MVBatch: a Matlab toolbox for batch process modeling and monitoring

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

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

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

[c]*Multivariate Statistical Engineering Group (GIEM), Department of Applied Statistics, Operational Research and Quality, Universitat Politècnica de València, Camino de Vera s/n Edificio 7A, 46022, Valencia, Spain*

Abstract

A novel user-friendly graphical interface for process understanding, monitoring and troubleshooting has been developed as a freely available MATLAB toolbox, called the MultiVariate Batch (MVBatch) Toolbox. The main contribution of this software package is the integration of recent developments in Principal Component Analysis (PCA) based Batch Multivariate Statistical Process Monitoring (BMSPM) that overcome modeling problems such as missing data, different speed of process evolution and length of batch trajectories, and multiple stages. An interactive user interface is provided, which aims to guide users in handling batch data through the main BMSPM steps: data alignment, data modeling, and the development of monitoring schemes. In addition, a small-scale non-linear dynamic simulator of the fermentation process of the *Saccharomyces cerevisiae* cultivation is available to generate realistic batch data under normal and abnormal operating conditions. This generator of synthetic data can be used for teaching purposes or as a benchmark to illustrate and compare the performance of new methods with sound techniques published in the field of BMSPM.

Keywords: Batch Multivariate Process Control, Batch Synchronization, Multi-phase Modeling, Principal Component Analysis, Monitoring, Fault Diagnosis

1. Introduction

Batch processes often exhibit batch-to-batch variation that are object of investigation, analysis and monitoring. The analysis of the available data in each batch is crucial to ensure safe operation, stable product quality and sustainable profit in batch processes [1]. There are four major objectives for analyzing batch data [2]: i) the analysis of variable trajectories from historical batches to gain process understanding and troubleshoot past abnormal operating conditions, with the ultimate goal of driving the process under Normal Operating Conditions (NOC) again; ii) the statistical process monitoring of incoming batches either after completion (post-batch) or during its progress (real-time); iii) the prediction of final product quality while the process evolves in real-time, and iv) the optimization and/or active control of batch operating conditions to reach the desired quality properties of the final product.

The design of statistical monitoring schemes involves two main phases: Phase I (process improvement and model building) and Phase II (model exploitation) [3]. In Phase I, the main objective is to assess and secure process stability [4]. For such purpose, it is required to understand the effects of varying initial and process conditions, which are associated with time-varying patterns in batch trajectories, on the performance of batches, and on the final product quality. In case there are unusual disruptions in the process, also known as assignable or special causes of variation, the process must be stabilized first, and optimized before implementing a final statistical model for monitoring purposes. With this objective, a number of modeling

*Corresponding author

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

steps are typically performed, namely i) data alignment, ii) data pre-processing and iii) transformation of the three-way array to one or several two-way arrays for the subsequent iv) bilinear batch modeling using projection methods to latent structures. When outliers are detected, the nature of the assignable causes is investigated, requiring the implementation of countermeasures if the special causes deteriorate the process performance. In contrast, if the assignable causes improve the process performance, they should be incorporated in the process [5]. Either case, new data should be collected once the process is stabilized, and subsequently analyzed following the four modeling steps. Thereafter, the understanding gained and the statistical model fitted are used to isolate and diagnose past poor operating conditions. Finally, once the process is operating stably, in-control statistical models are used to design process control systems for post-batch and/or real-time monitoring in Phase II.

This paper presents the MultiVariate Batch (MVBatch) Toolbox, a graphical user-friendly interface that integrates, on the one hand, the two phases for the design of monitoring schemes, and on the other hand, most of the recent developments in batch exploratory data analysis and monitoring. The MVBatch Toolbox addresses all the steps of the bilinear modeling cycle, from the missing data imputation and synchronization of batch trajectories, to the calibration of latent variable models based on Principal Component Analysis (PCA) and monitoring of historical batches. In the market, there exist commercial software packages addressing these modeling steps from different perspectives, such as ProMV Batch by Aspen [6], Unscrambler X Batch Modeling by CAMO [7] or SIMCA by Sartorius Stedim Biotech [8]. The intention of this toolbox is not to replace these commercial tools, but providing the scientific community with most of the recent analytical solutions that overcome limitations of methods already available. Besides, the MVBatch Toolbox has been designed and implemented as a free software package to analyze data from real batch processes. This toolbox also provides a simulator to generate batch data of a fermentation process of the *Saccharomyces cerevisiae* cultivation. Based on the hierarchical classification scheme presented in [9], this complementary tool to MVBatch can be categorized as a small-scale non-linear dynamic simulator, which can be used for teaching purposes in several statistical areas, such as Multivariate Statistical Process Control, BMSPM, and acceptance sampling. In addition, the simulator is very versatile to generate synthetic data as a benchmark to compare novel methods with sound methodology in BMSPM.

2. Software specification and requirements

The MVBatch Toolbox is a free software under the GNU 3 license, which is available at Github, one of the most used public repositories of free software. Both stable and beta versions of the MVBatch Toolbox are available for downloading. The first stable version can be downloaded at `https://github.com/jogonmar/MVBatch/releases`.This software package is compatible with different Matlab versions (R2013a-2017a) with no requirements for any other third party's utilities beyond the Multivariate Exploratory Data Analysis (MEDA) Toolbox [10], which is an open-source software. The latest stable version of the MEDA toolbox can be download at `https://github.com/josecamachop/MEDA-Toolbox/releases`. For further information on installation, readers are referred to the guidelines provided in the README text file.

The MVBatch Toolbox consists of a set of functions and scripts, which contain Matlab source code for user interfaces and algorithms for bilinear modeling of batch processes; a number of Matlab data files containing batch data from simulated and real processes; and a main Graphic User Interface (GUI) designed in Matlab, which integrates all the functions required to model batch data. There are two ways to work with the MVBatch Toolbox: using the graphical user interface (GUI) for starting users and using the command line for expert users. The main GUI, which is shown in the top center in Figure 1, invokes up to 7 different auxiliary user interfaces along the modeling cycle in a user-friendly manner for: i) variable and batch visualization and screening, ii) imputation of missing data within a batch, iii) batch synchronization, iv) calibration, v) data analysis and latent model exploration, vi) monitoring, and vii) fault diagnosis. The MVBatch GUI is launched by typing MVBatch in the Matlab command window. For expert users, the different parts of the bilinear modeling can be performed using Matlab functions via the command window. Each function provides helping information to guide users in the use of such functions in the MVBatch toolbox, which can be displayed by typing help <function> in the command window.

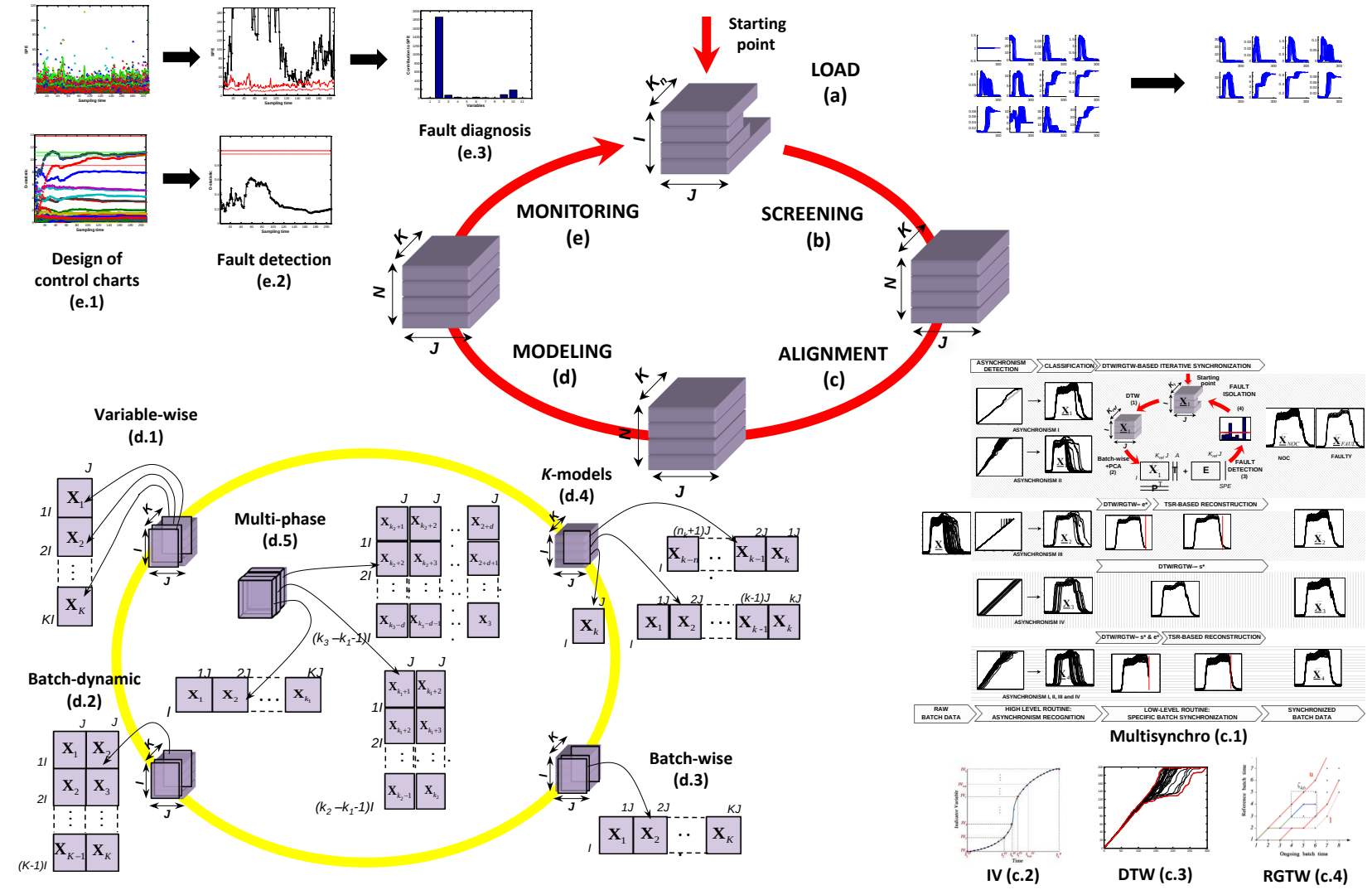


Figure 1: Modeling scheme in Batch Multivariate Statistical Process Control (MSPC) systems based on PCA and modified version of the front-end of MVBatch (top-center).

Table 1: Data sets provided with the MVBatch toolbox. Simulated data from the *Saccharomyces cerevisiae* cultivation process (SACCHA data sets) contain four types of asynchronisms according to [12].

Set	Three-way arrays	Description
SACCHA_A	$\underline{\mathbf{X}}_A(N_A \times J \times K_{n_A}),\ N_A = 40, J = 11, K_{n_A} \in [158, 330]$	NOC calibration batches:
SACCHA_B	$\underline{\mathbf{X}}_B(N_B \times J \times K_{n_B}),\ N_B = 40, J = 11, K_{n_B} \in [158, 330]$	NOC batches with: 10% missing data in J variables and in 2 batches, 10% missing data in 1 variable and in 3 batches
SACCHA_C	$\underline{\mathbf{X}}_C(N_C \times J \times K_{n_C}),\ N_C = 25, J = 11, K_{n_C} \in [129, 237]$	NOC test batches:
SACCHA_D	$\underline{\mathbf{X}}_D(N_D \times J \times K_{n_D}),\ N_D = 30, J = 11, K_{n_D} \in [166, 270]$	Fault #1 (modified k1l) Fault #2 (modified k6) Fault #3 (bias in biomass sensor)
NYLON	$\underline{\mathbf{X}}_{ny}(N_{ny} \times J \times K_{nny}),\ N_{ny} = 57, J = 10, K_{nny} \in [113, 135]$	NOC and faulty batches

A Simulink-based simulator of the fermentation of *Saccharomyces cerevisiae* cultivation is also provided with the toolbox. The simulator is based on the biological model of the aerobic growth of the yeast *Saccharomyces cerevisiae* on glucose limited medium [11]. In the simulation, batches under normal operating conditions -processed with slightly modified values of the internal kinetic constants- are simulated. For further details, the reader is referred to Appendix A.

A total of four data sets are provided with the MVBatch toolbox (see Table 1). The first three data sets contain NOC simulated batch data with four different types of asynchronism from the fermentation process of the *Saccharomyces cerevisiae* cultivation. These data sets are a combination of those used in [12]. The difference between SACCHA_A and SACCHA_B is that the former does not contain missing data and the latter does. The third data set SACCHA_C belongs to the same simulated process and aims to be used for the monitoring of NOC batches. The fourth data set SACCHA_D contains batches with three different abnormalities affected by four types of asynchronism. The last data set NYLON_A contains real batch data of a nylon polymerization process [13].

3. Modeling cycle of batch processes

The bilinear modeling of batch data, once performed variable/batch screening, comprises three main steps: data alignment, data modeling, and the development of the monitoring schemes. In the following, these steps implemented in MVBatch are briefly explained and the methods used in each are shortly described.

3.1. Data alignment

The data alignment step includes equalization of variables and batch synchronization. The aim of this stage is to obtain a three-way data structure where batch data are equalized (i.e. all the variables across batches are expressed at the same sampling rate) and synchronized (all the features of the variables trajectories are aligned across batches). The performance of this stage is crucial to obtain adequate synchronized batch data that can be handled by the modeling approaches, thereby improving the model parameter stability, and reducing the false positive and false negative alarm rates [12, 14]. The synchronization methods proposed in the literature can be roughly classified into three categories [15]: (i) methods based on compressing/expanding the raw trajectories using linear interpolation (TLEC family) either in the batch time dimension or in an indicator variable dimension; (ii) methods based on feature extraction (curve feature family); and (iii) methods based on stretching, compressing, and translating pieces of trajectories (SCT family). The most used commercial software by the chemometrics community for bilinear modeling of batch processes are Aspen ProMV by Aspen Technology [6] and SIMCA by Sartorius Stedim Biotech [8]. The former offers synchronization algorithms based on TLEC, both in the time and variable domain (IV), and includes ordinary SCT-based methods, which are ineffective in the presence of complex asynchronisms though. The latter uses TLEC family to make all batches equal in length, but recent work of the authors has shown that this strategy is not appropriate in cases of multiple asynchronisms [12]. To overcome these problems, the Multisynchro algorithm that successfully tackles these challenging scenarios of asynchronism is implemented in MVBatch (see Figure 1(c.1)). Other approaches available in the MVBatch toolbox are the

Indicator Variable (IV) [16] (see Figure 1(c.2)), the different versions of the Dynamic Time Warping (DTW) for batch synchronization [17, 18] (see Figure 1(c.3)), and the Relaxed Greedy Time Warping algorithm (RGTW) [19] (see Figure 1(c.4)).

3.2. Data modeling

Once batch data are aligned, the calibration of the model is carried out. Prior to the fitting of a multivariate model, batch data need to be pre-processed. The pre-processing strategies available in the software are [20]: i) trajectory centering, ii) trajectory centering and scaling, iii) trajectory centering and variable scaling, iv) variable centering, and v) variable centering and scaling. Depending on the nature of the batch data and the type of model to fit, the pre-processing approach may be different [21].

The aligned and pre-processed three-way array needs to be conveniently rearranged in a number of two-way arrays to apply PCA. There are at least three methods to transform the data structure. First, unfolding the three-way array into a single two-way array: batch-wise [16] -method implemented in Aspen ProMV [6]-, variable-wise [22] based on the implementation in SIMCA Release 14.1 [8], and batch-dynamic [22] (see Figures 1(d.1-d.3)). Second, using an adaptive approach where current and past information are combined, e.g. by using hierarchical models [23]. Finally, fitting K PCA local models [24], each one modeling the information corresponding to a sampling time point. Additionally, both the unfolding and splitting in K models -options i) and iii), respectively- can be combined in approaches such as the evolving modeling [24, 25] or the moving window approach [26] (see Figure 1(d.4)). A simplification of this approach is the multi-stage approach, which is based on the calibration of independent models for different stages of a batch process [27]. All the previous models use a specific and fixed modeling structure, no matter the dynamic nature of the process [28]. It is not surprising that many studies in the literature yield contradictory conclusions regarding the performance of these monitoring approaches. The reason is that the best monitoring approach is very dependent on the features of the process at hand [29]. Hence, experiments on different processes may lead to very different conclusions. Supporting this idea, recent investigations performed by the authors [14, 29] have shown that the use of an inappropriate modeling structure for a process has negative consequences in the model parameter stability and in the performance of a monitoring system. In an attempt to overcome these limitations, among most of the preceding approaches, MVBatch implements the Multi-Phase Framework (MPF) [30], which is aimed at identifying the convenient model structure for a specific process at hand, instead of using the same fixed modeling structure for every process. The MPF selects an appropriate model among a wide number of possibilities, as depicted in Figure 1(d.5).

For the exploration of data sets and latent structures, the MEDA Toolbox [10] is included in MVBatch. This complementary toolbox provides a set of multivariate analysis tools for such purpose, among which we highlight: GPCA [31], GPLS [32], MEDA [33], oMEDA [34], SVI plots [35], and score and loading plots.

3.3. Design of the monitoring scheme

One of the most important objectives for analyzing batch data is the statistical process monitoring of incoming batches. The statistical monitoring can be performed either after completion of the batch, which is called post-batch process monitoring and is segregated into end-of-batch and pseudo-online applications, or during the progress of the batch run, also called real-time process monitoring. In end-of-batch process monitoring, the aim is to discover sources of variability among batches, improve operation policies, and at the end of the batch diagnose the root causes of past abnormal and/or non-expected operating conditions. This is carried out by estimating statistical measures describing the behavior of the process during the entire batch duration. In contrast, pseudo-online process monitoring focuses on statistically evaluating the process at each sampling time point, leading to a more exhaustive, accurate, and thorough examination of the process.

In the design of monitoring schemes, two Shewhart control charts are usually developed: the D-statistic or Hotelling T^2 chart, and the Q-statistic or SPE chart [3, 36]. MVBatch estimates their control limits from NOC process data, and later adjusts these thresholds using cross-validation techniques for a given imposed significance level (ISL) [29] (see Figure 1(e.1)). Additionally, an unsupervised control chart based on the warping profiles from NOC batches (NOC-WICC) [37] is designed as a complementary tool to the aforementioned charts for end-of-batch and real-time batch process monitoring.

The multivariate statistics for a new batch are computed in on-line fashion to emulate a real-time application. For those models that require the imputation of the missing part of the variables trajectories at each time point k, the Trimmed Score Regression (TSR) [38] is used. If the batch remains under NOC, i.e. the statistics remain below the control limits except for punctual cases, we can assume that the quality will be within specifications, and therefore, the processing should continue. If, otherwise, several consecutive points are beyond the control limits, it means that the batch is behaving in an abnormal manner and the quality may be seriously affected (see Figure 1(e.2)). In this situation, the fault can be diagnosed (see Figure 1(e.3)). Overall and instantaneous contribution plots to the D-statistic and Q-statistic [39] are available in MVBatch for fault diagnosis. If there is any chance to recover the batch or drive the process under NOC again, a control action should be implemented [40]. Otherwise, the batch might be discarded, saving time and resources.

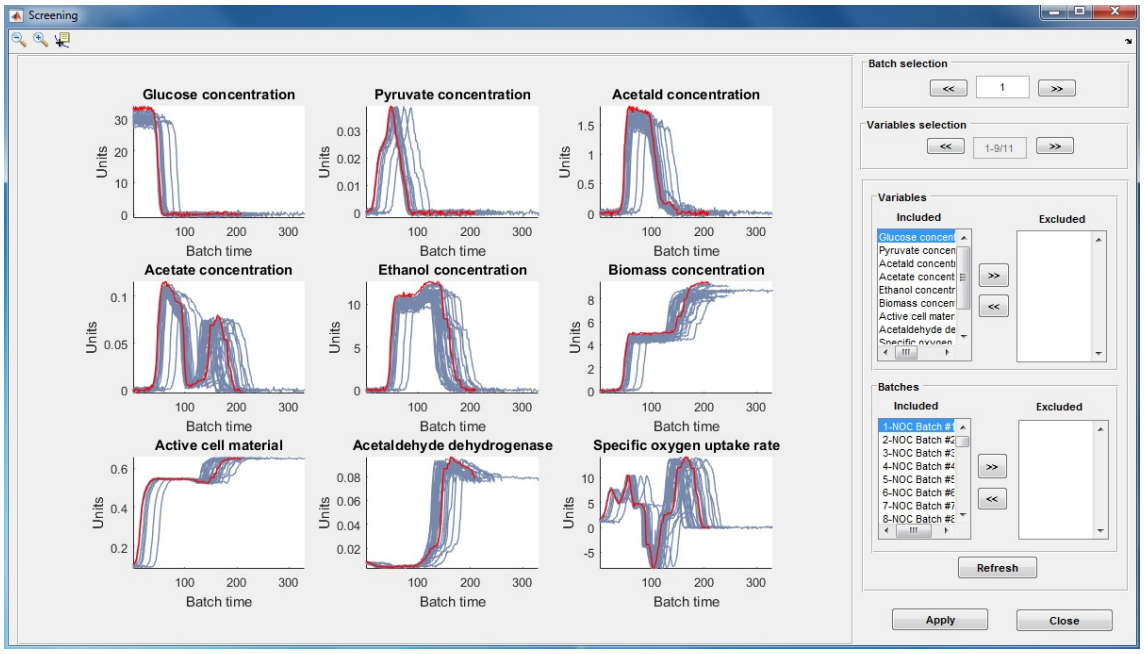


Figure 2: Interface for screening batches and variables for subsequent alignment and modeling.

4. A toolbox overview with a case study: *Saccharomyces cerevisiae*

In this section, a software demonstration of the MVBatch Toolbox is carried out by using the simulated data sets SACCHA_A and SACCHA_D for calibration and test, respectively.

The main interface of the software package integrates a top row with the File and About pull-down menus and four main modules, which are named 'Screening', 'Alignment', 'Modeling', and 'Monitoring'. We can save the results of the screening of the variables and batches, alignment, and modeling by clicking on the Save option of the File menu. Existent analysis can also be loaded via the Open option.

To initiate MVBatch, we need to load a data set through the Load option. This software package reads batch data contained in a Matlab file, which must contain three data structures: batch data, tagnames, and batch identifiers. For further details, the reader is referred to the Guidelines file included in the toolbox.

4.1. Screening

In the Screening module we can firstly visualize the raw process variables trajectories for all the batches (see Figure 2). MVBatch displays the trajectories for the calibration data set in grey color at the left-side of the interface, highlighting the trajectories of the first batch in red color. At this point, we can select any

batch of interest by moving forward or backward the batch index in the Batch Selection panel. If there are more variables than subplots depicted, we have the option of visualizing those not displayed by clicking on the Variable Selection panel. Based on prior knowledge or process understanding reached through previous analysis, we may be interested in including and/or excluding process variables and batches for subsequent steps in the modeling cycle. For this purpose, we can select the data in the Variables and Batches panels. To update the graphs of the batch trajectories, we click on the Refresh button. Once batch data are filtered out, we can move on to the alignment of the batches by clicking on the Apply button.

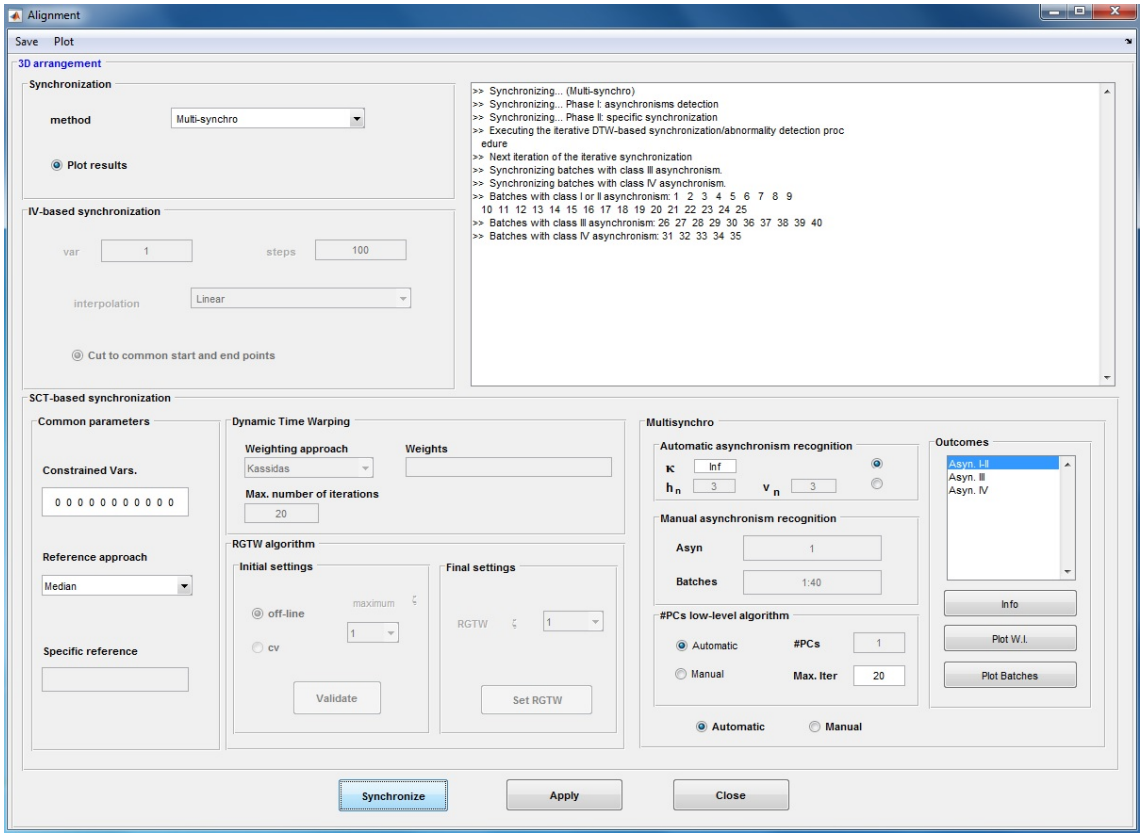


Figure 3: Interface for data synchronization.

4.2. Alignment

In this interface, we can synchronize batch trajectories by interacting with the 3D arrangement panel (see Figure 3). This panel provides a set of tools to synchronize batch trajectories, ensuring not only that the resulting batches have equal duration but also that the main process features are aligned across batches. The methods programmed in this toolbox are divided into two groups: TLEC-based method in the variable domain (Indicator Variable (IV) [39]), and SCT-based methods (Dynamic Time Warping (DTW) following Kassidas et al's [17] and Ramaker et al's [18] approaches, the Relaxed-Greedy Time Warping (RGTW) [19] and Multisynchro [15]).

For the application of IV, it is required to have a process variable which is a good indicator of the maturity of the batch. This variable has to be strictly monotonic and smooth, with the same starting and end point in all batches. The selection of this method is done in the Method pop-up menu located in the Synchronization panel. To perform this type of synchronization, the indicator variable and the number of sampling points must be indicated in the Var and Steps edit boxes, respectively, as well as the type of interpolation in the Interpolation pop-up menu (see IV-based synchronization panel at the center-left side

in Figure 3). In case that the variable selected as an indicator variable does not have the same starting and end points across batches, the algorithm can discard the preceding values of the initial common point or succeeding values to the final common point if the option 'Cut to common starting and end point' is enabled. Caution must be taken enabling this option since the system forces the data set to meet the requirements of the synchronization algorithm. It may cause distortion on data that can likewise lead to misleading results in the outcomes of the multivariate analysis, and a large type I and II error rates in the monitoring [12].

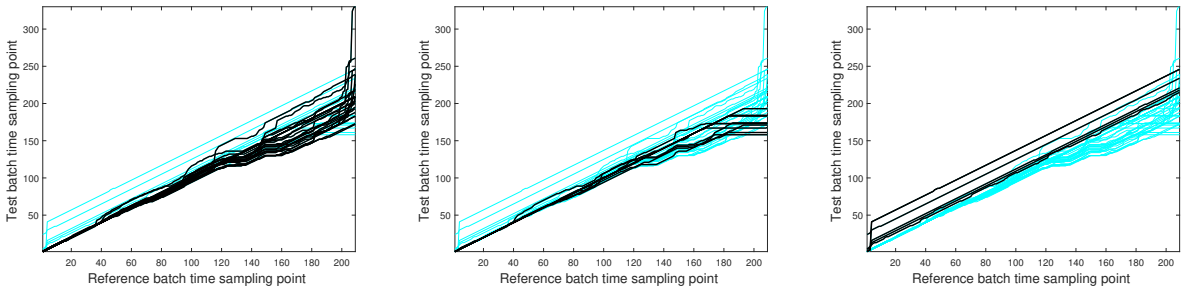
Synchronization based on any of the implemented SCT-based methods requires the selection of two common parameters (see common parameters panel at the bottom-left side in Figure 3). First, a batch of the calibration data set needs to be selected to align the remaining batches. If prior knowledge about the process is available, we can choose a specific batch as a reference. If not, the algorithm can select that batch whose length is the closest to the median or to the average duration of the collected batches. For this purpose, the Reference pop-up menu provides these options: select, median and average. To select a certain batch, the Reference edit box is enabled to type the number of the batch. After this, we need to specify constrained variables. Typically, there are process variables with flat profiles containing too few or no features for synchronization, and/or low signal-to-noise ratio that might negatively affect the quality of synchronization [41]. In this situation, these variables are recommendable to be discarded in the computation of the synchronization weights by typing 1 values in the edit box 'Constrained vars'. In contrast, if we want to include variables in the computation of the synchronization weights, we have to specify 0 values instead.

The next parameters to initialize depend on the SCT method selected in the Method pop-up menu. For the DTW-based synchronization, we need to specify the weights that will give more importance to certain process variables in the synchronization (see DTW panel at the bottom-center side in Figure 3). These weights can be computed in such a way that more importance is given to either variables more consistent batch-to-batch (Kassidas *et al.*'s approach), to variables containing more warping information (Ramaker *et al.*'s approach), or to variables satisfying both requirements (Geometric average of Kassidas *et al.* and Ramaker *et al.*'s weights). We can constrain the number of iterations in the synchronization based on Kassidas *et al.*'s approach by introducing the value in the edit box 'Max. number of iteration'. Another option is to indicate explicitly the weights if prior knowledge about the process is available. All these options can be selected in the Weighting Approach pop-up menu. If we click on the Select option, the Weight edit box will be enabled to introduce the weights. Note that the weights range from zero to the number of variables, and the sum of all the weights must be equal to the number of variables.

Furthermore, MVBatch gives the possibility to synchronize batch data in such a way that can be used for the design of a monitoring scheme for real-time applications. For this purpose, we need to execute the synchronization procedure based on the RGTW algorithm proposed in [19]. Note that the RGTW panel will only be enabled in the user interface when a synchronization based on DTW has been completed. Two parameters need to be initialized: the width of the sliding warping window ζ and the bands (see the RGTW algorithm panel at bottom-center side in Figure 3). On the one hand, we can set the bands using the warping information obtained from the off-line DTW synhronization for a specific window width γ (off-line radiobutton). The window width can be manually set by selecting the value in the RGTW γ pop-up menu. On the other hand, the RGTW parameters can be optimized by running the cross-validation procedure proposed in [19]. Upon the execution of the cross-validation using a maximum window width (see maximum γ pop-up menu), an Analysis of Variance (ANOVA) is performed on the Fisher Z-transformed[1] correlation coefficients for the different window widths γ. Based on statistical significant differences, we can choose the optimal window width from the RGTW γ pop-up menu and confirm the synchronization model by pressing the Set RGTW button.

In scenarios of multiple asynchronism, the previous synchronization strategies not only are inappropriate but also may produce misalignments and introduce artificial features. The reason why these methods are not accurate is because they do not take into consideration the different asynchronisms batch data may contain. To overcome this problem, the Multisynchro algorithm can be applied to batch data by selecting the Multisynchro option from the Method pop-up menu of the Synchronization panel. Automatically,

[1]The Fisher Z transformation is usually performed to approximate the distribution of the Pearson's correlation coefficients to a normal distribution.



(a) (b) (c)

Figure 4: Warping information derived from the Multisynchro-based synchronization for the three types of asynchronism found in the data set.

the software package will enable the Multisynchro panel (see panel at the bottom-right side in Figure 3). This synchronization algorithm consists of a high-level and low-level routine. The high-level routine aims to recognize the different types of asynchronous trajectories. The low-level routine is in charge of synchronizing the variable trajectories of each of the batches with a specific procedure based on the type of asynchronism. To initiate the algorithm, there are two ways: the automatic and manual asynchronism detection (see panels at the bottom-right side in Figure 3). As an introduction to this functionality, we will restrict to the automatic detection, where we can decide the thresholds for the number of consecutive horizontal and vertical transitions in the warping information (see parameters h_n and v_n in the Automatic Asynchronism Recognition panel). This is the procedure that the algorithm uses to identify the type of asynchronisms. Another parameter needed is the number of principal components (PCs) with the aim of detecting abnormal batches in the calibration data set. Either the algorithm automatically selects those PCs that explain more than 50% of the variability (option 'Automatic' in the panel '#PCs low-level algorithm'), or we explicitely indicate the number of PCs in the edit box 'Max. iter'. Internally, the algorithm iteratively synchronizes trajectories provided that there are abnormal batches still in the data set. For more details on the Multisynchro algorithm, the user is referred to the original research work [15].

As the experimental data under study contain different types of asynchronisms, let us synchronize these batches with Multisynchro by selecting the corresponding option in the Method pop-menu of the Synchronization panel, and keeping the default values. At the end of the execution, the algorithm returns the steps carried out and the classification of the batches by the nature of their asynchronisms in the console at the top-right of Figure 3. To visualize the information of the synchronization performed, we press the Info button. For the warping profiles of the variable trajectories, we press the Plot W.I. button. Finally, we can press the Plot Batches button to visualize the synchronized variable trajectories of the batches classified into one of the types of asynchronism. For the example data, the warping information for each class of asynchronism is shown in Figure 4. Warping profiles, which are associated with batches with different or similar pace, but without process events overlapping are displayed in Figure 4(a). Incomplete batches are also present, which produce warping profiles with horizontal transitions at the end of the batch run, displayed by black lines in Figure 4(b). A third category of asynchronism observed in data is batches shifted at early stage of the process. Their warping profiles typically show vertical transitions at the start of the batch, as shown by black lines in Figure 4(c). The resulting synchronized batch trajectories are displayed in Figure 5.

Afterward, the systems enables the Apply button located at the bottom side of the interface to proceed with the modeling of the synchronized data. Note that we can re-synchronize the raw batch data at any time, but previous synchronizations will be discarded.

4.3. Modeling

Once batch data have been synchronized, the Modeling module of the main interface is enabled. When we select this option, the Modeling interface pops up (see Figure 6). The interface contains a top row with

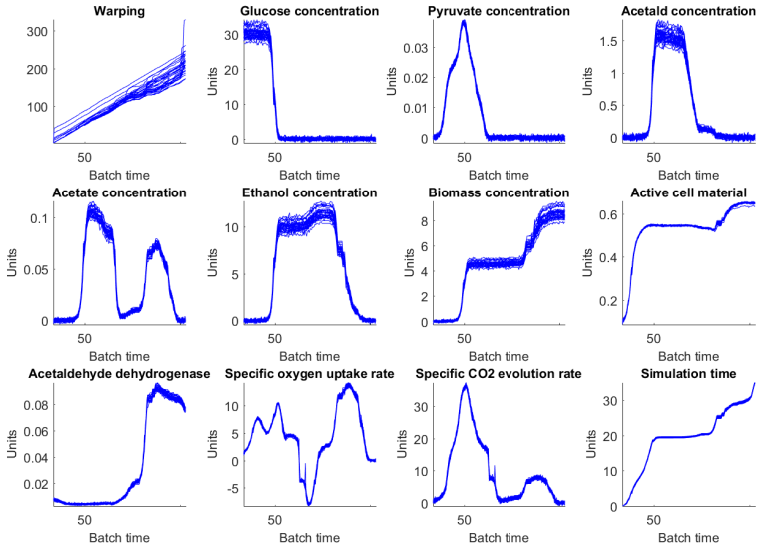


Figure 5: Resulting variable trajectories after synchronizing the trajectories using the Multisynchro algorithm. The variables shown in order from top to bottom, and from left to right are: warping information, concentrations (glucose, pyruvate, acetaldehyde, acetate, ethanol, and biomass), active cell material, acetaldehyde dehydrogenase, specific oxygen uptake rate, and specific carbon dioxide evolution rate.

the File, Preprocessing, Cross-validation pull-down menus. In the first option, we can save, load and print results of the multivariate analysis. In the second option, we can select any of the pre-processing methods described in Section 3.2. Finally, the third option provides different cross-validation methods: row-wise, element-wise, and column-wise k-fold [42].

The Manual Mode panel shown at the top side in Figure 6 groups a set of options to display the correlation of the process variables (see Correlation Maps panel), define statistical models (see Model panel) and study the process dynamics captured by the models (see Dynamics panel). Prior to the statistical modeling, it is recommended looking at the correlation maps because these tools are useful to reveal the time-varying process dynamics. Specific variables and time periods can be selected for analysis. Afterward, it is a good practice to fit a manual model using batch-wise modeling for a first exploratory data analysis. This type of models is very useful to get insight into the correlation among variables over the batch run, detect the most severe faulty batches and diagnose their main causes. For instance, define a batch-wise model of 3 PCs by modifying the LMVs parameter to 'Inf' (infinite) and press the Add button. We can also use the Explore button for a better selection of the number of PCs using the MEDA toolbox.

The Modeling user interface also provides the option of fitting multi-phase models automatically (see the 'MPPCA' panel at the center-left in Figure 6). The parameters required to automatically recognize the appropriate model structure for monitoring are: the unfolding approach specified in terms of the number of lagged measurement vectors or LMVs, the improvement threshold that controls when to include new PCs and new phases in the model (T), the gain parameters that control the trend of the algorithm to separate data into more or less phases (k), the minimum length of the phases (minL), and the initial number of PCs (a). The number of LMVs should be initialized according to the set of models one would like to explore in the LMV edit box. With 0 LMVs, we compute a static model -variable-wise, with 1 LMV a model of order 1 and so forth until we reach as many LMVs as the number of sampling time points minus one, which corresponds to the batch-wise unfolded model. The MPPCA can handle several LMVs at the same time, and therefore, consider several variants of unfolding. For this purpose, we can type several values in the LMV edit box, such as "0, 1, 2" or "0:2" to compute models from order 0 to 2, for instance. In order to yield parsionomious models, the number of LMVs recommended ranges from 0 to a low number of LMVs [29]. Notice that the

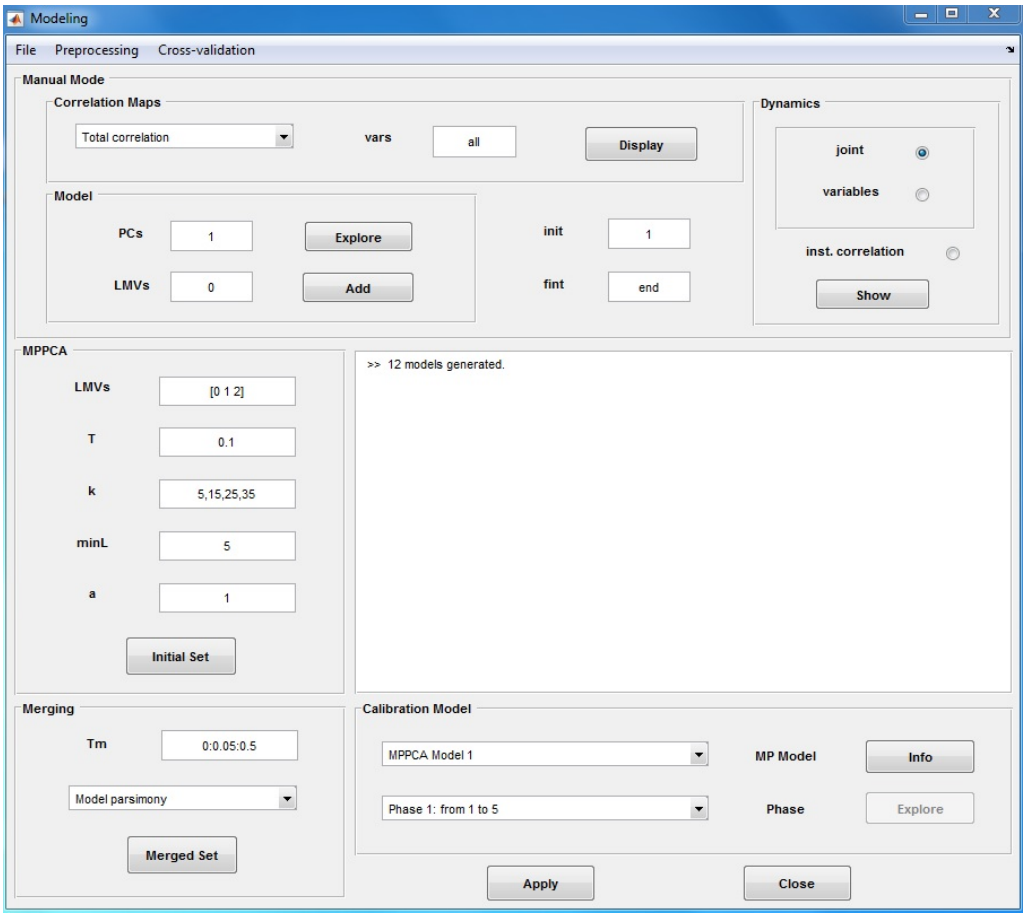


Figure 6: Interface for data modeling using the multi-phase framework.

features of a model depend very much on the number of LMVs [28]. The rest of parameters can be set to suggested values. Afterward, we can execute the automatic detection by clicking on the Initial Set button. Note that the detection of the phases is a time-consuming task; the larger the number of parameters to explore, the higher the computation time. During this execution, information on the task the algorithm is performing will appear in the console located at the center-right side of the interface shown in Figure 6.

For illustrative purposes, let us perform a multi-phase algorithm with a range of parameters, namely: LMVs = [0, 1, 2], T = 0.1, k = 5, 15, 25, 35, minL = 5, and a= 1. Upon execution, the algorithm returns twelve different multi-phase models, which are listed in the MP Model list box (see Calibration Model panel at the bottom-right side in Figure 6). We can display information of each model selected in the MP model pop-up menu by pressing the Info button. For instance, if we select the first multi-phase model named 'MPPCA Model 1', we will realize that the multi-phase algorithm segregates the batch data into 21 different phases with 1 PC each. The console also displays the PRESS and the parameters used in the calibration, including the pre-processing and cross-validation methods. For each phase, we also gather information on the number of PCs, the number of LMVs and the PRESS.

At the end of the execution, the Merging section at the bottom-left side in Figure 6 is enabled. This section aims to post-process and merge the information obtained in the previous step in an optimum manner, yielding mixtures of the previous models [28]. In addition, we can select different merging criteria in the pop-up menu located in the same panel: model parsimony, covariance matrix parsimony, minimize LMVs, minimize phases and minimize PCs. To proceed with the merging procedure, we click on the Merged Set button. This operation yields n multi-phase merged models, which are listed in the MP Model pop-up menu.

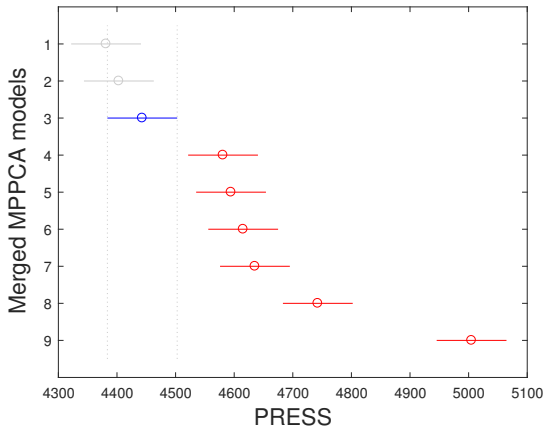


Figure 7: LSD intervals estimated for the merging procedure applied to the models found by the multi-phase algorithm.

The software package performs an ANOVA and computes the LSD plot to help in selecting the optimum model that does not produce a significant loss of prediction power. In the case of the current modeling, we obtain nine merged multi-phase models, and eventually, we select the parsimonious model as objective function. As can be appreciated from Figure 7, the simplification from the 4th and 9th merged models imply a statistically significant loss of prediction capability because the LSD intervals for these models do not overlap with the rest. According to this analysis, the best multi-phase model in terms of parsimony is the 3rd merged MPPCA model.

At this point, we can use the Explore button to analyze any of the resulting phases using the MEDA Toolbox and some built-in dynamic plots (see Calibration Model panel at the bottom-right side in Figure 6). Likewise the covariance maps, these tools can help to understand the dynamics of the process.

Once the multi-phase model is optimized, we can design the monitoring scheme by using the calibration data set and the model fitted in the next step of the modeling cycle by pressing the Apply button.

4.4. Monitoring

The interface for the design of a monitoring scheme comprises i) a menu bar with different options to save the results and import data, ii) four panels for the design of the monitoring schemes and the monitoring of batches named 'Model', 'Monitoring system', 'Post-batch off-line process monitoring' and 'Post-batch on-line process monitoring', and iii) one panel for the selection of the statistics and batches for fault diagnosis (see Figure 8).

The first step is to select one of the models we fitted in the previous interface in the MP model pop-up menu (see Model panel at the top-left side of Figure 8). One recommended step in the design of the monitoring scheme is to re-adjust the theoretical control limits using a leave-one-out cross-validation approach [12, 29, 43]. To execute this procedure, we press the CV monitoring system button. Depending on the number of samples points, the type of multi-phase model and the number of batches, the time required to compute the statistics and control limits may considerably vary. At the end of the execution, the post-batch online control charts for the D-statistic, Q-statistic and the warping information [37] are plotted jointly with their control limits in the panel 'Post-batch on-line process monitoring'. In case that a batch-wise PCA model is fitted, the overall D-statistic and Q-statistic control charts are also shown in the panel 'Post-batch off-line process monitoring', as is the case of the monitoring scheme designed in Figure 8. The red control limits are computed using theoretical approximations [39] and the green control limits are corrected by cross-validation. To monitor the batches of the calibration data set, we press the Monitor button. Upon execution, we can select a batch and visualize the multivariate control charts by clicking on the Plot button. To use the corrected control limits in the monitoring of new batches, we tick the CV limits radio button. In case that we want to monitor a new batch, we have to import new data sets from a data file, such as

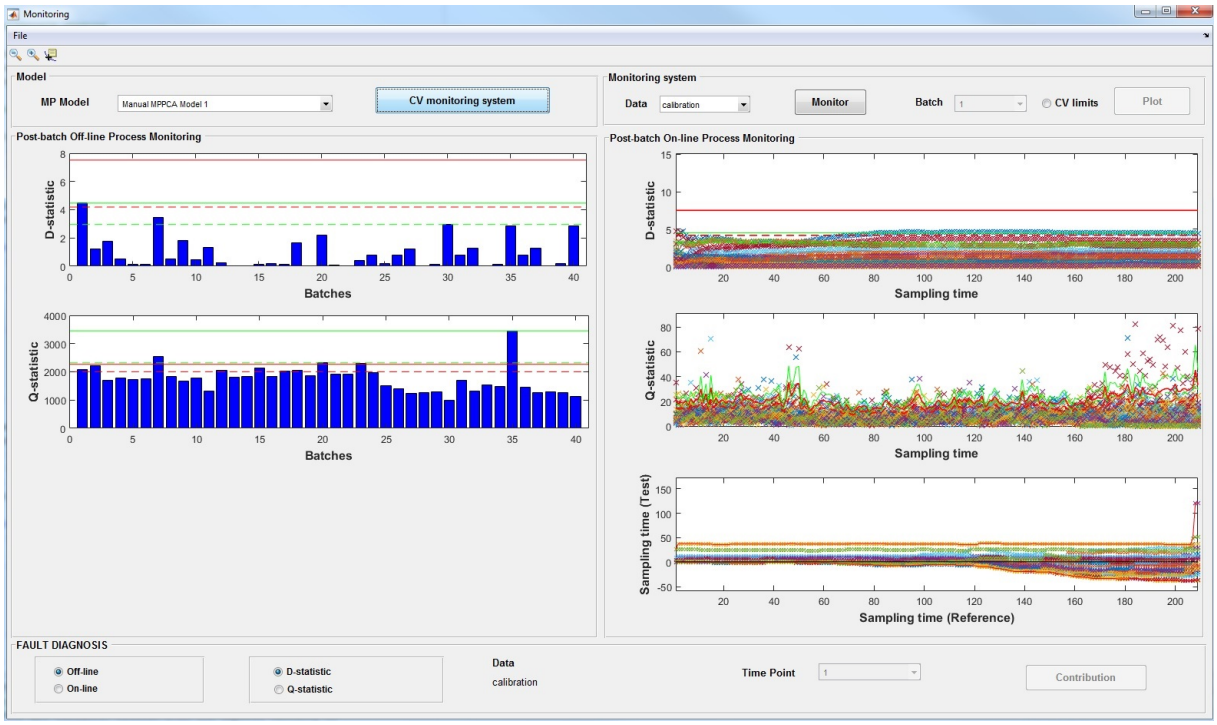


Figure 8: Interface for the design of a monitoring scheme and the monitoring of test batches.

SACCHA_D, by using the Import Data option of the File menu. Note that if there are missing values in any of the imported batches, the application will prompt us to impute the missing values via the Missing Imputation user interface (not shown). Afterward, we can project batches onto the model by pressing the Monitor button. Upon completion, we can visualize the corresponding control charts by pressing the Plot button.

In case of the appearance of out-of-control signals in any of the control charts, MVBatch can give insight into the potential root causes of the abnormalities through the Fault Diagnosis panel (see bottom side of the interface in Figure 8). Overall and instantaneous contributions to the D-statistic and SPE are available by marking the radio buttons 'Off-line' (overall contributions) or 'On-line' (instantaneous contributions), and 'D-statistic' or 'Q-statistic'. The selection of a batch is required by selecting the corresponding batch identifier in the Batch pop-up menu. For instantaneous contributions we also need to specify the time point at which the contribution to the chosen statistic will be calculated. For this purpose, we select the desired time point in the Time Point pop-up menu. To visualize the contributions, we press the Contribution button. Note that the overall contributions are only available if the fitted multi-phase model corresponds to a batch-wise PCA model.

Figure 9 shows the overall contribution plots to the SPE statistic for the first batch of the faulty data set, whose SPE value exceeds the control limits both in the post-batch offline and online control charts (graphs not shown). This first graph at the top-left side shows the overall contributions to the statistic per each variable and over time. The contribution for each variable is enclosed by black dashed lines. The second graph at the center-left side represents the accumulated contribution of each variable through the batch run to the statistic. Finally, the third graph at the bottom-left side depicts the contribution of all the variables per sampling point to the statistic. With these tools, the user is able to get insight into the causes of this abnormalities, not only what variables are affected, but also at which time interval the process behaved differently than expected. To confirm the findings, we can visualize the trajectories of a certain process variable for the calibration batches, the faulty batch and the average trajectory by selecting the variable of interest in the pop-up menu 'Variable #' and pressing the Display button (see right side of Figure 9). We

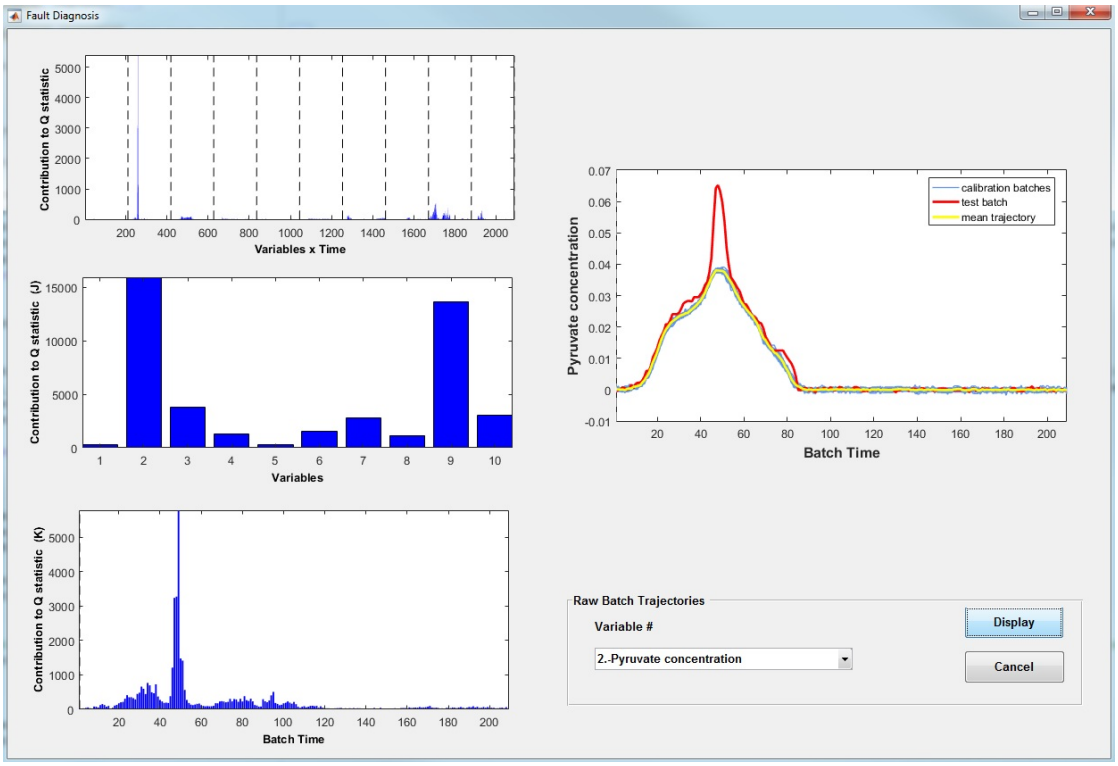


Figure 9: Interface for the diagnosis of faulty batches.

can always return to the monitoring interface to interrogate the multi-phase model on other test batches by clicking the Cancel button of the Fault Diagnosis interface.

5. Conclusions

The bilinear modeling of batch processes comprises the application of different steps to end up with a monitoring scheme that is able to detect and diagnose anomalies in the process. The MVBatch graphical user interface is a user-friendly tool designed to carry out the modeling steps: data alignment, data modeling, and the development of monitoring schemes. In contrast to commercial software packages, MVBatch offers not only the conventional methods in process chemometrics, but also the latest advances proposed in the literature that clearly outperform the former in a wide set of cases. This includes Relaxed Greedy Time Warping (RGTW) and Multisynchro for batch synchronization, and multi-phase modeling for batch data calibration. In addition, a small-scale non-linear dynamic simulator of the fermentation process of the *Saccharomyces cerevisiae* cultivation is provided to generate realistic batch data under normal and abnormal operating conditions. This versatile simulator is intended to close the gap between theory and practice in BMSPM courses for master and PhD students, professionals and practitioners in the field of Multivariate Statistics, Chemometrics and Data Science. In addition, the synthetic data can be used to illustrate the performance of novel methods with sound methodologies published in the literature.

6. Validation

Dr. Marina Cocchi. Department of Chemical and Geological Sciences, University of Modena and Reggio Emilia, Via G Campi, 183, 41125 Modena, Italy.

We tested the software on Mac OS X 10.11.6 Matlab R2015b and Windows 10, Matlab R2012b and Matlab R2016a. The interface is easy to use and considering the software from a general point of view, the

toolbox offers a neat and comprehensive way to exploit the potentiality of PCA for the analysis of Process Data, providing all the plots and features, which are necessary for an effective Multivariate Statistical Process Control framework implementation.

Acknowledgements

This work is partially supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds through the projects DPI2017-82896-C2-1-R and TIN2017-83494-R. Authors also acknowledge the volunteers to test MVBatch and report their impressions for this software tutorial.

Appendix A. Simulator of the fermentation process of the *Saccharomyces cerevisiae*

Saccharomyces cerevisiae is a yeast widely used in biotechnical and pharmaceutical industries for the production of proteins. The model of the aerobic growth of *Saccharomyces cerevisiae* on glucose limited medium introduced by Lei [11] is used as basis to generate data.

Fermentation is performed in four different phases in a batch mode: a) lag phase, b) first exponential growth, c) second exponential growth, and d) stationary phase (see Figure A.10). In the first phase, the yeast becomes acclimated to the heterogeneous culture media prior to the reproduction process, which typically elapses a couple of hours. In the first exponential growth, the glucose is in excess in the medium, and cells are not able to consume the whole amount of glucose. Hence, ethanol is produced together with the excretion of pyruvate and acetate. Later on, the initial amount of glucose is consumed by the growing cells. Before ethanol is consumed during the second exponential growth, the accumulated amount of pyruvate and acetate is consumed. During growth on ethanol, acetate is produced again.

The assumptions for a perfect abiotic subsystem of *Saccharomyces cerevisiae* cultivation have been considered to develop a simulator for batch data generation [11]. The metabolic reactions and stoichiometry of this microorganism have been implemented in Simulink. For the sake of accuracy in simulation, the biological variability of the yeast is taken into consideration to generate time-varying trajectories, and Gaussian noise of low magnitude is added to the initial conditions (10%) and measurements (5%) to simulate the typical errors in sensors. For each batch, measurements belonging to ten process variables are registered every sampling time point 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 array.

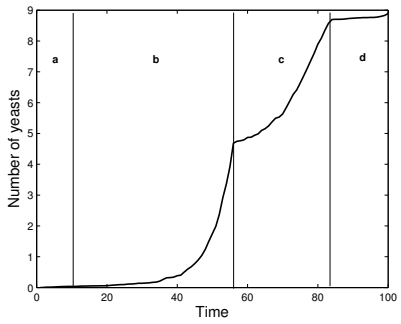


Figure A.10: Growth curve of the *Saccharomyces cerevisiae* cultivation: a) lag phase, b) first exponential growth, c) second exponential growth, and d) stationary phase.

Three different types of faults are designed: two process faults generated by modifying the internal constants *k1l* (associated with the reaction describing the glucose uptake system and the glycolytic pathway) and *k6* (associated with the reaction describing the formation of ethanol from acetaldehyde) and one sensor process fault representing a bias in the biomass concentration sensor. The first two faults do not illustrate abnormal behaviors related to specific biochemical changes in the metabolic network, but abnormal operating conditions that may produce changes in the kinetic parameters of the model. For this purpose, interference processes, i.e. factors that directly influence the maximum reaction rate (*Vmax*, *k1L* in the stoichiometric model) of the lumped biochemical reaction considered in the model, are simulated. Vmax represents the way in which the substrate is processed by the yeast in a glucose limited media. Although Vmax is biochemically based (highly efficient strains will be able to consume glucose more quickly, showing higher intrinsic Vmax values), this parameter may also be influenced by processes such as diffusion. For example, if the bioreactor is not correctly stirred or the viscosity of the mixture is too high, and nutrient diffusion is hindered, substrates may not be accessible for the microorganism, resulting in low consumption rates. When these operating conditions are overcome, a better material transport is expected, and hence, a higher maximum reaction

rate *Vmax*. To simulate these scenarios, the values of the kinetic constants *k1l* and *k6* can be accordingly modified in the stoichiometric equations. Modifying the constants, the consumption of glucose might be higher than in normal operating conditions, causing an excess of glucose in the microorganism (the so-called metabolic overflow). In this scenario, the rate of glycolysis exceeds a critical value resulting in by-product formation (ethanol, acetaldehyde, acetate) from pyruvate and ethanol (activation of the fermentation pathway). Consequently, the amount of carbon dioxide is also higher in media than in normal operating conditions. This has a direct effect on the duration of the second stage of the fermentation (from the 50th sampling time point -i.e. 20 hours after the batch started, approximately- onward), which takes longer than usual to reduce the amount of these products. The third fault revolves on a malfunctioning of the biomass concentration probe. This is simulated by adding a specific bias in the corresponding process variable.

The user can access the Simulink scheme by editing the file *saccha.mdl* [2]. To change the parameters of the simulation, such as number of NOC and faulty batches, magnitude of the failure, addition of new batches by modifying the stoichiometry of the reactions, the user must edit the Matlab script <*simule.m*>. For further information on customizing the simulation, users are referred to the help document provided in the Matlab file.

[2]The users must have the Simulink toolbox of Matlab installed to modify the biological model and generate data.

References

[1] J. MacGregor, M. Bruwer, I. Miletic, M. Cardin, Z. Liu, Latent variable models and big data in the process industries, IFAC-PapersOnLine 48 (2015) 520 – 524. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[2] S. Wold, N. Kettaneh-Wold, J. McGregor, K. Dunn, Batch Process Modeling and MSPC, Comprehensive Chemometrics - Chemical and Biochemical Data Analysis, Elsevier:Oxford 1 (2009) 163–195.

[3] A. Ferrer, Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process., Quality Engineering 19 (2007) 311–325.

[4] W. H. Woodall, Controversies and contradictions in statistical process control, Journal of Quality Technology 32 (2000) 341–350.

[5] A. Ferrer, Statistical control of measures and processes, Comprehensive Chemometrics - Chemical and Biochemical Data Analysis, Elsevier:Oxford 1 (2009) 97–126.

[6] AspenTech, Aspen ProMV, 2018.

[7] CAMO, Unscrambler x batch modeling Release 10.4, 2017.

[8] Sartorius Stedim Biotech, SIMCA Release 15.0.2, 2018.

[9] M. Reis, R. S. Kenett, A structured overview on the use of computational simulators for teaching statistical methods, Quality Engineering 29 (2017) 730–744.

[10] J. Camacho, A. Pérez-Villegas, R. A. Rodríguez-Gómez, E. Jiménez-Maas, Multivariate exploratory data analysis (meda) toolbox for matlab, Chemometrics and Intelligent Laboratory Systems 143 (2015) 49 – 57.

[11] F. Lei, M. Rotbøll, S. Jørgensen, A biochemically structured model for Saccharomyces cerevisiae, Journal of Biotechnology 88 (2001) 205–221.

[12] J. M. González-Martínez, R. Vitale, O. E. de Noord, A. Ferrer, Effect of synchronization on bilinear batch process modeling, Industrial & Engineering Chemistry Research 53 (2014) 4339–4351.

[13] K. Kosanovich, K. Dahl, M. Piovoso, Improved process understanding using multiway principal component analysis, Engineering Chemical Research 35 (1996) 138–146.

[14] J. M. González-Martínez, J. Camacho, A. Ferrer, Bilinear modeling of batch processes. Part III: parameter stability, Journal of Chemometrics 28 (2014) 10–27.

[15] J. M. González-Martínez, O. E. de Noord, A. Ferrer, Multisynchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms, Journal of Chemometrics 28 (2014) 462–475.

[16] P. Nomikos, J. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE Journal 40 (1994) 1361–1375.

[17] A. Kassidas, J. MacGregor, P. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE Journal 44 (1998) 864–875.

[18] H. Ramaker, E. van Sprang, J. Westerhuis, A. Smilde, Dynamic time warping of spectroscopic batch data, Analytica Chimica Acta 498 (2003) 133–153.

[19] 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.

[20] R. Bro, A. Smilde, Centering and scaling in component analysis., Journal of Chemometrics. 17 (2003) 16–33.

[21] 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.

[22] S. Wold, P. Geladi, K. Esbensen, J. Ohman, Multi-way principal components-and-PLS-analysis, Journal of Chemometrics 1 (1987) 41 – 56.

[23] S. Rännar, J. MacGregor, S. Wold, Adaptive batch monitoring using hierarchical PCA, Chemometrics and Intelligent Laboratory Systems 41 (1998) 73–81.

[24] 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.

[25] P. Nomikos, Statistical process control of batch processes, PhD Thesis, McMaster University, Hamilton, ON, 1995.

[26] B. Lennox, G. Montague, H. Hiden, G. Kornfeld, P. Goulding, Process monitoring of an industrial fed-batch fermentation, Biotechnology and Bioengineering 74 (2001) 125.

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

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

[29] 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.

[30] J. Camacho, J. Picó, A. Ferrer, Multi-phase analysis framework for handling batch process data, Journal of Chemometrics 22 (2008) 632–643.

[31] J. Camacho, R. A. Rodríguez-Gómez, E. Saccenti, Group-wise principal component analysis for exploratory data analysis, Journal of Computational and Graphical Statistics 26 (2017) 501–512.

[32] K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, P. Besse, A sparse pls for variable selection when integrating omics data, Statistical applications in genetics and molecular biology 7 (2008).

[33] J. Camacho, Missing-data theory in the context of exploratory data analysis, Chemometrics and Intelligent Laboratory Systems 103 (2010) 8 – 18.

[34] J. Camacho, Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models, Journal of Chemometrics 25 (2011) 592–600.
[35] J. Camacho, J. Picó, A. Ferrer, Data understanding with PCA: Structural and variance information plots, Chemometrics and Intelligent Laboratory Systems 100 (2010) 48 – 56.
[36] S. Joe Qin, Statistical process monitoring: basics and beyond, Journal of Chemometrics 17 (2003) 480–502.
[37] 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.
[38] F. Arteaga, A. Ferrer, Missing data, Comprehensive Chemometrics - Chemical and Biochemical Data Analysis, Elsevier:Oxford 3 (2009) 285–314.
[39] P. Nomikos, J. MacGregor, Multivariate SPC charts for monitoring batch processes, Technometrics 37 (1995) 41–59.
[40] Y. Yabuki, J. F. MacGregor, Product quality control in semibatch reactors using midcourse correction policies, Industrial & Engineering Chemistry Research 36 (1997) 1268–1275.
[41] J. González-Martínez, Advances on bilinear modeling of biochemical batch processes, Ph.D. thesis, Departament d'Estadstica i Investigaci Operativa Aplicades i Qualitat, 2015.
[42] J. Camacho, New Methods Based on the Projection to Latent Structures for Monitoring, Prediction and Optimization of Batch Processes, PhD Dissertation, Universidad Politcnica de Valencia, 2007.
[43] H.-J. Ramaker, E. N. M. Van Sprang, J. A. Westerhuis, S. P. Gurden, A. K. Smilde, F. H. Van Der Meulen, Performance assessment and improvement of control charts for statistical batch process monitoring, Statistica Neerlandica 60 (2006) 339–360.