and $N_2 = 10$ batches, respectively. The total length of batches corresponding to the first data set varies from 172 to 330 data points (*i.e.* $K_{n_1} \in [172, 330]$), and in the second data set from 173 to 294 data points (*i.e.* $K_{n_2} \in [173, 294]$).

The first data set is synchronized by using the DTW algorithm. For that, batch #12, the closest one to the median length from the first data set with $K_{ref} = 209$ sampling points, is selected as reference. The process variables are equally weighted to get a non-optimized synchronization, where the key process events are not completely aligned. The rest of conditions and constraints are set according to [29]. After synchronization, a three-way array $\underline{\tilde{\mathbf{X}}}_1$ ($N_1 \times J \times K_{ref}$) is obtained. The second data set is synchronized by using the DTW algorithm using batch #10 with $K_{ref} = 209$ sampling points as reference with the aforementioned parameters and constraints. The resulting three-way array $\underline{\tilde{\mathbf{X}}}_2$ ($N_2 \times J \times K_{ref}$) is derived.

Table 1: Batch data composing the four data sets with different asynchronisms.

Asynchronism case	Batch data	Explanatory text
#1	$\underline{\tilde{\mathbf{X}}}^{(1)} \subseteq \underline{\tilde{\mathbf{X}}}_1$, $N_1^{(1)} = 10$	Random cut sampling point of batches: #184, #193, #183, #173, #168, #141, #156, #185, #192 and #184.
	$\underline{\tilde{\mathbf{X}}}^{(2)} \subseteq \underline{\tilde{\mathbf{X}}}_1$, $N_1^{(2)} = 30$	No data manipulation
#2	$\underline{\tilde{\mathbf{X}}}^{(1)} \subseteq \underline{\tilde{\mathbf{X}}}_1$, $N_1^{(1)} = 10$	Random length of shift for batches: #2, #5, #7, #11, #13, #14, #22, #23, #28 and #37.
	$\underline{\tilde{\mathbf{X}}}^{(2)} \subseteq \underline{\tilde{\mathbf{X}}}^1$, $N_1^{(2)} = 30$	No data manipulation
#3	$\underline{\mathbf{X}}^{(1)} \subseteq \underline{\mathbf{X}}_1$, $N_1^{(1)} = 10$	Cut sampling point of each batch based on those set in case #1: #174, #171, #161, #158, #173, #128, #156, #169, #202 and #151
	$\underline{\mathbf{X}}^{(2)} \subseteq \underline{\mathbf{X}}_1$, $N_1^{(2)} = 30$	No data manipulation
#4	$\underline{\tilde{\mathbf{X}}}^{(1)} \subseteq \underline{\tilde{\mathbf{X}}}_1$, $N_1^{(1)} = 30$	No data manipulation
	$\underline{\tilde{\mathbf{X}}}^{(2)} \subseteq \underline{\tilde{\mathbf{X}}}_2$, $N_2^{(1)} = 10$	No data manipulation

Four different batch data sets with different patterns of asynchronism are created from the original and synchronized batch data (see Table 1). The resulting asynchronous batches are depicted in the acetate concentration variable in Figure 3. The first asynchronism case consists of a set of batch trajectories with different length due to incompletion of the batch run and key process events overlapping across batches (see Figure 3(a)). For the generation of this type of asynchronism, $N_1^{(1)} = 10$ batches randomly selected from $\underline{\tilde{\mathbf{X}}}_1$ are manipulated to have different length. Ten different end points are randomly generated and the batch trajectories corresponding to the $N_1^{(1)}$ batches are subsequently cut to these points (see case #1 in Table 1). The remaining $N_1^{(2)} = 30$ batches are arranged jointly with the $N_1^{(1)}$ incomplete batches into the three-way array $\underline{\mathbf{X}}_{c\#1}$ ($N_1 \times J \times K_{n_1}$). In the second case of asynchronism, batches have different length due to delay in the measurement collection but their trajectories show the same evolution pace over all batches (see Figure 3(b)). To generate this type of asynchronism, the $N_1^{(1)}$ batch trajectories are manipulated in the following way. Firstly, the duration of the delay for each batch is randomly generated in the range $[1, 50]$ sampling points. Secondly, data are generated for each one of the J process variables by following a normal distribution with mean and variance calculated in the first 5 sampling points from the start of the batches (see case #2 in Table 1). Finally, these measurements are added to each process variable and the resulting batch trajectories are arranged with the $N_1^{(2)}$ batches into the three-way array $\underline{\mathbf{X}}_{c\#2}$ ($N_2 \times J \times K_{n_2}$). In case #3, the batch trajectories show not only different duration due to incompletion of batches but also the key process events do not overlap at the same batch time across batches (see Figure 3(c)). For the generation of these asynchronism patterns, the $N_1^{(1)}$ raw batch trajectories are again manipulated. The cut points generated in case#1 in the domain of the synchronized time are chosen and their corresponding matching point in the actual batch time is reconstructed by using the warping information (see case #3 in Table 1). Afterwards, the $N_1^{(1)}$ raw batch trajectories are cut to these points. Finally, these batch data are arranged with the remaining $N_1^{(2)}$ raw batches into the three-way array $\underline{\mathbf{X}}_{c\#3}$ ($N_3 \times J \times K_{n_3}$). Concerning the fourth case of asynchronism, the batch trajectories have the same length but the evolution pace is different among batches (see Figure 3(d)). For this case, the synchronized batch trajectories from $\underline{\tilde{\mathbf{X}}}_1$ and $\underline{\tilde{\mathbf{X}}}_2$ are arranged

into the three-way array $\underline{\mathbf{X}}_{c\#4}$ $(N_4 \times J \times K_{ref})$. The final data sets containing the four different types of asynchronism are available in the Supporting Information.

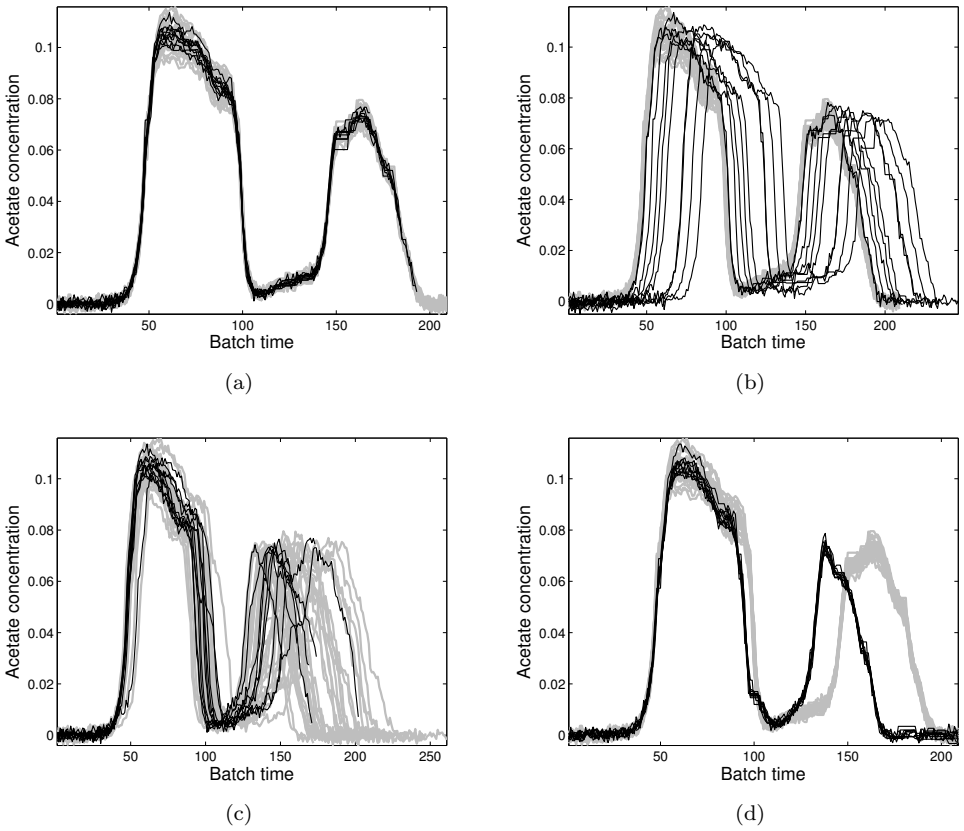


Figure 3: Trajectories of the process variable acetate concentration corresponding to 40 NOC batches in four different scenarios of asynchronism: a) case #1: different batch duration produced by incomplete batch runs and key process events overlapping at the same sampling point across batches; b) case #2: different batch duration produced by a delay in measurements collection (shift) and their trajectory profiles show the same evolution pace over all batches; c) case #3: different batch duration produced by natural variability and incomplete batch runs, and key process events not overlapping at the same sampling point across batches; and d) case #4: equal batch duration and key process events not overlapping at the same sampling point in the last process stage across batches. The batch trajectories with different asynchronism patterns for each scenario are distinguished by black and grey lines.

5. Results

The objective of this section is to illustrate i) the performance of the novel Multisynchro approach for batch synchronization in scenarios of multiple asynchronisms and ii) the effect of inappropriate synchronization on the batch trajectories.

Batches with four different types of asynchronism (see Table 1) are synchronized by using the Multisynchro approach. The high-level routine is executed for asynchronism detection. As a result of this step, a set of 40 warping profiles for each scenario of asynchronism is derived (see Figure 4). Looking at these profiles, in which every action taken by the synchronization algorithm is fingerprinted, insight into the nature of asynchronism present in batch data can be obtained.

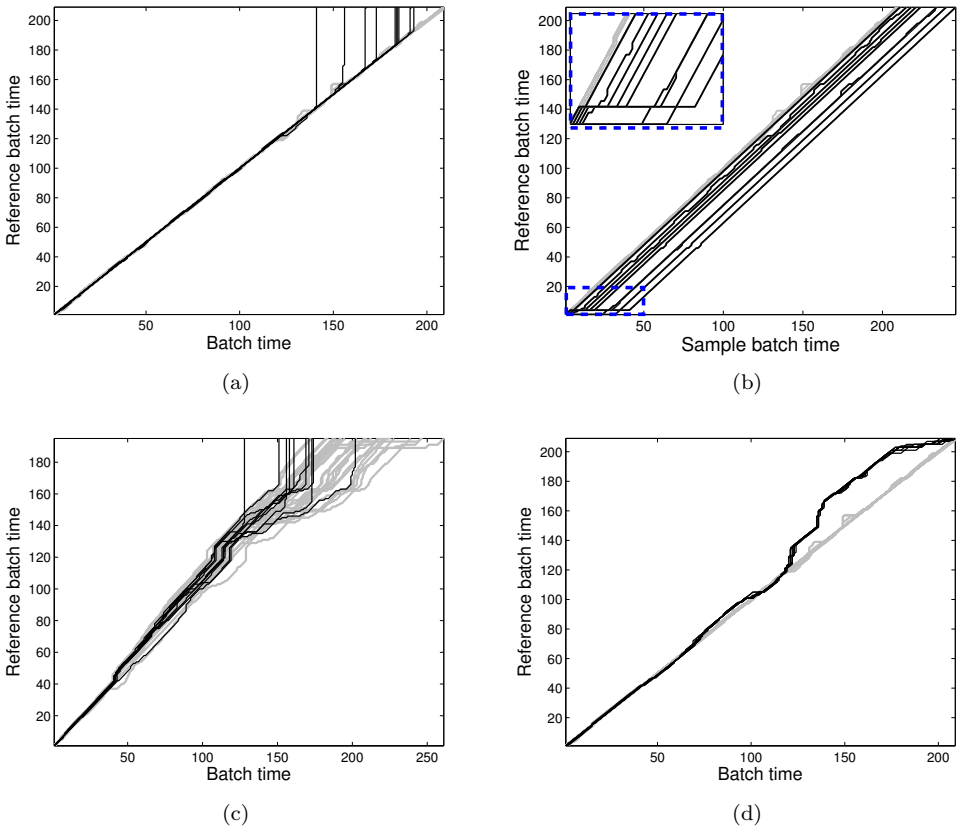
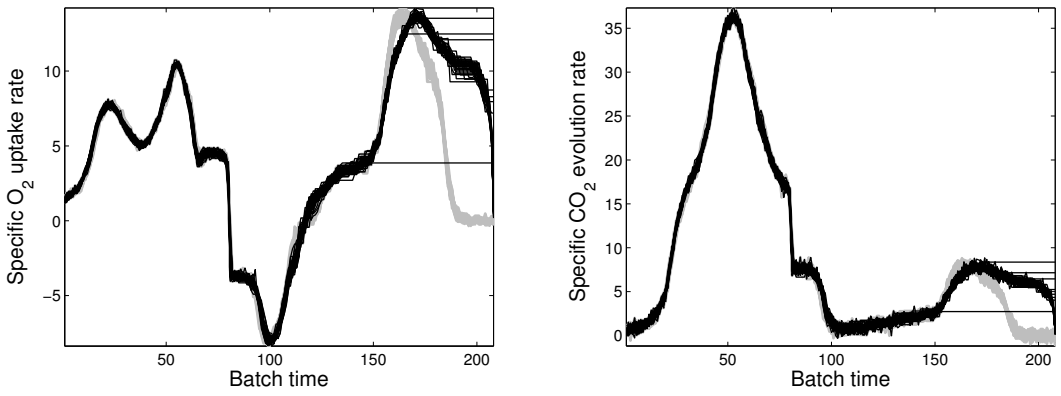


Figure 4: Warping information derived from the DTW-based synchronization of the raw batch trajectories for each one of the asynchronism scenarios: (a) case #1, (b) case #2, (c) case #3, and (d) case #4. The batch synchronization was performed weighting each process variable equally. The warping profiles belonging to batches with different asynchronism patterns for each scenario are distinguished by black and grey lines.

In cases #1, #2 and #4, the warping profiles belonging to the 30 out of 40 batches (see grey lines in Figure 4(a), Figure 4(b) and Figure 4(d), respectively) almost follow the main diagonal. Note that these batches have equal duration and apparently their key process events overlap at the same sampling points across batches (see Figure 3(a), Figure 3(b) and Figure 3(d), respectively). Nonetheless, the slight deviations observed from the diagonal profile denote that even though most of the batches have equal duration, the main events are not perfectly synchronized. This supports the claim that the batch synchronization is required even when the variable trajectories show the same evolution pace. Concerning the warping profiles corresponding to the rest of batches (see black lines in Figure 4), a different asynchronism pattern is recognized in each case.

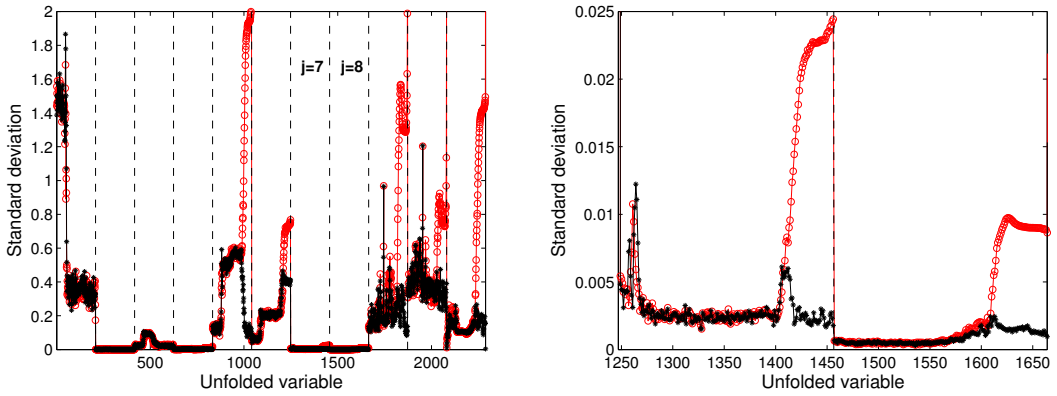
In Figure 4(a) and Figure 4(c), 10 out of the 40 warping profiles (black lines) show an excessive number of vertical transitions in comparison to the rest (grey lines). The difference between both cases is that the batch trajectories in the latter are not synchronized from the beginning (see the warping profiles notably deviating from the main diagonal in Figure 4(c)). This pattern is directly related to the presence of batches that were not totally completed. In common practice, these incomplete batches are usually taken into consideration for batch synchronization, causing severe and undesirable changes in the profiles of the process variables. To illustrate the effect of applying a general synchronization approach in these batches, the three-way batch data array $\underline{\mathbf{X}}_{c\#3}$ is synchronized by using the DTW algorithm in a regular way. Also, the low-level routine of the Multisynchro approach is applied on the raw batch trajectories for comparison purpose. For the sake of simplicity, batch data corresponding to case #1 are not used since it is a particular case of case #3. The outcomes of the application of both synchronization procedures are illustrated in Figure 5 by showing two out of the 11 registered process variables.



(a) (b)

Figure 5: Batch trajectories belonging to the process variables specific oxigen uptake rate (a) and specific carbon dioxide evolution rate (b) after synchronization. Black lines represent trajectories synchronized by using the DTW algorithm without taking into account the type of asynchronism and grey lines those batch trajectories synchronized by the Multisynchro approach.

When the batches are not completed, the DTW algorithm correctly synchronizes the batch trajectories from the initial point $(1, 1)$ to the optimum last matching point (k_{ref}, K_i) (the last closest point of the black lines to the diagonal profile in Figure 4(c)). From the $(k_{ref} + 1)$-th to the K_{ref}-th sampling point of the reference batch, the last point of the i-th batch is matched, leading to the vertical transitions observed (see Figure 4(c)). This would lead to expansions of the batch trajectories, *i.e.* the addition of replicated values of the K_i-th sampling point in the i-th batch. Consequently, flat profiles in the process variables (*i.e.* replicated values of the last actual value) are introduced (see Figure 5). This is an artifact since the batches were not actually finished and the remaining trajectory till completion is computed in an inappropriate way. In addition, these inaccuracies are inherited in the synchronization of that stage. Note that when a batch is



(a) Standard deviations (SD)

(b) SD: zoom corresponding to Variable #7 (active cell material) and Variable #8 (acetaldehyde dehydrogenase)

Figure 6: Comparison of the standard deviation vectors obtained from batch data synchronized by using the DTW algorithm without taking into consideration the asynchronous patterns from case#3 (red empty circles line) and by using the Multisynchro approach (black stars line).

finished earlier than the historical batches, the addition of artifacts in data may be higher. This would cause a possible change of the trajectory profile. In the batch data simulated, the largest cut was approximately 60 sampling points. As can be observed in Figure 5, it produces changes in the shape of the profiles in the second half of the batch runs and, consequently, in the normal process pace.

The higher the addition of artifacts, the higher the uncertainty inherited. This variability may severely affect the interpretation of the subsequent multivariate statistical model and, therefore, the performance of the monitoring scheme. An indicator of this is the variability of the resulting s ynchronized batch trajectories around their mean trajectory. This can be measured by the standard deviation vector after the average mean is subtracted and the resulting batch data is scaled to unit variance at every sampling point (the so-called Trajectory centring and scaling). The lower the difference among standard deviation vectors, the higher the synchronization quality.

In order to study the improvement reached by the application of the Multisynchro approach versus traditional synchronization policies (DTW-based synchronization applied to all batches) in case #3, the standard deviation vectors of the corresponding synchronized batch trajectories are computed and shown in Figure 6. Figure 6(a) reveals that when the incomplete batches are treated separately from the rest in the batch synchronization, the resulting standard deviation values are lower (black stars lines) than for the classical approach (red empty circles lines). These differences are more prominent in Variables #5, #6, #9 and #10 (see trajectories of Variables #9 and #10 in Figure 5), in particular at the last stage of the process (last 60 sampling points from the batch runs), where somes batches are incomplete or cut. Even though the standard deviation values seem to be similar for the rest of variables, these differences are also observed at the same batch time period but at less extent (see Figure 6(b) for Variables #7 and #8). This is a clear indicator that synchronizing the batch trajectories without taking into consideration this type of asynchronism seriously affects the resulting trajectories, decreasing the signal-to-noise ratio.

Concerning the case #2, 10 out of 40 batches (those at which a shift type asynchronism was introduced) show a diagonal warping profile that is parallel to the main diagonal. This pattern is characteristic in the cases where similar values of the J process variables are registered at the start of the batch. It leads to an excessive number of horizontal transitions in the synchronization, as can be seen in the blue dashed rectangle in Figure 4(b). The higher the duration of the shift from the start, the higher the number of horizontal transitions in the warping profile. In this case, the DTW algorithm shrinks the corresponding batches at that time interval by averaging the measurements of the J process variables. Note that this compression procedure needs to be done carefully since the presence of severe artifacts in these starting periods may affect

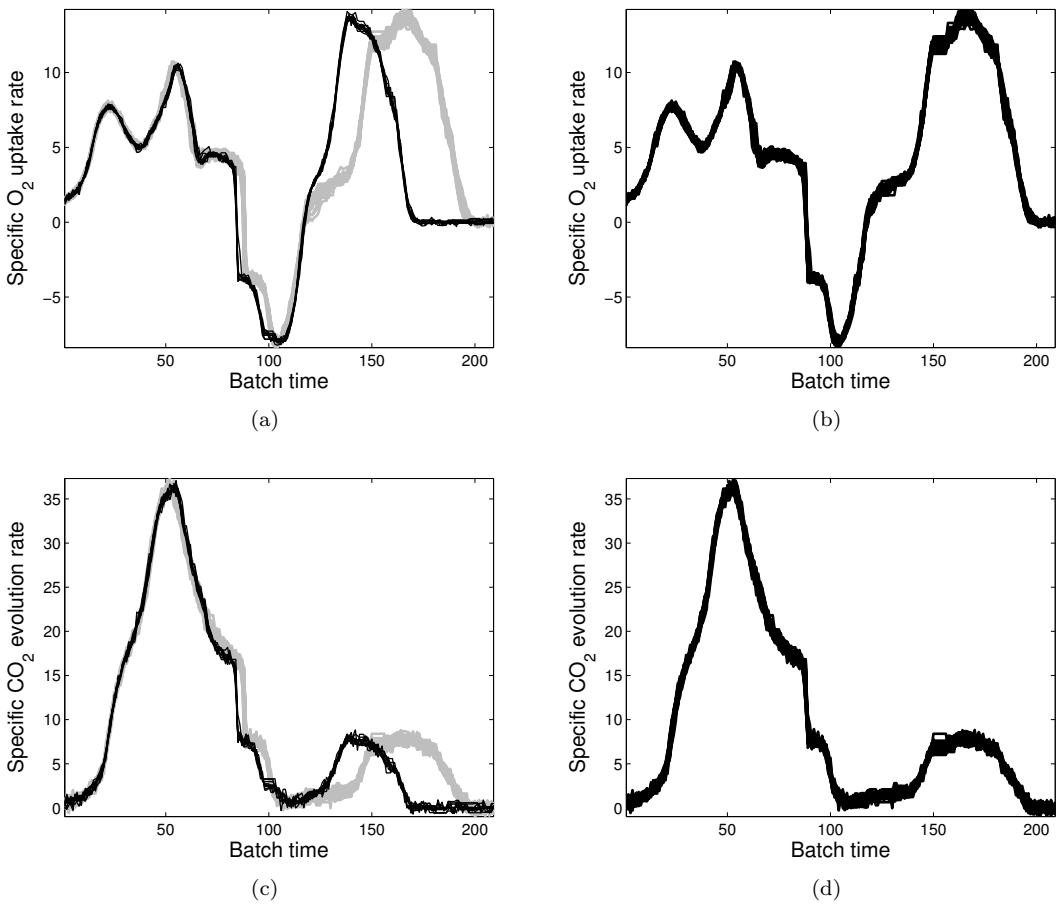


Figure 7: Batch trajectories belonging to the process variables specific oxigen uptake rate (a, b) and specific carbon dioxide evolution rate (c, d) without applying any synchronization (a and c, respectively) and applying the Multi-sinchro approach (b and d, respectively). The black lines in (a) and (c) represent the raw trajectories belonging to 10 out of 40 batches with the case #4 asynchronism embedded.

the synchronization quality. To avoid this problem, the Multisynchro approach performs the synchronization from the optimum match point at the start of the batch till the end. Hence, the resulting warping profiles do not show these horizontal transitions since no compression is carried out (not shown).

As was explained, there is a wrong conception about the importance of asynchronism. When some key process events are not totally aligned, regardless of the batch time duration, the batch trajectories need to be synchronized. This is clearly shown in case #4 represented in Figure 3(d). As can be seen, the acetate concentration trajectory shows that the second half of the batch run (from the 120th sampling point onwards) has a different pace for the two groups of batches, denoted as black and grey lines. From the start to the 90th sampling point, the main process phenomena apparently occur at the same time point across batches. In order to ensure that the key process events are actually synchronized, batch synchronization should be applied to batch data. In Figure 4(d), the resulting warping profiles from the synchronization at the high-level step are depicted. As can be seen, there are two groups of profiles clearly distinguished, those corresponding to the 10 batches where the different process pace was forced (black profiles) and the rest of batches (grey lines). Looking at the profiles corresponding to the 10 batches with asynchronism, one can observe two main time periods of large deviation from the main diagonal: from the 75th to the 100th sampling point and from the 120th to the end of the batch, the deviation being smaller in the

former than in the latter. In both time periods, these warping profiles have a higher number of vertical transitions than horizontal transitions. This is the reason why these warping profiles are beyond the main diagonal. It indicates that the batches with asynchronism had slower process pace than the rest. Hence, the synchronization algorithm needs to expand the corresponding batch trajectories at the aforementioned batch time periods.

In case #4, batch synchronization is seldomly applied because batches have already the same duration. There is commercial software for batch process monitoring, *e.g.* SIMCA Release 13.0.3 [43], that only demand the synchronization of the batch trajectories when they have different length[4] In order to emphasize the importance of this step, the raw batches trajectories $\underline{\mathbf{X}}_{c\#4}$ are compared with those obtained from the batch synchronization by using the Multisynchro approach. For comparative purposes only two process variables are illustrated (see Figure 7). As can be observed, the raw trajectories of the process variable specific oxygen uptake rate (see Figure 7(a)) and specific carbon dioxide evolution rate (see Figure 7(c)) belonging to 10 out of the 40 raw batches (black lines) differ with those corresponding to the rest of batches (grey lines). Mainly, these differences are shown at the last stage of the process, from the 120th sampling point onwards. This reflects that the fermentation at the second half of the process took less time than in the rest of the batch trajectories. Hence, the synchronization of this stage is needed for subsequent analysis. Once the Multisynchro approach is applied to batch data, the resulting 40 profiles not only have equal length but also the segments of profile corresponding to the last process stage overlap across batches (see the synchronized trajectories of the process variables specic oxigen uptake rate and specic carbon dioxide evolution rate in Figure 7(b) and Figure 7(d), respectively).

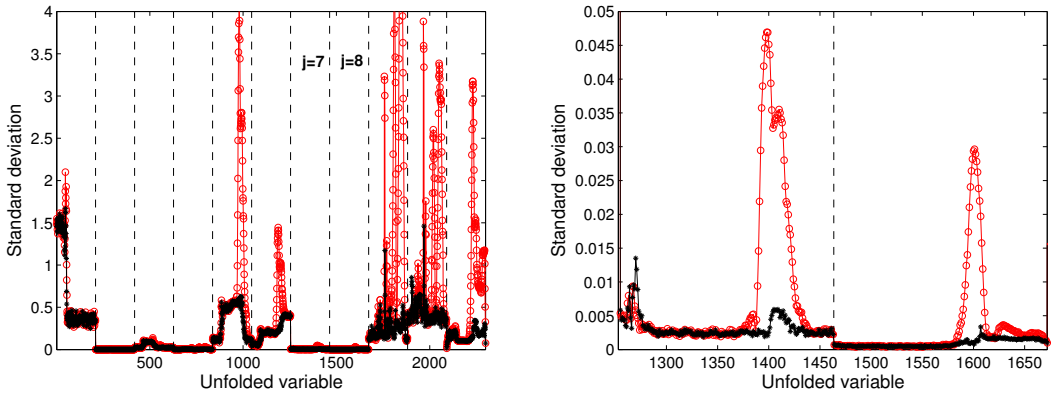
Again, the standard deviation vector is estimated from the raw and synchronized batch data to study the improvement achieved when the Multi-Synchro approach is applied in comparison to take no action for synchronization (see Figure 8). If the batch trajectories are not synchronized, the standard deviation vector derived contains more variability (see red empty circles line in Figure 8(a)) in comparison to that derived from the data synchronized with the Multisynchro approach (see black stars line in Figure 8(a)). These differences are more prominent in Variables #5, #6, #9 and #10 (see trajectories of Variables #9 and #10 in Figure 7), but also existent in the rest of process variables (see Figure 8(b)). In this case, these differences are mainly found between the 120th onwards, time period at which the batch profiles are clearly not synchronized. Note that the variation from the main trajectory is approximately 8 times higher when the key process events are not aligned in comparison to when batch data are synchronized with the Multisynchro approach. This again supports the idea that the type of asynchronism needs to be taken into consideration in batch synchronization, not only to focus the multivariate statistical anaylsis on the same point of process evolution but also to reach better synchronization quality.

6. Conclusions

This paper addresses the problem of batch trajectories with multiple types of asynchronism. Prior to bilinear batch modeling, batch trajectories must be synchronized in such a way that not only equal batch length is ensured, but also the key process events overlap at the same batch time points in all batches. Even though batch profiles show similar shape and equal length, batch synchronization needs to be always carried out.

The application of the same synchronization procedure to batches with asynchronisms of different nature may cause the addition of extreme artifacts, affecting seriously the synchronization quality. Based on the original DTW and RGTW algorithms, a novel synchronization approach called Multisynchro that takes into consideration the multiple asynchronisms present in batch data is proposed. The new proposal is composed of two routines. The first one (high-level routine) is devoted to detect the different patterns of asynchronism of each particular batch based on the warping information derived from the Relaxed Greedy

[4]The main synchronization procedure used in SIMCA Release 13.0.3 is the called the time linear expanding/compressing (TLEC)-based method, which is based on linearly expanding and/or compressing pieces of variable trajectories in the local batch time dimension [44]. In case the differences in batch length is greater than 20%, a maturity variable is used as the basis of batch synchronization instead of the local batch time [44].



(a) SD

(b) SD: zoom corresponding to Variable #7 (active cell material) and Variable #8 (acetaldehyde dehydrogenase)

Figure 8: Comparison of the standard deviation vectors obtained from raw batch data from case#4 (red empty circles line) and from synchronized bata data derived from the Multi-sinchro-based synchronization (black stars line).

Time Warping (RGTW) or Dynamic Time Warping (DTW). The second one (low-level routine) performs the batch synchronization using specific procedures based on the nature of the asynchronism. The new approach also includes a procedure that performs abnormality detection and batch synchronization in an iterative way. This avoids batch abnormalities to affect synchronization quality. The multisynchro approach outperforms the standard approach of applying the same synchronization procedure, no matter the type of batch asynchronism.

The conclusions drawn in this paper are in line with those derived from the comparative study performed in [45]. Inappropriate synchronization affects not only the quality of batch synchronization, but also the subsequent steps of bilinear modeling. When the key process events do not overlap at the same point of process evolution ensuring the same process pace in all batches, the capability of monitoring schemes for fault detection is dramatically reduced. The novel Multisynchro algorithm is a promising synchronization technique that mitigates the influence of multiple asynchronisms on the batch modeling cycle.

Acknowledgements

This research work was partially supported by the Spanish Ministry of Economy and Competitiveness under the project DPI2011-28112-C04-02. Part of this research work was carried out during an internship of the corresponding author at Shell Global Solutions International B.V. (Amsterdam, the Netherlands). The authors also thank the anonymous referees for their comments, which greatly helped to improve the text.

Supporting Information

Four data sets composed of 40 NOC batches each affected by different types of asynchronisms are available as supplementary material: (1) batches of varying length due to incompletion of batch runs and key process events overlapping at the same sampling point across batches; (2) batches of varying length produced by a delay in measurement collection with the same evolution pace in all batches; (3) batches of varying length caused by natural process variability and incomplete batch runs, with process features not overlapping at the same sampling point across batches; and (4) batches of equal length and key process events not overlapping at the same sampling point in the last process stage in all batches. This material is available free of charge via the Internet at http://onlinelibrary.wiley.com/.

APPENDIX: LIST OF SYMBOLS

α	confidence level used to estimate the limits of SPE control chart.
$\boldsymbol{h}$	$(N \times 1)$ array containing the number of consecutive compressions performed by the synchronization algorithm at the first time period in N batches.
$\boldsymbol{v}$	$(N \times 1)$ array containing the number of consecutive expansions performed by the synchronization algorithm at the last time period in N batches.
κ	heuristic fraction used to estimate the threshold.
ψ	threshold used to discriminate among types of asynchronisms.
$\boldsymbol{\Xi}$	$(K_{ref} \times J)$ matrix of averages (i.e. average trajectory of each of the J process variables).
$\boldsymbol{\Omega}$	$(K_{ref} \times J)$ matrix of standard deviations of J process variables estimated at each K_{ref} sampling points.
$\mathbf{f}_n$	$(K_{w_n} \times 2)$ one of the possible warping paths that can be derived from the DTW/RGTW-based synchronization.
$\mathbf{f}_n^*$	$(K_{w_n} \times 2)$ optimum warping paths derived from the DTW/RGTW-based synchronization.
$\underline{\mathbf{F}}$	$(N \cdot 2 \cdot K_{w_n})$ three-way array containing the warping paths for N batches.
$\mathbf{d}$	$(K_{ref} \times K_n)$ local distance matrix calculated in the DTW/RGTW-based synchronization.
$\mathbf{D}$	$(K_{ref} \times K_n)$ cumulative weighted distance matrix calculated in the DTW/RGTW-based synchronization.
e^*	best matching between the last point of the test batch and the reference batch.
$\mathbf{E}$	$(N \times JK_{ref})$ residual matrix.
$\mathbf{P}_A$	$(JK_{ref} \times A)$ loading matrix.
s^*	best matching between the rst point of the reference batch with the test batch.
$\mathbf{T}_A$	$(N \times A)$ score matrix.
$\mathbf{W}$	$(J \times J)$ nonnegative diagonal matrix containing the weights of the J process variables for synchronization.
$\underline{\mathbf{X}}$	$(N \times J \times K_n)$ three-way array containing the measurements of J process variables collected at K_n different sampling points.
$\underline{\tilde{\mathbf{X}}}$	$(N \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} sampling points.
$\underline{\mathbf{X}}_1$	$(N_1 \times J \times K_{n_1})$ three-way array containing the measurements of J process variables measured at K_1 different sampling points in N_1 batches with class I and/or II asynchronism.
$\underline{\mathbf{X}}_2$	$(N_2 \times J \times K_{n_2})$ three-way array containing the measurements of J process variables measured at K_2 different sampling points in N_2 batches with class III asynchronism.
$\underline{\mathbf{X}}_3$	$(N_3 \times J \times K_{n_3})$ three-way array containing the measurements of J process variables measured at K_3 different sampling points in N_3 batches with class IV asynchronism.
$\underline{\mathbf{X}}_4$	$(N_4 \times J \times K_{n_4})$ three-way array containing the measurements of J process variables measured at K_4 different sampling points in N_4 batches with class III and IV asynchronism.
$\underline{\tilde{\mathbf{X}}}_1$	$(N_1 \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} smapling points in N_1 batches.
$\underline{\tilde{\mathbf{X}}}_2$	$(N_2 \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} sampling points in N_2 batches.
$\underline{\tilde{\mathbf{X}}}_3$	$(N_3 \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} sampling points in N_3 batches.
$\underline{\tilde{\mathbf{X}}}_4$	$(N_4 \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} smapling points in N_4 batches.
$\underline{\mathbf{X}}_B$	$(B_L \times J \times K_b)$ three-way array containing the original measurements of J process variables measured at K_b sampling points in B_L faulty batches, which were isolated in the L iterations of the iterative batch synchronization/abnormalities detection procedure.

$\underline{\tilde{\mathbf{X}}}_B$ $(B_l \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} sampling points in B_l faulty batches, which were isolated at the l-th iteration of the iterative batch synchronization/abnormalities detection procedure.

$\underline{\mathbf{X}}_{c\#1}$ $(N_1 \times J \times K_{n_1})$ three-way array containing the simulated measurements of J process variables measured at K_1 different sampling points in N_1 batches with class III asynchronism.

$\underline{\mathbf{X}}_{c\#2}$ $(N_2 \times J \times K_{n_2})$ three-way array containing the simulated measurements of J process variables measured at K_2 different sampling points in N_2 batches with class IV asynchronism.

$\underline{\mathbf{X}}_{c\#3}$ $(N_3 \times J \times K_{n_3})$ three-way array containing the simulated measurements of J process variables measured at K_3 different sampling points in N_3 batches with class II and III asynchronism.

$\underline{\mathbf{X}}_{c\#4}$ $(N_4 \times J \times K_{ref})$ three-way array containing the simulated measurements of J process variables measured at K_{ref} sampling points in N_4 batches with class I asynchronism.

$\tilde{\mathbf{X}}_n$ $(K_{ref} \times J)$ matrix containing the synchronized batch trajectories of the n-th batch.

$\mathbf{X}_n$ $(K_n \times J)$ matrix containing the J batch trajectories measured at K_n sampling points of the n-th batch.

$\underline{\mathbf{X}}_G$ $(G \times J \times K_g)$ three-way array containing the measurements of J process variables measured at K_G sampling points in G normal batches.

$\underline{\tilde{\mathbf{X}}}_G$ $(G \times J \times K_{ref})$ three-way array containing the measurements of J process variables synchronized at K_{ref} sampling points in G normal batches.

$\mathbf{X}_{ref}$ $(K_{ref} \times J)$ matrix containing the J batch trajectories measured at K_{ref} sampling points in the batch selected as reference for synchronization.

References

[1] T. Kourti, 4.02 - multivariate statistical process control and process control, using latent variables, in: Comprehensive Chemometrics, Elsevier, Oxford, 2009, pp. 21 – 54.

[2] S. Wold, N. Kettaneh-Wold, J. MacGregor, K. Dunn, 2.10 - batch process modeling and mspc, in: Comprehensive Chemometrics, Elsevier, 2009, pp. 163 – 197.

[3] T. Kourti, Abnormal situation detection, three-way data and projection methods; robust data archiving and modeling for industrial applications, Annual Reviews in control 27 (2003) 131–139.

[4] S. Lakshminarayanan, R. Gudi, S. Shah, Monitoring batch processes using multivariate statistical tools: extensions and practical issues, in: Proceedings of lFAC Worm Congress, pp. 241–246.

[5] M. Zarzo, A. Ferrer, Batch process diagnosis: Pls with variable selection versus block-wise pcr, Chemometrics and Intelligent Laboratory Systems 73 (2004) 15–27.

[6] D. Louwerse, A. Smilde, Multivariate statistical process control of batch processes based on three-way models, Chemical Engineering Science 55 (2000) 1225–1235.

[7] J. Westerhuis, T. Kourti, J. MacGregor, Comparing alternative approaches for multivariate statistical analysis of batch process data, Journal of Chemometrics 13 (1999) 397–413.

[8] P. Nomikos, J. MacGregor, Monitoring batch processes using multiway principal components, AICHE Journal 40 (1994) 1361–1375.

[9] C. Ündey, S. Ertunç, A. Çinar, Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis, Industrial and Engineering Chemical Research 42 (2003) 4645–4658.

[10] D. Neogi, C. Schlags, Multivariate statistical analysis of an emulsion batch process, Industrial and Engineering Chemistry Research 37 (1998) 3971–3979.

[11] T. Kourti, J. Lee, J. MacGregor, Experiences with industrial applications of projection methods for multivariate statistical process control, Computers & Chemical Engineering 20 (1996) S745–S750.

[12] C. Duchesne, T. Kourti, J. MacGregor, Multivariate spc for startups and grade transitions, AIChE Journal 48 (2002) 2890–2901.

[13] Y. Zhang, M. Dudzic, V. Vaculik, Integrated monitoring solution to start-up and run-time operations for continuous casting, Annual Reviews in Control 27 (2003) 141 – 149.

[14] S. G. Rothwell, E. B. Martin, A. J. Morris, Comparison of methods for handling unequal length batches, in: Proceedings of IFAC DYCOPS5, Corfu, Greece, pp. 66–71.

[15] S. García-Muñoz, T. Kourti, J. MacGregor, Troubleshooting of an industrial batch process using multivariate methods, Industrial and Engineering Chemistry Research 42 (2003) 3592–3601.

[16] S. Wold, N. Kettaneh, H. Friden, A. Holmberg, Modelling and diagnostics of batch processes and analogous kinetic experiments, Chemometrics and Intelligent Laboratoy Systems 44 (1998) 331–340.

[17] N. Kaitsha, C. F. Moore, Extraction of event times in batch profiles for time synchronization and quality predictions, Industrial & Engineering Chemistry Research 40 (2001) 252–260.

[18] J. Ramsay, B. Silverman, Functional data analysis, New York:Springer-Verlag, 1997.

[19] C. Ündey, A. Çinar, Statistical monitoring of multistage, multiphase batch processes, IEEE Control Systems Magazine 22 (2002) 40–52.

[20] S. W. Andersen, G. C. Runger, Automated feature extraction from profiles with application to a batch fermentation process, Journal of the Royal Statistical Society: Series C (Applied Statistics) 61 (2012) 327–344.
[21] R. Srinivasan, M. S. Qian, Off-line temporal signal comparison using singular points augmented time warping, Industrial & Engineering Chemistry Research 44 (2005) 4697–4716.
[22] R. Srinivasan, M. S. Qian, Online fault diagnosis and state identification during process transitions using dynamic locus analysis, Chemical Engineering Science 61 (2006) 6109 – 6132.
[23] R. Srinivasan, M. S. Qian, Online temporal signal comparison using singular points augmented time warping, Industrial & Engineering Chemistry Research 46 (2007) 4531–4548.
[24] J. Chen, J. Liu, Post analysis on different operating time processes using orthonormal function approximation and multiway principal component analysis, Journal of Process Control 10 (2000) 411 – 418.
[25] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on acoustics, speech, and signal processing 26 (1978) 43–49.
[26] N. Nielsen, J. Carstensen, J. Smedsgaard, Aligning of single and multiple wavelength chromatographic profiles for chemometrics data analysis using correlation optimised warping, Journal of Chromatography 805 (1998) 17–35.
[27] V. Pravdova, B. Walczak, D. Massart, A comparison of two algorithms for warping of analytical signals, Analytica chimica acta 456 (2002) 77–92.
[28] G. Tomasi, F. van den Berg, Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data, Journal of Chemometrics 18 (2004) 231–241.
[29] A. Kassidas, J. MacGregor, P. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE Journal 44 (1998) 864–875.
[30] K. Gollmer, C. Posten, Supervision of bioprocesses using a dynamic time warping algorithm, Control Engineering Practice 4 (1996) 1287–1295.
[31] H. Ramaker, E. van Sprang, J. Westerhuis, A. K. Smilde, Dynamic time warping of spectroscopic batch data, Analytica Chimica Acta 498 (2003) 133–153.
[32] M. Fransson, S. Folestad, Real-time alignment of batch process data using cow for on-line process monitoring., Chemometrics and Intelligent Laboratory Systems 84 (2006) 56–61.
[33] J. M. González-Martínez, A. Ferrer, J. A. Westerhuis, Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping, Chemometrics and Intelligent Laboratory Systems 105 (2011) 195–206.
[34] G. Gins, P. Van den Kerkhof, J. F. M. Van Impe, Hybrid derivative dynamic time warping for online industrial batch-end quality estimation, Industrial & Engineering Chemistry Research 51 (2012) 6071–6084.
[35] Y. Zhang, T. F. Edgar, A robust dynamic time warping algorithm for batch trajectory synchronization, in: Proceedings of American Control Conference, pp. 2864–2869.
[36] J. M. González-Martínez, J. A. Westerhuis, A. Ferrer, Using warping information for batch process monitoring and fault classification, Chemometrics and Intelligent Laboratory Systems 127 (2013) 210–217.
[37] T. Kourti, Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions, Journal of Chemometrics 17 (2003) 93–109.
[38] J. M. González-Martínez, J. Camacho, A. Ferrer, Bilinear modeling of batch processes. part III: parameter stability, Journal of Chemometrics 28 (2014) 10–27.
[39] J. Camacho, A. Ferrer, Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects, Chemometrics and Intelligent Laboratory Systems 131 (2014) 37 – 50.
[40] F. Arteaga, A. Ferrer, Dealing with missing data in mspc: several methods, different interpretations, some examples, Journal of Chemometrics 16 (2002) 408–418.
[41] F. Lei, M. Rotbøll, S. Jørgensen, A biochemically structured model for saccharomyces cerevisiae, Journal of Biotechnology 88 (2001) 205–221.
[42] J. Camacho, J. González-Martínez, A. Ferrer, Multi-phase (mp) toolbox, http://mseg.webs.upv.es/Software.html, 2013.
[43] UMETRICS, SIMCA 13.0.3, info@umetrics.com: www.umetrics.com, Umea, Sweden (2013).
[44] L. Eriksson, E. Johansson, N. Kettaneh-Wold, J. Trygg, C. Wikström, S. Wold, Multi- and Megavariate Data Analysis, 13, Umetrics AB.
[45] J. M. González-Martínez, R. Vitale, O. E. de Noord, A. Ferrer, Effect of synchronization on bilinear batch process modeling, Industrial & Engineering Chemistry Research (2014) DOI:10.1021/ie402052v.