Bilinear Modeling of Batch Processes. Part III: Parameter Stability

J.M. González-Martínez[a,*], J. Camacho[b], A. Ferrer[a]

[a] *Department of Applied Statistics, Operational Research and Quality, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, Spain*

[b] *Department of Signal Theory, Networking and Communications, Universidad de Granada, 18071, Granada, Spain*

Abstract

A paramount aspect in the development of a model for a monitoring system is the so-called parameter stability. This is inversely related to the uncertainty, *i.e.* the variance in the parameters estimates. Noise affects the performance of the monitoring system, reducing its fault detection capability. Low parameters uncertainty is desired to ensure a reduced amount of noise in the model. Nonetheless, there is no sound study on the parameter stability in Batch Multivariate Statistical Process Control (BMSPC). The aim of this paper is to investigate the parameter stability associated to the most used synchronization and PCA-based BMSPC methods. The synchronization methods included in this study are: Indicator Variable, Dynamic Time Warping, Relaxed Greedy Time Warping and Time Linear Expanding/Compressing-based methods. In addition, different arrangements of the 3-way batch data into 2-way matrices are considered, namely: single-model approaches, K-models approaches and hierarchical approaches. Results are discussed in connection with previous conclusions in the first two papers of the series.

Keywords: Stability, uncertainty, multivariate statistical process control, unfolding, principal component analysis, synchronization.

1. Introduction

Batch processing plays an important role in the production of high value-added products, such as in the pharmaceutical, food, semiconductor, and biochemical industries, among others. The final goal of a monitoring scheme in a batch process is safe and stable operation, to maintain the release of high quality product and to minimize the waste of product in off-spec batches. For this purpose, these schemes must be designed in such a way that faults, failures and disturbances can be accurately and early detected, allowing the subsequent diagnosis of their potential causes. Once these causes have been diagnosed, actions in the process can be taken, restoring the faulty operation to a normal operating condition (NOC).

For the design of monitoring schemes, the measurements of J process variables collected at K different sampling points over I batches run under NOC are used. Setting a BMSPC system becomes a challenging task due the nature of batch data [1, 2]: high volume of data (high dimensionality); unequalized batch trajectories; uneven and unsynchronized batch trajectories; non-linear and time-varying dynamics; presence of noise, collinearity and outliers, variables of different magnitude and variance, and missing data. In this context, Latent Structures-based methods, like Principal Component Analysis (PCA) and Partial Least Squares (PLS), combined with the adequate preprocessing methods are frequently used for the generation of empirical models [2, 3]. Using this type of methods, process understanding can be gained and process operating problems can be troubleshooted in a timely manner. From this off-line investigation based on historical data, a monitoring system can be designed (the so-called model building phase), allowing real-time fault detection and diagnosis on the basis of incoming batch data (the so-called exploitation model phase) [4].

*Corresponding author

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

A principal concern when designing BMSPC systems based on PCA should be the stability of the model parameters -*i.e.* the preprocessing parameters (means and standard deviations) and the loadings. The parameter stability is inversely related to the uncertainty, *i.e.* the variance in the parameters estimates. The assessment of the parameter stability is relevant for almost any purpose PCA is applied for. If PCA is used to develop a monitoring scheme, low parameters uncertainty is desired to assure a reduced amount of noise in the model. Noise affects the performance of the monitoring system, reducing the fault detection capability. From the statistical point of view, it is well known that the higher the number of observations in the calibration the better the parameters estimation and so the lower the parameters uncertainty. There is a second element which affects the uncertainty in the parameters of PCA: the more different the eigenvalues in the model, the more stable the loadings [5].

The application of bilinear models like PCA to batch data requires the rearranging of the 3-way data matrix in a number of 2-way matrices. This transformation can be performed following a number of different approaches. This is the third paper of a series devoted to study and compare several of these approaches from different perspectives: process dynamics modeling, on-line prediction and parameter stability. In the first paper [6], a theoretical discussion on the capability to capture the process dynamics based on the structure of the covariance matrices was presented. In the second paper of the series [7], PLS modeling approaches were compared in the on-line estimation (soft sensor) of a key variable in a batch process. The main motivation of this paper is to complement the companion papers. For that, a comparison of the most used modeling approaches and synchronization methods in terms of parameter stability is performed.

This paper is organized as follows. The state of the art concerning the development of BMSPC systems based on PCA is introduced in Section 2. Section 3 presents the materials and methods of the research work. Section 4 illustrates the effect of the batch synchronization on the parameter stability. Section 5 is devoted to present and discuss the results of the comparison of the different rearranging methods under study. Finally, conclusions are drawn in Section 6.

2. State of the art

In model building for process monitoring, a number of steps are typically performed, namely: i) data alignment, ii) data preprocessing and iii) transformation of the 3-way data matrix to one or several 2-way data matrices for the subsequent iv) bilinear batch modeling (see Figure 1). These steps are iteratively repeated whereas outliers are detected and isolated from the calibration data set.

The data alignment step includes equalization of variables and batch synchronization. The aim of this stage is to obtain a 3-way data structure from the data collected through the net of process sensors with multiple sampling rates and for batches of possibly different duration and/or processing pace. Batch synchronization is one of the most important steps prior to batch modeling and process monitoring. The accuracy of both empirical models and the subsequent monitoring schemes in terms of fault detection and fault diagnosis is highly dependent on the synchronization quality [8]. A number of proposals for dealing with the most complex synchronization problems can be found in the literature. The approaches for synchronizing batch data can be roughly classified into three categories. The first category are the methods based on compressing/expanding the raw trajectories using linear interpolation. This interpolation can be performed in the time dimension, which is referred to as the Time Linear Expanding/Compressing (TLEC)-based method. The TLEC can be applied to the entire batch run [9], which is the technique implemented in SIMCA Release 13.0.3 -Umetrics software- [10], or within stages that are defined by key process events [11, 12], which is one of the synchronization techniques coded in ProMV Batch Edition Release 13.02 -ProSensus software- [13]. Other linear interpolation-based strategies also exist [12, 14]. Additionally, the linear interpolation can be applied in an indicator variable dimension, following the so-called Indicator Variable-based synchronization, IV [15]. A second strategy is formed by methods based on features extraction [2, 16, 17, 18]. Finally, a third category are the methods based on Stretching, Compressing and Translating pieces of trajectories (the SCT-based methods), such as Dynamic Time Warping (DTW) [19] and Relaxed Greedy Time Warping (RGTW) [20]. In [19], an end-of-batch version of DTW for batch synchronization was proposed and some guidelines to carry out the real-time synchronization were also presented. Nonetheless, this real-time version was proved to be inappropriate in BMSPC due to the false alarms produced in process

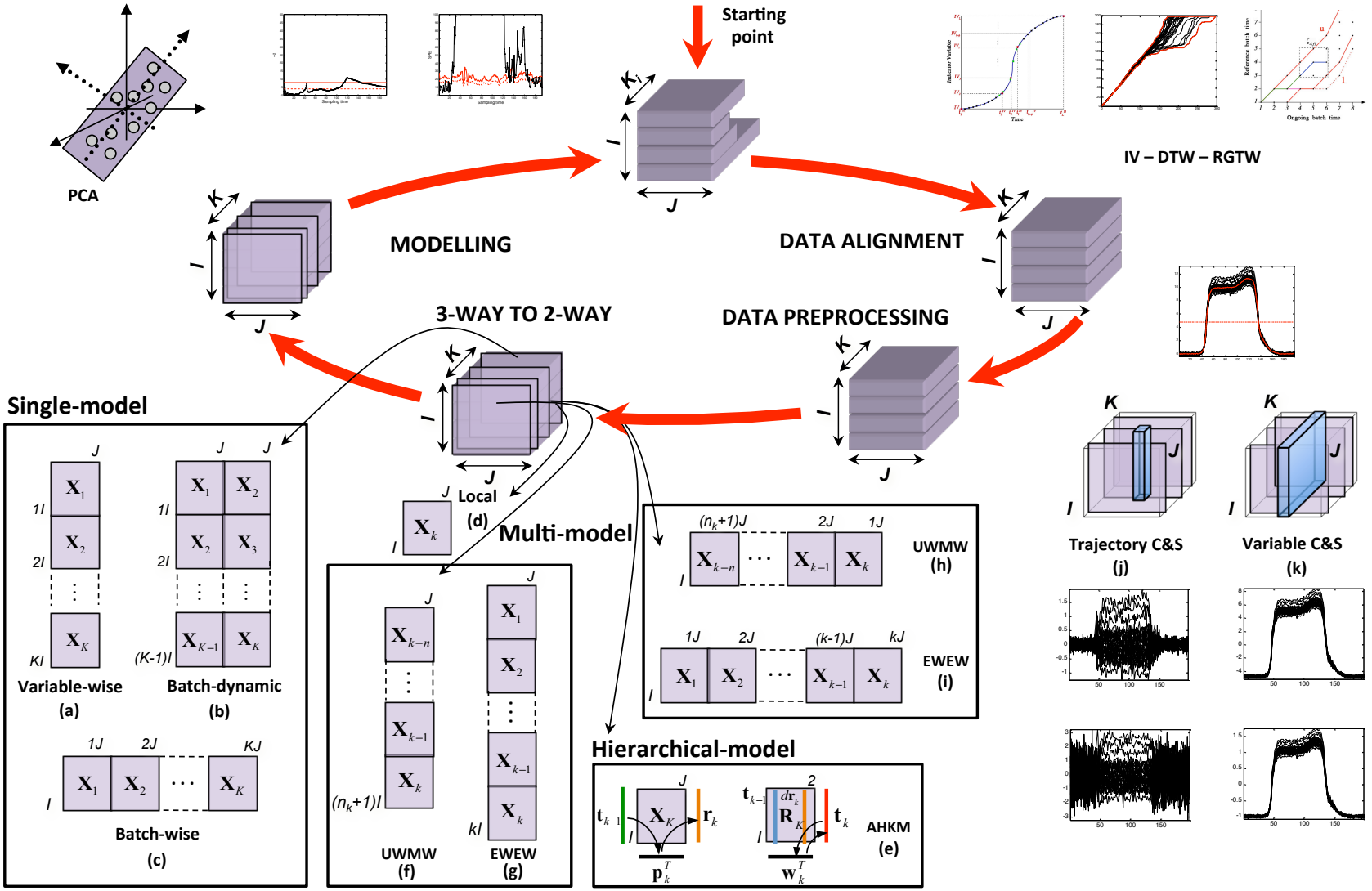


Figure 1: Modelling scheme in BMSPC systems based on PCA.

monitoring, being the RGTW a solution to overcome this problem [20]. Other SCT-based methods were proposed in the literature for batch synchronization in off-line and real-time applications [21, 22].

After synchronization, a preprocessing step is required before model calibration. Depending on the nature of batch data and the type of model to be fitted, the preprocessing approach may be different [23]. Two main preprocessing methods are widely used in process chemometrics: trajectory centering and scaling (Trajectory C&S) and variable centering and scaling (Variable C&S). The former consists of mean centering and scaling to unit variance the data corresponding to each j-th process variable at each k-th sampling point, *i.e.* each vector $\mathbf{x}_{jk}$ is mean-centered and scaled to unit variance (see Figure 1(j)). Provided the synchronized 3-way data structure contains J variables, K sampling points and I batches, this means that $J \cdot K$ averages and standard deviations are computed from I batches each. These averages are then subtracted to the corresponding data and, then, the I observations corresponding to the j-th process variable at the k-th sampling point are scaled to unit variance. This normalization allows each process variable at each time to have the same weight in the multivariate analysis. Variable C&S performs mean centering and scaling to unit variance of the data corresponding to each j-th process variable. This means that each lateral slab $\mathbf{X}_j$ is mean-centered and scaled to unit variance (see Figure 1(k)). Hence, J averages and standard deviations are computed from $I \cdot K$ observations each. Again, these averages (also called grand means) are subtracted and, subsequently, the centered data is scaled to unit variance. With this normalization, time periods with more variability will be weighted more and periods with lesser variability (*e.g.* under tight control) will get a small weight in the multivariate analysis. Two main advantages of Trajectory C&S over Variable C&S makes the former more suitable than the latter for BMSPC: i) Trajectory C&S models the variability around the average trajectory, which is actually the type of variability of interest to monitor a batch process [24]; and ii) the non-stationary problem is transformed into a stationary problem since the average trajectory is removed from batch [1]. The discussion about which of these two choices is more adequate has been present in the literature [24, 25] since the two main pioneer research work in BMSPC [26, 27] selected one of them each. Nomikos and MacGregor [26] performed Trajectory C&S whereas Wold *et al.* [27] used Variable C&S. After the desired variability is kept on data, the transformation of the 3-way data matrix to a 2-way data matrix can be carried out.

In model calibration, the aligned and preprocessed 3-way data matrix $\underline{\mathbf{X}}$ needs to be rearranged in a number of 2-way submatrices to fit bilinear models, such as PCA or PLS. The different approaches to perform this transformation can be classified into three categories: the single-model approach, the K-model approach and the hierarchical approach.

In the single-model approach, the 3-way matrix is unfolded in a single 2-way matrix. There are several unfolding choices, which differ in the number of process variables lagged in time (the so-called Lagged Measurement Vectors, LMV): variable-wise [27], batch dynamic [28] and batch-wise unfolding [15, 26] (see Figure 1(a), Figure 1(b) and Figure 1(c), respectively). Batch dynamic unfolding can be seen as a generalization of the traditional unfolding procedures [6]: if no LMV is added, the resulting matrix is the same as the one after variable-wise unfolding - *i.e.* $\mathbf{X}$ $(IK \times J)$; if all possible LMV are added, the resulting matrix is the same as the one after batch-wise unfolding - *i.e.* $\mathbf{X}$ $(I \times KJ)$. The addition of a certain number of LMVs depends on two factors: the order of the dynamics that need to be modeled and/or how the correlation structures changes throughout the batch run, *i.e.* the way process variables are related with each other and in time [6].

The K-models approach is based on generating as many bilinear models as sampling points in a batch. Several proposals can be found in the literature, which differ in the data used in the generation of the sub-models. If each sub-model only incorporates measurements collected at the current sampling point, then it is called local model [29] -*i.e.* $\mathbf{X}$ $(I \times J)$ (see Figure 1(d)). If measurements registered from the beginning of the batch to the current sampling point k are taken into account in each sub-model -*i.e.* $\mathbf{X}$ $(I \times kJ)$-, then it is known as evolving model [29]. This approach can be seen as a local model at the k-th sampling time where all the possible LMV are included as additional variables [6]. Special cases

[1] Provided the batch process is under tight control so that the process can be considered to be stationary in the batch dimension

of evolving models are Uniformly Weighted Moving Window (UWMW)[30] and Exponentially Weighted Evolving Window (EWEW) models, which are used when not all the past information (lagged information) is of interest or has the same importance in bilinear modeling [6]. UWMW models are based on modeling the information contained into a window of width n_k, *i.e.* the measurements collected at the k-th current sampling point with those of the immediate n_k LMVs. This information can be also seen as a local model at the k-th sampling time where n_k LMVs are included as observations -*i.e.* $\mathbf{X}$ $(n_k I \times J)$- (see Figure 1(f)) or as variables -*i.e.* $\mathbf{X}$ $(I \times n_k J)$- (see Figure 1(h)). In contrast, EWEW models incorporate all the lagged measurements to the k-th current sampling point, which are weighted following an exponentially decreasing profile associated to the weighting factor $\lambda_k \in [0, 1]$. With this factor, the measurements are losing importance over the batch duration and their contribution to the covariance matrix is down-weighted [6]. The weight of the measurement-vector collected at time $k - d$, for the generation of the sub-model at time k, is $(\lambda_k)^d$, being the weight of the current measurements always $(\lambda_k)^0 = 1$. This is equivalent to a local model at the k-th sampling point where all the possible LMV are added either as observations -*i.e.* $\mathbf{X}$ $(kI \times J)$- (see Figure 1(g)) or as variables -*i.e.* $\mathbf{X}$ $(I \times kJ)$- (see Figure 1(i)) and exponentially weighted. One of the advantages of these K-model approaches is that they are capable of capturing varying dynamics of certain order. The main drawback is the generation and maintenance of a high number of sub-models. For the reduction of sub-models, some authors proposed to calibrate independent linear models for each one of the process stages (the so-called multi-stage approach) [2, 31] or separately model segments of batch data that are well approximated by a linear model (PCA or PLS) [32]. For more detail on the structure of the different K-model approaches, the interested reader is referred to the first paper of the series [6].

The hierarchical approach is based on combining the past and current information at each sampling point with an adaptive hierarchical PCA model (see Figure 1(e)) [33]. Firstly, a PCA model is fitted on the information belonging to the first sampling point, *i.e.* $\mathbf{X}_1$ $(I \times J)$. At sampling time point k, the overall score vector $\mathbf{t}_{k-1}$, which summarizes previous process variation up to the sampling time point $k-1$, is used together with matrix $\mathbf{X}_k$ to estimate the block scores $\mathbf{r}_k$. Afterwards, this score vector is weighted by the forgetting factor d (adaptive nature) and placed together with the previous overall score vector $\mathbf{t}_{k-1}$ in the consensus matrix $\mathbf{R}_k$. This matrix is then used to calculate the overall scores vector $\mathbf{t}_k$, which represents the total process variation up to the sampling point k. For more details, the reader is referred to [33].

Once batch data have been properly prepared, calibrated and outliers have been isolated, a monitoring scheme can be built. Typically, two Shewhart control charts based on the Hotelling-T^2 and Squared Prediction Error (SPE) statistics are designed. Their control limits (thresholds) are estimated from NOC process data. Also, it is recommended to readjust these limits using cross-validation techniques for an imposed significance level (ISL) [29, 34, 35]. Once the scheme is designed, the measurements from a new batch can be projected onto the latent subspace, yielding to the aforementioned multivariate statistics, to check for the correct performance of the process.

3. Material and methods

The different modeling approaches under study are compared in terms of parameter stability using data from realistic simulations of a fermentation process of the *Saccharomyces cerevisae* cultivation. Two data sets were generated based on the biological model of the aerobic growth of *S. Cerevisae* on a glucose limited medium [36] (available in the MP toolbox [37]), using Simulink for Matlab release 2010a ® (©The MathWorks, Inc). Parameter stability is assessed by modeling the batch data of both data sets using the approaches under study and comparing the model parameters fitted.

In the simulation of batch data, physical uncertainty caused by the biological variability is taken into consideration. Slightly modified values of constants of the first principles model are introduced into the parametric space. Also, Gaussian noise of low magnitude is added in the initial conditions (10%) and measurements (5%) to simulate the typical errors in sensors is added. Furthermore, the simulation achieved here takes into account the biological variability of yeasts. In fermentation processes, characterized by a duration of days, some microorganisms may have different generation times, with a significant influence on biomass growth and quality features. This is the main cause why this type of process presents different release times for different batches.

A total of 30 unsynchronized batches for each data set are simulated under normal operating conditions and in two different simulation sequences to ensure independency[2]. Ten process variables are measured 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. Also, the original time of processing from simulation is added to the batch data matrix. The total length of the batches belonging to the first data set varies from 172 to 330 data points, and in the second data set from 173 to 297 data points.

Prior to proceeding with the comparative study, both data sets need to be synchronized. For this purpose, methods working in the domain of the batch time (SCT-based and TLEC-based methods) and in the domain of an indicator variable (IV) are used. For the sake of simplicity, only two SCT-based methods (DTW and RGTW algorithms) are chosen (see Table 1). For the DTW-based synchronization on raw batch data, the reference batch selected in both data sets is is that whose batch length is the closest to median length from the first data set: batch #12. This is also the reference batch for the RGTW-based synchronization on raw batch data. The rest of conditions and constraints, both for the classical DTW and the RGTW algorithm, are set according to [19, 20]. The TLEC is carried out in raw batch data by linearly interpolating 209 data points (the length of the reference batch, batch #12 belonging to the first data set) in each batch. In order to check to what extend TLEC correctly synchronizes the batch trajectories (*i.e.* the key process events overlap in all batches ensuring the same process evolution), the TLEC-based synchronized batch trajectories are re-synchronized, *i.e.* synchronized once again, with SCT-based methods. In particular, a second synchronization using the DTW algorithm (TLEC-DTW) and the RGTW algorithm (TLEC-RGTW) with the aforementioned parameters is performed. Finally, the TLEC-based synchronization between stages defined by key process events (TLEC events) is carried out. For the sake of comparison, batch #12 from the first data set is selected as reference batch. A total of 10 key events placed at sampling points #23, #38, #54, #65, #89, #96, #104, #119, #140 and #166 in the reference batch are extracted by examining its variable trajectories. Afterwards, time linear interpolation is performed between time periods limited by the defined key process events across batches, yielding a set of synchronized trajectories with 209 sampling points. Concerning IV-based synchronization, the biomass concentration is selected as indicator variable given its monotonic and increasing behavior. To fulfil the requirements of IV, a start and end point in the biomass concentration variable is selected across batches. A total of 209 data points are obtained by linear interpolation. In addition, a second synchronization on the IV-based synchronized trajectories using the DTW algorithm (IV-DTW) and the RGTW algorithm (IV-RGTW) with the parameters specified above is carried out. The purpose of this re-synchronization is again to check to what extend IV properly synchronizes the key process events.

The comparison of the PCA-based MSPC approaches in terms of the parameter stability is organized in three categories: single-model approaches, K-model approaches and hierarchical-model approaches (see Table 2). Among the single-models, variable-wise (VW), batch dynamic (BD) and batch-wise (BW) models are studied. The approaches VW-TCS and VW-VCS represent a variable-wise unfolding where Trajectory C&S and Variable C&S[3] are performed, respectively. BD1 denotes a batch-dynamic model where 1 LMV is added as new variables and Trajectory C&S is applied. BW represents a batch-wise model where Trajectory C&S is applied. Regarding the K-model approaches, local K-models and evolving models in their different variants are studied. LM represents local K-models with Trajectory C&S. The approaches UWMW 1LMV-var and UWMW 1LMV-obs denote Uniformly Weighted Moving Window models with Trajectory C&S generated by adding 1 LMV as new variables and observations, respectively. EWEW-var and EWEW-obs correspond to Exponentially Weighted Evolving Window models generated by adding all the possible LMVs at the k-th sampling time as new variables and observations, respectively. Also, Trajectory C&S is applied and a weighting factor $\lambda_k \in [0, 1]$ is used, where $\lambda = 0.97$. In addition, the adaptive approach of the local K-models with $d = 0.2$ and $d = 50$, *i.e.* AHKM, is also included in the study.

[2]The seed used in the simulation differs for each data set to obtain different sequences of random numbers, which are used to generate Gaussian noise and the length of batches

[3]The application of Variable C&S is only meaningful in VW. Hence, this preprocessing approach is not taken into consideration for the rest of BMSPC approaches in this study

Table 1: Synchronization approaches used in the study of the parameter stability to synchronize batch data.

Domain	Approach	Model	Parameters
Time	Stretching/Compressing/Translating (SCT)-based method	DTW	Reference: batch #12 (209 time points), constraints according to [19]
		RGTW	Reference: batch #12 (209 time points), constraints according to [20]
	Time Linear Expanding/Compressing (TLEC)-based method	TLEC	209 interpolation points
		TLEC-events	209 interpolation points, key processes events at sampling points: #23, #38, #54, #65, #89, #96, #104, #119, #140 and #166
	(TLEC & SCT)-based method	TLEC-DTW	Parameters from TLEC and DTW models
		TLEC-RGTW	Parameters from TLEC and RGTW models
Variable	IV-based method	IV	Indicator variable: variable #6 (Biomass concentration)
	(IV & SCT)-based method	IV-DTW	Parameters from IV and DTW models
		IV-RGTW	Parameters from IV and RGTW models

Table 2: BMSPC approaches used in the study of the parameter stability. M represents the number of PCA models fitted in each modeling approach.

Approach	Model	Structure	Preprocessing	# Parameters per loading vector
Single-model ($M = 1$)	BW	Batch-wise	Trajectory C&S	$J \cdot K$
	VW-TCS	Variable-wise	Trajectory C&S	J
	VW-VCS	Variable-wise	Variable C&S	J
	BD1	Batch-dynamic with 1LMV	Trajectory C&S	$J \cdot (1 + LMV)$
Multi-model ($M = K$)	LM	Local K-model	Trajectory C&S	J
	UWMW 1LMV-var	Uniformly Weighted Moving Window K-model with 1LMV in the variables	Trajectory C&S	n_k
	UWMW 1LMV-obs	Uniformly Weighted Moving Window K-model with 1LMV in the observations	Trajectory C&S	J
	EWEW-var	Exponentially Weighted Evolving Window K-model with 1LMV in the variables and λ_k	Trajectory C&S	$k \cdot J$, for k from 1 to K
	EWEW-obs	Exponentially Weighted Evolving Window K-model with 1LMV in the observations and λ_k	Trajectory C&S	J
Hierarchical-model ($M = K$)	AHKM	Adaptive hierarchical K-model with $d = 0.2$ and $d = 50$	Trajectory C&S	J

A priori, there are clear equivalences and an important interplay between the parameter stability in the preprocessing and in the unfolded model. To compare the parameter stability of each one of the calibration and monitoring approaches, the Normalized Squared Difference (NSD) between the different parameter vectors (averages, standard deviations, sum of squares and loadings) is computed as follows:

$$NSD_{\theta} = \sum_{j=1}^{J} \left(\frac{\theta_j^{(1)}}{\left\| \boldsymbol{\theta}^{(1)} \right\|} - \frac{\theta_j^{(2)}}{\left\| \boldsymbol{\theta}^{(2)} \right\|} \right)^2 \tag{1}$$

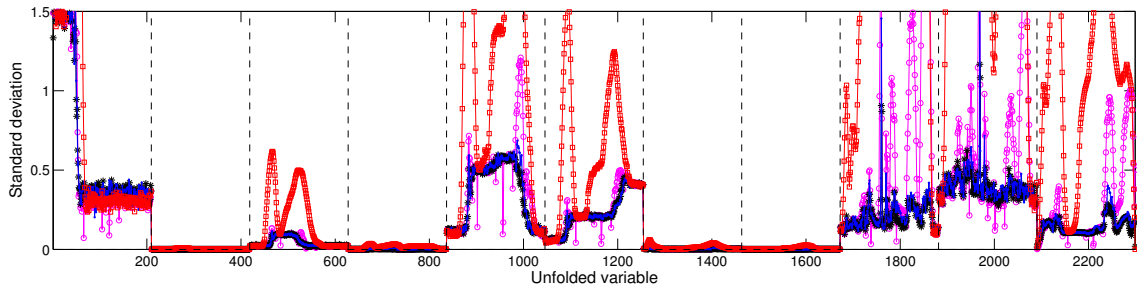
where $\theta_j^{(1)}$ and $\theta_j^{(2)}$ correspond to the j-th value in the parameter vectors $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ for the first and second data set, respectively. To make the NSD values of the loadings comparable across approaches, two factors need to be taken into account in the estimation: the number of PCA models and the number of parameters. As illustrated in Table 2, $M = 1$ and $M = K$ different models are obtained from PCA-based bilinear modeling in the single-model and multi-model approach, respectively. The size of the loading vectors in each model depends on the number of LMV added as new variables. To make all the models comparable, the NSD values are estimated as an average of the NSD values calculated on the loadings corresponding to each sampling point k (*i.e.* $NSD_{\theta} = \sum_{k=1}^{K} NSD_{\theta_k}/K$, where NSD_{θ_k} is assessed by following Equation (1)). When including LMVs, exception made for BW models, data from a specific sampling time are used more than once to fit parameters in the same (BD) or different submodels (*e.g.* UWMW). When this occurs, parameters in the form of LMVs are not considered to compute the NSD. To properly estimate the NSD values in loadings, the sign change of loadings due to the rotational ambiguity in PCA is taken into account. For this purpose, each loading vector $\mathbf{p}_a$ is corrected by the sign of the absolute maximum loading. Note that the averaged NSD value allows us to compare the NSD values of single models including the complete batch history (BW/AKHM), single models where the batch history is averaged (VW-VCS/VW-TCS), singles models with LMVs (BD) and K-models with LMVs as observations (UWMW-obs/EWEW-obs) and as variables (UWMW-vars/EWEW-vars).

Batch data synchronized by all the synchronization approaches under study (see Table 1) are employed to study the effect of batch synchronization on parameter stability in Section 4. To proceed with the comparison of the rearranging methods in terms of parameter stability in Section 5, for the sake of easy understanding only the two data sets synchronized by using the RGTW algorithm are used.

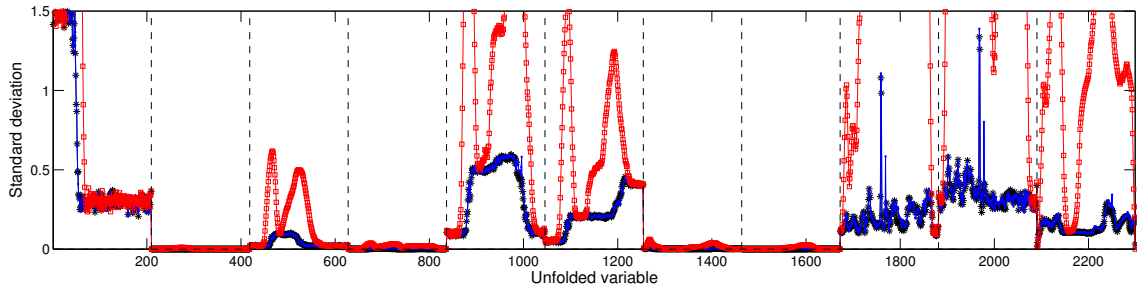
4. Effects of batch synchronization on parameter stability

A critical factor in the modeling of batch data is the synchronization quality, *i.e.* the accuracy of the synchronization approach to overlap the key process events across batches. An indicator of this factor is the variability of the resulting synchronized batch trajectories around their mean trajectory. This can be measured by the standard deviation vector after trajectory C&S. The lower the difference among standard deviation vectors, the higher the synchronization quality.

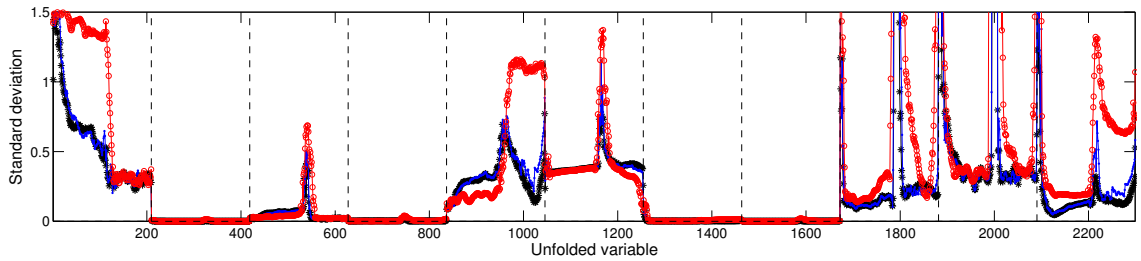
In order to compare the methods, the average of the standard deviation vectors of the corresponding synchronized batch trajectories of both data sets are computed and shown in Figure 2. This figure reveals that when SCT-based methods are applied in batch data, the resulting standard deviation values are lower (blue dots and black asterisks lines in Figure 2(a), 2(b) and 2(c), respectively) than for the rest of synchronization methods. This implies that SCT-based methods outperform the other approaches in terms of reduction of trajectory variability. Note that the differences are more prominent in Variables #1, #5, #6, #9, #10 and #11 in all the comparisons. Concerning the TLEC-based methods, TLEC-based synchronized batch trajectories yield standard deviation values much higher (red empty squares lines in Figure 2(a)) than those synchronized with TLEC-events (magenta empty circles lines in Figure 2(a)). Hence, the latter synchronizes the batch trajectories with more accuracy, reducing the variability in comparison with the former. Another issue worth being highlighted is that the standard deviation vectors calculated from the batch trajectories synchronized and re-synchronized by SCT-based methods do not differ much each other (compare black, blue and red empty circles -*i.e.* DTW, TLEC-DTW and IV-DTW- with black, blue and red dots -*i.e.*



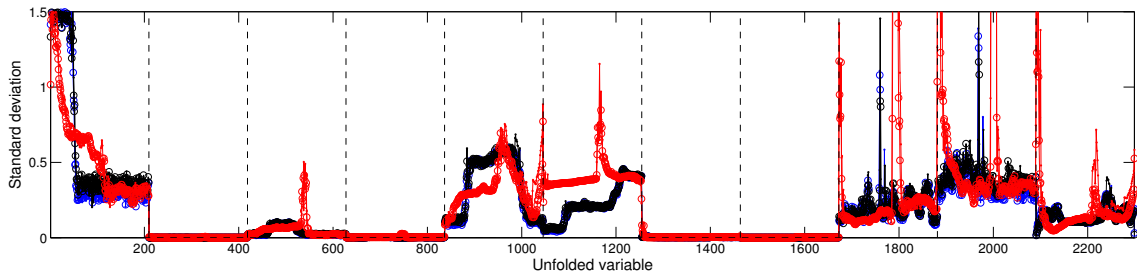
(a)



(b)



(c)



(d)

Figure 2: Standard deviation vector of the synchronized trajectories (average from both data sets) using the synchronization methods under study: (a) comparison between the SCT-based and TLEC-based methods: blue dots (RGTW), black asterisks (DTW), magenta empty circles (TLEC-events) and red empty squares lines (TLEC); (b) comparison between the (TLEC & SCT)-based and TLEC-based methods: blue dots (TLEC-RGTW), black asterisks (TLEC-DTW) and red empty squares (TLEC) lines; (c) comparison between the (IV & SCT)-based and IV-based methods: blue dots (IV-RGTW), black asterisks (IV-DTW) and red empty circles (IV) lines; and (d) comparison between SCT-based methods and (TLEC/IV & SCT)-based methods: black empty circles (DTW), black dots (RGTW), blue empty circles (TLEC-DTW), blue dots (TLEC-RGTW), red empty circles (IV-DTW) and red dots (IV-RGTW).

RGTW, TLEC-RGTW and IV-RGTW-, respectively, in Figure 2(d)). This denotes a similar performance among synchronization methods (RGTW and DTW) in terms of synchronization. Notice, however, that RGTW performs the synchronization in real-time, while DTW requires to wait until the batch has finished to perform the synchronization.

This outcome shows that when synchronization is focused on aligning the key process events (SCT-based synchronization), the addition of artefacts and, consequently, the amount of noise, is reduced. Hence, the variability of the resulting batch trajectories is lower that those where the key process events are not properly synchronized (TLEC-based and IV-based synchronization). The resulting standard deviation vectors after applying a second SCT-based synchronization in trajectories already synchronized by TLEC (see blue dots and black asterisks lines in Figure 2(b)) and by IV (see blue dots and black asterisks lines lines in Figure 2(c)) contain lower values than those derived from the synchronized trajectories by TLEC and IV (see red empty squares line in Figure 2(b) and see red empty circles line in Figure 2(c), respectively). The enhancement of the synchronization (*i.e.* the difference of standard deviation among synchronization approaches) are clearly higher in the TLEC approach than in the IV approach.

Table 3: Comparison of the different preprocessing and synchronization approaches (a), and the different modeling and synchronization approaches (b) under study using the NSD values. NSD: normalized squared differences between the average and standard deviations vectors (a) and between the first loading vector (b) of the two simulated data sets.

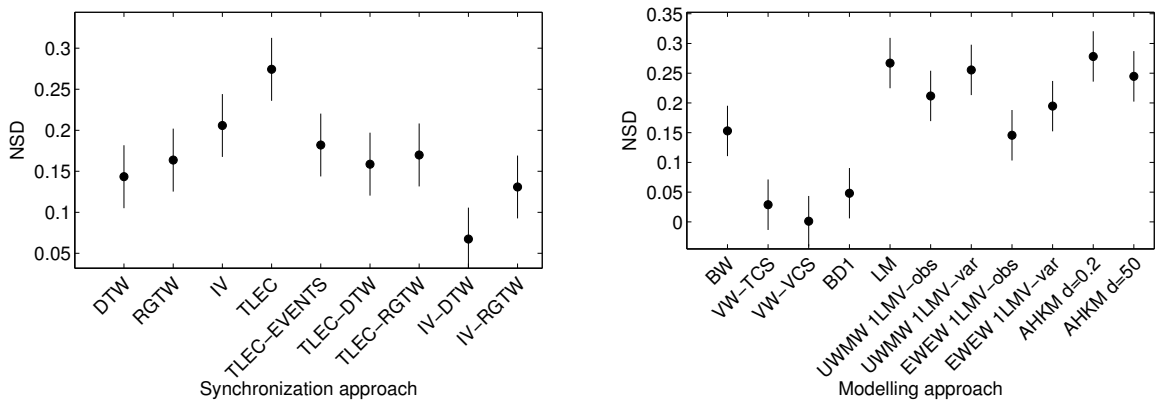
	Trajectory C&S		Variable C&S	
Synchronization method	Mean	Standard deviation	Mean	Standard deviation
IV-RGTW	3.804e-04	1.989e-02	8.404e-06	2.196e-05
IV-DTW	1.739e-03	2.546e-02	2.613e-05	1.697e-04
IV	3.383e-04	1.695e-02	7.508e-06	3.022e-05
TLEC-RGTW	2.925e-05	1.856e-02	8.181e-06	8.363e-06
TLEC-DTW	4.184e-05	2.127e-02	7.630e-06	9.321e-06
TLEC	1.600e-03	3.162e-02	3.065e-04	8.980e-05
TLEC-events	1.308e-04	3.493e-02	9.388e-06	1.493e-05
RGTW	7.012e-05	2.000e-02	7.412e-06	5.007e-06
DTW	3.010e-05	2.077e-02	6.958e-06	3.022e-06

(a)

	Single Model					Multi-Model				Hierarchical-Model	
Synchronization method	BW	VW-TCS	VW-VCS	BD1	LM	UWMW 1LMV-var	UWMW 1LMV-obs	EWEW-var	EWEW-obs	AHKM $d = 0.2$	AHKM $d = 50$
IV-RGTW	7.906e-02	8.458e-03	8.289e-05	1.753e-03	9.671e-02	1.305e-01	8.401e-02	1.374e-01	4.143e-02	3.912e-01	9.672e-02
IV-DTW	8.311e-02	1.034e-02	2.349e-04	2.141e-02	9.782e-02	1.871e-01	7.686e-02	1.196e-01	2.904e-02	1.738e-02	9.786e-02
IV	1.033e-01	2.037e-01	3.144e-05	3.388e-01	2.038e-01	2.431e-01	1.975e-01	1.932e-01	9.886e-02	4.769e-01	2.038e-01
TLEC-RGTW	1.400e-01	1.532e-02	4.749e-06	3.071e-02	3.297e-01	2.822e-01	2.616e-01	1.643e-01	1.534e-01	1.615e-01	3.296e-01
TLEC-DTW	1.439e-01	4.960e-03	5.598e-06	8.606e-03	3.145e-01	2.791e-01	2.248e-01	1.649e-01	1.241e-01	1.661e-01	3.144e-01
TLEC	3.551e-01	6.736e-03	9.028e-03	1.428e-02	3.478e-01	3.244e-01	3.119e-01	3.433e-01	3.004e-01	6.571e-01	3.477e-01
TLEC-events	1.747e-01	2.398e-03	1.265e-04	3.866e-03	3.338e-01	3.177e-01	2.911e-01	2.879e-01	6.504e-02	2.912e-01	2.337e-01
DTW	1.454e-01	1.934e-03	8.069e-06	3.301e-03	3.199e-01	2.719e-01	2.159e-01	1.703e-01	5.817e-02	1.712e-01	2.197e-01
RGTW	1.524e-01	5.137e-03	3.699e-06	1.019e-02	3.582e-01	2.640e-01	2.405e-01	1.715e-01	6.858e-02	1.706e-01	3.580e-01

(b)

Comparing SCT-based synchronization and re-synchronization, some findings are worth being highlighted. No important differences are found between the standard deviations derived from batch data after applying an SCT-based synchronization (see black empty circles and dots Figure 2(d)) and those derived after applying an SCT-based synchronization in trajectories already synchronized by TLEC (see blue empty circles and dots Figure 2(d)). In contrast, notable differences are observed between the resulting standard



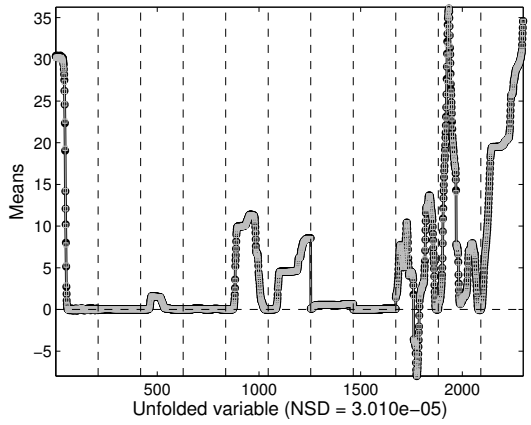
(a) (b)

Figure 3: LSD intervals (95 % confidence) for the NSD values estimated from the first loading vector of both data sets for (a) the synchronization method and (b) the modeling approach.

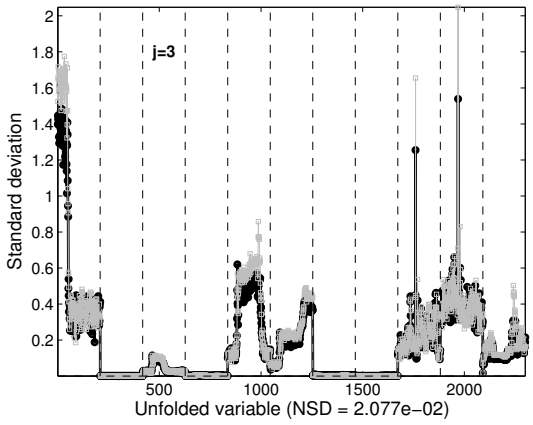
deviations after an SCT-based synchronization in trajectories already synchronized by IV (see red empty circles and dots Figure 2(d)) and the other synchronization approaches. These differences are originated by the change of shape of the original batch trajectories, which are caused by the IV-based synchronization. Consequently, the associated standard deviations differ from those obtained after applying an SCT-based synchronization to either the original or the trajectories already synchronized by TLEC. It leads to differences that are not comparable each other.

An Analysis of Variance (ANOVA) was performed on the NSD values of each preprocessing parameter i.e. mean and standard deviation- (see Table 3(a)) using the preprocessing and synchronization approach as factors. The objective of this analysis is to determine if there exist statistical differences among approaches in stability. The outcome of the ANOVA on the means suggests that the simple effect of the preprocessing is statistically significant (p-value = 0.06). In terms of parameter stability, Variable C&S shows better results (statistically lower NSD values on average, $NSD_{mn,VCS}$ = 4.312e-05) in comparison to Trajectory C&S (statistically higher NSD values on average, $NSD_{mn,TCS}$ = 4.844e-04). The ANOVA on the standard deviations yielded that the simple effect of the preprocessing approach is statistically significant (p-value $<$ 0.05). The NSD values corresponding to Variable C&S are statistically lower on average ($NSD_{std,VCS}$ = 3.918e-05) than those from Trajectory C&S ($NSD_{std,TCS}$ = 2.329e-02), showing an outperformance of the former compared to the latter in terms of stability. Note that the uncertainty in the preprocessing parameters is inherited in the loadings (see Table 3(b)). This will be discussed in detail in next section.

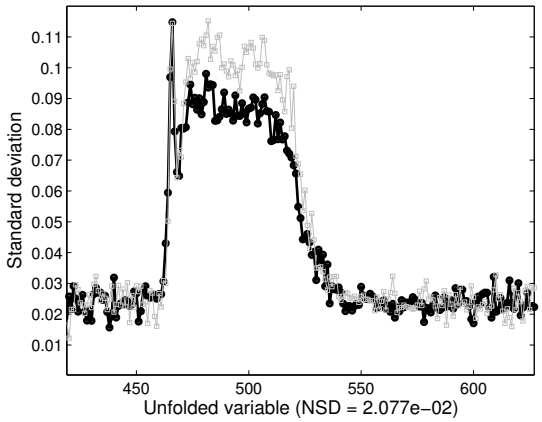
In order to check if there are statistically differences among modeling and synchronization approaches, an ANOVA was performed on the NSD values of the PCA modeling parameters -i.e. first loading vector- (see Table 3(a)). This yielded that both, both the effects of the synchronization and the modeling approach are statistically significant (p-value $<$ 0.05). In order to find out specific differences, the 95% confidence Least Significant Differences (LSD) intervals are computed (see Figure 3). The NSDs corresponding to batch data synchronized by the group of SCT-based methods are statistically lower on average (NSD_{DTW} = 1.434e-01 and NSD_{RGTW} = 1.635e-01) than those synchronized by using TLEC-based method (NSD_{TLEC} = 2.743e-01). The TLEC method is also outperformed by TLEC-events (statistically lower NSD values on average, $NSD_{TLEC-events}$ = 1.820e-01). Re-synchronization with SCT-based methods provides statistically significant improvements for both TLEC (statistically lower NSD values on average: $NSD_{TLEC-DTW}$ = 1.587e-01 and $NSD_{TLEC-RGTW}$ = 1.700e-01 in comparison with NSD_{TLEC} = 2.743e-01) and IV (statistically lower NSD values on average: NSD_{IV-DTW} = 6.734e-02 and $NSD_{IV-RGTW}$ = 1.309e-01 in comparison with NSD_{IV} = 2.057e-01). Hence, the better the key process events are synchronized, the higher stability in the



(a) Means



(b) Standard deviations (SD)



(c) SD: zoom corresponding to Variable #3 (acetaldehyde concentration)

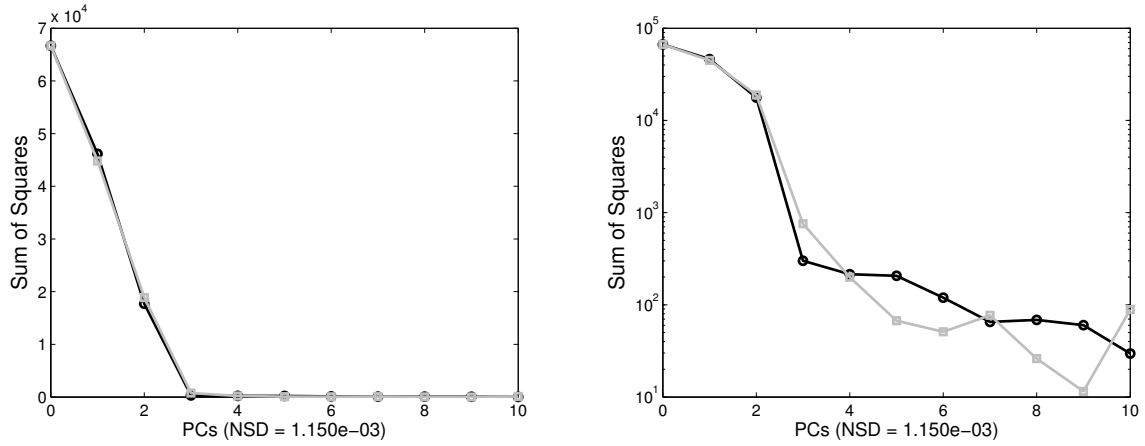
Figure 4: Batch-wise unfolding and Trajectory C&S. Comparison of the preprocessing parameters (means and standard deviations) obtained from the two simulated data sets batch-wise unfolded, after applying Trajectory C&S. NSD: normalized squared differences between the average and standard deviations vectors of the two simulated data sets.

loadings. Finally, similar results in terms of parameter stability are found for RGTW and DTW. Therefore, the RGTW algorithm seems to be an adequate procedure to be used both in real-time and end-of-batch process monitoring in terms of parameter stability.

From these results, the application of other SCT-based synchronization methods (*e.g.* [21, 22]) may deserve further research.

5. Effect of the rearranging methods on parameter stability

In this section, the study of the parameter stability associated to the most used rearranging methods is carried out. The discussion on the single-model approaches -*i.e.* BW, VW and BD- is introduced in Subsections 5.1, 5.2 and 5.3, respectively. In addition, the study on the K-model approaches -*i.e.* LM, UWMW and EWEW- is presented in Subsection 5.4. Finally, the parameter stability of the hierarchical approaches



(a) Linear scale

(b) Logarithmic scale

Figure 5: Batch-wise unfolding and Trajectory C&S. Explained Sum of Squares (SS) vs #PCs extracted in batch-wise unfolding from the two simulated data sets. Note that PC #0 corresponds to the sum of squares remained after applying Trajectory C&S on batch data. NSD: normalized squared differences between the sum of squares vector captured by each PC of the two simulated datasets.

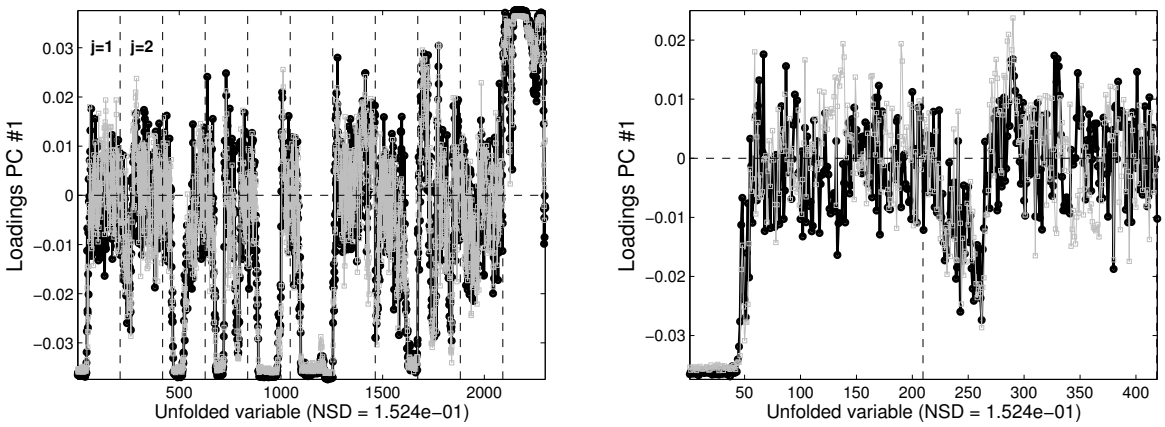
-*i.e.* AHKM- is studied in Subsection 5.5. For the sake of simplicity, the two data sets synchronized by using the RGTW algorithm are used.

5.1. Batch-wise unfolding

As was stated in Section 1, parameter stability depends on two main factors. Firstly, precise identification relies on the availability of a sufficiently large calibration data set. Secondly, the more different the sum-of-squares captured by each PC, the more stability in the model parameters [38].

The first question is the amount of calibration data which is enough to identify the parameters accurately. In Figure 4, the preprocessing information (*i.e.*, means and standard deviations) corresponding to the two data sets generated is compared. At first glance, the preprocessing parameters identified seem to be identical. Nonetheless, the zoom performed in Figure 4(c) shows that there are slight differences. The reason for this is that a high number of means ($J \cdot K$) and standard deviations ($J \cdot K$) is identified using only I batches, which is in principle a low number compared to the number of estimated parameters. For instance, in the present example, $J \cdot K = 2090$ means and standard deviations are computed from $I = 30$ batches. This uncertainty can be also checked by the NSD values computed for the means and the standard deviations: 3.010e-05 and 2.077e-02, respectively. As can be seen, there is variability in the preprocessing statistics between the two data sets, being lower in the mean than in the standard deviations. When the standard deviations are computed, the uncertainty from the mean is inherited. Hence, the resulting uncertainty is higher due to the accumulation of variability in the preprocessing parameters.

Concerning the second factor, if the sum-of-squares extracted in each PC is different enough in comparison to subsequently extracted PCs, low uncertainty in the parameters estimation is expected for a large calibration data set. In some situations and for some applications, it is not a problem to have several PCs with a similar amount of sum-of-squares captured. All of them can be included in the PCA model, or otherwise discarded and left in the residuals. Nonetheless, it is important to be aware of the uncertainty introduced in the loadings when this occurs. For this reason, it is always recommended to have a look at the sum-of-squares captured by each PC. Figure 5 shows the plot of the explained sum-of-squares vs #PCs extracted for the current example assuming batch-wise unfolding and Trajectory C&S. For the sake of visualization, both the linear and logarithmic scales are presented. As it can be seen, the sum of squares captured by PC#1 ($SS_1 \approx 4.500\text{e} + 04$) explains a high percentage of the sum of squares remained after



(a) Complete loading vectors

(b) Zoom corresponding to Variables #1 and #2

Figure 6: Batch-wise unfolding and Trajectory C&S. Comparison of the loading vector corresponding to the first PC obtained from the two simulated data sets batch-wise unfolded, after applying Trajectory C&S. NSD: normalized squared differences between the first loading vector of the two simulated data sets.

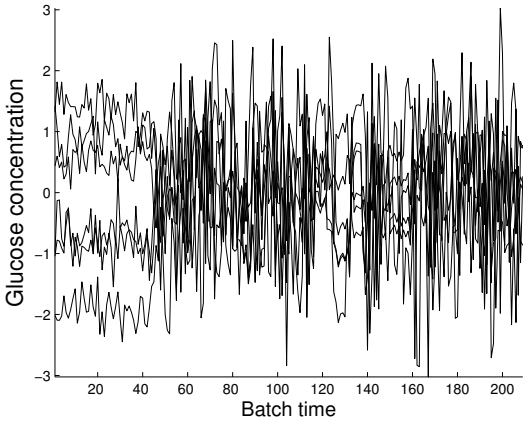


Figure 7: Batch-wise unfolding and Trajectory C&S. Trajectory of Glucose concentration (variable #1) after trajectory centering and scaling in some of the batches of the second simulated data set.

applying Trajectory C&S on batch data, *i.e.* SS_0, (approximately 70%) and differs enough from that captured by PC#2 ($SS_2 \approx 1.800\text{e} + 04$). Consequently, the loadings of the first PC are expected to be stable. Note that the sum of squares captured from PC#3 onwards are similar and, therefore, their corresponding loadings are not expected to be stable. In the present investigation, we will only focus on those PCs which are expected to be stable in order to draw conclusions about the effect of applying one specific BMSPC method in the stability of the parameters. Thus, parametric instability motivated by a specific BMPSC structure is distinguished from that due to PCs with similar captured variance, which is expected to affect the PCA models independently of the BMSPC method of choice.

There is a comment in due regarding the use of a plot like the one in Figure 5 to check for stability of the model parameters. The sum-of-squares in the curves are a pool of the data corresponding to the different sampling times and process variables. Nonetheless, this pool may not be representative of some parts of the data and should be checked with the loading vectors and, subsequently, with the raw batch trajectories.

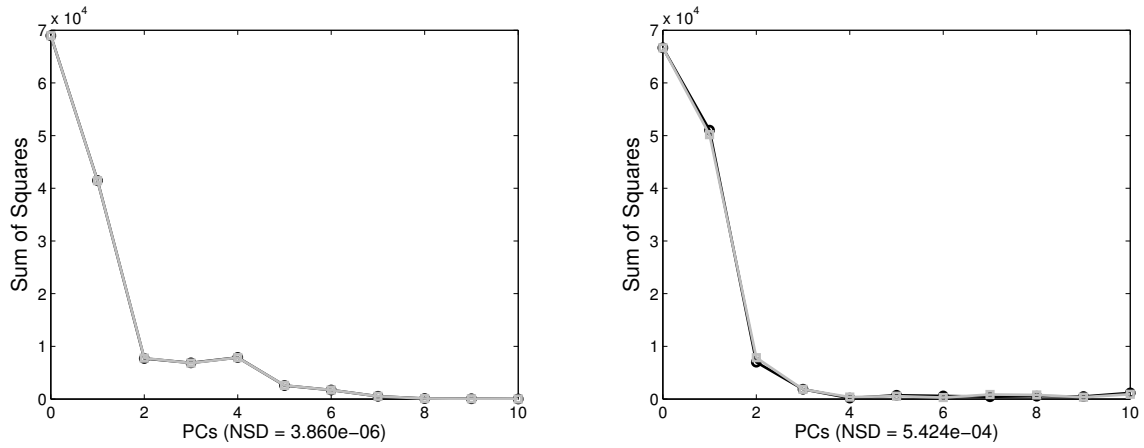
In Figure 6, the two loading vectors corresponding to the first PC obtained for the two data sets generated are shown. Inaccuracies in the preprocessing estimation are inherited in the identification of the PCs. In particular, the NSD value corresponding to the first loading vector of both data sets is equal to 1.524e-01, denoting an increasing instability with respect to the preprocessing parameters. Furthermore, each PC contains $J \cdot K$ parameters, the same number of means or standard deviations estimated previously. The parameters are, again, estimated from I observations each. It is clear that there is a parallelism between trajectory centering and scaling, and batch-wise unfolding from the point of view of uncertainty estimation. In the zoom of Figure 6(b), the loadings corresponding to the glucose concentration (variable #1) and pyruvate concentration (variable #2) are shown. Several loadings have such uncertainty that they present different sign for the two data sets. Nonetheless, most of this variability is due to noise since most of loadings take values around zero (*e.g.*, from the 60th to the 209th loading and from the 260th to the 418th loading belonging to variable #1 and #2, respectively, see Figure 6(b)). Despite the fact that with batch-wise unfolding a very complete modeling structure can be estimated [6], the noisy loadings shown in Figure 6 suggest model over-parametrization (*i.e.* overfitting). In any case, important parts are captured. For instance, the loadings of high magnitude in the interval [1,50] are reflecting the high auto-correlation of the first variable (glucose concentration) during that period in the aligned data sets (see Figure 7).

5.2. Variable-wise unfolding

As already discussed, a factor where the parameter stability relies on is the amount of observations used in the parameter estimation. In Variable C&S, a total of J means and J standard deviations are identified using $I \cdot K$ observations. In this example, $J = 10$ and $I \cdot K = 6270$. Due to the fact the number of parameters-to-the number of observations ratio in Trajectory C&S ($R_{TCS} = \frac{J \cdot K}{I} = \frac{2090}{30}$) is much higher than in Variable C&S ($R_{VCS} = \frac{J}{I \cdot K} = \frac{10}{6270}$), the uncertainty in the estimation in the former is also higher than in the latter. This was also observed in the results of Section 4.

In Figure 8, the explained sum-of-squares vs #PCs extracted for the current example assuming variable-wise unfolding and Trajectory C&S (see Figure 8(a)) and Variable C&S (see Figure 8(b)) are shown. Note that the sum-of-squares at PC#0 remaining after Variable C&S for both data sets (SS_0 =6.896e+04) is slightly higher than after Trajectory C&S (SS_0 =6.667e+04). This has nothing to do with stability and it is due to the different type of preprocessing carried out. In the former, the remaining sum-of-squares is equal to $SS_0 = (I \cdot K - 1) \cdot J = (30 \cdot 209 - 1) * 11 = 6.896\text{e}{+04}$ units whereas in the latter is equal to $SS_0 = (I - 1) \cdot K \cdot J = (30 - 1) * 209 * 11 = 6.667\text{e}{+04}$.

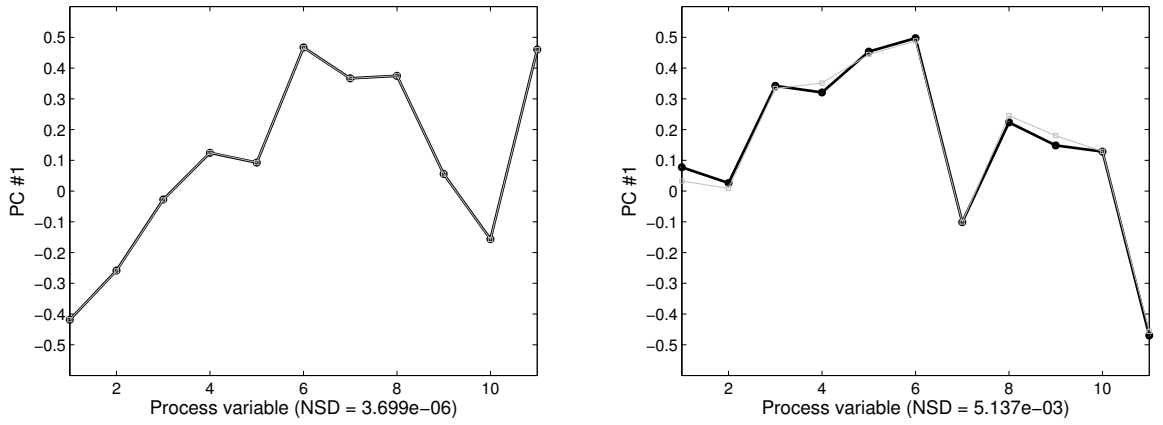
Again, the model parameter stability is studied by assessing how different the sum-of-squares captured by each PC are. Firstly, in the case of VW with Variable C&S (Figure 8(a)), the sum of squares captured by PC#1 ($SS_1 \approx 4.145\text{e} + 04$) explains a high percentage of the sum of squares remained after applying Variable C&S on batch data, *i.e.* SS_0, (approximately 60%) and it is different enough to that captured by PC#2 ($SS_2 \approx 7.700\text{e} + 03$). Consequently the loadings of the first PC are expected to be stable. The sum of squares of PC#2, PC#3 and PC#4 seem to be quite similar and therefore their loadings may not be stable. Notice that this result is specific of the data set at hand, and not a feature of the modeling and/or preprocessing method. Secondly, in the case of VW with Trajectory C&S (Figure 8(b)), the sum of squares



(a) Variable C&S (b) Trajectory C&S

Figure 8: Variable-wise unfolding. Explained Sum of Squares (SS) vs #PCs extracted in variable-wise unfolding from the two simulated data sets. Note that PC #0 corresponds to the sum of squares remained after applying Variable C&S (a) and Trajectory C&S (b). NSD: normalized squared differences between the sum of squares vector captured by each PC of the two simulated datasets.

captured by PC#1 ($SS_1 \approx 5.100\text{e} + 04$)) explains a high percentage of the sum of squares remained after applying Trajectory C&S on batch data, *i.e.* SS_0, (approximately 75%) and again it is different enough to that captured by PC#2 ($SS_2 \approx 7.500\text{e} + 03$). As a consequence, the loadings of the first PC are expected to be stable. Note that the sum of squares captured from PC#2 onwards are similar, so their corresponding loadings are not expected to be stable. Also, uncertainty measured in the residual sum-of-squares by PC of each of the VW models through the NSD values (NSD=3.860e-06 and NSD=5.424e-04 for VW-Variable C&S and for VW-Trajectory C&S, respectively) confirms that Variable C&S outperforms Trajectory C&S in terms of parameter stability.



(a) Variable C&S (b) Trajectory C&S

Figure 9: Variable-wise unfolding. Comparison of the loading vector corresponding to the first PC obtained from the two simulated data sets. NSD: normalized squared differences between the first loading vector of the two simulated data sets.

In order to compare the stability of the first PC, the corresponding loadings for both preprocessing methods are shown in Figure 9. In terms of NSD, the uncertainty observed in the loadings after Variable C&S (NSD=3.699e-06) is approximately three orders of magnitude lower than after Trajectory C&S (NSD=5.137e-03). It is inherited from a similar difference in the uncertainty of the preprocessing parameters. Hence, stability of the loadings of PC#1 in Variable C&S is higher than in Trajectory C&S.

The results in terms of stability should be interpreted with care and in connection with other features of the models, as those discussed in the companion papers [6, 7]. It should be remarked that the parameters present low uncertainty does not guarantee the model is adequate. Note that the variability of interest in BMSPC is the deviation of a batch from the common trend (*e.g.* the average trajectory) of the process. When the average trajectory is not extracted in the preprocessing, like in Variable C&S, the associated variability remains in the data. If the data are subsequently unfolded VW, that specific variability turns into non-linear relationships which cannot in general be captured with a linear model, such as PCA. Therefore, VW after Variable C&S is not suited to capture the variability of interest in BMSPC.

5.3. Batch-dynamic unfolding

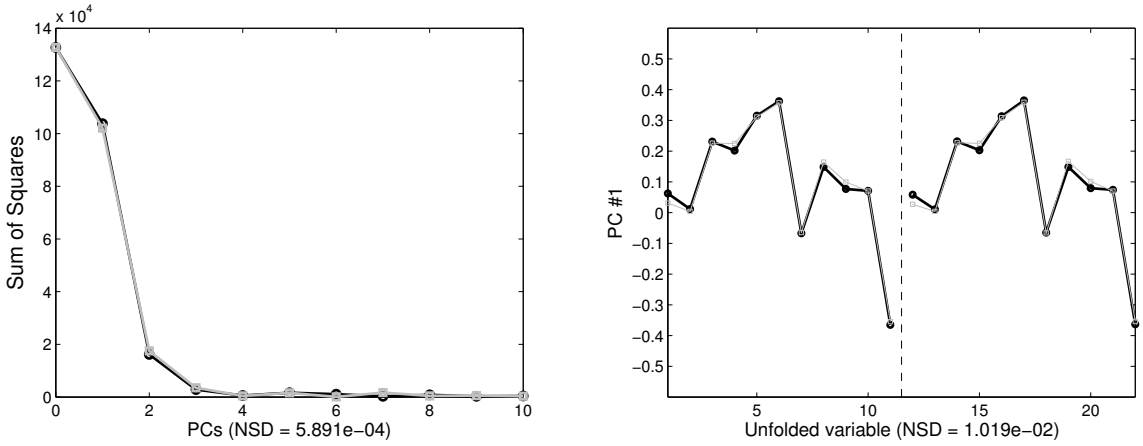
Figure 10 shows the explained sum of squares vs #PCs and the loading vectors corresponding to the first PC for the two data sets after batch-dynamic unfolding with 1 LMV and Trajectory C&S. These results are quite similar to those obtained for variable-wise unfolding and Trajectory C&S. Hence, Figures 10(a) and 8(b) present a very similar shape, being the main difference that the former doubles the latter in explained sum-of-squares. This is the logical consequence of doubling the number of variables by adding one LMV. Also, Figures 10(b) and 9(b) present essentially the same relationships among variables, but again the former shows these relationships twice. Concerning the loadings stability, this approach yields an intermediate uncertainty between variable-wise and batch-wise unfolding. In particular, variability in batch-dynamic is lower (NSD=1.019e-02) than in batch-wise after Trajectory C&S (NSD=1.524e-01) and, higher than in variable-wise after Trajectory C&S (NSD=5.137e-03) and after Variable C&S (NSD=3.699e-06). This result is expected since batch-dynamic is a generalization of variable-wise and batch-wise (its number of parameter-to-number of observation ratio is higher than variable-wise, but lower than batch-wise). Figure 10(b) also shows that the auto-correlation in the data is so high that the loadings for one variable and its lagged version are almost identical.

5.4. K-models

Figure 11 displays the loading vectors of the first PC for a) a local model, b) a UWMW model with 1 LMV in the variables, c) a UWMW model with 1 LMV in the observations, d) an EWEW model with LMVs in the variables and $\lambda = 0.97$, and e) an EWEW model with LMVs in the observations and $\lambda = 0.97$. All the models shown correspond to sampling time $k = 10$ in the data sets and in all the cases data were Trajectory C&S.

Essentially, the instantaneous relationships captured in the models are the same (*i.e.* the loading vector profiles are basically similar). Nevertheless, this does not necessarily has to generalize for other processes or numbers of LMV. In Figure 11, the NSD between the loadings corresponding to both data sets are also included. As previously discussed, in the approaches where 1 or all the possible LMVs are added as new variables, the NSD value is computed on the loading vector defined by the last J loadings (corresponding to the the k-th current sampling time) instead of all the loadings (like in the single-model approaches). This is done to make comparison between K-model approaches and with the rest of possible approaches.

Comparing the addition of LMVs as new variables with the addition of LMVs as new observations both in UWMW and EWEW, the former presents higher uncertainty (NSD_{UWMW}=2.640e-01 and NSD_{EWEW}=1.715e-01) than the latter (NSD_{UWMW}=2.405e-01 and NSD_{EWEW}=6.858e-02) (see Figure 11(b) and Figure 11(d) in comparison with Figure 11(c) and Figure 11(e)). Hence, when LMV are added as new variables, there is a negative effect in terms of parameter fitting as a consequence of increasing the number of parameters to be estimated. This means that adding new parameters–adding LMV as new variables–affects negatively the estimation of the parameters already in the model–those for instantaneous correlations (*i.e.* for the loadings corresponding to the current sampling point). On the other hand, adding LMV as new observations has



(a) (b)

Figure 10: Batch-dynamic unfolding and Trajectory C&S. Explained Sum of Squares (SS) vs #PCs extracted (a) and first loading vector (b) in batch dynamic unfolding with 1 LMV from the two simulated data sets. Note that PC #0 corresponds to the sum of squares remained after applying Trajectory C&S on batch data. NSD: normalized squared differences between (a) the sum of squares vector captured by each PC and (b) the first loading vector of the two simulated data sets.

a positive effect in the parameter stability in such a way that it reduces the uncertainty on parameters estimation, as a consequence of increasing the number of observations to estimate each parameter.

It should be noted that the local models show a higher NSD than the EWEW-var and UWMW-var approaches, for the present data set and the metaparameters selected (number of LMV and λ). This can be explained by the fact that autocorrelation and lagged cross-correlation has also a smoothing effect on loadings, which reduces the uncertainty. A similar effect can be seen by comparing the NSD of the loadings corresponding to the first PC for BW ($NSD = 1.524e-01$, see Figure 6) and local models. In both cases, a total of $J \cdot K$ parameters are estimated from the data of I batches. However, a BW PCA model takes into account the autocorrelation and lagged cross-correlation to improve the model estimation, while local PCA models do not. The result is a lower uncertainty in the former than in the latters. Therefore, the inclusion of LMVs as variables has a double and contradictory effect on the uncertainty. Generally speaking, the increase in the number of parameters augments the uncertainty. This happens unless that increase is justified by a high level of correlation in the data. This supports the claim that the approach for transforming 3-way data in 2-way should be selected depending on the data at hand [39]

5.5. Adaptive K-models

Firstly, the identification of the PCA model parameters is studied through the sum of squares captured for each PC. Figure 12 shows the explained sum-of-squares (SS) vs #PCs for an adaptive hierarchical K-models (AHKM) approach by using weighting factors $d = 0.2$ (see Figure 12(a)) and $d = 50$ (see Figure 12(b)). Weighting factor d is used to give less or more importance to the information collected at the current sampling time with regard to the past information. This factor plays the same role as the exponential weighting factor in an EWMA model [33]. For low values of d, the adaptation of the model is slow, while for high values of d, the adaptation is fast. For values of d close to 0, the adaptive hierarchical K-model approach uses memory of the past information and, therefore, this approach becomes similar to batch-wise unfolding. As d grows further than one, the adaptive hierarchical K-model approach converges to the local K-models approach since the adaptive model down-weights the memory of any previous information. In the PCA with $d = 0.2$ (see Figure 12(a)), the sum-of-squares captured by PC#1 ($SS_1 \approx 4.550\text{e} + 04$) explains roughly 70% of the sum of squares remained after applying Trajectory C&S on batch data, *i.e.* SS_0, and differs

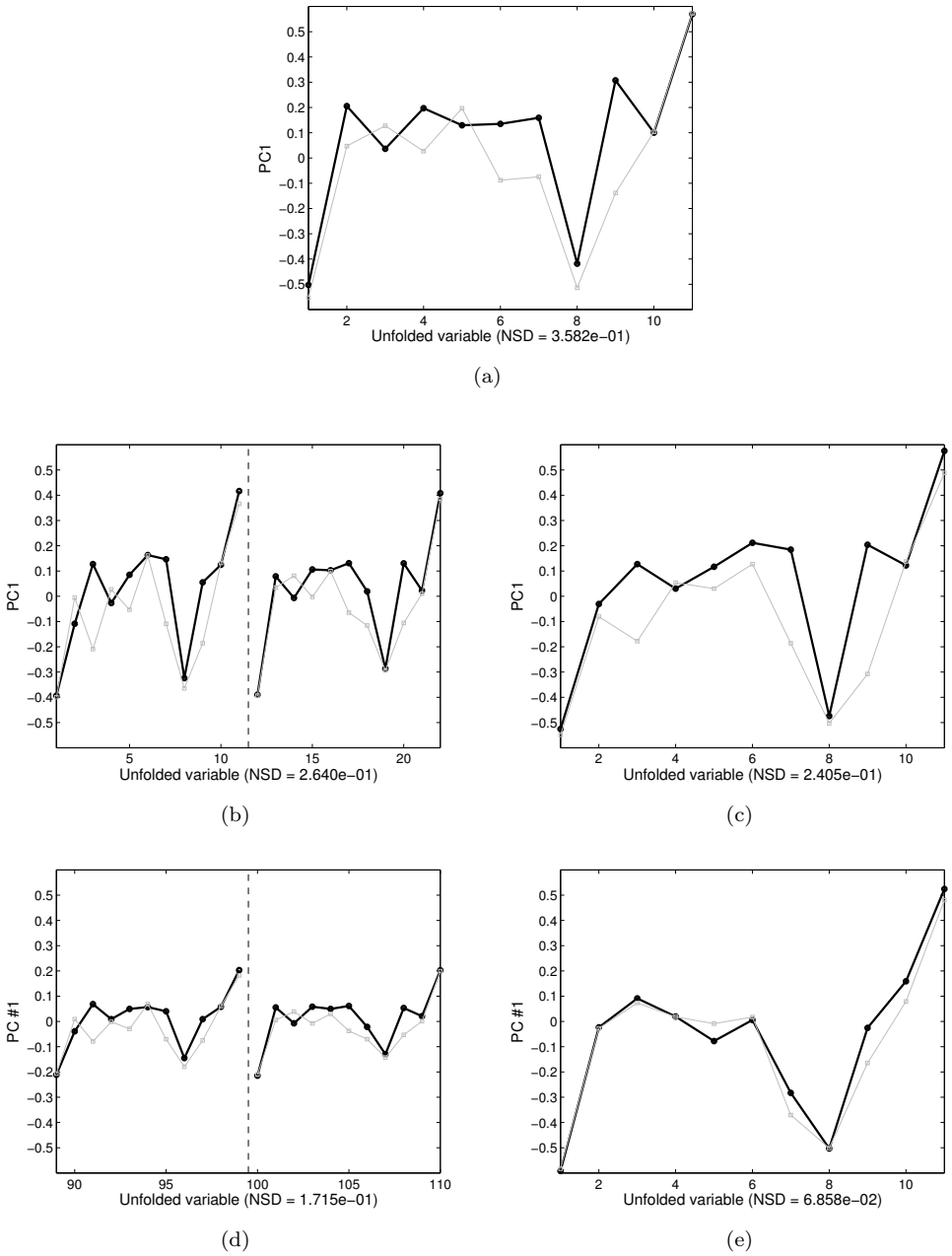
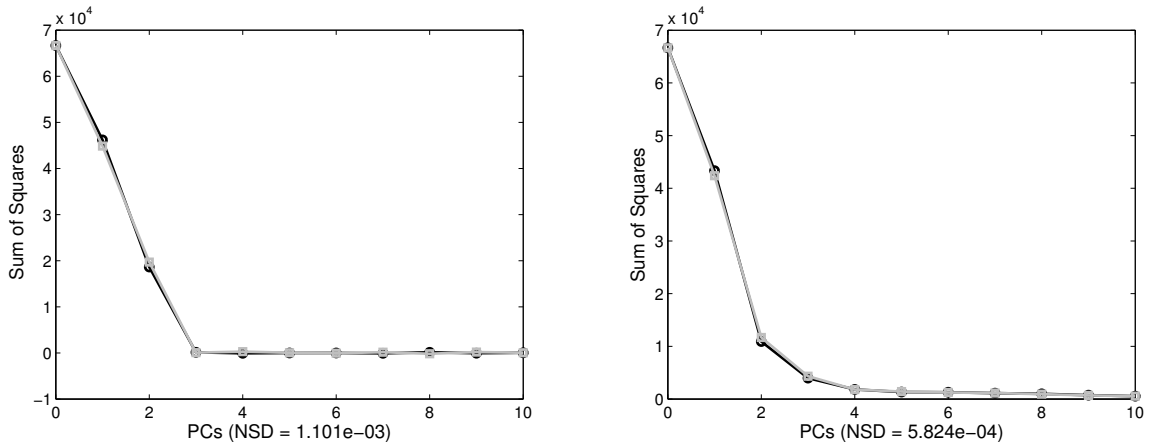


Figure 11: Multi-models and Trajectory C&S. First loading vector for the two data sets at the k-th sampling time: (a) local model, (b) UWMW model with 1 LMV in the variables, (c) UWMW model with 1 LMV in the observations, (d) EWEW model with LMVs in the variables and $\lambda = 0.97$ (only the loadings corresponding to the $k-$ and $(k-1)$-th sampling time are shown for the sake of comparison) and (e) EWEW model with LMVs in the observations and $\lambda = 0.97$. NSD: normalized squared difference between the first loading vector of the two simulated data sets. In this approach, NSD is estimated as the average of the NSD values calculated at each k sampling point.



(a) (b)

Figure 12: Adaptive multi-models and Trajectory C&S. Explained Sum of Squares (SS) vs #PCs extracted in the adaptive hierarchical K model (AHKM) with (a) $d = 0.2$, and (b) $d = 50$. Note that PC #0 corresponds to the sum of squares remained after applying Trajectory C&S on batch data. NSD: normalized squared differences between the sum of squares vector captured by each PC of the two simulated datasets

.

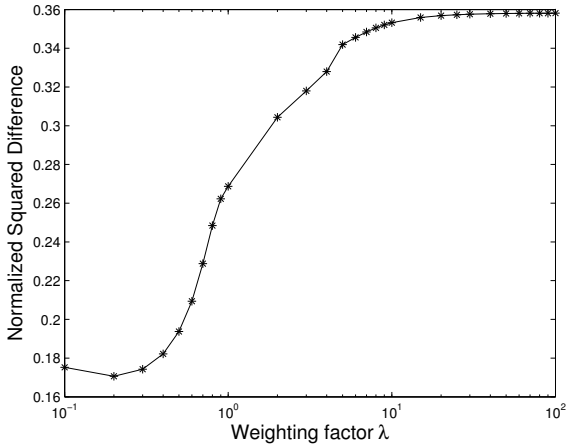


Figure 13: Adaptive K-models and Trajectory C&S. Normalized squared differences (NSD) between the first loading vector of the two data sets as a function of the weighting factor d.

enough to that captured by PC#2 ($SS_2 \approx 1.920\text{e}+04$). Hence, the corresponding loadings are expected to be stable, like in batch-wise unfolding (see Figure 5 for comparison). Again, from PC#3 onwards, the explained sum of squares are similar and, consequently, their corresponding loadings are not expected to be stable. Regarding the AHKM model with $d = 50$, a progressive decay of the explained sum of squares as a function of the number of PCs can be observed (see Figure 12(b)). The sums-of-squares captured by the first 2 PCs ($SS_1 \approx 4.290\text{e}+04$ and $SS_2 \approx 1.130\text{e}+04$) differ each other enough to consider the loadings of the corresponding PCs stable. In contrast, the sum of squares captured from PC#3 onwards are similar, so their corresponding loadings are not expected to be stable.

With the aim of studying the effect of the weighting factor in terms of parameter stability, AHKM was performed for the two data sets varying the weighting factor from $d = 0.1$ (roughly non-adaptive model) to

$d = 100$. The corresponding NSD computed for the loadings of the first PC are shown in Figure 13. As can be seen, AHKM using $d = 0.2$ reduces the differences found between the loadings of the first PC for the two data sets ($NSD = 1.706\text{e-}01$). Note that the value of d that minimizes parameter stability may not be the same for different data sets and/or number of PCs. Another fact worth being highlighted is that the differences among loadings obtained for the two data sets are stabilized for $d \geqslant 20$ (e.g. $NSD = 1.706\text{e-}01$ for $d = 50$), due to the adaptive hierarchical-model approach converges to the classical local K-models approach (the curve of Figure 13 converges to the NSD value of Figure 11(a)). It apparently suggests that the lower the weighting factor, the more stable the model parameters in the first loading vector. This is coherent with the results observed for BW and local models and the discussion at the end of previous section.

6. Conclusions

This is the third paper of a series devoted to study the properties of bilinear batch modeling approaches. The first companion paper [6] presents a theoretical analysis of the principal modeling approaches focused on how the process -possibly changing- dynamics are captured. In the second companion paper [7], a comparison of several PLS modeling approaches in terms of the on-line estimation of a key variable is performed. In the present paper, the importance of parameter stability in PCA-based BMSPC is addressed. To obtain accurate PCA models for process monitoring, low variability (*i.e.* stability) on the model parameters is desired. The existence of uncertainty in both the preprocessing statistics and the latent variables yields a considerable amount of noise in the model that may affect the performance of the monitoring systems in terms of fault detection and diagnosis.

Parameter stability depends on the synchronization method, the type of preprocessing performed in batch data, and the type of model and unfolding used to transform the 3-way data structure to 2-way. More specific conclusion in these issues are drawn below:

- Synchronization. Accuracy in batch synchronization has been proved to have a profound impact on the loadings stability. The group of SCT-based methods (DTW and RGTW) outperforms the group of TLEC-based methods (TLEC and TLEC-events) in terms of synchronization quality, *i.e.* accuracy in synchronizing the key process events. Also, SCT-based methods outperform the rest of synchronization techniques in terms of stability in the loadings. Hence, the better the synchronization of key process events, the better the model parameter stability.

- Preprocessing. One of the factors that parameter stability depends on is the size of the calibration data set. Trajectory C&S performs a mean centering of the batch data corresponding to each j-th process variable at each k-th sampling time point. This means that JK averages and JK standard deviations are computed from I batches. In contrast, in Variable C&S a mean centering and scaling of the batch data belonging to each j-th process variable is performed. Hence, J averages and J standard deviations are computed from IK observations. Comparing both preprocessing approaches, the number of parameters-to-number of observations ratio is much higher in Trajectory C&S than in Variable C&S. As was expected, the parameter stability found in this study was lower in the former than in the latter.

- Rearranging method. Uncertainty found in the preprocessing parameters is directly inherited in the loadings, decreasing their stability. Depending on the type of rearranging method performed on the 3-way batch data matrix, this uncertainty is considerably changed. Those methods that introduce more variables in the model (BW, BD, UWMW and EWEW in its variable-wise version, and AHKM, being the latter a particular case due to its adaptive nature) showed less stability in comparison to those methods that introduce more observations (VW, UWMW and EWEW in its observation-wise). As a side reserve effect, when a number of LMVs are added, the underlying autocorrelation and lagged cross-correlation in data may slightly reduce the uncertainty in the loadings, as a smoothing effect. However, in general speaking, the less LMV as new variables, the more stability in loadings.

Although this paper has been focused on the parameter stability of the different synchronization and modeling approaches, there is a paramount comment which is in due. For those modeling approaches where the number of parameters depends on the number of sampling points throughout the batch, the sampling frequency may be seen as a method to artificially modify the parametric uncertainty. Moreover, the lower the sampling frequency, the smaller the difference among modeling approaches in terms of parameter stability. This fact must not mislead practitioners in the decision-making about the modeling approach and the sampling frequency to use. Also, the fact that the parameters present low uncertainty does not guarantee the corresponding model is adequate for the specific process at hand and the model goal. For instance, Variable C&S, although yielding stable parameters, is not focused on the source of variability of interest in BMSPC (the deviation from the common trend). In addition, models with a low number of LMV may provide poor prediction performance. Hence, the modeling approach must not be selected from the consideration of the parameter stability alone. The findings of the present paper need to be combined with those from the companion papers for a proper choice. Finally, note that the case study presented is limited to a specific batch process, the fermentation of the *Saccharomyces cerevisiae*.

This series of papers have studied three critical factors in the design of accurate monitoring/prediction schemes: the source of variability remaining after preprocessing, process dynamics and parameter stability. The setting of these factors should be balanced in such a way that PCA and PLS models are accurate in fault detection and diagnosis and/or in on-line prediction.

Acknowledgements

This research work was partially supported by the Spanish Ministry of Economy and Competitiveness under the project DPI2011-28112-C04-02. Authors also acknowledge anonymous reviewers for their comments to improve the article.

APPENDIX: LIST OF ABREVIATIONS

AHKM	Adaptive Hierarchical K model
BD	Batch dynmic unfolding
BD1	Batch dynamic unfolding adding 1 lagged measurement vector
BMSPC	Batch Multivariate Statistical Process Control
BW	Batch-wise unfolding
C&S	Centring and Scaling
DTW	Dynamic Time Warping
EWEW	Exponentially Weighted Evolving Window
EWEW-obs	Exponentially Weighted Evolving Window in the observation domain
EWEW-var	Exponentially Weighted Evolving Window in the variable domain
ISL	Imposed Significance Level
IV	Indicator Variable
IV & SCT	Synchronization performed using a SCT-based method after synchronizing batch data with IV
IV-DTW	DTW-based synchronization after performing a IV-based synchronization
IV-RGTW	RGTW-based synchronization after performing a IV-based synchronization
LM	Local K-model
LMV	Lagged Measurement Vector
NOC	Normal Operating Conditions
NSD	Normalized Squared Difference
NSD_{DTW}	average NSD values for the DTW synchronization approach
NSD_{EWEW}	NSD values for the EWEW model
NSD_{IV}	average NSD values for the IV synchronization approach
NSD_{IV-DTW}	average NSD values for the combined IV-DTW synchronization approach

$NSD_{IV-RGTW}$	average NSD values for the combined IV-RGTW synchronization approach
$NSD_{mn\|TCS}$	average NSD value estimated for means in Trajectory C&S
$NSD_{mn\|VCS}$	average NSD value estimated for means in Variable C&S
NSD_{RGTW}	average NSD values for the RGTW synchronization approach
$NSD_{std\|TCS}$	average NSD value estimated for standard deviations in Trajectory C&S
$NSD_{std\|VCS}$	average NSD value estimated for standard deviations in Variable C&S
NSD_{TLEC}	average NSD value for the TLEC synchronization approach
$NSD_{TLEC-events}$	average NSD value for the TLEC-events synchronization approach
$NSD_{TLEC-DTW}$	average NSD value for the combined TLEC-DTW synchronization approach
$NSD_{TLEC-RGTW}$	average NSD value for the combined TLEC-RGTW synchronization approach
NSD_{UWMW}	NSD values for the UWMW model
PC	Principal Component
PCA	Principal Component Analysis
PLS	Partial Least Squares
R_{TCS}	Number of parameters-to-the number of observations ratio in Trajectory C&S
R_{VCS}	Number of parameters-to-the number of observations ratio in Variable C&S
RGTW	Relaxed Greedy Time Warping
SCT	Stretching, Compressing and Translating
SPE	Squared Prediction Error
SS	Explained Sum of Squares
TLEC	Time Linear Expanding/Compression
TLEC & SCT	TLEC-based synchronization after synchronizing batch data with a SCT-based method
TLEC-DTW	DTW-based synchronization after performing a TLEC-based synchronization
TLEC-events	TLEC-based synchronization among stages defined by key process events
TLEC-RGTW	RGTW-based synchronization after performing a TLEC-based synchronization
UWMW	Uniformly Weighted Moving Window
UWMW 1LMV-obs	Uniformly Weighted Moving Window generated by adding 1 LMV in the observations
UWMW 1LMV-var	Uniformly Weighted Moving Window generated by adding 1 LMV in the variables
VW	Variable-wise unfolding
VW-TCS	Variable-wise unfolding after Trajectory centering and scaling
VW-VCS	Variable-wise unfolding after Variable centering and scaling

References

[1] T. Kourti, Process analysis and abnormal situation detection: from theory to practice, IEEE Control Systems Magazine 22 (2002) 10–25.

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

[3] T. Kourti, Application of latent variable methods to process control and multivariate statistical process control in industry, Int. J. Adapt. Control Signal Process. 19 (2005) 213–246.

[4] S. Wold, N. Kettaneh-Wold, J. MacGregor, K. Dunn, Comprehensive Chemometrics, volume 2, Elsevier, pp. 163–195.

[5] J. Jackson, A User's Guide to Principal Components, Wiley-Interscience, England, 2003.

[6] J. Camacho, J. Picó, A. Ferrer, Bilinear modelling of batch processes. part i: Theoretical discussion, Journal of Chemometrics 22 (2008) 299–308.

[7] J. Camacho, J. Picó, A. Ferrer, Bilinear modelling of batch processes. part ii: A comparison of pls soft-sensors, Journal of Chemometrics 22 (2008) 533–547.

[8] J. González-Martínez, R. Vitale, O. de Noord, A. Ferrer, Does synchronization matter in bilinear batch process monitoring? - (submitted) –.

[9] L. Eriksson, E. Johansson, N. Kettaneh-Wold, J.Trygg, C. Wikström, S. Wold, Multi- and Megavariate Data Analysis Part I: Basic Principles and Applications, Umetrics Inc, Ume, Sweden, 2006.

[10] Umetri, Simca release 13.0.3 for windows, graphical software for multivariate process modeling, Umea, Sweden (2013).

[11] 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.

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

[13] D. Wallace, Prosensus multivariate v12. 02 (2010).
[14] D. J. Louwerse, A. A. Tates, A. K. Smilde, G. L. M. Koot, H. Berndt, Pls discriminant analysis with contribution plots to determine differences between parallel batch reactors in the process industry, Chemometrics and Intelligent Laboratory Systems 46 (1999) 197 – 206.
[15] P. Nomikos, J. MacGregor, Monitoring batch processes using multiway principal components, AICHE Journal 40 (1994) 1361–1375.
[16] 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.
[17] J. Ramsay, B. Silverman, Functional data analysis, New York:Springer-Verlag, 1997.
[18] 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.
[19] A. Kassidas, J. MacGregor, P. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE Journal 44 (1998) 864–875.
[20] J. González-Martínez, A. Ferrer, J. 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.
[21] Y. Zhang, T. F. Edgar, A robust dynamic time warping algorithm for batch trajectory synchronization, in: Proceedings of American Control Conference, pp. 2864–2869.
[22] 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.
[23] S. Gurden, J. Westerhuis, S. Bijlsma, A. Smilde, Modelling of spectroscopy batch process data using grey models to incorporate external information, Journal of chemometrics 15 (2001) 101–121.
[24] 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.
[25] J. Westerhuis, T. Kourti, J. MacGregor, Comparing alternative approaches for multivariate statistical analysis of batch process data, Journal of Chemometrics 13 (1999) 397–413.
[26] P. Nomikos, J. MacGregor, Multivariate spc charts for monitoring batch processes, Technometrics 37 (1995) 41–59.
[27] 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.
[28] J. Chen, K. Liu, On-line batch process monitoring using dynamic pca and dynamic pls models, Chemical Engineering Science 57 (2002) 63–75.
[29] H. Ramaker, E. van Sprang, J. Westerhuis, A. Smilde, Fault detection properties of global, local and time evolving models for batch process monitoring, Journal of Process Control 15 (2005) 799–805.
[30] B. Lennox, G. Montague, H. Hiden, G. Kornfeld, P. Goulding, Process monitoring of an industrial fed-batch fermentation, Biotechnology and Bioengineering 74 (2001) 125.
[31] 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.
[32] J. Camacho, J. Picó, Multi-phase principal component analysis for batch processes modelling, Chemometrics and Intelligent Laboratory Systems 81 (2006) 127–136.
[33] S. Rännar, J. MacGregor, S. Wold, Adaptive batch monitoring using hierarchical pca, Chemometrics and Intelligent Laboratory Systems 41 (1998) 73–81.
[34] J. Camacho, J. Picó, A. Ferrer, On-line monitoring of batch processes based on pca: Does the modelling structure matter?, Analytica chimica acta 642 (2009) 59–69.
[35] E. van Sprang, H. Ramaker, J. Westerhuis, S. Gurden, A. Smilde, Critical evaluation of approaches for on-line batch process monitoring, Chemical Engineering Science 57 (2002) 3979–3991.
[36] F. Lei, M. Rotbøll, S. Jørgensen, A biochemically structured model for saccharomyces cerevisiae, Journal of Biotechnology 88 (2001) 205–221.
[37] J. Camacho, J. González-Martínez, A. Ferrer, Multi-phase (mp) toolbox, http://mseg.webs.upv.es/Software.html, 2013.
[38] I. T. Jolliffe, Principal Component Analysis, Springer, 2002.
[39] J. Camacho, J. Picó, A. Ferrer, Multi-phase analysis framework for handling batch process data, Journal of Chemometrics 22 (2008) 632.