



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

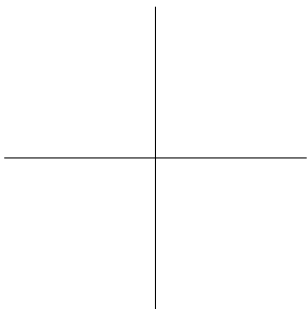
Universitat Politècnica de València
Department of Applied Statistics, Operations Research and Quality
PHD PROGRAM IN STATISTICS AND OPTIMIZATION

Advances on bilinear modeling of
biochemical batch processes

Author:
J.M. González-Martínez

Ph.D. supervisor:
Prof. A. Ferrer

Valencia, September 2015



A mis seres queridos.

This thesis has been partially supported by the Spanish Ministry of Economy and Competitiveness under the projects DPI2008-06880-C03-03 and DPI2011-28112-C04-02.

Part of the content of this thesis has been awarded the 4th Siemens Process Analytics Prize by the German Working Group "Prozessanalytik" in cooperation with Siemens.

In recognition to the innovation and technical challenge on transferring part of the developments of this thesis to industry, Shell Global Solutions International B.V. has granted the 1st Analysis and Measurements Award.

Abstract

This thesis is aimed to study the implications of the statistical modeling approaches proposed for the bilinear modeling of batch processes, develop new techniques to overcome some of the problems that have not been yet solved and apply them to data of biochemical processes. The study, discussion and development of the new methods revolve around the four steps of the modeling cycle, from the alignment, preprocessing and calibration of batch data to the monitoring of batches trajectories. Special attention is given to the problem of the batch synchronization, and its effect on the modeling from different angles: synchronization quality, changes in the correlation structure, capture of the process dynamics, parameters stability and accuracy to detect abnormal situations.

The manuscript has been divided into four blocks. First, a state-of-the-art of the latent structures based-models in continuous and batch processes and traditional univariate and multivariate statistical process control systems is carried out. In addition, a theoretical revision of the different process modeling approaches and their effects on capturing the process dynamics is presented.

The second block of the thesis is devoted to the preprocessing of batch data, in particular, to the equalization and synchronization of batch trajectories. The first section addresses the problem of the lack of equalization in the variable trajectories. The different types of unequalization scenarios that practitioners might find in batch processes are discussed and the solutions to equalize batch data are introduced. The performance of these proposals for equalization is dependent on the type of process and on the degree of unequalization the process variables are affected with. In the second section, a theoretical study of the nature of batch processes and of the synchronization of batch trajectories as a prior step to bilinear modeling is carried out. The drawbacks of the most used synchronization methods in process chemometrics are explored. The topics under discussion are i) whether the same synchronization approach must be applied to batch data in presence of different types of asynchronisms, and ii) whether synchronization is always required even though the length of the variable

trajectories are constant across batches. To answer these questions, a thorough study of the most common types of asynchronisms that may be found in batch data is done. Furthermore, two new synchronization techniques are proposed to solve the current problems in post-batch and real-time synchronization. First, a synchronization strategy that copes with variable trajectories affected by multiple asynchronisms in an optimum way, maximizing synchronization quality and reducing false alarms in the monitoring schemes is presented. The second synchronization method is the so-called Relaxed Greedy Time Warping (RGTW), which enables the real-time synchronization of batch trajectories. To improve fault detection and classification, new unsupervised control charts and supervised fault classifiers based on the information generated by the batch synchronization are also proposed.

In the third block of the manuscript, a research work is performed on the parameter stability associated with the most used synchronization methods and principal component analysis (PCA)-based Batch Multivariate Statistical Process Control methods. The results of this study have revealed that accuracy in batch synchronization has a profound impact on the PCA model parameters stability. Also, the parameter stability is closely related to the type of preprocessing performed in batch data, and the type of model and unfolding used to transform the three-way data structure to two-way. The setting of the parameter stability, the source of variability remaining after preprocessing and the process dynamics should be balanced in such a way that multivariate statistical models are accurate in fault detection and diagnosis and/or in online prediction.

Finally, the fourth block introduces a graphical user-friendly interface developed in Matlab code for batch process understanding and monitoring. This toolbox integrates, on the one hand, the two phases of the design of a monitoring scheme (model building and exploitation), and on the other hand, the four main steps in the bilinear modeling of batch processes (alignment, preprocessing, calibration and monitoring). Additionally, a simulator to generate batch data of a fermentation process of the *Saccharomyces cerevisiae* cultivation is provided. In this version of the software package, the bilinear modeling is performed using the three-way structure that contains the variable trajectories of historical batches. To perform multivariate analysis, the last developments in process chemometrics, including the methods proposed in this thesis, are implemented.

Resumen

La presente tesis doctoral tiene como objetivo estudiar las implicaciones de los métodos estadísticos propuestos para la modelización bilineal de procesos por lotes, el desarrollo de nuevas técnicas para solucionar algunos de los problemas más complejos aún por resolver en esta línea de investigación y aplicar los nuevos métodos a datos provenientes de procesos bioquímicos para su evaluación estadística. El estudio, la discusión y el desarrollo de los nuevos métodos giran en torno a las cuatro fases del ciclo de modelización: desde la sincronización, ecualización, preprocesamiento y calibración de los datos, a la monitorización de las trayectorias de las variables del proceso. Se presta especial atención al problema de la sincronización y su efecto en la modelización estadística desde distintas perspectivas: la calidad de la sincronización, los cambios en la estructura de correlación, la captura de las dinámicas del proceso, la estabilidad paramétrica y la precisión por parte de los sistemas de monitorización para detectar situaciones anómalas.

El manuscrito se ha dividido en cuatro grandes bloques. En primer lugar, se realiza una revisión bibliográfica de las técnicas de proyección sobre estructuras latentes para su aplicación en procesos continuos y por lotes, y del diseño de sistemas de control basados en modelos estadísticos multivariantes. Además, se presenta una discusión teórica de los diferentes enfoques de modelización para procesos por lotes y sus efectos en la captura de las dinámicas del proceso.

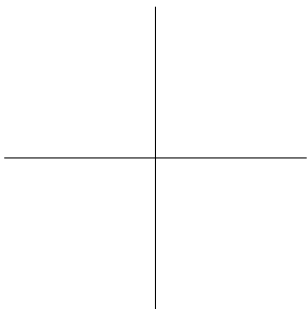
El segundo bloque del documento versa sobre el preprocesamiento de los datos, en concreto, sobre la ecualización y la sincronización. La primera parte aborda el problema de la falta de ecualización en las trayectorias de las variables. Se discuten las diferentes políticas de muestreo que se pueden encontrar en procesos por lotes y las soluciones para ecualizar las variables. La efectividad de estas propuestas son dependientes del tipo de proceso y del número de frecuencias de muestreo en las que se registran las medidas variables. En la segunda parte de esta sección, se realiza un estudio teórico sobre la naturaleza de los procesos por lotes y de la sincronización de las trayectorias como paso previo a la modelización bilineal. Se analizan en profundidad las desventajas del uso de los métodos de sincronización más

empleados en la quimiometría de procesos. Los temas bajo discusión son: i) si se debe utilizar el mismo enfoque de sincronización en lotes afectados por diferentes tipos de asincronismos, y ii) si la sincronización es siempre necesaria aún y cuando las trayectorias de las variables tienen la misma duración en todos los lotes. Para responder a estas preguntas, se lleva a cabo un estudio exhaustivo de los tipos más comunes de asincronismos que se pueden encontrar en este tipo de datos. Además, se proponen dos nuevas técnicas de sincronización para resolver los problemas existentes en aplicaciones post-mortem y en tiempo real. En primer lugar, se presenta una estrategia de sincronización que hace frente a las trayectorias de las variables afectadas por múltiples asincronismos de una manera óptima, maximizando la calidad de la sincronización y la reducción de la tasa de falsas alarmas en los sistemas de monitorización. El segundo método de sincronización es el denominado Relaxed Greedy Time Warping (RGTW), que permite la sincronización en tiempo real de las trayectorias en procesos por lotes. Para mejorar la detección de fallos y la clasificación, también se proponen nuevos gráficos de control no supervisados y clasificadores de fallos supervisados en base a la información generada por la sincronización de los lotes.

En el tercer bloque del manuscrito se realiza un estudio de la estabilidad de los parámetros asociados a los métodos de sincronización y a los métodos estadístico multivariante basados en el Análisis de Componentes Principales (PCA) más utilizados para el control de procesos. Los resultados de este estudio revelan que la precisión de la sincronización de las trayectorias tiene un impacto significativo en la estabilidad de los parámetros de los modelos PCA. Además, la estabilidad paramétrica está estrechamente relacionada con el tipo de preprocesamiento realizado en los datos de los lotes, el tipo de modelo ajustado y el despliegue utilizado para transformar la estructura de datos de tres a dos dimensiones. El ajuste de la estabilidad de los parámetros, la fuente de variabilidad que queda después del preprocesamiento de los datos y la captura de las dinámicas del proceso deben ser ajustados de forma equilibrada de tal manera que los modelos estadísticos multivariantes sean precisos en la detección y diagnóstico de fallos y/o en la predicción en tiempo real.

Por último, el cuarto bloque del documento describe una interfaz gráfica de usuario que se ha desarrollado en código Matlab para la comprensión y la supervisión de procesos por lotes. Esta herramienta informática integra, por un lado, las dos fases del diseño de un sistema de monitorización (construcción y explotación del modelo), y por otro lado, las cuatro fases principales en el modelado bilineal de procesos por lotes (alineamiento, preprocesamiento, calibración and monitorización). Además, un simulador está accesible para generar datos de lotes de un proceso de fermentación del cultivo de *Saccharomyces cerevisiae*. En esta versión del software, la

modelización bilineal se realiza a partir de la estructura tridimensional que contiene las trayectorias de las variables de lotes históricos. Para llevar a cabo los análisis multivariantes, se han implementado los últimos desarrollos en la quimiometría de procesos, incluyendo los métodos propuestos en esta tesis doctoral.



Resum

Aquesta tesi doctoral té com a objectiu estudiar les implicacions dels mètodes de modelització estadística proposats per a la modelització bilineal de processos per lots, el desenvolupament de noves tècniques per resoldre els problemes encara no resolts en aquesta línia de recerca i aplicar els nous mètodes a les dades dels processos bioquímics. L'estudi, la discussió i el desenvolupament dels nous mètodes giren entorn a les quatre fases del cicle de modelització, des de l'alineació, preprocessament i el calibratge de les dades provinents de lots, a la monitorització de les trajectòries. Es presta especial atenció al problema de la sincronització per lots, i el seu efecte sobre el modelatge des de diferents angles: la qualitat de la sincronització, els canvis en l'estructura de correlació, la captura de la dinàmica del procés, els paràmetres de l'estabilitat i la precisió per detectar situacions anòmales.

El manuscrit s'ha dividit en quatre grans blocs. En primer lloc, es realitza una revisió bibliogràfica dels principals mètodes basats en tècniques de projecció sobre estructures latents en processos continus i per lots, així com dels sistemes de control estadístics multivariats. A més, es presenta una revisió teòrica dels diferents enfocaments de modelat de processos i els seus efectes sobre la captura de la dinàmica del procés.

El segon bloc del document es dedica a la preprocessament de les dades provinents de lots, en particular, l'equalització i la sincronització. La primera part aborda el problema de la manca d'equalització en les trajectòries de les variables. Es discuteixen els diferents tipus d'escenaris en que les variables estan mesurades a distints intervals i les solucions per equalitzar-les en processos per lots. L'efectivitat d'aquestes propostes d'equalització són dependents del tipus de procés i del nombre de freqüències de mostreig en què es registren les mesures de les variables. A la segona part d'aquesta secció es porta a terme un estudi teòric de la naturalesa dels processos per lots i de la sincronització de les trajectòries de lots com a pas previ al modelatge bilineal. Els inconvenients dels mètodes de sincronització més utilitzats en la quimiometria de processos són analitzats en profunditat. Els temes en discussió són: i) si el mateix enfocament de sincronització ha de ser aplicat a les dades del lot en presència de diferents tipus de

asincronismes, i ii) si la sincronització sempre es requereix tot i que la longitud de les trajectòries de les variables són constants en tots els lots. Per respondre a aquestes preguntes, es du a terme un estudi exhaustiu dels tipus més comuns de asincronismes que es poden trobar en les dades provinents de lots. A més, es proposen dues noves tècniques de sincronització per resoldre els problemes existents la sincronització post-mortem i en temps real. En primer lloc, es presenta una estratègia de sincronització que fa front a les trajectòries de les variables afectades per múltiples asincronies d'una manera òptima, maximitzant la qualitat de la sincronització i la reducció de falses alarmes en els sistemes de monitorització. El segon mètode de sincronització és l'anomenat Relaxed Greedy Time Warping (RGTW), que permet la sincronització en temps real de les trajectòries de procés per lots. Per millorar la detecció i la classificació de anomalies, també es proposen nous gràfics de control no supervisats i classificadors de falla supervisats dissenyats en base a la informació generada per la sincronització de lots.

En el tercer bloc del manuscrit es realitza un treball de recerca sobre l'estabilitat dels paràmetres associats als mètodes de sincronització i als mètodes estadístics multivariats basats en l'Anàlisi de Components Principals (PCA) més utilitzats per al control de processos. Els resultats d'aquest estudi revelen que la precisió en la sincronització per lots té un profund impacte en l'estabilitat dels paràmetres dels models PCA. A més, l'estabilitat paramètrica està estretament relacionat amb el tipus de preprocessament realitzat en les dades provinents de lots, el tipus de model i el desplegament utilitzat per transformar l'estructura de dades de tres a dos dimensions. L'ajust de l'estabilitat dels paràmetres, la font de variabilitat que queda després del preprocessament i la captura de la dinàmica de procés ha de ser equilibrada de tal manera que els models estadístics multivariats són precisos en la detecció i diagnòstic de fallades i/o en la predicció en línia.

Finalment, el quart bloc del document introdueix una interfície gràfica d'usuari que s'ha dissenyat e implementat en Matlab per a la comprensió i la supervisió de processos per lots. Esta ferramenta informàtica integra, d'una banda, les dues fases del disseny d'un sistema de monitorització (construcció i explotació del model), i per altra banda, els quatre passos principals en el modelatge bilineal de processos per lots (alineament, preprocessament, calibratge i monitorització). A més, aquesta ferramenta informàtica proporciona un simulador per generar les dades del lot d'un procés de fermentació del cultiu de *Saccharomyces cerevisiae*. En aquesta versió del programari, la modelització bilineal es realitza a partir de l'ús de l'estructura de tres dimensions que conté les trajectòries de les variables registrades en lots històrics. Per dur a terme aquestes anàlisis multivariats, s'han implementat els últims desenvolupaments en la quimiometria de processos, incloent-hi els mètodes proposats en aquesta tesi.

Agradecimientos

Mi primer agradecimiento va dirigido al que ha sido mi mentor, maestro, amigo y compañero de aventuras tanto en el ámbito profesional como en el personal, mi supervisor de tesis, Alberto Ferrer. Nunca podré agradecerle todo lo que has hecho por mí. Muchísimas gracias por estar siempre cuando lo he necesitado. ¡Jamás lo olvidaré!

Special mention deserve my colleagues and office roommates I had during my two research stays at University of Amsterdam and Shell. First, I would like to thank Johan Westerhuis for his commitment to supervise my research work on batch MSPC, and for his comradeship and affection shown during my stay at BDA group. Likewise, I would like to extend my acknowledgements to all my colleagues and friends who have made my life abroad easier and joyful during my stay in Shell and nowadays, and all those who have supported me in my down moments, which there have been plenty of them.

¡Qué decir de mis compañeros de aventuras en el despacho de doctorandos! Sin vosotros, la larga y dura travesía de la tesis doctoral no hubiera sido la misma, ni mucho menos amena. Me gustaría recordar de forma cariñosa a José Manuel. Tú pasaste de ser mi profesor de estadística en la carrera, a ser compañero de trabajo en el grupo de investigación, amigo y confidente, a ser como un hermano para mí. Siempre tendrás un hueco en mi corazón aunque estemos lejos. ¡Gracias por todo, hermano! Una mención especial también merece José Camacho, quién ha sido compañero de peripecias científicas en una parte de esta tesis doctoral y en el proyecto del libro que nos embarcamos hace ya hace unos años. Gracias por tu ayuda, consejo, contribución en esta tesis, y sobretodo, ¡por tu amistad!

A nivel más personal, quisiera agradecer a todos mis amigos y seres queridos, empezando por mis amigos de la niñez y adolescencia Nacho, Conxi, Emilio, Juan Carlos, Víctor, Patricia y a todos aquellos que pasaron en algún momento por mi vida, por todo el apoyo recibido y por su amistad.

Un párrafo aparte merece las personas más importantes de mi vida: mi familia. Papá y mamá, siempre os tendré en mis recuerdos y en mi corazón. Desgraciadamente os perdí durante el doctorado, os fuisteis para no volver.

Me disteis la mejor herencia que me podrías haber dado, educación, amor, cariño y las herramientas para dirigir mi futuro con honradez y tenacidad. Ojalá estuvierais aquí para compartir conmigo el final de esta etapa de mi vida. Os quiero y querré por siempre. También a mi hermano por estar ahí siempre, por darme el apoyo y los ánimos en los momentos difíciles y desconsuelo que he tenido. *With special gratefulness I would like to acknowledge my Dutch family (including my lovely Lizzy), which has given me what I have missed since my parents passed away. Without you all, I would have not reached this moment. Thanks all for you unconditional support.* Por último, y no por ello menos importante, todo lo contrario, agradecer a mi pareja Emma, quien ha sido artífice del último empujón que le he dado a mi tesis doctoral. Gracias por todos los momentos que hemos compartido con amor y respeto. Te quiero muchísimo y espero compartir muchísimos más momentos felices contigo. *Ik hou van jou, mijn lieverd!*

Contents

Justification, Objectives, Summary and Contributions	xvii
I Introduction	1
1 State-of-the-art	3
1.1 Batch processes	4
1.2 Batch process modeling	6
1.2.1 Knowledge-based models	8
1.2.2 Data-driven models	8
1.2.3 Hybrid models	9
1.3 Bilinear modeling cycle of batch processes	10
1.3.1 Data alignment	11
1.3.2 Data preprocessing	12
1.3.3 Bilinear modeling	14
1.3.4 Monitoring	21
2 Material and Methods	33
2.1 Hardware	33
2.2 Software	33
2.3 Simulated Processes	33
II Preprocessing of Batch Data	39
3 Batch data equalization	41
3.1 Introduction	42
3.2 Challenges in batch equalization	43
3.3 Equalization of variables within a batch	47
3.3.1 Discarding intermediate values	50
3.3.2 Estimating missing values	52
3.3.3 Rearranging data	59

3.4	Multi-rate system	61
3.5	Conclusions	66
4	Batch synchronization	69
4.1	Introduction	70
4.2	Synchronization approaches	72
4.2.1	Indicator Variable	74
4.2.2	Time Linear Expanding/Compressing	80
4.2.3	Dynamic Time Warping	84
4.3	Warping Information	97
4.4	Comparative study of the synchronization techniques	97
4.4.1	Effect of synchronization on the correlation structure	99
4.5	Conclusions	109
5	Relaxed Greedy Time Warping for BMSPC	113
5.1	Introduction	114
5.2	Relaxed Greedy Time Warping algorithm	115
5.2.1	Enhanced global constraints	116
5.2.2	The RGTW algorithm	118
5.2.3	Cross-validation for the estimation of the RGTW parameters	122
5.2.4	Trade-offs of RGTW and comparison with online DTW	124
5.3	Warping information for enhanced BMSPC	125
5.3.1	NOC warping information-based control charts	126
5.3.2	Fault classification procedures	127
5.4	Application of batch synchronization	131
5.4.1	Offline synchronization	133
5.4.2	Online synchronization	138
5.5	Use of warping information	144
5.5.1	NOC-WICC for process monitoring	146
5.5.2	Supervised faulty WICC	147
5.5.3	PLSDA-based classifier	150
5.5.4	SIMCA-based classifier	152
5.6	Conclusions	153
6	Batch synchronization in scenarios of multiple asynchronisms	157
6.1	Introduction	158
6.2	Multisynchro approach for batch synchronization	159
6.2.1	Asynchronism detection	161
6.2.2	Specific batch synchronization	163
6.3	Iterative batch synchronization/abnormalities detection procedure	167

6.4 Failure modes and complex non-linear interactions 170
 6.4.1 Failure modes of the asynchronism categorization . . 171
 6.4.2 Failure modes of the iterative batch synchronization/abnormalities detection procedure 173
 6.5 Material and methods 175
 6.6 Results 178
 6.6.1 Performance of synchronization based on DTW and Multisynchro 178
 6.6.2 Synchronization of batches with equal duration . . . 185
 6.7 Conclusions 189

III Modeling of Batch Process Data 191

7 Parameters stability on bilinear process modeling 193
 7.1 Introduction 194
 7.2 Material and methods 195
 7.3 Effects of batch synchronization 200
 7.4 Effect of the rearranging methods 207
 7.4.1 Batch-wise unfolding 207
 7.4.2 Variable-wise unfolding 211
 7.4.3 Batch-dynamic unfolding 214
 7.4.4 *K*-models 215
 7.4.5 Adaptive hierarchical *K*-models 217
 7.5 Conclusions 220

8 Implications of batch synchronization in process monitoring 223
 8.1 Introduction 224
 8.2 Material 226
 8.3 Effects of asynchronisms in synchronization quality 231
 8.4 Effects of synchronization in process monitoring 232
 8.5 Conclusions 241

IV Application 243

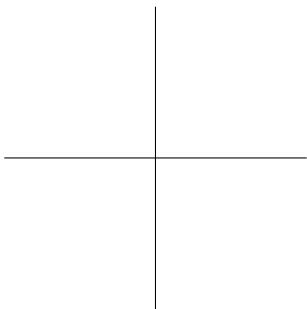
9 Modeling process dynamics through multivariate models and control charts. 245
 9.1 Introduction 246
 9.2 Nature of data in continuous phosphorus removal processes 249
 9.3 Batch modeling for continuous processes 251
 9.4 Conclusions 263

10 MVBatch Toolbox for bilinear batch process modeling	265
10.1 Introduction	266
10.2 Modeling cycle of batch processes	267
10.2.1 Data alignment	267
10.2.2 Data modeling	267
10.2.3 Design of the monitoring scheme	269
10.3 Software specification and requirements	271
10.4 Data set	271
10.5 Description of the MVBatch Toolbox	272
10.5.1 Screening	273
10.5.2 Alignment	274
10.5.3 Modeling	282
10.5.4 Monitoring	291
10.6 Conclusions	296
V Conclusions	297
11 Conclusions	299
VI Appendices	315
A Projection methods to latent structures for BMSPC	325
A.1 Bilinear models	325
A.1.1 Principal Component Analysis	325
A.1.2 Partial Least Squares	327
A.1.3 Principal Component Regression	328
A.2 <i>N</i> -way modeling	329
A.2.1 Parallel Factor Analysis	329
A.2.2 Tucker-3	331
A.2.3 Other models	332
A.3 Selection of the number of principal components	332
A.4 Online application of projection methods to latent structures	337
B Variance-covariance maps for process dynamics visualization	343
C Monitoring of multiple asynchronous batches	347
C.1 Synchronization and monitoring of NOC batches	348
C.2 Synchronization and monitoring of type I faulty batches	358
C.3 Synchronization and monitoring of type II faulty batches	368
C.4 Synchronization and monitoring of type III faulty batches	378

References

389





Justification, Objectives, Summary and Contributions

Batch processes can be defined as finite duration processes that involve well-defined multiple process operations in a predetermined sequence, procedure, and time. First, a specific mix of raw materials or reactants are initially charged in a vessel, which has an internal stirrer to keep reactants well mixed. Then, materials are processed during a finite duration with a time-varying behavior. Optionally, intermediate reactants can be added to the vessel at a proper time. Once raw materials have been converted into desired products, they are finally discharged from the vessel. This manufacturing mode has some inherent advantages, such as processing flexibility, repetitive nature, ability to handle variations in market demand patterns and handling hazardous ingredients or final products, that make batch processes well suited for the production of low volume of high value-added products. These features are in line with the ultimate goals of hi-tech and biochemical industries, which are to reduce time-to-market and manufacturing costs, comply with regulatory and strict requirements, improve yields and quality, increase throughput and reduce energy usage.

One of the most critical challenges current industries must address in the next decades is ensuring an acceptable product quality and productivity. The lack of in-depth real-time knowledge about the process state, forces manufacturers to operate at too conservative, suboptimal and not intensified regimes. To overcome this situation two main objectives need to be pursued: i) better process understanding to facilitate risk-based regulatory decisions and innovation, and ii) better handling of product quality assessment and accomplishment. These goals are in line with the guidelines released by the *United States Food and Drugs Administration* (FDA) in 2004, which encourage industries to detect the presence of variation, calculate the degree of variation, understand the sources of variation, and determine to what extent this variation may impact the process and ultimately the product features and quality.

Batch processes are more complex to deal with than continuous pro-

cesses. This fact is owing to the mode of operation and the feature of the control systems. In contrast to continuous processes, batch processes have no steady-state operating conditions. The setpoint and control signals correspond to time-varying profiles -so called recipes-, and both initial and processing conditions have to be controlled, requiring a higher effort to understand, model, control, and monitor this kind of processes. However, batch processes are very flexible in such a way that production of different products using the same industrial hardware and only modifying the recipe of raw materials can be achieved. Also, it permits to transfer the underlying scientific knowledge of the research quickly, thereby yielding early benefits to recover the research costs. The latter issues together with the possibility to produce high value-added products make batch processes interesting to be used at large scale in industry.

For the analysis of the performance of batch processes, typically three types of data are collected: initial conditions, process variable trajectories, and quality and productivity measurements. The first category includes initial charge recipe and any process measurements registered before the start of the batch, which may have an impact on product quality. Second, measurements of process variables sampled at a relatively fast sampling frequency is available for each batch. Commonly, the duration of batches is variable to a certain extent, depending on the nature of the process and the manufacturing conditions. Finally, a set of measurements belonging to quality features are optionally recorded; typically at the end of each batch because of their high cost. As a consequence of the complex characteristics of batch data, such as high correlation, rank deficiency, low signal-to-noise ratio, missing values and/or nonlinear and time-varying dynamics, online quality prediction and control in batch processes suffer from lack of reproducibility due to batch-to-batch variations. In this context, the application of classical *Statistical Process Control* (SPC) techniques for process modeling is inadequate.

In the light of meeting the FDA's requirements, the application of multivariate statistical projection methods in the *Batch Multivariate Statistical Process Control* (BMSPC) framework that copes with the challenges of batch processes plays a crucial role. This provides more reliable process modeling and sophisticated tools to design online monitoring schemes that are capable of detecting complex batch abnormalities incipiently and safely. Despite the work done in the last years, there are still some topics in BMSPC that deserve research. They have been the drivers to define the following main objectives of this dissertation.

I. Identify the limitations of the different preprocessing and bilinear process modeling approaches.

The most used bilinear batch modeling methods will be thoroughly studied and compared to establish which approach is more appropriate based on the characteristics of the process and the goals pursued. In particular, this milestone includes:

- Theoretical study of the preprocessing and synchronization techniques.
- State-of-the-art of bilinear modeling methods for analyzing batch data.
- Study of the effects of synchronization and bilinear batch process modeling on process monitoring.

II. Propose new techniques that overcome the limitations pinpointed.

With the aim of overcoming the limitations found in the modeling cycle of batch data, new methods will be proposed and rigorously described along this dissertation. The main focus will be on the synchronization of batch trajectories and its effects on the rest of the modeling steps. Special emphasis will be given to the validation of the statistical parameters as well as the automation of the algorithms for its straightforward application to real processes.

III. Apply and compare the methods developed by using different industrial scenarios.

One of the most important goals of this work is to test the methods proposed in this dissertation, providing a comparison with the existing methods. For this purpose, several simulated data sets from two different types of biochemical processes will be used in order to highlight the differences in comparison to state-of-the-art methods, focusing the attention on bilinear process modeling and monitoring.

Based on the objectives of the thesis, the manuscript is structured as follows. The first part of the document is devoted to introduce the state-of-the-art of the bilinear statistical modeling cycle for the design of monitoring schemes for post-batch and real-time process monitoring (Chapter 1). First, the focus is on the different steps for model building: data alignment (including data equalization and synchronization), data preprocessing, transformation from three-way to two-way structure,

and statistical calibration and monitoring. The most used chemometrics strategies and techniques in each of the modeling steps will be described, stressing the advantages, drawbacks and limitations of their use in different contexts in batch processes. Thereafter, an extended review of the projection techniques to latent structures used for batch process understanding, monitoring, prediction and optimization is provided in Appendix A in connection to the first chapter. Chapter 2 describes the hardware and software tools used to implement the algorithms proposed in this work and those proposed in the literature for subsequent comparisons. A detailed description of the data sets employed along the manuscript to illustrate the performance of these novel methods is presented.

Along the second part, the problem of the equalization and synchronization of batch trajectories are addressed. In Chapter 3 the differences between non-equalized and asynchronous trajectories are stressed, and the equalization of batch data is studied in depth. The different types of unequalization scenarios that practitioners might find in batch processes are discussed: different sampling rate per stage and same policy for all variables and batches, multi-rate sampling in process variables and similar policy among batches, and multi-rate sampling both in process variables and batches. Three different strategies to solve the problem of different sampling frequencies among process variables are introduced: discarding intermediate values, estimating missing values and rearranging data. The application of these techniques to batch data for equalization is crucial to continue the bilinear modeling with the next step: the batch synchronization.

Chapter 4 is aimed to study the nature of batch processes and the problem of batch synchronization. First, a study of the state-of-the-art in batch synchronization is done. The methods most used in process chemometrics are categorized and reviewed: i) methods based on compressing/expanding the raw trajectories using linear interpolation either in the batch time dimension or in an indicator variable dimension; ii) methods based on feature extraction; and iii) methods based on stretching, compressing and translating pieces of trajectories. The drawbacks of these methods are explored. In particular, the application of these techniques in real-time applications are analyzed. The simulation study shows how the inaccuracy of the synchronization techniques to synchronize batch trajectories may have severe effects on the correlation structure, the actual process variability, the parameter stability and the accuracy of the monitoring schemes to detect abnormalities. In this chapter, a comparative study of the selected synchronization methods on accurately capturing the time-varying process dynamics is performed. This study provides insight into the importance of precisely aligning the main features of the variables to mitigate the addition of artifacts in the trajectories, which may affect the actual relationships of

the variables over time.

One of the most critical aspects in batch synchronization is the application of these techniques in real-time applications. To overcome the problems of the post-batch proposals for synchronization, a new time warping algorithm called Relaxed Greedy Time Warping (RGTW) is proposed in Chapter 5. This new technique is based on a relaxed greedy strategy to find the optimal combinations of non-linear modifications of the trajectories that ensure a suboptimal synchronization. The deployment of this synchronization algorithm in the monitoring systems allows us to successfully perform the batch synchronization in such a way that the uncertainty in the monitoring statistics and predictions are reduced, and therefore, the false alarm rate as well. Furthermore, the use of the warping information obtained from the RGTW-based batch synchronization both for batch process monitoring and supervised fault classification are addressed. Two new proposals are introduced to strengthen the use of the warping information in the monitoring schemes. First, an unsupervised control chart based on the warping profiles from batches ran under normal operating conditions (NOC) (NOC-WICC) is proposed as a complementary tool to the traditional multivariate statistical control charts. Second, supervised warping information-based control charts (faulty WICC) and the design of classifiers based on the warping information are presented. The use of these complementary statistical tools enables a better monitoring of incoming batches by providing more information to technologists to interrogate the statistical models in abnormal manufacturing situations.

Other topics under discussion in this thesis are 1) whether the same synchronization approach must be applied to batch data in presence of different types of asynchronisms, and 2) whether synchronization is always required even though the length of the variable trajectories are constant across batches. To answer these questions, a thorough study of the most common types of asynchronisms that may be found in batch data is done. Namely, there are four classes of asynchronisms: (i) batches with equal duration but key process event not overlapping at the same time point in all batches; (ii) batches with different duration and process pace caused by external factors; (iii) batches with different duration due to incompleteness of the last process stage, irrespective of whether the process pace is the same or not across batches; and (iv) batches with different duration due to delay at the start but batch trajectories showing the same evolution pace after. To cope with the complex number of asynchronisms batches may be affected with, Chapter 6 describes a novel synchronization strategy called Multisynchro, which also maximizes synchronization quality and minimizes the false alarm rate. The new approach also includes a procedure that performs abnormality detection and batch synchronization in an iterative way. This prevents batch abnormalities from affecting

synchronization quality. The Multisynchro approach is proved to clearly outperform the standard approach of applying the same synchronization procedure, irrespective of the type of batch asynchronism.

The third part of this document is devoted to thoroughly study the entire "value chain" of BMSPC systems based in *Principal Component Analysis* (PCA) models: from data alignment, preprocessing, transformation, modeling to monitoring. In previous research works, the most used batch processes modeling approaches were studied from two different perspectives: process dynamics and online prediction. In the first section of this block, these studies will be extended to the so-called parameter stability, i.e. the estimates of the model parameters. This is a paramount and often overlooked aspect in the development of a monitoring system. When a PCA model is built in BMSPC, low model parameters uncertainty is desired to ensure a reduced amount of noise in the model. Noise affects the performance of the monitoring system, reducing its detection capability. Hence, the study of the effects of bilinear batch process modeling on the parameter stability, and consequently on the performance of monitoring schemes, is needed. For this purpose, a comparison of the most used modeling approaches and synchronization methods in terms of parameter stability is performed in Chapter 7. The different methods for rearranging the three-way batch data array in a number of two-way arrays are taken into consideration: i) to unfold the three-way array in a single two-way array, e.g. variable-wise unfolding, batch dynamic and batch-wise; ii) to fit K PCA models, iii) to use an adaptive approach where the current and past information are combined. In addition, the effect of the batch synchronization on the parameter stability is studied in batch data synchronized by the most used synchronization methods by the chemometric community: the Indicator Variable (IV), Dynamic Time Warping (DTW) and Relaxed Greedy Time Warping (RGTW) algorithms, and Time Linear Expanding/-Compressing (TLEC)-based method throughout both the batch run and time periods limited by key process events. Results are discussed in connection with conclusions from previous research that addressed the capture of the process dynamics and the online prediction of quality properties.

Another paramount subject under discussion in the second section of this block are the effects of batch synchronization in batch monitoring. Several publications express the belief that batch synchronization is only required when the duration of the variable trajectories varies across batches. In fact, commercial software packages as SIMCA Release 13.0.3 only requires the synchronization of batch trajectories when they differ in time. Also, ProMV Batch Edition Release 13.02 does not compulsorily demand the application of synchronization algorithms when batch data have equal length. However, batch data can only be considered synchronized if and only if the main features of the trajectories (or key process events) are

overlapped across batches, irrespective of whether the batch duration is the same for all batches or not. A number of proposals for dealing with asynchronous batches can be found in the literature: from methods based on stretching/expanding the raw trajectories using linear interpolation either in the batch time dimension or in an indicator variable dimension to methods based on stretching, compressing and translating pieces of trajectories. Time stretching/expanding is one of the most used synchronization techniques in process chemometrics and implemented in commercial and scientific software because of its simplicity (e.g. in SIMCA Release 13.0. as unique synchronization procedure and in ProMV Batch Edition Release 13.02 as one of the provided synchronization methods). Nonetheless, the application of this approach may be dangerous due to the underlying assumptions that are seldom fulfilled in batch processes: i) linear process pace, ii) all batch runs are completed, and iii) batches with equal duration are considered as synchronized. In Chapter 8, a comparative study between the two main families of synchronization methods in scenarios of multiple asynchronisms is carried to get insight into three key aspects: i) importance of ensuring the overlap of the key process events, ii) the severity of the type of asynchronism present in batch data to select the type of synchronization method, and iii) the effect of inadequate synchronization in scenarios of multiple asynchronism on process monitoring. The results of this research reveal that the Multisynchro approach proposed in Chapter 6 is a promising technique to overcome the complex asynchronisms in batch processes, becoming a clear outperforming alternative to the techniques implemented in commercial software packages that can be used in post-batch and real-time applications.

The fourth part of this manuscript is devoted to apply some of the most successful state-of-the-art approaches used in process chemometrics as well as the methodologies proposed in this thesis. First, the importance of the modeling approach and its influence in online fault detection, isolation and diagnosis using projection methods to latent structures is discussed in Chapter 9. The application of all the modeling steps of batch processes explained along the thesis is illustrated in Chapter 10. For this purpose, a Matlab graphical user-friendly interface called MVBatch is designed for batch process understanding and monitoring, which is entirely described in Chapter 10. This tool integrates, on the one hand, the two phases for the design of a monitoring scheme (model building -exploratory data analysis and post-batch process monitoring- and model exploitation -real-time process monitoring), and on the other hand, the four main steps in the bilinear modeling of batch processes: data screening, alignment, calibration and monitoring. All the novel algorithms are accessible through the main window of the software package that represents the modeling phases. In addition, the simulator of batch data of a fermentation process

based on the biological model of the yeast *Saccharomyces cerevisiae* is provided. The simulated data used to illustrate the performance of the new techniques hereby proposed and used in the comparative studies of this thesis are generated by this tool. Finally, a case study of batch multivariate statistical process control based on principal component analysis is provided to illustrate the performance of MVBatch, emphasizing the differences with other commercial software packages.

To sum up, the main contributions of this thesis are:

- The comparative study of the most used synchronization techniques in process chemometrics from different aspects: synchronization quality, capture of the actual process dynamics, parameter stability, and accuracy of the monitoring schemes in fault detection in scenarios of multiple asynchronisms.
- The development of a novel synchronization algorithm called RGTW that enables the design of monitoring schemes for real-time deployment in batch processes characterized by the presence of asynchronisms. This new algorithm notably reduces the false alarm rate produced by the inaccuracy of traditional synchronization algorithms.
- The implications of the use of the warping profiles derived from synchronization in BMSPC. From this study, guidelines are provided to use this information for building unsupervised control charts and fault classifiers when a rich faulty batch database is available.
- The design of a novel synchronization strategy called Multisynchro that copes with asynchronous batch trajectories affected by complex combinations of asynchronisms in an automatic manner. The use of this synchronization framework reduces the addition of artifacts affecting the correlation structure of the variables, the parameter stability and the accuracy of the monitoring systems for detecting abnormal situations in batch processes.
- The comparison and study of the most used preprocessing and principal component analysis-based Batch Multivariate Statistical Process Control methods (namely single-model, K -models, and hierarchical model approaches) from a perspective not yet studied, but crucial for the performance of monitoring schemes: the model parameter stability. This study provides insight into the parameters that must be properly tuned up to obtain optimum multivariate models for

process understanding, optimization, online prediction and accurate monitoring systems for process monitoring.

- The design and implementation of a Matlab user-friendly interface named MVBatch that integrates the recent developments in process chemometrics for analyzing, optimizing and monitoring batch processes.

The following written contributions have been produced from the research work conducted in this thesis:

Refereed Journal Papers

- [1] J.M. González-Martínez, A. Ferrer and J.A. Westerhuis. Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping, *Chemometrics and Intelligent System Laboratory*, 105:195–206, 2011.
- [2] J.M. González-Martínez, J.A. Westerhuis and A. Ferrer. Using warping information for batch process monitoring and fault classification, *Chemometrics and Intelligent System Laboratory*, 127:210–217, 2013.
- [3] J.M. González-Martínez, J. Camacho and A. Ferrer. Bilinear modelling of batch processes. Part III: Parameter Stability, *Journal of Chemometrics*, 28(1):10–27, 2014. **Awarded with the 4th Siemens Process Analytics Prize for an outstanding publication in the field of Process Analytics by the German Working Group "Prozessanalytik" in cooperation with Siemens.**
- [4] J.M. González-Martínez, O.E. de Noord and A. Ferrer. Multisynchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms, Special Issue in honor to Bruce Kowalski in *Journal of Chemometrics*, 28: 462:475, 2014.
- [5] J.M. González-Martínez, R. Vitale, O.E. de Noord and A. Ferrer. Effect of synchronization on bilinear batch process modeling, *Industrial & Engineering Chemistry Research*, 53(11): 4339–4351, 2014.
- [6] J.M. González-Martínez, J. Camacho and A. Ferrer. MVBatch Toolbox: a MATLAB graphical interface for bilinear batch process modeling. In elaboration.
- [7] J.M. González-Martínez, J. Camacho and A. Ferrer. A comparison of synchronization methods based on the correlation structure of batch data. In elaboration.

- [8] J. Camacho, J.M. González-Martínez and A. Ferrer. Equalization of batch data: challenges and solutions. In elaboration.
- [9] J.M. González-Martínez, J. Camacho and A. Ferrer. Modeling Time-varying Process Dynamics through Latent Models and Control Charts. In elaboration.

Chapter of books

- [10] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 1: Introduction to Batch Processing. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [11] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 2: Latent Structures based models. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [12] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 3: Batch Process Data. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [13] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 4: Bilinear Modeling of Batch Process Data. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [14] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 5: Batch Process Analysis and Understanding. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [15] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 6: On-line Batch Process Monitoring. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.

Conference presentations and posters

- [16] J.M. González-Martínez and A. Ferrer. A comparison of different methods for synchronization of batch trajectories. *In proceedings of 11th Scandinavian Symposium on Chemometrics*, page 83, Loen (Norway), 2009

- [17] A. Ferrer, J. Camacho and J.M. González-Martínez. Issues on Batch Multivariate Statistical Process Control. *In proceedings of 2010 Eastern Analytical Symposium and Exposition*, page 56, New Jersey (USA), 2010.
- [18] J.M. González-Martínez, A. Ferrer and J.A. Westerhuis. Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping. *In proceedings of 12th Conference on Chemometrics in Analytical Chemistry*, pages 229-230, Antwerpen (Belgium), 2010.
- [19] J.M. González-Martínez, A. Ferrer, F. Arteaga, D. Aguado and J. Ribes. Error-proof Latent-based Multivariate Process Control of a Continuous Biological Removal Process. *In proceedings of 12th Conference on Chemometrics in Analytical Chemistry*, pages 149-150, Antwerpen (Belgium), 2010.
- [20] J.M. González-Martínez, A. Ferrer, F. Llaneras, M. Tortajada and J. Picó. Metabolic Flux understanding of *Pichia Pastoris* grown on heterogenous culture media. *In proceedings of 12th Conference on Chemometrics in Analytical Chemistry*, pages 86, Antwerpen (Belgium), 2010.
- [21] J.M. González-Martínez and A. Ferrer. Metabolic flux understanding: a grey modeling approach. *In proceedings of the Workshop on Chemometrics for Young Researchers*, pages 47-38, A Coruña (Spain), 2011.
- [22] J.M. González-Martínez and A. Ferrer. Batch synchronization: a paramount step before bilinear modeling in batch multivariate statistical process control. *In proceedings of the Workshop on Chemometrics for Young Researchers*, page 32, A Coruña (Spain), 2011.
- [23] A. Ferrer, J. Camacho and J.M. González-Martínez. Chemometrics tools for process understanding and monitoring with bilinear models: a review of novel proposals. *In proceedings of the 5th International Chemometrics Research Meeting*, page 11, Nijmegen (The Netherlands), 2011.
- [24] J.M. González-Martínez, A. Ferrer and J.A. Westerhuis. Relaxed-Greedy Time Warping: a new tool for real-time synchronization of batch trajectories for batch MSPC. *In proceedings of the 5th International Chemometrics Research Meeting*, page 32, Nijmegen (The Netherlands), 2011.

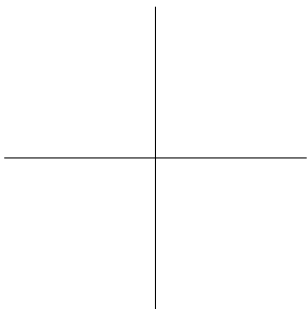
- [25] J.M. González-Martínez, A. Ferrer, F. Arteaga, D. Aguado and J. Ribes. Multivariate Statistical Process Control of a Continuous Biological Removal Process: Designing efficient monitoring schemes robust to sensor malfunctioning. *In proceedings of the 2nd European Conference on Process Analytics and Control technology (EUROPACT)*, page 149, Glasgow (UK), 2011.
- [26] O.E. de Noord and J.M. González-Martínez. Recent developments in multivariate data analysis and monitoring of chemical manufacturing processes. *In proceedings of the 2nd African-European Conference on Chemometrics (AFRODATA 2012)*, Stellenbosch (South Africa), 2012.
- [27] J.M. González-Martínez, J. Camacho and A. Ferrer. Enhancement of batch process understanding and monitoring: a matter of parameters stability. *In proceedings of the 13th Conference on Chemometrics in Analytical Chemistry*, page 39, Budapest (Hungary), 2012.
- [28] J.M. González-Martínez, O.E. de Noord and A. Ferrer. A novel approach for batch synchronization in scenarios of multiple asynchronies. *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 28, Djurönäset (Sweden), 2013.
- [29] A. Ferrer, J.M. González-Martínez and J. Camacho. Practical implications of synchronization, preprocessing and bilinear modelling of batch processes for MSPC. *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 26, Djurönäset (Sweden), 2013.
- [30] J.M. González-Martínez, R. Vitale, O.E. de Noord and A. Ferrer. Does synchronization matter in Batch Multivariate Statistical Process Control? *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 58, Djurönäset (Sweden), 2013.
- [31] J.M. González-Martínez, J. Camacho, O.E. de Noord and A. Ferrer. Equalization and data-driven compression as a prior step to batch modelling. *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 59, Djurönäset (Sweden), 2013.

Software

- [32] J. Camacho, J.M. González-Martínez and A. Ferrer. MVBatch Toolbox for Matlab, <https://mseg.webs.upv.es/Software.html> [accessed 2014].

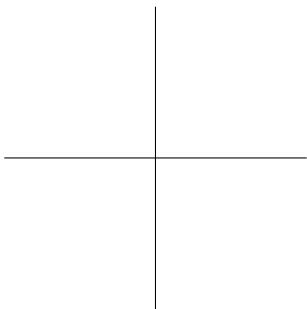
Additional contributions derived from the research work done during the PhD program are:

- [33] J.M. González-Martínez and O.E. de Noord. Enhanced Process Understanding and Monitoring of Manufacturing Batch Processes of Specialty Chemicals, internal report for Shell Global Solutions International B.V., 2013.
- [34] J.M. González-Martínez, A. Folch-Fortuny, F. Llaneras, M. Tortajada, J. Picó and A. Ferrer. Metabolic Flux Understanding of *Pichia pastoris* Grown on Heterogeneous Culture Media, *Chemometrics and Intelligent Laboratory Systems*, 134: 89–99, 2014.



Part I

Introduction



State-of-the-art

Part of the contents of this chapter has been included in the following publications:

- [10] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 1: Introduction to Batch Processing. *Batch Processes: Monitoring and Process Understanding*, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [11] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 2: Latent Structures based models. *Batch Processes: Monitoring and Process Understanding*, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [13] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 4: Bilinear Modeling of Batch Process Data. *Batch Processes: Monitoring and Process Understanding*, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [15] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 6: Online Batch Process Monitoring. *Batch Processes: Monitoring and Process Understanding*, *Wiley-VCH Verlag GmbH*, publication due in 2016.

1.1 Batch processes

Industrial procedures involving raw material and mechanical steps in the manufacture of a product are usually carried on a very large scale and in different ways. They can be broadly classified as continuous or batch procedures. In both cases the goal is to operate the process so that desired end-product specifications are met. The achievement of these specifications involves technical decisions at the process operational level, which are related to reaching defined values in a set of relevant process variables through control systems.

Batch processes play an important role in the production of high added-value specialties at key economic sectors such as industry, agriculture and biotechnology. The manufacture of these products needs a more sophisticated production strategy than the one required in a continuous process. In general, batch processes can be defined as finite duration processes consisting of several actions (see Figure 1.1): a specific mix of raw materials or reactants are initially charged in a vessel, which has an internal stirrer to keep reactants well mixed. Then, materials are processed during a finite duration with a time-varying behavior. Optionally, intermediate reactants can be added to the vessel at a proper time. Once raw materials have been converted into desired products, they are discharged from the vessel.

This complex type of process is experiencing a renaissance since last decade as products-on-demand and first-to-market strategies impel the need for flexible and specialized production methods [35]. Several reasons make batch processes preferable over continuous processes. In practice, keeping steady state conditions over prolonged periods of time is difficult in some processes. For instance, in biotechnological and pharmaceutical industry using microorganisms, keeping their genetic stability over time is a tough problem. Keeping sterile conditions over prolonged periods of time is another typical problem to face in many biochemical processes. Also, frequent tuning of process settings for low-volume productions may be economically unfeasible in many processes. Thus, batch processing is usually preferred for low production volumes below 10,000 Mt/year while continuous ones are predominant for production volumes higher than 100,000 Mt/year in industries such as petrochemical, chemical, polymer, and food [36]. Batch processing has been preferred over continuous processing in the pharmaceutical industry to boost the manufacturing of high-added value and innovative products for decades. However, increases in competition, further increases in proportion of generic utilization, opening of new markets, and the current socio-economic pressures for price controls, appear to be creating new needs in the sector. The FDA released some guidelines [37] to help manufacturers to develop and implement new efficient tools

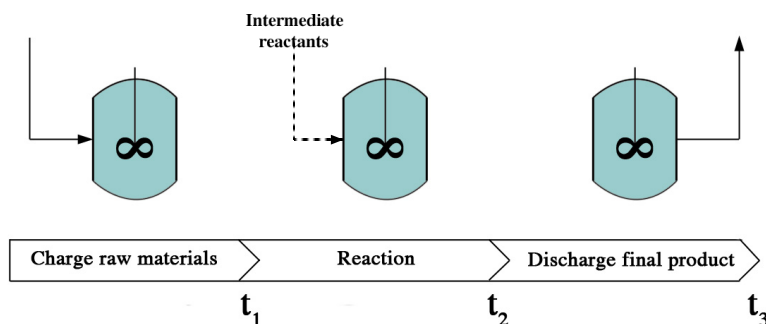


Figure 1.1. Batch process scheme representing the main stages to perform certain reaction for the production of a product.

for use during pharmaceutical development, manufacturing, and quality assurance while maintaining or reducing manufacturing costs. The usage of new approaches to pharmaceutical manufacturing, while maintaining the paradigm that quality cannot be tested into a product, but must be built in by design, leads to the concept of continuous processing. In particular, the draft guidance document [37] states: "Facilitating continuous processing to improve efficiency and manage variability." These regulatory statements, among others, further encourage pharmaceutical manufacturers to begin to shift from batch processing to continuous processing.

For control purpose, a set of measurements belonging to different process variables are recorded at every sampling time point in each batch. Regarding the sampling frequency to measure the variables, prior knowledge about the process may determine such frequency as a function of the type of variable or the importance of certain phases in the process, for instance. Additionally, the quality of a batch can be quantified by collecting online quality measurements provided by probes, as is the case of ammonia probe in wastewater treatment processes. Another possibility to assess the quality is analyzing a sample of the final product in laboratory upon batch completion. These measurements can be seen as end-product quality indices that represent how good the batch was performed. As a consequence of the high cost to obtain these measurements, the frequency at which they are sampled is lower than for engineering measurements (e.g. temperatures, flows, levels). It may happen that problems raised in a batch are reproduced in the subsequent batches before the problems are detected and diagnosed once the results of the laboratory are available.

In batch processes, once a batch is finished, the history of its performance is fingerprinted in the trajectories of each of the process variables. Hence, process stability can be defined based on a nominal or reference

trajectory. In other words, batches that approximately follow the reference trajectory and yield products on specifications can be considered as batches performed under *Normal Operating Conditions* (NOC). By analyzing the trajectories of the process variables over time, a better understanding of the process can be gained and a more accurate troubleshooting can be performed. The strategy is to model historical NOC and faulty batches, with the aim to determine the main mechanisms driving the process and to diagnose the potential root causes of the abnormal situations. From these learnings, changes in the process may be introduced and/or best operational practices may be defined to avoid future abnormal deviations. The next step would be to design a monitoring system that uses the in-control model to ensure that the variable trajectories of incoming batches are within their expected operational ranges. This monitoring scheme would permit detecting abnormalities before the quality measurements are available and taking actions to drive the process under normal operation conditions when deviating.

This chapter describes the state-of-the-art in batch process modeling and monitoring. Section 1.2 briefly describes the two major paradigms for modeling batch processes. In Section 1.3, the entire modeling cycle of batch processes is reviewed and rigorously discussed. It will establish the basis to hereafter present the new innovations and scientific discussions that this thesis brings out.

1.2 Batch process modeling

Two main modeling paradigms exist in literature [38]: the mechanistic (or white-box) models inherited from the field of the classical mechanics, and data-driven (or black-box) models originally proposed in the field of engineering. The former are developed from the first principle knowledge of the process in form of mathematical or biochemical relations, such as reaction kinetics, stoichiometries, mass/energy balances, that capture the behavior of the process. In contrast, the latter are models that are calibrated solely from input-output data. The final goal is to build models that best describe the nature or sources of variation over time. The choice of the model depends on the purpose of the model (exploratory analysis, optimization, monitoring and/or control), and the type of process [39].

The complexity of the processes in industry has fostered the development of a large number of model structures and modeling methodologies spanning the whole range between both paradigms, which can be classified into three categories [40]: knowledge-based or white-box models, data-driven or black-box models, and hybrid or grey-box models. Namely these models aim to decompose the process data into three (at least) well separated sources of

variation:

1. A white or hard part, related to a deterministic behavior or an prior process knowledge, such as mathematical relations describing chemical or physical properties.
2. A black or soft part, formed by the systematic variation that has not been taken into account in the white part of the model, usually related to the unknown sources of variation.
3. Noise or residual/unsystematic variation of the process, i.e. part of the data not explained by the previous parts.

The two types of variation can be segregated by using prior knowledge of the processes in the models, or extracting the information related to the theoretical/ideal models. In the particular case of batch modeling, the external or prior information can be divided into two parts [39]:

- Batch run specific information. This is the knowledge or additional measurements associated with an individual batch run (e.g. quality, reactor conditions, cleaning of the reactor, environmental conditions, batch duration, etc.).
- Process specific information. This information relates to the nature and sources of information within the process itself, that is, the underlying physico-chemical causes influencing all the batches are the same (e.g. the set of ordinary differential equations which describe the relationships among the variables; and the kinetics and/or stoichiometrics for one or more process reactions).

The separation of the different types of variation in process modeling is crucial for process understanding, optimization, monitoring and prediction. White-box models do not necessarily fit process data and explain batch-to-batch variation as well as black-box model do. Hence, the use of data-driven models that are capable of capturing not only the known but also the unknown variation sources is highly recommended. However, the use of hybrid models is a good alternative to black-box models if process knowledge is available. These models use the prior information (white part) to separate the major variation in data that is mainly generated by known physico-chemical interactions. It permits discovering new phenomena within data that are less abundant (black part). The remaining variability is kept into residuals, being subjected to be modeled separately.

1.2.1 Knowledge-based models

In this category, both the mathematical models obtained from physico-chemical laws -first principles or fundamental models- and those obtained from the expert knowledge of a process -expert systems- are included. The application of these modeling strategies to batch processes has received much attention in the scientific community for many years, both using dynamical models based on states formulations [41] and rule-based expert systems [42, 43]. These models allow us to estimate the underlying theoretical states of a complex process, such as the growth of a bacteria in a certain environment. Take the emergent field of systems biology as an example. In such area, the development of kinetic models of cellular metabolic networks based on quantitative experiments becomes a challenging and primordial task for the biotechnological industry. Such models are used to understand the principles that govern cellular behavior and to achieve a predictive understanding of cellular functions for subsequent modeling of fermentations [34, 44].

Nonetheless, such models are usually difficult to be used accurately for tasks related to monitoring or prediction in batch processes. This is due to the fact that such models are based on assumptions, which may not be true or adequate for the majority of the process states. Additionally, some unknown reactions may exist. Hence, the uncertainties produced by the unknown knowledge not taken into account in the models may cause inaccuracies in fault detection and predictions.

1.2.2 Data-driven models

Current industrial processes, both continuous and batches, collect an enormous amount of measurements belonging to hundreds or even thousands of process variables. The analysis of such data becomes a challenging task for industries in order to distinguish between in-control and out-of-control situations. Often, such data are highly correlated (process variables are collinear) with low signal-to-noise ratios and the existence of missing measurements in some variables [45]. In such environment, methods to extract the main information that explains the majority of the variability of process data is required [46, 47]. Projection techniques to latent structures (black-box or empirical models), such as PCA or *Partial Least Squares* (PLS), are useful to overcome all the above problems, apart from providing an analysis tool that is easy to understand from a statistical point of view. In statistical batch process control, a calibration data set composed by NOC batches that are representative of the common process variation is usually used for model building. Afterwards, a test data set is typically used to validate the model parameters for subsequent use in process optimization,

monitoring or prediction. In Appendix A, a brief description of the most used methods based on latent structures is provided.

1.2.3 Hybrid models

The use of knowledge-based models has the drawback that the unknown part of the process is not represented as well as some of the underlying assumptions (e.g. reaction kinetics, unknown dynamics, values of the model parameters, objective functions), which may not be valid for all the possible states of the process [48, 49]. To overcome this problem, hybrid models that combine knowledge-based models, which fit the theoretical behavior, and empirical models, which fit any remaining systematic variation, can be used [50].

Different approaches to decompose the data into the three types of variation (known causes, unknown causes and residuals) have been proposed in the literature. There are three categories where these methods can be classified:

- Based on known constraints. There exist general frameworks that enable to impose very specific constraints on each type of information, e.g. observed experimental information [51] or transformations on the original variables [52]. By using these decompositions, it is possible to segregate the information in four blocks: i) the information of the prior knowledge related to the observations and to the variables, ii) the part of the unknown information related to the observations and to the known information associated with the variables, iii) the part of the information of the prior knowledge related to the observations and the unknown information linked to the variables, iv) the part of the unknown information related to the observations and to the variables. When decomposing each of these well separated blocks by using multivariate projection techniques, it is possible to better interpret the latent structure.

The methodology in batch processes consists of using the prior knowledge of the process to impose different constraints in the multivariate statistical model. Interesting examples in which Tucker3 models are used to decompose the data structures can be found in [39, 53, 54]. More recently, *Grey Component Analysis* (GCA) was proposed [55], which uses a soft penalty approach to gently force the decomposition towards the direction of the prior information –a chemically or biologically meaningful solution. In [34], a grey modeling approach is introduced combining data-driven and first principles information at different scales, developed for *Pichia pastoris* cultures grown on different carbon sources. This hybrid framework has been proved

to be a useful tool to analyze and describe the most important biochemical processes, turning out a promising strategy for the design of real-time monitoring systems.

- Focused on model parameter estimation. Another option is to use techniques based on introducing the prior knowledge by means of mathematical relations that describe the system behavior or dynamics. The starting point is some specific structure based on first principles, where some functions have to be estimated. Different tools can be used to calculate these functions, such as artificial *Neural Networks* (NN) [56] or Kalman filters [57, 58].
- Constraints on the algorithms. Knowledge can be also incorporated in a model by imposing constraints on the modeling algorithms, such as using restricted Tucker3 models, or modeling a data structure by using *Multivariate Curve Resolution* (MCR) or *PARAllel FActor* (PARAFAC) models [59]. In the first case, constraints can be imposed by not allowing loadings to be negative when dealing with spectroscopic data, as in MCR [60] or PARAFAC models, or by imposing zero value on certain elements in the core matrix \mathbf{G} of Tucker3 models [61].

1.3 Bilinear modeling cycle of batch processes

Batch processes often exhibit batch-to-batch variations that are subject to investigation, analysis and monitoring: differences in the chemical composition, types and levels of impurities in raw materials, gradual operational changes, disturbances in the normal processing that affect the quality of the final product, etc. The analysis of data available in each batch is crucial to ensure safe operation, stable product quality and sustainable profit in batch processes. There are four major objectives for analyzing batch data [62]: i) the analysis of the variable trajectories of historical batches in the light of gaining process understanding and troubleshooting past abnormal operating conditions; ii) the statistical process control of incoming batches either after completion (so-called post-batch process monitoring, which can be segregated into end-of-batch and pseudo-online applications¹) or during its progress (real-time process monitoring); iii)

¹In end-of-batch process monitoring, the aim is to discover sources of variability among batches, improve operation policies, and diagnose the root causes of past abnormal and/or non-expected operating conditions at the end of the batch. 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.

the prediction of the 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 data available in batch processes for statistical analysis mainly fall into three categories [63]: i) L initial conditions available before the start of the batch for N batches (raw material properties, preprocessing times such as stages duration and waiting times, information on the shifts, any process measurements taken before the batch starts and expected to have an influence on quality, etc.), which are arranged into a two-way array \mathbf{Z} ($N \times L$), ii) J time-varying variable trajectories and/or online analytical sensor responses measured at K different sampling time points for all the N batches, which are arranged in three way array \mathbf{X} ($N \times J \times K$), and iii) M quality and productivity measurements stored after batch completion in the two-way array \mathbf{Y} ($N \times M$). These quality properties can be collected at K_y different sampling time points over the production of each batch leading to the three-way array \mathbf{Y} ($N \times M \times K_y$) though.

In the design of monitoring schemes, two phases are involved [47]: model building (exploratory data analysis and post-batch process monitoring) and model exploitation (real-time process monitoring). In the former, understanding the nature of the effects of varying initial conditions and process operating trajectories on the performance of the batches and on the final product quality is pursued [62]. Thereafter, the gained understanding and the statistical models are used to isolate and diagnose past poor operating conditions and set up SPC schemes for monitoring purpose in the second phase. 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 three-way array to one or several two-way arrays for the subsequent iv) bilinear batch modeling (see Figure 1.2). These steps are iteratively repeated whereas outliers are detected and isolated from the calibration data set.

1.3.1 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 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 [3]. 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 include the methods based on compressing/expanding the raw trajectories using linear interpolation. This interpolation can be performed in the time dimension, which is named the *Time Linear Expanding/Compression* (TLEC)-based method. The TLEC can be applied to the entire batch run [64], which is the technique implemented in SIMCA Release 13.0.3 -Umetrics software- [65], or within stages that are defined by key process events [66, 67], which is one of the synchronization techniques coded in ProMV Batch Edition Release 13.02 -ProSensus software- [68]. Other linear interpolation-based strategies also exist [67, 69]. Additionally, the linear interpolation can be applied to an indicator variable dimension, following the so-called Indicator Variable-based synchronization, IV [70]. A second category is formed by methods based on features extraction [71, 72, 73, 74, 75]. Finally, a third category is composed of the methods based on *Stretching, Compressing and Translating* (SCT) pieces of trajectories (the SCT-based methods), such as *Dynamic Time Warping* (DTW) [76] and *Relaxed Greedy Time Warping* (RGTW) [1]. In [76], a pseudo-online 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 shown to be inappropriate in BMSPC due to the false alarms produced in process monitoring, being the RGTW algorithm a solution to overcome this problem [1]. Other SCT-based methods were proposed in the literature for batch synchronization in offline and real-time applications [77, 78]. For an extensive review of the state-of-the-art in batch synchronization and the description of the most used synchronization approaches in the chemometric community, readers are referred to Chapter 4.

1.3.2 Data preprocessing

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 [39]. 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 time point, i.e. each vector \mathbf{x}_{jk} is mean-centered and scaled to unit variance (see Figure 1.2(j)). Provided the synchronized three-way data structure contains J variables, K sampling time points and N batches, this means that $J \cdot K$ averages and standard deviations are computed from N batches each. These averages are then subtracted from

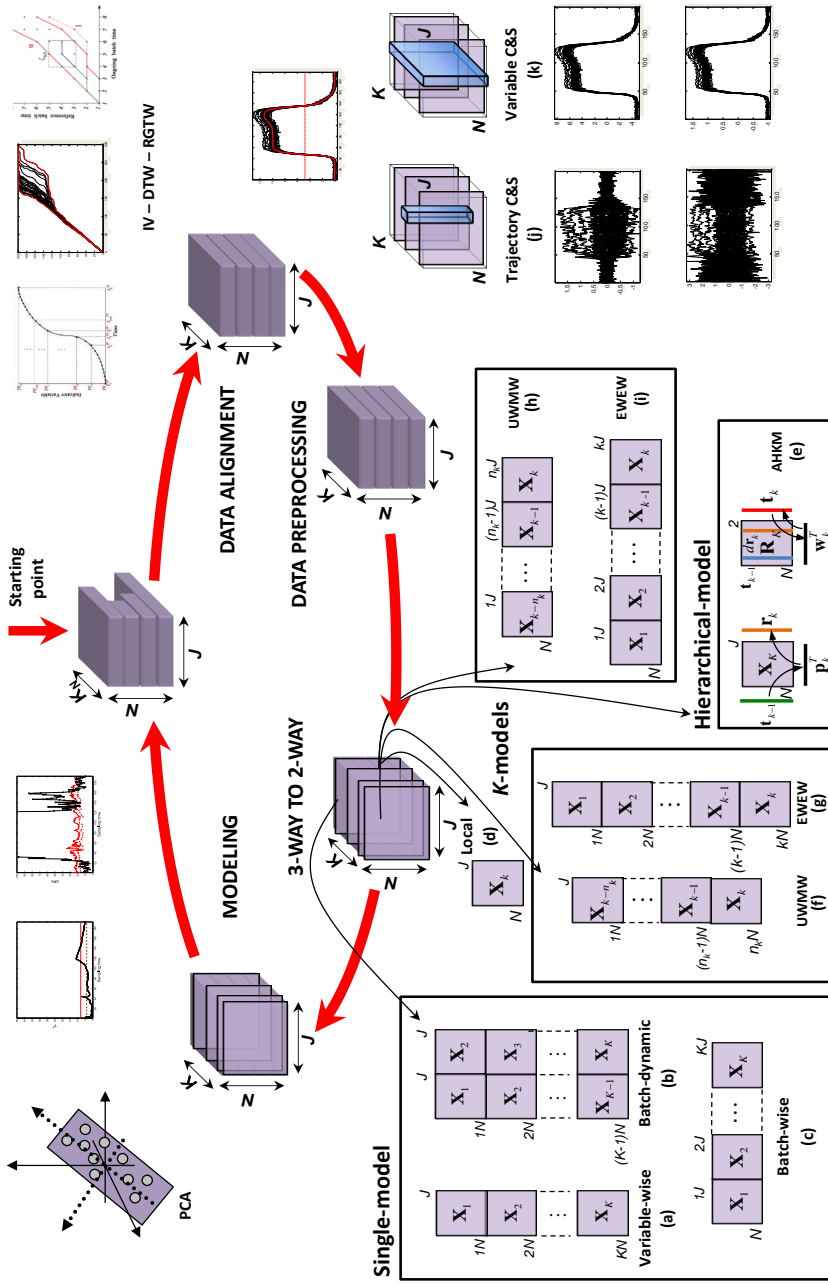


Figure 1.2. Modeling scheme in BMSPC systems based on PCA.

the corresponding data, and then, the N observations corresponding to the j -th process variable at the k -th sampling time 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 1.2(k)). Hence, J averages and standard deviations are computed from $N \cdot K$ observations each. Again, these averages (also called grand means) are subtracted, and subsequently, the centered data are scaled to unit variance. With this normalization, time periods with more variability will be weighted more and periods with less 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 make 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 [79]; and ii) the non-stationary problem is transformed into a stationary problem since the average trajectory is removed from the batch². The discussion about which of these two choices is more adequate has been present in the literature [79, 80, 81] since the two main pioneer research studies in BMSPC [82, 83] selected one of them each. Nomikos and MacGregor [82] performed Trajectory C&S whereas Wold et al. [83] used Variable C&S. After the desired variability is kept on data, the transformation of the three-way data array to a two-way data array can be carried out.

1.3.3 Bilinear modeling

In model calibration, the aligned and preprocessed three-way data array \mathbf{X} needs to be rearranged in a number of two-way subarrays 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 -models approach and the hierarchical-model approach (see Figure 1.2).

Single-model approach

In the single-model approach, the three-way array is unfolded in a single two-way array. There are several unfolding choices, which differ in the number of process variables lagged in time (the so-called *Lagged Measurement Vectors*

²Provided the batch process is under tight control so that the process can be considered to be stationary in the batch dimension.

(LMVs): *Variable-Wise unfolding* (VW) [83] (also called *Observation Wise Unfolding* (OWU) [62]), *Batch Dynamic* (BD) [84, 85, 86] and *Batch-wise* (BW) unfolding [70, 82] (see Figure 1.3(a), Figure 1.3(b) and Figure 1.3(c), respectively). BD unfolding can be seen as a generalization of the traditional unfolding procedures [87]: if no LMVs are added, the resulting matrix is the same as the one after VW unfolding - i.e. \mathbf{X} ($NK \times J$); if all possible LMVs are added, the resulting matrix is the same as the one after BW unfolding - i.e. \mathbf{X} ($N \times KJ$). 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 change throughout the batch run, i.e. the way process variables are related to each other and in time [87].

In the case of batch-wise unfolding, all the possible time-varying dynamics can be studied through the bilinear modeling of the resulting ($N \times KJ$) two-way array, despite some authors' assertions [86, 88]. In the light of confirming this claim, the resulting ($JK \times JK$) variance-covariance matrix after applying the batch-wise unfolding to data is investigated (see Figure 1.3(c)³). On the one hand, the sub-matrices $\{\mathbf{VC}_1, \mathbf{VC}_2, \dots, \mathbf{VC}_K\}$ located in the main diagonal of the variance-covariance matrix represent the variances and instantaneous cross-covariances of the variables at every sampling time point, i.e. the instantaneous relationships of the process variables. On the other hand, the sub-matrices $\mathbf{VC}_{k,k+d}$ for $k = 1, \dots, K-d$ represent the auto-covariances and lagged cross-covariances of order d , i.e. the dynamic relationships of the process variables. Hence, the relationships among all variables at different sampling time points are captured by the multivariate model fitted on the variance-covariance matrix.

In contrast to BW, the VW approach does not capture the dynamics of the process. This statement can be confirmed by studying the variance-covariance matrix after variable-wise unfolding. In Figure 1.3(a), the ($J \times J$) variance-covariance matrix of the VW unfolded matrix $\tilde{\mathbf{X}}$ ($NK \times J$) is shown. The former is composed of the sum of the sub-matrices $\{\mathbf{VC}_1, \mathbf{VC}_2, \dots, \mathbf{VC}_K\}$ weighted by the factor $1/K$. Hence, only the variances and instantaneous cross-covariances of the process variables are captured and not the dynamics of the process. This implies that the relationships between a pair of variables in the covariance matrix is an average of their relationships throughout the batch duration [87]. Consequently, this modeling approach assumes that there are no dynamics and the correlation structure of the process does not change over time. This is a strong assumption that is rarely accepted in batch processes, where the process dynamics are time-varying. The application of VW unfolding

³Note that only the auto-covariances and lagged cross-covariances of order 1 and 2 are shown for simplicity.

	Unfolding	Variance-covariance matrix
Variable-wise (a)		$\frac{1}{K} \cdot \left(\mathbf{VC}_1 + \dots + \mathbf{VC}_k + \dots + \mathbf{VC}_K \right)$
Batch dynamic (b)		$\frac{1}{(K-1)} \cdot \left(\begin{array}{cc} \mathbf{VC}_1 & \mathbf{VC}_{1,2} \\ \mathbf{VC}'_{1,2} & \mathbf{VC}_2 \end{array} + \dots + \begin{array}{ccc} \mathbf{VC}_k & \mathbf{VC}_{k,k+1} & \\ \mathbf{VC}'_{k,k+1} & \mathbf{VC}_{k+1} & \\ & \mathbf{VC}'_{k+1,k+2} & \mathbf{VC}_{k+2} \end{array} + \dots + \begin{array}{ccc} \mathbf{VC}_{K-1} & \mathbf{VC}_{K-1,K} & \\ & \mathbf{VC}'_{K-1,K} & \mathbf{VC}_K \end{array} \right)$
Batch-wise (c)		

Figure 1.3. Single model approaches to transform the three-way array into a two-way array with their associated variance-covariance matrices: (a) variable-wise (VW), (b) batch dynamic (BD), and (c) batch-wise (BW) unfolding.

in this scenario may have a notable impact on prediction [89] and fault detection [90].

As mentioned, the BW and VW unfoldings are particular cases of the BD approach (see Figure 1.3(b)). The number of LMVs to add to the two-way array must be carefully assessed. The more LMVs, the more dynamics are captured. However, special caution should be taken with the overparameterization of the models caused by an excessive and unnecessary addition of LMVs. Take as an example the situation represented in Figure 1.3. Considering that the dynamics of the process are invariant for $K-1$ sampling time points, the three-way array is variable-wise unfolded by adding 1 lag to all the variables. Hence, the instantaneous dynamics of the process is explained by the sub-matrices $\{\mathbf{VC}_1, \mathbf{VC}_2, \dots, \mathbf{VC}_{K-1}, \mathbf{VC}_K\}$ whereas the dynamics of order 1 are captured by the closest sub-matrices $\mathbf{V}_{1,2}, \mathbf{V}_{2,3}, \dots, \mathbf{V}_{K-1,K}$ to the sub-matrices \mathbf{VC}_k in the variance-covariance matrix. If batch data are modeled after batch-wise unfolding the three-way array, the resulting model would be overparameterized. Consequently, the derived variance-covariance matrix would contain much noise because only two main diagonals capture the dynamics whereas the remaining diagonals only noise.

K -models approach

The K -models approach is based on generating as many bilinear models as there are sampling time 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 time point, then it is called local model [91] -i.e. \mathbf{X} ($N \times J$) (see Figure 1.2(d)). If measurements registered from the beginning of the batch to the current sampling time point k are taken into account in each sub-model -i.e. \mathbf{X} ($N \times kJ$)-, then it is known as evolving model [91]. This approach can be seen as a local model at the k -th sampling time point where all the possible LMVs are included as additional variables [87]. Special cases of evolving models are *Uniformly Weighted Moving Window* (UWMW) [92] and *Exponentially Weighted Evolving Window* (EWEW) models, which are used when not all past information (lagged information) is of interest or has the same importance in bilinear modeling [87]. UWMW models are based on modeling the information contained in a window of width n_k , i.e. the measurements collected at the k -th current sampling time point with those of the immediate n_k LMVs. This information can also be seen as a local model at the k -th sampling time point where n_k LMVs are included as observations -i.e. \mathbf{X} ($n_k N \times J$)- (see Figure 1.2(f)) or as variables -i.e. \mathbf{X} ($N \times n_k J$)- (see Figure 1.2(h)). In contrast, EWEW models incorporate all the lagged measurements to the k -th current sampling time point, which

are weighted following an exponentially decreasing profile associated with 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 [87]. This can be problematic for processes where the final quality of the batch is strongly dependent on phenomena at the beginning of the batch - see [79] and [93] for a discussion on this topic in emulsion polymerization. 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 time point where all the possible LMVs are added either as observations -i.e. $\mathbf{X} (kN \times J)$ - (see Figure 1.2(g)) or as variables -i.e. $\mathbf{X} (N \times kJ)$ - (see Figure 1.2(i)) and exponentially weighted. One of the advantages of these K -model approaches is that they are capable of capturing varying dynamics of a certain order when all the possible LMVs are added as new variables (see Figure 1.4(a)), in contrast to when they are added as new observations (see Figure 1.4(b)). Since there is one model for every sampling time point, changing -dynamic or instantaneous- relationships are modeled. In evolving models, all the dynamics are included while the process evolves whereas no dynamics are captured by local models, only the instantaneous correlations at the k -th sampling time point (see Figure 1.4). Moving Window models are an intermediate case. In UWMW models where LMVs are added as observations, variances and instantaneous cross-covariances of the process variables within the selected window of samples of width n_k are captured. When LMVs are added as variables, the process dynamics of the n_k sampling time points are captured. In EWEW models, the same dynamics are captured as in UWMW models for the two versions with the difference that these relationships are weighted by the factor $(\lambda_k)^d$. The main drawback of the K -models approach 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 process stage or phase. Three main paradigms can be found to separate data into phases [94, 95]. First, the methods based on expert knowledge, which are based on modeling batch data according to different processing units or operation phases inside each unit [71], either applying multivariate models in each phase [96, 97, 98] or multi-block models [99]. Second, the methods based on distinguishing the different phases through analysis, either by detecting the landmark events [100] or singular events [101], or by the comparison of variable trajectories across batches using synchronization techniques. Finally, the methods based on automatically detecting the phases based on statistic optimization functions, such as the maximization of the correlation changes along the operating time in each batch [102] or on the prediction abilities

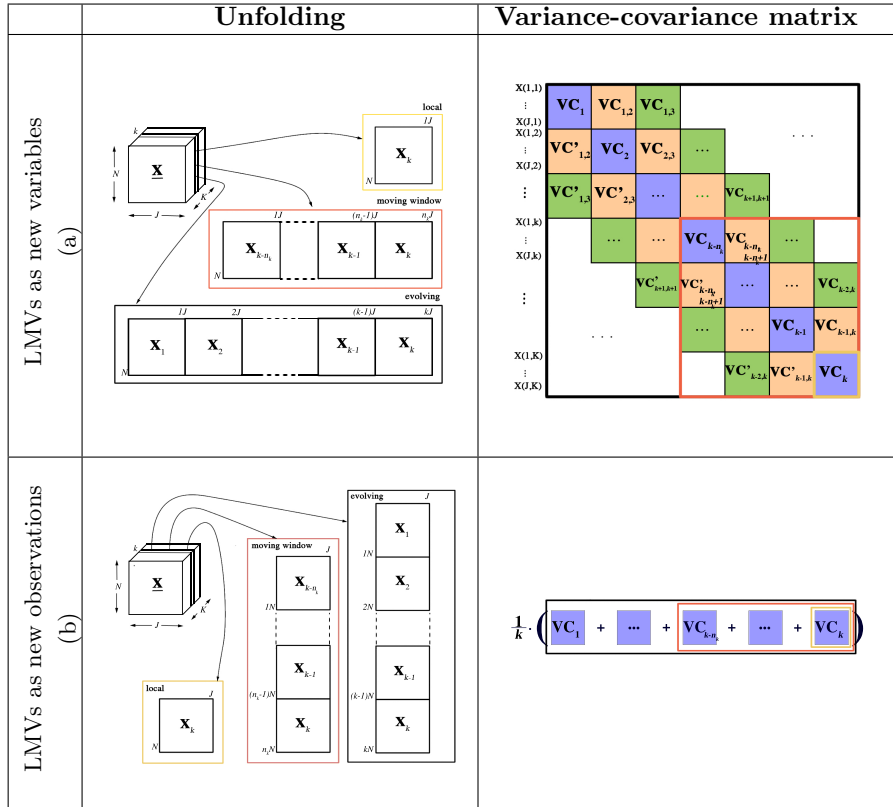


Figure 1.4. K -model approaches with all the possible LMVs added as new variables (a) and as new observations (b) with their associated variance-covariance matrices: local (yellow), moving window (red), and evolving (black).

of multivariate models [94]. The latter finds the single model (variable-wise, batch-dynamic and batch-wise) that best approximates the segments of batch data by linear models (PCA or PLS) (see Figure 1.5(a)). The resulting covariance matrix of the sub-model of three phases for variable-wise, batch-dynamic and batch-wise data in the entire batch is shown in Figure 1.5(b). As can be observed, the orders of the dynamics captured correspond to those of the single-model approaches explained previously.

Hierarchical-model approach

The hierarchical-model approach is based on combining the past and current information at each sampling time point with an adaptive hierarchical

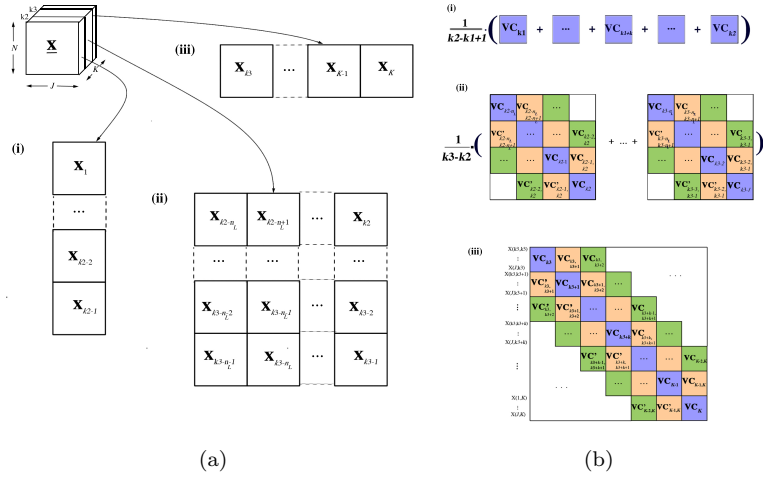


Figure 1.5. Sub-models in the multi-phase approach for variable-wise (i), batch dynamic (ii), and batch-wise (iii), and their respective associated variance-covariance matrices.

PCA model, i.e. *Adaptive Hierarchical K-Models* (AHKM) (see Figure 1.2(e)) [103]. First, a PCA model with one PC is fitted on the batch data corresponding to the first time slice $\tilde{\mathbf{X}}_1$, leading to the scores and loadings \mathbf{t}_{a1} and \mathbf{p}_{a1} corresponding to the a -th PC. The hierarchical part of the algorithm starts from the 2nd sampling time point. At sampling time point k ($k = 2, \dots, K$), the overall score vector $\mathbf{t}_{a(k-1)}$, which summarizes previous process variation up to the sampling time point $k - 1$, is used as starting vector to estimate the block scores \mathbf{r}_{ak} derived from the matrix $\tilde{\mathbf{X}}_k$ corresponding to time slice k . Afterward, this score vector is weighted by the weighting factor λ and placed in the consensus matrix \mathbf{R}_k together with the previous overall score vector \mathbf{t}_{k-1} .

The weighting factor λ is similar in nature to the exponential weighting factor in an *Exponentially Weighted Moving Average* (EWMA) model. It is used to give more or less importance to information collected at the current sampling time point k with regard to the past information. For high values of λ , the model is adapted quickly, while for low values of λ , the adaptation is slow. As λ grows further than one, the adaptive hierarchical K -model approach converges to the local K -models approach since the adaptive model does not use memory of any previous information.

The consensus matrix \mathbf{R}_k is then used to calculate the overall scores vector \mathbf{t}_k , which represents the total process variation up to the sampling

time point k . For this purpose, the weights vector \mathbf{w}_k is calculated by multiplying the consensus matrix \mathbf{R}_k by the overall score vector $\mathbf{t}_{a(k)}$. Then, the new overall score vector $\mathbf{t}_{a(k)}$ is calculated by multiplying the consensus matrix \mathbf{R}_k by the weight array \mathbf{w}_k . These estimations are repeated until the overall score vector \mathbf{t}_k reaches convergence. This procedure is repeated for all the sampling time points and all the *Principal Components* (PCs) to extract. For further details about the algorithm, readers are referred to [103].

The AHKM approach has been proved to work out well in terms of fault detection in a multi-stage context [71]. Nevertheless, one of the drawbacks of this approach is the need of estimating a model for each sampling time point, making this difficult to be implemented for online applications. In addition, a proper weighting factor may not be found for the whole batch run due to transitions among phases. In this case, the use of hierarchical K -models would not be the best approach for process monitoring. The analysis of this approach from the covariance matrix is not trivial and is not carried out here.

1.3.4 Monitoring

The design of a monitoring scheme in batch processes is crucial to ensure safe and stable operation, and the production of high quality product meeting defined specifications. In the following, the two main paradigms of the statistical control of measures and processes are briefly reviewed: univariate and multivariate process control. The drawbacks of the former and the differences with the latter will be stressed. Special emphasis is given to multivariate process control using methods to latent structures, which is the approach used in this thesis.

1.3.4.1 Univariate Statistical Process Control

The aim of SPC is to monitor the performance of the process over the time in order to verify whether the process is in-control or not, and to detect unusual disturbances that may occur. By finding the root causes of such abnormalities, some actions can be carried out to correct them (or implementing them to the process if they are beneficial), yielding an improvement in the quality of the end-product. In other words, it is expected that the process behaves with normality, having the minimum variability as possible in order to release end-products with good quality.

At this point, it is worth nothing the difference between the common and special variation cause since it is important to distinguish among them in order to implement an SPC scheme. The variation presented in a stable process reflects the common cause of the variation that is inherent

to the process itself, which cannot be removed easily without fundamental changes in the process [104]. As long as the common variation of the process is remaining, the process is considered to be in-control. Special cause variations are those produced by external causes (environmental conditions, fault in the sensors, changes in the quality of raw materials, etc.). These variations are desirable to be detected and diagnosed in time to correct them.

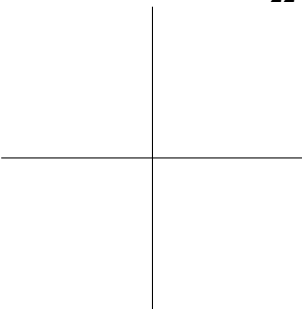
In order to implement an SPC scheme, industries have used some univariate control charts to monitor one or a few quality variables or key process variables that are suspected to be related in some way to the final product quality. The SPC charts most used in the literature have been the Shewhart, *CUmulative SUM* (CUSUM) and EWMA charts. In the following, a brief overview of such control charts is given.

Average Run Length

The *Average Run Length* (ARL) is defined as the average number of points that will be plotted on a control chart before an out-of-control condition is indicated when one or several points exceed the stated control limits. This measure is useful to compare the performance and efficacy of SPC control charts in terms of fault detection. It is well-known that even when a process is in-control, there is a probability associated to a false alarm signal without actually being a fault (so-called Type I error). When the control limits are set to 3-sigmas and assuming normally distributed data, this probability is equal to 0.27%. Hence, the in-control ARL, which is estimated as $ARL = 1/\alpha$, being α the Overall Type Risk I, is equal to 370. It means that, on average, every 370 samples plotted on the SPC control chart, an out-of-control signal will be spotted while the process is under control. Additionally, the ARL when the process is out of control can be estimated as $ARL = 1/(1 - \beta)$, where β represents the probability of a Type II error, i.e. the percentage of samples detected as normal when the process is out of control.

Shewhart Control Chart

In the 1920's, W.S. Shewhart proposed the use of a control chart to plot a certain statistic that describes the behavior of a single process variable V_j in time order. Figure 1.6 shows a typical example. This chart contains a center line (green line), which represents the in-control average value of the sample statistic of the variable V_j subjected to be monitored. Additionally, two lines (red lines) are depicted and represent the *Upper Control Limit* (UCL) and *Lower Control Limit* (LCL). The values of such control limits are chosen in such a way that when the process is under



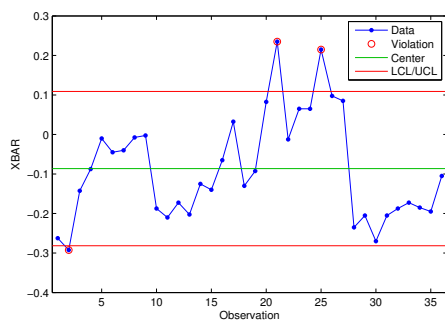


Figure 1.6. A typical Shewhart control chart

control, an expected fraction of the statistic values beyond the control limits takes a predetermined α , assuming the sample statistic is normally distributed. Hence, a $100 \times (1 - \alpha)$ percentage of the sample statistic values plotted are expected to fall within the confidence region limited by the upper and lower control limits.

One of the most common Shewhart control charts is the $\bar{x} - s$ control chart, which is designed from two different charts. The \bar{x} chart uses the sample mean ($\bar{x}_t = \sum_{i=1}^n x_{ti}/n$) to monitor the process mean, whereas the s

chart uses the sample standard deviation ($s_t = \sqrt{\sum_{i=1}^n (x_{ti} - \bar{x}_t)^2 / (n - 1)}$)

to monitor the process standard deviation at each sampling time point (where n , is the sample size). Control limits for both charts are estimated from collected data when the process was under control. Usually $M = 25$ sampling time points are considered. Special care should be taken when data are selected to estimate such control limits. The existence of outliers would cause wider control limits, thereby reducing the detection capability of the control charts. Hence, a previous step consisting of detecting potential outliers in data needs to be taken prior to the estimation of the control limits.

Traditionally, the control limits on the \bar{x} chart are set to $\bar{\bar{x}} \pm A_3 \bar{s}$, where $\bar{\bar{x}} = \sum_{t=1}^M \bar{x}_t / K$ is estimated over all the K sampling time points. The center line of the s chart is plotted on the average standard deviation \bar{s} , $\bar{s} = \sum_{t=1}^M s_t / K$, and the upper (UCL) and lower (LCL) control limits are set to $B_3 \bar{s}$ and $B_4 \bar{s}$, respectively. Assuming the process is under control, the standard deviation of the monitored process variable can be assessed as \bar{s}/c_4 . Values of A_3, B_3, B_4 and c_4 are tabulated for various sample sizes

[105].

It is worth noting that there are many situations in industry in which the sample size is 1, hence, control charts for individual measurements are used. In case quality features of a process are monitored, control charts for attributes, such as the p chart and the u chart need to be designed. An overview of these control charts for SPC can be found in [104].

CUSUM control chart

The CUSUM chart was originally designed by Page [106]. The basic idea is to plot at each stage t the CUSUM of past and present deviations of the selected sample statistic z_t over its target (in-control) value θ_0 :

$$C_t = \sum_{k=1}^t (z_k - \theta_0) \quad (1.1)$$

CUSUM charts are more effective than Shewhart charts for detecting persistent shifts in the process parameter θ , since the former accumulates information of several samples. This control chart is also effective with samples of size 1. When the process is under control, the CUSUM statistic C_t will fluctuate around 0 as a random walk. In the case θ_0 shifts to θ_1 , the CUSUM control chart will signal an upward or a downward trend. Care should be taken in the interpretation of the trends since it may happen that the process parameter θ is on target ($\theta = \theta_0$) but the CUSUM value C_t is far from 0, giving the appearance that there has been a process shift. Control limits in the form of a V-mask were proposed in the original CUSUM control chart to identify statistically significant changes in the slope.

An alternative to the V-mask-based CUSUM control chart is the so-called tabular CUSUM. This involves two statistics, C_t^+ and C_t^- , which are the sum of deviations above the target (referred as one-sided upper CUSUM) and below the target (referred as one-sided lower CUSUM), respectively. Both statistics are expressed as

$$\begin{aligned} C_t^+ &= \max\{0, z_t - (\theta_0 + K) + C_{t-1}^+\} \\ C_t^- &= \max\{0, (\theta_0 - K) - z_t + C_{t-1}^-\} \end{aligned} \quad (1.2)$$

where K is the 'reference value' to detect a change in the process parameter. This is usually set to the difference between the target value θ_0 and the out-of-control value θ_1 that we are aiming to detect quickly. The reference value can be also expressed in terms of δ units of standard deviations as $K = (\delta/2)\sigma$. The starting value of the aforementioned statistic is $C_t^+ = C_t^- = 0$. When any of the two statistics exceeds a stated threshold H , the process is considered to be out of control. ARL based methods are

often used to find the appropriate values of the parameters H and K . The proper selection of both parameters is crucial for the good performance of the control chart in terms of fault detection [107].

EWMA control chart

The EWMA control chart can be useful for detecting small shifts in the process and was first introduced by Roberts [108]. The control statistic to be charted is an EWMA of present and past values of the selected sample statistic z_t :

$$E_t = \lambda z_t + (1 - \lambda)E_{t-1} \quad (1.3)$$

where λ is a smoothing constant ($0 < \lambda \leq 1$). Considering that the initial value E_0 is equal to the process target θ_0 , Equation 1.3 can be expressed as

$$E_t = (1 - \lambda)^t E_0 + \lambda \sum_{k=1}^t (1 - \lambda)^{t-k} z_k \quad (1.4)$$

The latter expression shows the weights $\lambda(1 - \lambda)^{t-k}$ decreasing geometrically with the time at which the observations were registered. Hence, the parameter λ determines the memory of EWMA, i.e. the rate of weighting of past information. When $\lambda = 1$, the chart becomes a Shewhart control chart. On the contrary, if λ is close to zero, the EWMA performs like a CUSUM. The selection of the parameter λ should be chosen based on the magnitude of the shift to be detected. Usual values for this parameter are $0.05 \leq \lambda \leq 0.25$.

As commented before, the goal of this chart is to improve the detection of small shifts in the monitored process parameter. Typically, this chart

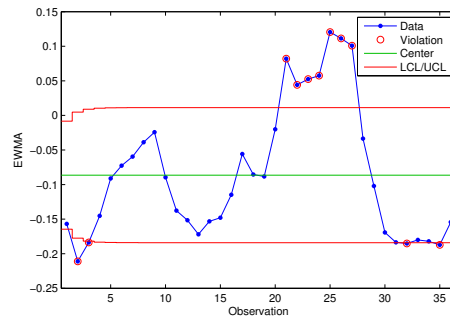


Figure 1.7. EWMA control chart

has two control limits (red lines depicted in Figure 1.7), upper (UCL) and lower (LCL) control limits, which define the region where the process can be considered under control. When one or more values of E_t exceed the control limits, the process is considered to be out of control. The green center line represented in Figure 1.7 is the process target θ_0 . The EWMA control limits are estimated as

$$\begin{aligned} UCL &= \theta_0 + L\sigma_z \sqrt{\frac{\lambda}{2-\lambda} [1 - (1-\lambda)^{2t}]} \\ LCL &= \theta_0 - L\sigma_z \sqrt{\frac{\lambda}{2-\lambda} [1 - (1-\lambda)^{2t}]} \end{aligned} \quad (1.5)$$

where L is the width of the control limits and σ_z is the standard deviation of the sample statistic z . Usually, the parameter L is typically set to values between 2.6 and 3. For further details, readers are referred to [109].

1.3.4.2 Multivariate Statistical Process Control (MSPC)

Classical MSPC

In traditional quality control of multivariate processes, several quality variables are separately monitored using univariate control charts. The problem of this approach is that quality data are often highly correlated, i.e. variables are not independent of each other, and do not separately define the quality of the end product [45]. Due to these interactions among quality variables, data might be cross-correlated and auto-correlated over time. Hence, monitoring a single quality variable at a time might cause that faults affecting the multivariate correlation structure are not detected [46]. This phenomenon can be appreciated from in Figure 1.8, where two quality variables (y_1, y_2) are monitored using two different univariate control charts and a two-dimensional control chart that is built by aligning one univariate control chart perpendicular to the other. Assume that the control limits of the univariate control charts are set to a 99% confidence level. As observed, all the samples plotted lie within the in-control region limited by the UCL and LCL. The ellipsoid shown at the top-left in Figure 1.8 represents the control limits associated with the in-control bivariate process behavior at 99% confidence level. Let us assume that measurements of quality variables y_1 and y_2 corresponding to an off-spec product are collected and plotted on the control charts (marked by red squares in the univariate and the two-dimensional control charts). These points clearly lie outside the in-control region represented by the ellipsoid, which indicates that the quality of the product is deviating from historical normal records. Nonetheless, this abnormality is not detected by the univariate control charts. This supports the claim that a monitoring scheme needs to capture

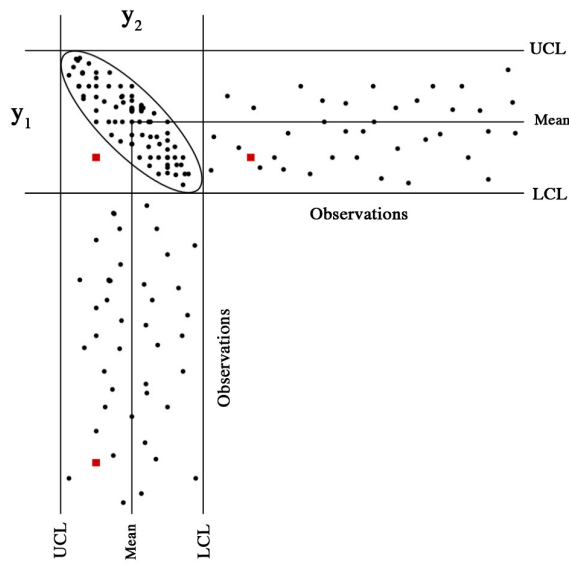


Figure 1.8. Univariate vs bivariate control charts

the time-varying correlations among variables to be capable of detecting severe abnormalities affecting the multivariate data structure.

One of the drawbacks of monitoring quality variables is the low frequency at which these variables are usually collected, which is mainly caused by the high cost associated with the collection and analysis of the samples. If the outcome of this analysis indicates that the process was out of control and producing off-spec products, there might not be time enough to drive the process under normal operating conditions prior to the release of the product. As a consequence, it might cause a waste of resources, raw materials and end product, manufacturing time, and hence, a loss of opportunities for the manufacturer.

By monitoring only some quality properties, potential information available from process variables which are measured more frequently than quality variables, is neglected. This might undermine the accuracy of the monitoring scheme to detect perturbations in the process. Unlike quality variables, process variables provide valuable information associated with the state of the process in real-time. Hence, an efficient process monitoring system should exploit the information available in the two types of variables because abnormal occurrences in the process might be solely explained by a unique source of information. It would allow us not only to detect the

loss of product quality but also abnormal process behaviors produced by the malfunctioning of sensors such as drifts, shifts or missing data.

In traditional statistical process control, several multivariate extensions of Shewhart, CUSUM and EWMA⁴ control charts have been proposed in the literature to monitor several quality or key process variables. One of the most successful control charts to monitor the stability of the process mean is the Hotelling T^2 control chart [113]. When a new multivariate sample is available (typically belonging to some quality or key process variables), Hotelling T^2 statistic can be calculated as

$$T^2 = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad (1.6)$$

where \mathbf{x} is a $(J \times 1)$ vector of measurements belonging to J process variables, $\boldsymbol{\mu}$ is the in-control $(J \times 1)$ mean vector and \mathbf{S} is the in-control $(J \times J)$ covariance matrix estimated from the NOC data. The underlying idea of this chart is to check whether the process mean remains stable or not, provided that the new measurements follow a multivariate normal distribution and the covariance matrix is constant over time.

Nonetheless, *Multivariate Statistical Process Control* (MSPC) based on the original variables suffers from lack of applicability in data-rich environments, typical in modern processes. The main serious drawback comes from the fact that the Hotelling T^2 statistic needs the inversion of the covariance matrix. To avoid this problem, the number of samples should be greater than the number of process variables ($N \gg J$), and the covariance matrix must be well conditioned, i.e. variables slightly correlated. These requirements are seldom met in the modern automated processes, where an enormous amount of process information is available, consisting of hundreds of highly correlated process variables registered at a high frequency rate (seconds or even milliseconds). Such measurements are often missing due to deficiencies in the good performance of the sensing systems. Add to it, the information contained in the variables is low due to the low signal-to-noise ratio.

Kourti and MacGregor [114] suggested the use of the multivariate statistical projection techniques to treat large and ill-conditioned data sets for process monitoring, leading to the basis of MSPC based on latent variables. Multivariate statistical methods such as PCA are used to reduce the dimensionality of the monitoring space by projecting the information associated with the original process variables onto an A -dimensional space. By taking into consideration all information contained in all the process

⁴For the sake of simplicity, these multivariate control charts will not be explained in detail on this document, with the exception of the multivariate extension of the univariate x Shewhart control chart, the Hotelling T^2 control chart. For a complete overview on this topic, readers are referred to [110, 111, 112].

variables, MSPC on latent variables (MSPC-PCA if PCA is used) is more powerful for detecting out-of-control conditions.

Multivariate Statistical Process Control based on latent models

Two phases are involved in the design of a monitoring scheme, Phase I (model building) and Phase II (model exploitation) [47, 104]. In Phase I, process data are modeled and outliers are removed from data. Thereafter, monitoring control charts based on latent variables are designed by using the in-control data under the assumption that the process remains stable. In Phase II the designed control charts are used to monitor the latent variables obtained from the projection of the new measurements onto the latent subspace. When an abnormal behavior is detected, diagnosis tools are used to identify the root causes of the out-of-control signals. Both modeling phases are considered both in continuous and batch processes if the objective is to design a monitoring scheme.

Focusing on Phase I, let us assume that a data matrix \mathbf{X} ($N \times J$) is collected with NOC data, where N is the number of sampling time points at which the values belonging to J process variables are recorded. The goal in this phase is to develop a latent model using PCA to model the in-control process data and thereby improve process understanding, detect possible abnormal behaviors not identified by the operator, and apply the proper corrective actions to remove such abnormalities from data.

Prior to fitting a PCA model on \mathbf{X} , process data should be mean-centered to improve the interpretation of the latent subspace. If the process variables are measured in different units, a proper scaling is needed. Provided that there is not prior knowledge on the process data, all the process variables should have the same weight in the PCA model by scaling to unit variance. Once process data are preprocessed, a PCA model is built, yielding the following latent structure: a $(J \times A)$ loadings matrix \mathbf{P} containing the eigenvectors associated with the A largest eigenvalues of the covariance matrix of the preprocessed data matrix $\tilde{\mathbf{X}}$, and the $(N \times A)$ scores matrix \mathbf{T} containing the coordinates of the N samples in the A -dimensional latent variable sub-space (see Section A.1.1). Recall that the preprocessed data matrix $\tilde{\mathbf{X}}$ can be reconstructed with minimum mean square error as $\tilde{\mathbf{X}} = \mathbf{T} \cdot \mathbf{P}^T$. The residuals are arranged in the $(N \times J)$ residual matrix \mathbf{E} .

From the scores vector $\boldsymbol{\tau}_n$ and residuals vector \mathbf{e}_n associated with the n -th sample of matrix $\tilde{\mathbf{X}}$, two orthogonal and independent statistics are computed: the Hotelling T_A^2 and the sum of *Squared Prediction Error* (SPE). The value of the Hotelling T_A^2 for the n -th observation can be expressed as

$$T_{A_n}^2 = \boldsymbol{\tau}_n^T \cdot \boldsymbol{\Theta}^{-1} \cdot \boldsymbol{\tau}_n = \sum_{a=1}^A \frac{\tau_{a_n}^2}{\lambda_a} \quad (1.7)$$

where $\boldsymbol{\Theta}$ ($A \times A$) is the variance-covariance matrix of the scores matrix \mathbf{T} . This statistic represents the estimated Mahalanobis distance from the center of the latent subspace to the projection of the sample onto the A -dimensional subspace.

The SPE statistic for the n -th sample is given by:

$$SPE = \mathbf{e}_n^T \mathbf{e}_n = \mathbf{x}_n^T (\mathbf{I} - \mathbf{P}_A \mathbf{P}_A^T) \mathbf{x}_n \quad (1.8)$$

This statistic represents the squared Euclidean distance of the sample from the low-dimensional latent space.

In order to develop a monitoring system, two multivariate control charts based on the aforementioned statistics, T_A^2 and SPE, need to be designed by estimating their corresponding control limits. When the aim of this monitoring system is to monitor incoming samples, the UCL for the Shewhart T_A^2 control chart at significance level α (type I risk) is given by [110]:

$$UCL(T_A^2)_\alpha = \frac{A(N^2 - 1)}{N(N - A)} F_{A, (N-A), \alpha} \quad (1.9)$$

where $F_{A, (N-A), \alpha}$ is the $100 \times (1 - \alpha)\%$ percentile of the corresponding F distribution.

Regarding the UCL of the Shewhart SPE chart, several procedures can be applied. Jackson and Mudholkar [115] showed that an approximate SPE critical value at significance level α is given by

$$UCL(SPE)_\alpha = \theta_1 \left[\frac{z_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right] \quad (1.10)$$

where $\theta_k = \sum_{j=A+1}^{rank(\mathbf{X})} (\lambda_j)^k$, $h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2}$, λ_j are the eigenvalues of the PCA residual covariance matrix $\mathbf{E}^T \mathbf{E} / (N - 1)$, and z_α is the $100(1 - \alpha)\%$ standardized normal percentile. Alternatively, one can use an approximation based on the weighted chi-squared distribution ($g\chi_h^2$) proposed by Box [116]. For further details, an extended review can be found in [47].

Let us assume that a new multivariate sample \mathbf{x}_n ($J \times 1$) is available. After centering and scaling the new measurements, \mathbf{x}_n is projected onto the PCA model, yielding the corresponding values for the Hotelling T_A^2 and SPE statistics. Afterward, these statistics are plotted on the Shewhart T_A^2 and SPE control charts, respectively, and checked against their corresponding

control limits [117]. When the SPE value exceeds the control limit, it means that the process is out-of-control. If the value of the T_A^2 statistic lies beyond the limits, a different process performance is detected without breaking the correlation structure. Hence, when one of the control charts detects an out-of-control signal, a fault diagnosis is required to identify the potential group of responsible variables of such abnormality. For this purpose, contribution plots are useful tools [114].

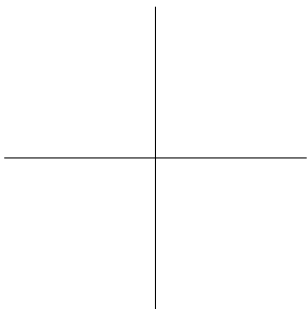
Let us assume that an abnormal event has been signaled by the Shewhart SPE control chart at the n -th sample point. The contribution of each original j -th process variable to the high value in the SPE statistic is calculated as follows:

$$c_{n,j}^{SPE} = e_{n,j}^2 \quad (1.11)$$

If the abnormality has been detected by the Shewhart T_A^2 control chart, the diagnosis is carried out in two steps: i) a bar plot of the squared normalized scores for that observation $(\tau_{a,n}/\sqrt{\lambda_a})^2$ is plotted and the a -th score with the highest normalized value is investigated; ii) the contribution plot of each j -th process variable to this a -th score at this abnormal observation is then estimated as:

$$c_{n,j}^{T_A^2} = x_{n,j} \cdot \frac{\tau_{a,n}}{\lambda_a} \cdot p_{a,j} \quad (1.12)$$

where $\tau_{a,n}$ is the score value of the new sample \mathbf{x}_n , $x_{n,j}$ is the value of the j -th process variable, and $p_{a,j}$ is the loading of the j -th process variable at the a -th principal component. Variables on this plot with high contributions but with the same sign as the score should be investigated (contributions of the opposite sign, will only make the score smaller). When there are some scores with high squared normalized values, an overall average contribution per variable can be calculated over all selected scores [114]. Contribution plots are a powerful tool for fault diagnosis. They provide a list of process variables that contribute numerically to the out-of-control condition (i.e. they are no longer consistent with NOCs), but they do not reveal the actual cause of the fault. Those variables and any variables highly correlated with them should be investigated. Incorporation of technical process knowledge is crucial to diagnose the problem and discover the root causes of the fault [104].



Material and Methods

2.1 Hardware

All computations carried out along this thesis have been performed with a notebook Intel Core 2 Duo, CPU 2.4 Ghz with 4GB 1067 MHz DDR3.

2.2 Software

The software packages used are:

- ▷ Mac OS X 10.5.8 (English), Darwin 9.8.0 kernel version (2009).
- ▷ Matlab 7.4 and R2010a (©The MathWorks, Inc., Natick, MA, USA).

All functions, algorithms and scripts used in this thesis are own code implemented in Matlab code and eventually integrated into MVBatch, the software package derived from Multi-Phase Framework (toolbox developed by José Camacho Ph.D.).

2.3 Simulated Processes

Two different simulated data sets are used along this manuscript to illustrate the features of the methods proposed in the literature and the performance of the different methods and techniques developed in this thesis.

***Saccharomyces cerevisiae* cultivation.** *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*

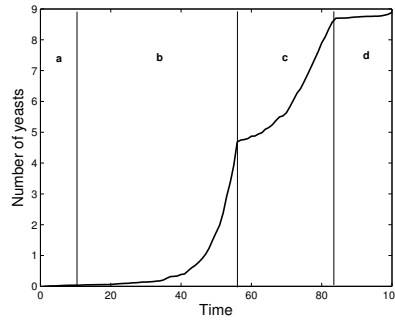


Figure 2.1. Growth curve of the *Saccharomyces cerevisiae* cultivation: a) lag phase, b) first exponential growth, c) second exponential growth, and d) stationary phase.

cerevisiae on glucose limited medium introduced by Lei [118] 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 2.1). 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. For more details about theoretical assumptions and additional information concerning the first principles model, readers are referred to [118].

For the sake of accuracy in simulation, the biological variability of the yeast is taken into consideration to generate time-varying variable 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.

Three different types of faults are designed: two process faults generated by modifying the internal constants k_{11} (associated with the reaction describing the glucose uptake system and the glycolytic pathway) and k_6 (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 (V_{max} , k_{11} in the stoichiometric model) of the lumped biochemical reaction considered in the model, are simulated. V_{max} represents the way in which the substrate is processed by the yeast in a glucose limited media. Although V_{max} is biochemically based (highly efficient strains will be able to consume glucose more quickly, showing higher intrinsic V_{max} 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 V_{max} . To simulate these scenarios, the values of the kinetic constants k_{11} and k_6 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.

Five different sets of simulations under different conditions and data treatments are generated (see Table 2.1). In set #1, two different simulation sequences are performed to ensure independency. For this purpose, the seed used in the simulation is different for each data set to obtain different sequences of random numbers, which are used to generate Gaussian noise and the length of batches. For the rest of the sets, the same seed is used to generate the batch data. Ten process variables are measured every sampling

Table 2.1. Simulated data sets used along the manuscript.

Set	Three-way arrays	Description
#1	$\underline{\underline{\mathbf{X}}}_1(N_1 \times J \times K_{n_1}), N_1 = 30, J = 11, K_{n_1} \in [172, 332]$	NOC
	$\underline{\underline{\mathbf{X}}}_2(N_2 \times J \times K_{n_2}), N_2 = 30, J = 11, K_{n_2} \in [173, 297]$	NOC
#2	$\underline{\underline{\mathbf{X}}}_3(N_3 \times J \times K_{n_3}), N_3 = 50, J = 10, K_{n_3} \in [180, 264]$	NOC
	$\underline{\underline{\mathbf{X}}}_4(N_4 \times J \times K_{n_4}), N_4 = 3, J = 10, K_{n_4} \in [179, 215]$	modified k_{11} , k_6 and k_{10}
#3	$\underline{\underline{\mathbf{X}}}_5(N_5 \times J \times K_{n_5}), N_5 = 85, J = 10, K_{n_6} \in [172, 220]$	NOC
	$\underline{\underline{\mathbf{X}}}_6(N_6 \times J \times K_{n_6}), N_6 = 44, J = 10, K_{n_6} \in [171, 210]$	modified k_{11}
	$\underline{\underline{\mathbf{X}}}_7(N_7 \times J \times K_{n_7}), N_7 = 44, J = 10, K_{n_7} \in [195, 222]$	modified k_6
#4	$\underline{\underline{\mathbf{X}}}_8(N_8 \times J \times K_{n_8}), N_8 = 40, J = 11, K_{n_8} \in [172, 330]$	NOC
	$\underline{\underline{\mathbf{X}}}_9(N_9 \times J \times K_{n_9}), N_9 = 10, J = 11, K_{n_9} \in [173, 294]$	NOC
#5	$\underline{\underline{\mathbf{X}}}_{10}(N_{10} \times J \times K_{n_{10}}), N_{10} = 60, J = 11, K_{n_{10}} \in [173, 298]$	NOC
	$\underline{\underline{\mathbf{X}}}_{11}(N_{11} \times J \times K_{n_{11}}), N_{11} = 10, J = 11, K_{n_{11}} \in [173, 262]$	modified k_{11}
	$\underline{\underline{\mathbf{X}}}_{12}(N_{12} \times J \times K_{n_{12}}), N_{12} = 10, J = 11, K_{n_{12}} \in [173, 243]$	modified k_6
	$\underline{\underline{\mathbf{X}}}_{13}(N_{13} \times J \times K_{n_{13}}), N_{13} = 10, J = 11, K_{n_{13}} \in [177, 236]$	bias in biomass sensor

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. Additionally, the original time of processing from simulation is added to the three-way arrays of sets #1, #4 and #5, leading to a total of eleven variables in these cases. For further information, readers are referred to Table 2.1.

Continuous Wastewater Treatment Process. Data from a continuous wastewater treatment process (WWTP) for the biological removal of organic matter and nutrients have been simulated. The simulated layout consisted in a nutrient removing WWTP based on a modified UCT scheme (see Figure 2.2). The plant was simulated using the software DESASS [119], which includes the mathematical model BNRM1 [120]. This model considers the most important physical, chemical and biological processes taking place in a WWTP. The plant layout was designed to treat $12000 \text{ m}^3 \cdot \text{d}^{-1}$, with 18h of Hydraulic Retention Time (HRT) and 8 days of Sludge Retention Time (SRT). It includes three reactors (anaerobic, 30%; anoxic, 10%; and aerobic, 60%) and a reactive secondary settler (including biological processes taking place in it).

In order to simulate the important variations of the influent loading and flow rate taking place in a WWTP, the standardized influent file for dry weather proposed by Copp [121] was used in this simulation study. The simulation strategy consisted in a steady-state simulation to obtain

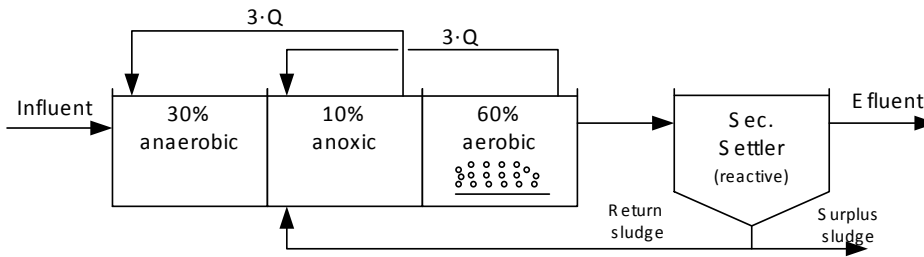


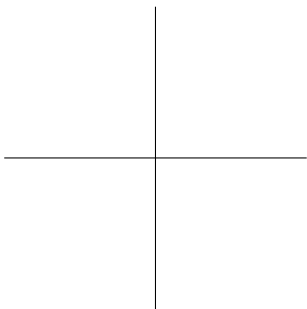
Figure 2.2. Layout of the simulated WWTP.

proper initial conditions followed by 28 days dynamic simulations. Online measurements of the main parameters, which are available in a WWTP, were simulated with DESASS and sampled every 15 minutes. Table 2.2 shows the measurements in each reactor/stream that were considered in this study. All the measurements included white noise with different standard deviations according to the type of sensor.

Table 2.2. Online measurements in each reactor/stream considered in this study.

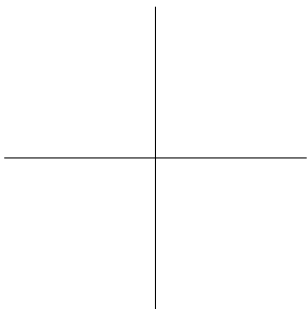
Reactor/Stream	Influent	Anaerobic Reacto	Anoxic Reactor	Aerobic Reactor	Return Sludge
Simulated measurements	pH, flow rate	pH, NO_3	pH	K_{L_a} , NO_3 , NH_4 , PO_4	pH

In addition, several faults in nutrients probes installed in the aerobic reactor were simulated: (1) a slow drift in the nitrate and ammonium probes from the ninth day at 12 am to the end of the simulation; (2) a shift in the nitrate probe on the ninth day from 12 am to 4.30 pm with signal stabilization with such deviation up to the end of the simulation; (3) lack of collecting measurements in the phosphorous probes due to technical problems at different time intervals: (i) from the fourteenth day at 12 pm to the fifteenth day at 8 am, (ii) from the eighteenth day at 12 am to the twenty-first day at 12 am and (iii) from the twenty-third day at 9 am to the twenty-fourth day at 12 am. The latter probe fault is simulated by imposing the value 0 to each time interval indicated above.



Part II

Preprocessing of Batch Data



Batch data equalization

Part of the contents of this chapter has been included in the following publications:

- [8] J.M. González-Martínez, J. Camacho and A. Ferrer. Equalization of batch data: challenges and solutions. *In elaboration for Journal of Process Control*.

- [12] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 3: Batch Process Data. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.

- [31] J.M. González-Martínez, J. Camacho, O.E. de Noord and A. Ferrer. Equalization and data-driven compression as a prior step to batch modeling. *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 59, Djurönäset (Sweden), 2013.

3.1 Introduction

The success of multivariate statistical analyses and monitoring systems relies on the quality of data collected by historian software packages [79]. Typically, process data are manipulated by compression algorithms to reduce the costs of long term storage of manufacturing records, which is a common practice in pharmaceutical industries to achieve compliance with the demands of governmental authorities. Also, the size of databases is usually reduced when the costs of transmission of data through telecommunication networks from distanced production assets to research centers are too high, as is the case with the oil and gas industries [122]. The application of univariate compression techniques to data of multivariate nature might represent however a threat to the reliability of the outcomes in BMSPC [123].

Most industrial data historians use piecewise linear trending for data compression, although other methods based on feature extraction were proposed in the literature [124, 125, 126]. For instance, AspenTech software uses an adaptive method based upon the box-car/backward slope (BCBS) method [127] in their data historian while OSIsoft uses a variant of swinging door compression algorithm [128] involving a compression deviation blanket with a width equal to twice the compression deviation specification [129]. Both algorithms contain no mechanism to post-process the data in order to change the stored data point, therefore, both approaches can be considered as a filter. The main drawback of these algorithms is that the archived variables can be approximated by a piece-wise linear function, whereby variable values within a threshold window around the line segments are not stored. This assumption affects the way to reconstruct the signal, which is based on minimizing the deviation of the reconstructed signal from the actual signal and not aimed towards preserving the variance of the signal or the correlation between the variables. The basis of these linear compression methods jointly with the need of tuning the parameters of the algorithms might negatively impact the trade-off between the data compression ratio and the approximation error, and hence, distort the multivariate data structure [130].

As a consequence of compression, the measurements belonging to the process variables might be collected at different sampling time points, which yields a lack of equalization among variables, and potentially among batches. In addition, process variables might be collected at a different sampling frequency within and/or across batches for manufacturing reasons. In this scenario, the bilinear modeling for

the design of a monitoring system cannot be conducted because the pattern of evolution of a batch under NOC cannot be identified. Remind that batch modeling is aimed at uncovering the effect of process variables at specific points during the evolution of the process, regardless the time the process needs to reach this process point. This implies that the values of the collected process variables must be representative of the same points of evolution for all batches [131]. Hence, if batch data do not meet these requirements, equalization of batch data must be conducted prior to the statistical modeling.

Since batch processes are dynamic processes that produce multivariate time series data, it is crucial that the sampling frequency of the measured variables is greater than, or equal to, twice the maximum frequency content of the signal being sampled, which is the minimum dictated by the Nyquist-Shannon sampling theorem [132, 133]. Otherwise the data will be aliased and will not be a true representation of the measured analogue variables. This is the domain of Digital Signal Processing but has implications for equalization policies such as discarding intermediate values. Especially, when the resampling is done automatically by software without reference to the parameters of the data acquisition system.

In this chapter, the equalization of batch data is addressed. In Section 3.2, the problem of the lack of equalization in the variable trajectories and the different scenarios that can be found in batch processes are described. Section 3.3 introduces the approaches to solve the different sampling policies can be found: i) discarding intermediate values, ii) estimating missing values, and iii) rearranging batch data. The implication of each method on the modeling of batch data is illustrated by using the simulated data of the fermentation of *Saccharomyces cerevisiae* cultivation. The most complex situation of unequalization is studied and discussed in Section 3.4. Finally, some conclusion on this matter will be provided in Section 3.5.

3.2 Challenges in batch equalization

In order to model the pattern of evolution in a batch under NOC, calibration batches should be properly equalized so that they show a similar behavior at the same points of evolution during the processing.

In industrial processes, different sampling policies can be found mostly due to three main reasons: (i) importance of some stages of the process to release products of good quality, ii) sensor or actuator speed constraints, and iii) the cost of high-rate sampling. In the case that

none of the previous constraints are given in the process, all process variables can be collected throughout the batch run at the same sampling rate over all batches (see Figure 3.1(a)). Hence, no action is required since process data are just equalized. Nevertheless, when the process variables are measured following a non-common sampling rate variable-to-variable and/or batch-to-batch, the equalization of the measurements of process variables must be performed.

A typical case of non-uniform sampling frequency is when an intermediate reactant is added to the reactor in a fed-batch chemical process, for instance, the manufacturing of specialty chemicals such as polymers and surfactants. Typically, the feed rate or the accumulated amount of intermediate reactant added to the reactor is only monitored during this stage, whereas no value is recorded in the remaining stages (see Figure 3.1(b)). Note, however, that this type of processing might not represent a case of lack of equalization itself although having an important consequence in the way data might be arranged prior to modeling. If some process variables are sampled at different frequencies throughout the stage and/or among batches, thereby causing a clear event of lack of equalization, data must be equalized. Otherwise, if the sampling frequency of the process variables collected only at a specific stage is the same over this period and across batches, and likewise for the rest of variables across batches, no action is needed because data are already equalized. To continue the bilinear modeling in this case, data can be batch-wise unfolded in such a way that each time slice contains a different number of variables [134].

Different complex sampling rates can be set up in batch processes. Usually, some stages of the manufacturing process are crucial to release products of good quality, and they are measured at higher sampling frequency than others (see Figure 3.1(c)). Take as an example the process used in [79], the emulsion polymerization. The particle size depends mainly on the nucleation period, which occurs at the first time interval of the reaction. The following process stages will also affect the particle size, but neither to the same extent nor degree as in the nucleation. Sampling more frequently than normal at this stage on all process variables may be advisable in order to monitor in detail such reaction batch-to-batch. In this case, the variables are not equalized but the batches are. Although the data collected may look anarchic, there is a very simple way to arrange the data in order to develop the monitoring system of the process, which is again batch-wise unfolding the three-way data array (see Figure 3.1(c)). This data arrangement is recommended for this type

of processes, where only process variables are not equalized, in order to avoid equalization issues.

More complicated cases of sampling policy can be found, the so-called multi-rate systems. One of the multi-rate sampling cases is when all process variables are sampled in different sampling rates but this is accurately repeated in each batch (see Figure 3.1(d)). This phenomenon is common in sequencing batch reactors for wastewater treatment. In this kind of processes, measurement belonging to pH, electric conductivity, oxidation reduction potential, dissolved oxygen, etc., are measured at different sampling rates throughout the batch run based on its importance in the different phases of the batch, e.g. anaerobic, aerobic and anoxic. Additionally, other measurements derived from laboratory analysis are usually obtained at lower frequency to measure the performance of the process and the quality of the water treatment (phosphorus concentration, ammonium concentration, etc.). Hence, a set of process variables measured at multi-rate sampling time points repeatedly over all batches are available. Another case of multi-rate systems is when process variables are measured following different sampling policy between variables and batches (see Figure 3.1(e)). This multi-scale non-uniform sampling frequency is mainly caused by the application of compression algorithms by historians to reduce the size of archived data files. This type of unequalization problem may be also found in industries where the process of a plant A must be scaled-up to another plant B, having different operative conditions. In these cases, it may be interesting to monitor engineering process variables (temperatures, pressures, etc.) at different sampling rates in phases of the process depending on its importance in the manufacturing of high quality products. In addition, several quality variables may be measured at lower sampling rates than the engineering variables. Due to the difficulties to obtain desired products related to the scaling-up, this sampling policy may be different among batches in order to monitor the process more exhaustively. In the complex context of multi-rate systems, the use of imputation techniques are needed to overcome the difference of sampling rate within and/or among batches, transforming the process data to a data structure similar to the one depicted in Figure 3.1(a).

For the equalization of process variables, the between-batch variation is not used but the within-batch variation instead because there is no certainty that batch data are synchronized, even though the batch trajectories have equal length. Hence, the time-varying correlation structure across batches cannot be exploited, only the dynamics within batches. In the next sections, specific solutions are described

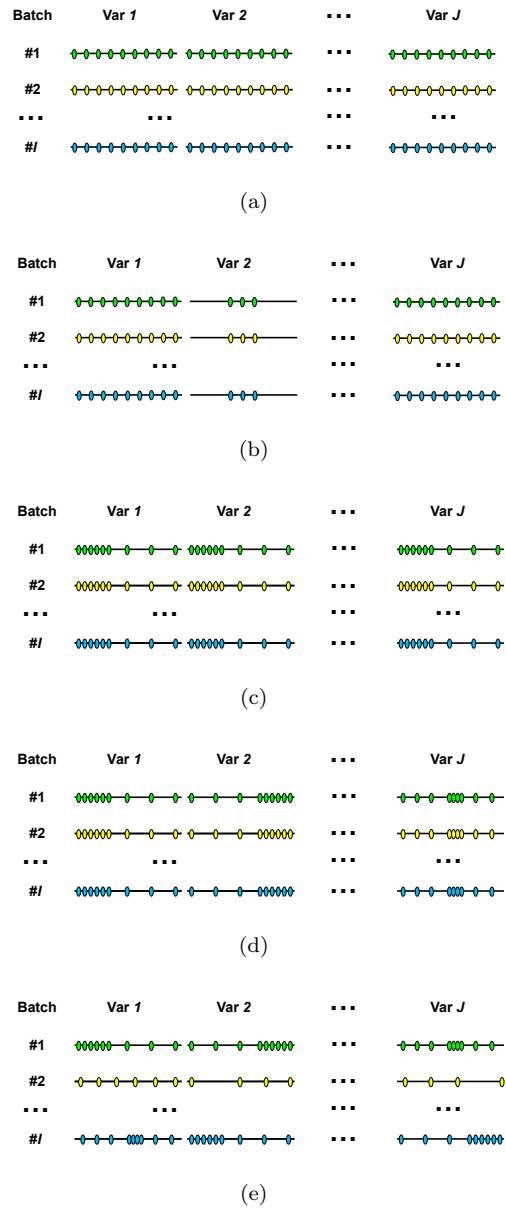


Figure 3.1. Sampling rate scenarios in batch processes: (a) same sampling rate for all variables and batches, (b) same sampling rate among variables and batches, except for a variable that is only measured at certain phase, (c) different sampling rate in different process stages but same policy for all variables and batches, (d) multi-rate sampling in process variables and similar policy among batches, and (e) multi-rate sampling both in process variables and batches.

for the different scenarios of unequalization we can find in batch data.

3.3 Equalization of variables within a batch

A relevant step in the development of a monitoring system is how to deal with a data set in which process variables are measured at different rates (see Figure 3.2(a)). Unfortunately, it is also one of the less considered problems in the literature. Probably, the easiest way to build the monitoring system in multi-rate systems is by equalizing all the process variables to a common rate (see Figure 3.2(b)). Any equalization implies the estimation of the value of one or several variables at a number of sampling time points where the variables were not actually measured. In the following we will assume that data acquisition cannot be modified to a sampling scheme similar to Figure 3.2(b), and consequently, estimation techniques are necessary.

The equalization of variables can be done straightforwardly by using interpolation. In interpolation, previous and subsequent values of a variable are used to estimate its value at sampling time point t according to a certain function -e.g. splines functions- which approximates the data. Recall that during model exploitation, in an online setup, only measurements previous to t are available for estimation.

Two principal decisions for the equalization of variables are the between-observations interval (the time elapsed between two observations) and the measurement vector (MV) -i.e. the set of variables which constitutes an observation. The arrangement shown in Figure 3.2(b) is the simplest one that can be thought of. Nonetheless, in some processes a different arrangement may be desirable. To illustrate this idea, a simple example in which only one variable has a different -slower- sampling rate than the rest will be used. Afterward, the more general problem of multi-rate systems will be discussed. Take the example of a batch process aimed at the synthesis of Polyhydroxyalkanoates (PHAs), in which the content of biomass is measured every 7 minutes whereas temperature, pH and gas flows are measured every minute. To equalize the data, three options are suggested:

- i) Discarding intermediate values. Provided that the Nyquist-Shannon sampling theorem is obeyed for the resampled data set, data may be reduced to a 7 minutes interval between observations, so that intermediate measurements of temperature, pH and gas flows are simply discarded from the design (Figure 3.3(a)). This approach is the simplest. Simplicity is a desired

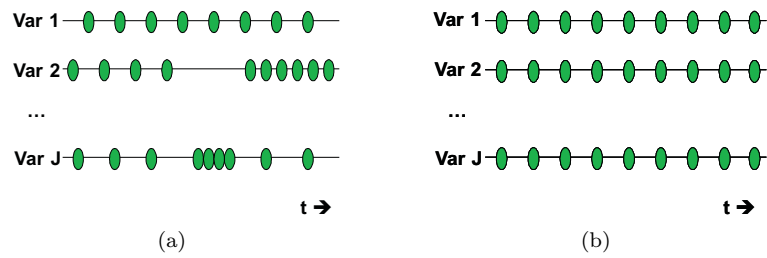


Figure 3.2. Example of non-equalized variables (a) and equalized variables (b).

feature taking into account that this processing will be performed online in the monitoring system. Nevertheless, it has the drawback that any fault appearing during the 7 minutes interval can only be detected at the end of this period whereas the discarded data may contain the information to detect the fault earlier. Depending on the process, after 7 minutes have elapsed it may be too late to fix the problem, if this was possible at all. Furthermore, if the content of biomass was measured using manual techniques instead of in-line sampling devices, the biomass may be measured every hour or even less frequently. This is the case in many real setups. In these cases, discarding all the intermediate measurements is not feasible. Finally, there is a potential loss of dynamic information which may be of interest for process understanding when following this approach.

- ii) Estimating missing values. A second choice is to estimate the value of the biomass content every minute (Figure 3.3(b)). Interpolation can be used for this purpose since it might be a simple and suitable technique when the evolution of the variable trajectories -the dynamics- are smooth, although we do not exploit the multivariate nature of the data. This technique might work properly provided that measurements are auto-correlated, or otherwise that the measurements are not very spaced in time considering process dynamics. However, special caution should be taken when using univariate interpolation so that the multivariate structure of the data is not broken. To prevent the risk of inadvertently destroying this structure, the exploitation of the time-varying within-batch dynamics is strongly recommended for imputation. For this purpose, the missing data recovery ability of multivariate projection methods to latent structures can be used. Note that the performance of the resulting monitoring

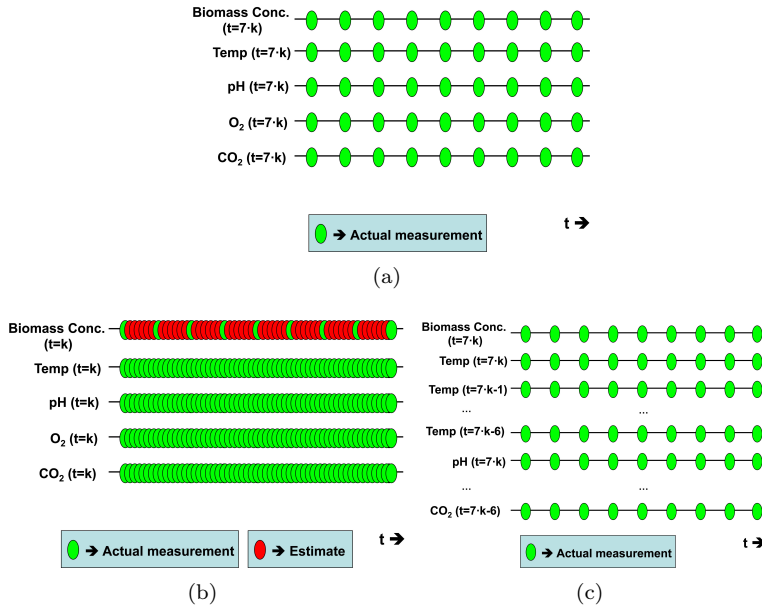


Figure 3.3. Equalization options for the fermentation process in which the biomass concentration is measured every 7 minutes whereas temperatures, pH and gas flows are measured every minute: (a) discarding intermediate values, (b) estimating intermediate values, and (c) rearranging intermediate values.

system will clearly depend on the estimation technique.

- iii) Rearranging data. A third possibility is to rearrange the data collected from temperatures, pHs and gas flows so that all intermediate measurements are treated as different variables (Figure 3.3(c)). This option has the same between-observations interval as the first choice (7 minutes). This equalization enriches the original measurement vectors by adding lagged variables, therefore, no information is discarded as in the first option. In case that a more frequent sampling rate is desired, missing data imputation techniques can be applied to estimate intermediate values taking advantage of the process dynamics captured by adding lagged variables. The drawback of this approach is that may lead to more complex models to interpret.

In the preceding example, although the variables were collected at

different rates, they were equalized. Thus, the biomass content was measured every seven measurements of pH and the others. Nonetheless, the example is also valid for situations when there is no equalization. For instance, this would be the case if temperatures, pH and gas flows were measured every 2 minutes. A possible solution is to interpolate these process variables flows to a 7 minutes rate.

On the other hand, option i) is not applicable to the more complicated case of multi-rate systems. In this situation, all process variables may be collected at different sampling time points. Therefore, hardly any complete observation may be found and the complete data set would be discarded according to option i). Also, option iii) is not applicable in this case. Option ii) is the most generally applicable method. If the missing data algorithms inherent in multivariate projection methods to latent structures are used for the estimation in option ii), instead of univariate interpolation, the data matrix may be enhanced with additional columns with lagged and even future variables, similar as in option iii), to take into account dynamic information for the missing data estimation. Also, multi-model approaches may be necessary. The implications of the application of these solutions are studied by using batch data of the simulator of the *Saccharomyces cerevisiae* fermentation process in the following subsections, in particular the first ten process variables of the first batch of the three-way array $\tilde{\mathbf{X}}_1$ corresponding to Set #1 (see Chapter 2). For the sake of simplicity, the case of one variable collected at lower rate than the rest of the process variables is considered.

The comparison of the equalization techniques is conducted by comparing the latent structures of the multivariate models fitted on within-batch data, which is the only source of accurate information available at this modeling stage. Under this premise, the multivariate models built in this section are not aimed to be used for process understanding, troubleshooting or monitoring, but to illustrate how the latent structures (summary of the variances and instantaneous cross-covariances of the process variables) for a particular batch is affected after being equalized.

3.3.1 Discarding intermediate values

Similar to the example proposed in the previous section, we will consider that the biomass concentration is measured every 10 measurements of the other variables, as depicted in Figure 3.4(a). Variables equalized according option i) -discarding intermediate values-

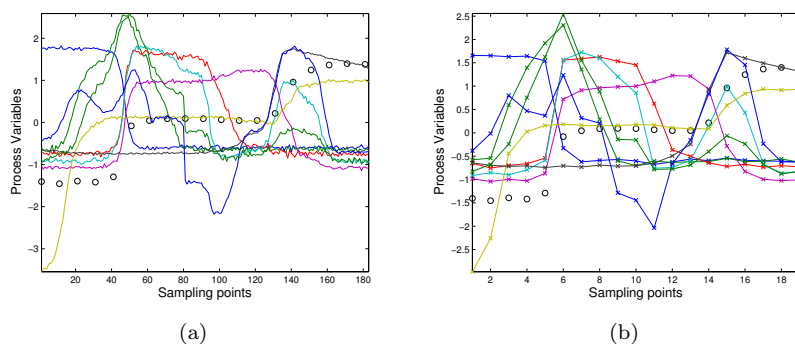


Figure 3.4. Measurements of the process fermentation where the biomass concentration (open circles) is measured every 10 measurements of the other variables (a) and equalization following option i) discarding intermediate values (b).

are shown in Figure 3.4(b). In addition to potential aliasing problems caused by the resampling (failing to obey the Nyquist-Shannon sampling theorem), it can be seen that there is an important loss of information in the equalization. In particular, there exist more abrupt changes in the shape of the equalized variable (see Figure 3.4(b)) than in the unequalized variables (see Figure 3.4(a)). Note that the less frequent a variable is sampled and subsequently equalized, the more changes are in the shape of the variable trajectories. Modifications of the original variable may mask mild changes and abnormal situations, which may not be detected by the monitoring system.

In Figure 3.5, the scores, loadings and residuals of a PCA model -2 PCs- calibrated from the original data (where biomass concentration is measured at the same sampling rate as the other variables (see Figure 3.4(a)) is compared with those corresponding to the model calibrated from the equalized data obtained by discarding intermediate values (see Figure 3.4(b)) after autoscaling the process variables. There are differences between both models as can be appreciated from Figure 3.5. The score trajectories -i.e. the trajectories formed by the score values obtained for each sampling time point of the selected batch corresponding to the original data and the data equalized by choice i) do not overlap (see Figure 3.5(a)). Likewise, the location of the loadings in the plot shown in Figure 3.5(b) differs among models although the differences are not large. Looking at the residuals of the biomass concentration from both PCA models in Figure 3.5(c), slight

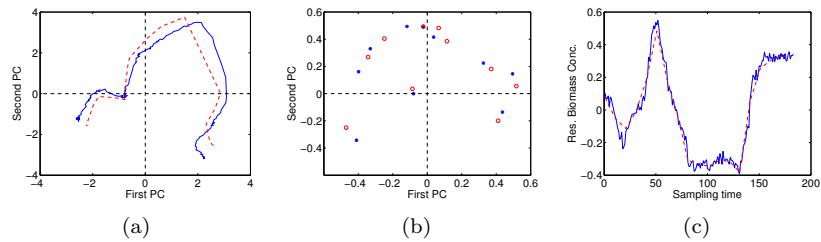


Figure 3.5. Comparison of the PCA models from the original data -no missing values- (blue lines and markers) and the data after discarding the intermediate values (red lines and markers): (a) scores, (b) loadings and (c) residuals.

differences are observed again. These differences may be relevant for the performance of a fault detection system. In case that the purpose of the analysis is to design a monitoring scheme, this effect should be rigorously studied. This resampling of the data is only valid if the Nyquist-Shannon sampling theorem is obeyed.

Unfortunately, in a real situation the model built from the original data is not available for comparison. A possible alternative is to compare the models obtained from data equalized following options i) and ii).

3.3.2 Estimating missing values

There are a number of approaches to impute missing data, from the simplest approach, such the univariate interpolation of intermediate values, to the most complex and powerful methods based on the exploitation of the correlation among variables using projection methods to latent structures. The former consists of calculating new data points within a specified range of a discrete set of sampled points by using low-degree (spline interpolation) or high-degree (polynomial interpolation) polynomials in each of the intervals, choosing the polynomial pieces such that they fit smoothly together [73]. The latter are estimation methods based on PCA models, which can be classified in two groups: non-regression-based and regression-based methods [135]. For model building, the calibration based on the *Trimmed Score Regression* (TSR) and the *Iterative Algorithm* (IA) is used due to its accurate results in missing data imputation¹ [136, 137].

¹Note that the resulting model from the TSR-IA procedure is only used for process understanding and comparative purposes, and not for designing a monitoring system.

In the TSR-IA algorithm, three steps are iteratively repeated until convergence: i) filling the missing positions of the new observation with an initial estimate, ii) calculating the estimated vector of scores for observations using the TSR algorithm and the latent structure from the PCA model fitted, and iii) re-estimating the missing values by reconstructing the original data matrix with the estimated vector of the TSR-scores and the loading vector of the PCA model. For details in this imputation technique, readers are referred to the original work [137]. In the following, the differences between the two aforementioned approaches for estimating the missing values in the biomass concentration are studied.

3.3.2.1 Interpolation

A univariate interpolation based on cubic splines is carried out to impute the nine intermediate values between samples (see Figure 3.6(a)). As can be seen, the estimation seems to be accurate since the original and equalized trajectories almost overlap, however, oscillations have been created in the interpolation. In Figure 3.6(b) the sampled and equalized values of the biomass concentration are depicted for comparison. The results of this particular simulation of a particular case of multi-rate sampling indicate that the cubic interpolation works out well enough if the criteria are that predicted versus sampled values roughly lie in a diagonal line and the *Sum of Squared Residuals* (SSR) of the biomass concentration is low ($SSR = 0.506$). However, a low SSR is not a decisive factor to confirm that interpolation meets the main requirement to continue the modeling of batch data, which is the preservation of the original time-varying covariance structure.

At this point it is worth warning against the danger of applying univariate interpolation on dynamic data, and the effects on the relationships of the process variables captured in the covariance matrix. The main problem polynomial interpolation might generate is the so-called Runge's phenomenon, which is a problem of oscillation that occurs when using polynomials of high degree over a set of equispaced interpolation points [138]. These oscillations can be found at any interval, although is more frequent at the edges of the intervals (start and end points), and with more severity, at the surrounding points of an outlier [134].

An example of the oscillatory problem can be found in the interpolated variable depicted in Figure 3.6(a). Comparing the actual and interpolated values of the biomass concentration, there exist damped oscillations produced by the polynomial interpolation at the

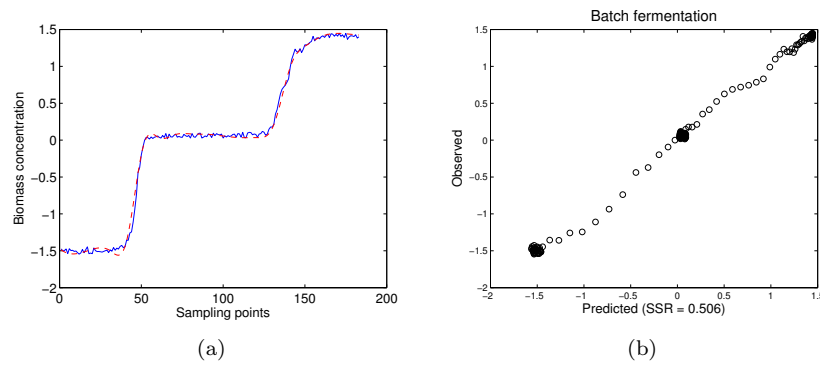


Figure 3.6. Comparison of the original data (blue line) versus data resulting from the estimation of the biomass concentration using interpolation based on cubic splines (red line). Note that SSR is calculated as $SSR = (\mathbf{x}_j - \hat{\mathbf{x}}_j)^2$, where \mathbf{x}_j and $\hat{\mathbf{x}}_j$ are the column vectors containing the sampled and estimated samples of the biomass concentration $j = 6$ of the selected batch.

first stage of the fermentation -the first 40 sampling time points-, which is an artificial behavior not observable in the original variable (see solid blue and red dashed lines in Figure 3.6(a)). In addition, these oscillations are reproduced in the transition between the two exponential growth stages -between sampling time points #50 and #130. The difference between this oscillation and the one at the start of the batch lies on the generated trend. As can be observed from sampling time point #60 to #130, a negative trend is created by the univariate polynomial interpolation as a consequence of the change in the signal induced by the second exponential growth stage. This behavior might cause a change on the time-varying dynamics, and hence, on the instantaneous and dynamic relationships of the process variables. As a result, the bilinear modeling of batch data might be severely jeopardized. Unrealistic correlations might be generated, thereby potentially affecting the interpretation of the statistical model, the prediction of quality properties, the imputation of missing data through the use of statistical soft-sensors, and the detection and diagnosis of abnormal occurrences in the process. Though some authors claim that the estimation accuracy is not the main concern when the aim is to detect faults [90], the addition of artificial correlations might rise the false alarm rate, or even mask failures in the process. A sensible way to proceed is to compare the model obtained after different equalizations and check whether the models are similar

or not. In the case of existing differences, the underlying reasons should be investigated. Also, it is useful to have a list of all the steps performed in the model building, since unexpected results in the monitoring system may be caused by decisions at the very beginning of the bilinear modeling cycle, for instance those related to the data equalization. If there are suspicions that the multivariate correlation structure among batches is affected, the resulting equalization should be discarded.

3.3.2.2 Missing data recovery

Another possibility tailored to batch process modeling needs is to use the missing values recovery ability of the projection models to latent structures, which exploits the multivariate nature of the data for imputation. The outcomes of the application of this procedure to the data set under study are shown in Figure 3.7. In terms of preservation of the dynamic data structure, Figure 3.7(a) shows that the damped oscillations created by polynomial interpolation are not originated. The results are improved because the correlation of the remaining process variables are used to accurately impute the missing values without distorting the covariance matrix, unlike interpolation. In Figure 3.7(b), the residuals of the intermediate samples are neglected in the PCA model fitted in the TSR-IA algorithm by imposing these values to the average value (0's values when data are mean-centered in the first iteration of the TSR-IA algorithm). Note that the variable trajectories of the fermentation process are characterized by having long periods of noise with low variability (e.g. glucose, acetate and biomass concentrations), and periods of high variability containing common cause process variation. To prevent noise from having more weight and signal from having less weight in the model, which would negatively affect the quality of the missing data imputation, the batch trajectories are not auto-scaled. To overcome the problem of the different scales of the process variables, more principal components are extracted to account those components explaining less variability. The estimation of the intermediate values using this procedure outperforms the interpolation method ($SSR = 0.468$ for missing data imputation by TSR-IA versus $SSR = 0.506$ for polynomial interpolation). This is mainly due to the high collinearity and data redundancy present in the batch trajectories, which permits accurately imputing the missing values and reduce the SSR. However, as explained above, the main difference lies on the preservation of the time-varying dynamics

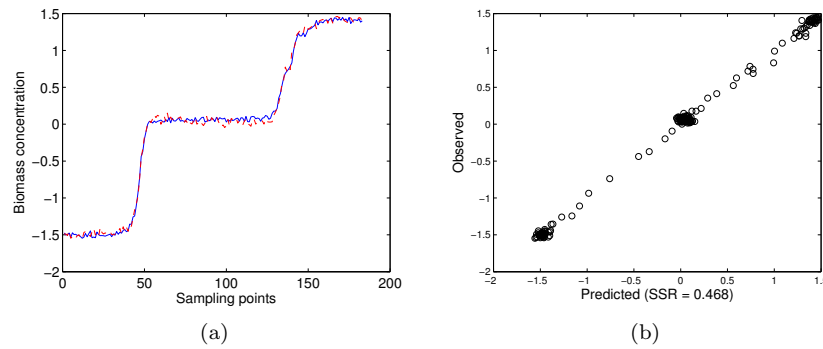


Figure 3.7. Comparison of the original data (blue line) versus data resulting from the estimation of the biomass concentration using the TSR-IA algorithm (red line). Note that SSR is calculated as $SSR = (\mathbf{x}_j - \hat{\mathbf{x}}_j)^2$, where \mathbf{x}_j and $\hat{\mathbf{x}}_j$ are the column vectors containing the sampled and estimated samples of the biomass concentration $j = 6$ of the selected batch.

within the batch run when missing values are imputed, unlike when missing values are interpolated.

A drawback of the estimation shown in the previous PCA model using a single-phase approach is that the dynamics in the variables are not taken into account. As commented in Chapter 1, a method to include dynamic information in the estimation is to add lagged variables as additional columns. For instance, if one order of dynamics is desired, each row of the resulting matrix of data must contain the measurement of the process variables at time k and $k - 1$. By adding information of several orders, the estimates of the missing values of a variable are not only based on instantaneous values of other variables, but also on past values of the variable itself and the other variables. The result of using this idea for the whole batch is shown in Figure 3.8 - 2 LMVs are added and residuals are set to 0. Similar to the non-dynamic single-phase model approach, the exploitation of the multivariate data structure avoids the addition of oscillations that potentially might affect the outcomes of later steps in the modeling cycle (see Figure 3.8(a)). The dynamic single-phase model approach slightly outperforms the non-dynamic single-phase approach, taking as criterion the difference between observed and predicted values ($SSR = 0.457$ for dynamic versus $SSR = 0.468$ for non-dynamic approach). The addition of lagged variables, and hence the capture of more dynamics, shows that the accuracy of the multivariate model to

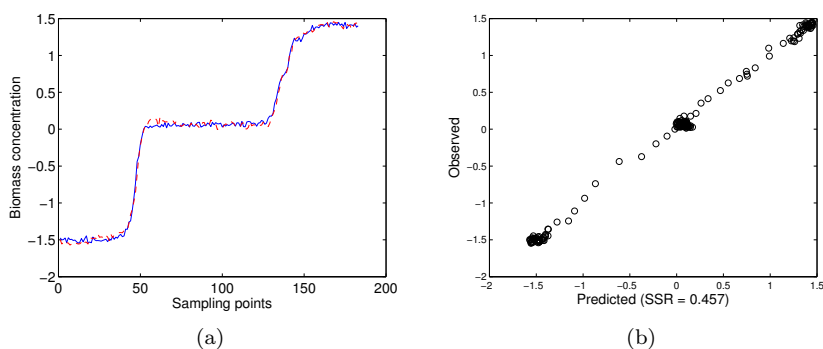


Figure 3.8. Comparison of the original data (blue line) versus data resulting from the estimation of the biomass concentration using the TSR-IA algorithm in a single-phase dynamic model approach with 2 LMVs (red line). Note that SSR is calculated as $SSR = (\mathbf{x}_j - \hat{\mathbf{x}}_j)^2$, where \mathbf{x}_j and $\hat{\mathbf{x}}_j$ are the column vectors containing the sampled and estimated samples of the biomass concentration $j = 6$ of the selected batch.

impute missing values can be enhanced. Hence, both methods clearly outperform interpolation results, not only in terms of accuracy in prediction, but also and more importantly, in the preservation of the covariance matrix.

In the application of the projection methods to latent structures we are assuming that the relationship among process variables remains the same during the evolution of the batch. However, batch processes are characterized by having time-varying relationships among process variables. In addition, the limitation of using the within-batch data hampers the removal of non-linearities in data. Since the bilinear PCA method is used for equalization, the accuracy of the imputation might be seriously affected because not all the complex relationships can be captured. Let us have a look at the residuals of the first PC of the fitted PCA model on the original data in Figure 3.9 to observe this effect. The residuals are clearly showing a changing non-linear behavior during the processing. This is a phenomenon that has been reported several times in the literature [87, 102, 139, 140] and has been discussed in Chapter 1 as well. The relationships among the ten process variables, which must be captured with PCA, change along the batch processing. Thus, if a single model is fitted, the relationships captured in the model are simply an average which may not be representative of some parts -or even any part- of the process.

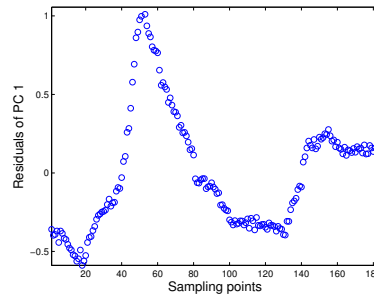


Figure 3.9. Residuals of of the biomass concentration after extracting the first PC of the PCA model fitted on the original data.

A possible solution is to properly split the data in sub-models in such a way that the process behavior can be well approximated with linear techniques and the changing relationship structure can be captured. Based on the prior knowledge on the fermentation process, let us divide the batch trajectories into the following three phases: $[1, 40]$, $[41, 127]$, and $[128, 183]$. The TSR-IA method is applied to each of these phases to impute the missing values neglecting the residuals in the computation -i.e. setting the residuals to 0 when data are mean-centered. The results of the phase-based estimation of the intermediate values of the biomass concentration are depicted in Figure 3.10(a). As can be observed, it yields better results in terms of prediction since the associated $SSR = 0.275$ is much lower than in the imputation by TSR-IA ($SSR = 0.468$). These results support the claim that the quality in the estimation of a model depends very much on how the time-varying process dynamics are captured.

3.3.2.3 Comparison of equalization methods based on latent structures

Often, the accuracy of the estimation is not the most important aspect in the analysis or the design of a monitoring scheme. The main objective pursued is that the missing values do not affect the covariance structure present in the available data, and hence, the calibration of the model. In Figure 3.11, the model obtained from the original data -full sampling rate of biomass concentration- is compared to the model obtained after the estimation of the biomass concentration by using the different approaches. Slight differences can be appreciated in the resulting latent structures, in particular,

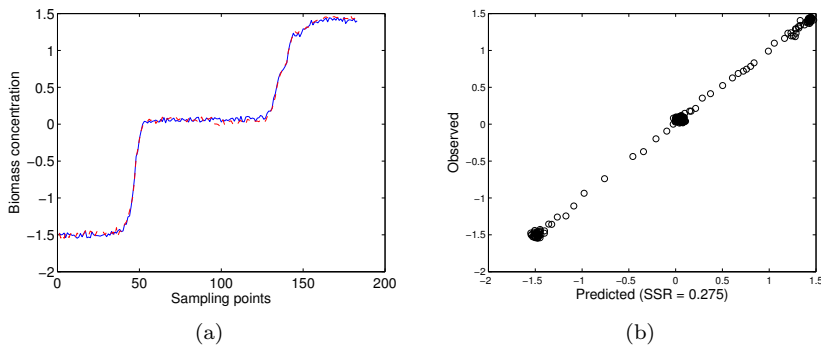


Figure 3.10. Comparison of the original data versus data resulting from the estimation of the biomass concentration using the TSR-IA algorithm in a multi-phase-based approach. Note that SSR for each of the equalization methods is calculated as $SSR = (\mathbf{x}_j - \hat{\mathbf{x}}_j)^2$, where \mathbf{x}_j and $\hat{\mathbf{x}}_j$ are the column vectors containing the sampled and estimated samples of the biomass concentration $j = 6$ of the selected batch.

between the model obtained on interpolated data (see Figure 3.11(a)) and the models using the data imputed by using the correlation structure of the process variables (see Figure 3.11(b), Figure 3.11(c) and Figure 3.11(d)). Among the models built on imputed data, the multi-phase model shows more accuracy on replicating the latent structure of the model built on the original data (see score and loading plots in Figure 3.11(d)), and on minimizing the difference in the residuals (see residual plot in Figure 3.11(d)). Apart from visualizing the resulting models fitted on the sole data available due to synchronization problems -i.e within-batch data, the results of the imputation should be carefully checked with fields experts. The most important objective is to ensure that the imputation is accurate and the covariance structure is not distorted. In this regard, the missing data recovery techniques based on the exploitation of the correlation of the process variables over time have shown better performance than interpolation.

3.3.3 Rearranging data

Data in this example are rearranged according to option iii), so that all the intermediate measurements are treated as additional variables -columns. The resulting data set is plotted in Figure 3.12.

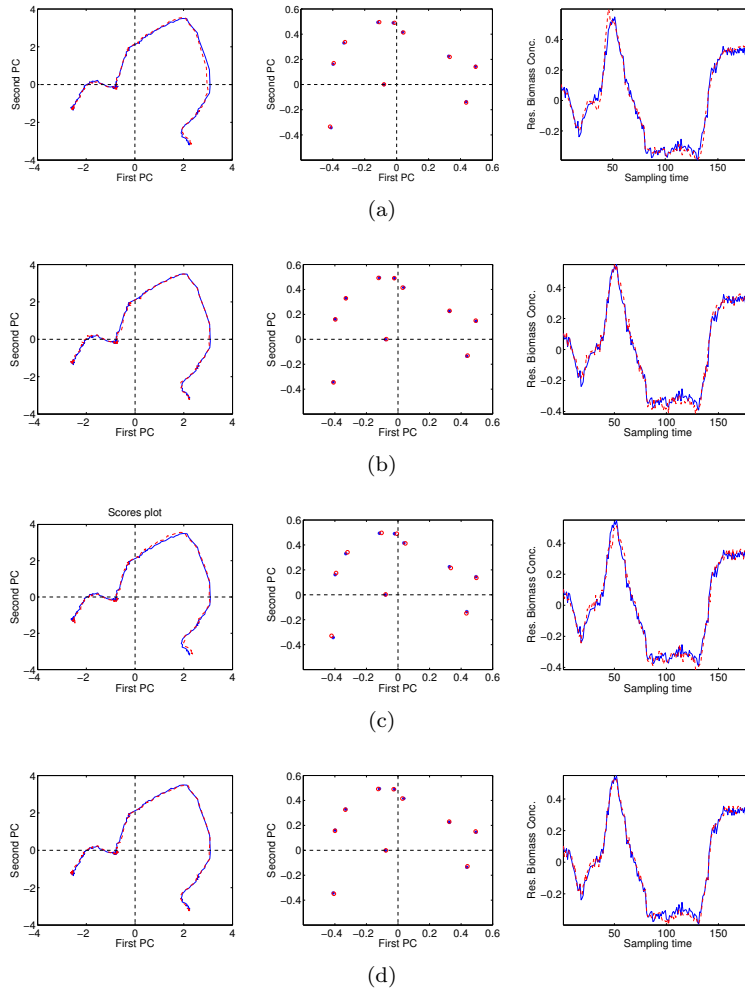


Figure 3.11. Comparison of the PCA models from the original data (blue lines and markers) and the estimated data with different approaches (red lines and markers): (a) cubic spline interpolation, (b) TSR-IA (no residuals), (c) single-phase dynamic model approach, and (d) multi-phase-based approach. First column (scores); second column (loadings); third column (residuals of the biomass concentration).

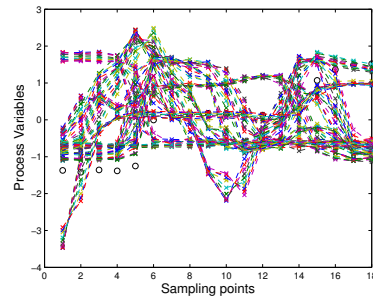


Figure 3.12. Data resulting from the rearrangement of the data.

A quick look at the figure leads to think that this approach will complicate the understanding and interpretation of the model. This idea is enforced when the model of the original data is compared with the one obtained from the rearranged data in Figure 3.13. We have a completely different model and interpretation may become challenging.

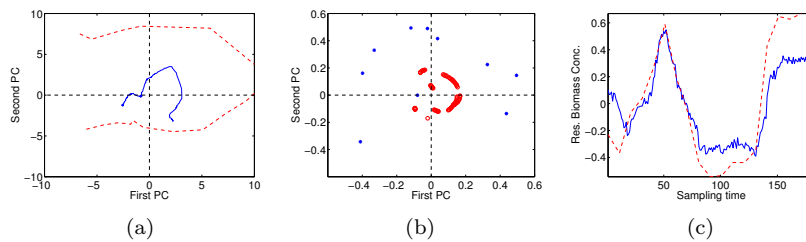


Figure 3.13. Comparison of the PCA models from the original data -no missing values- (blue lines and markers) and the data rearranged (red lines and markers). First column (scores); second column (loadings); third column (residuals of the biomass concentration).

3.4 Multi-rate system

A multi-rate sampling scenario commonly found in industry is the one produced by compression algorithms. The main difference between this unequalization case and the other sampling scenarios considered in this chapter is that the number of samples among variables and phases differ across batches. The reason of this non-uniform

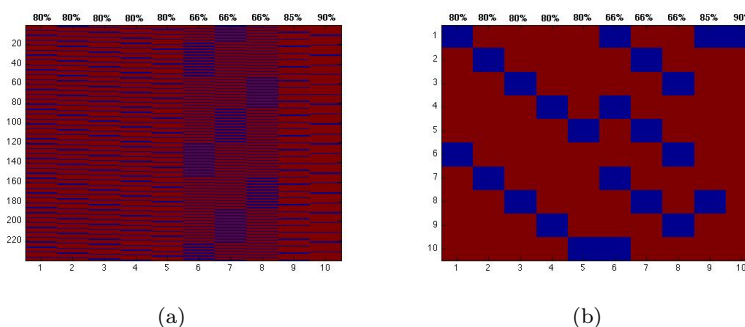


Figure 3.14. Missing data map showing the available (blue) and missing (red) measurements for the ten process variables of the selected batch for the whole batch run (a) and for the first 10 sampling time points (b). The percentage of missing data per variable is shown in the columns of the maps.

sampling is the policy implemented in the compression algorithm. Typically, a sample of a specific variable is archived only when a time interval is elapsed, or when the difference between the current and previous value exceeds a threshold commonly associated with the calibration of the instrument [125]. A good example of this scenario was reported in [131], where data of the manufacturing of an active pharmaceutical ingredient were used. The objective was to extract valuable information for process understanding and improvement from compressed historical data. In this case study, the median number of samples per variable for the reaction step varied between 65 and 1000 samples, whereas for the distillation step, between 31 and 2001 samples. With the aim of preserving the number of samples and the sampling frequency in each stage, a re-sampling mechanism was proposed prior to bilinear modeling. This procedure consisted of re-sampling each variable as many times as the ratio between the median number of total samples and the median fraction of total time spent in each stage using interpolation. The authors claimed that this methodology allowed them to equalize the batch data and also reach some degree of alignment. However, interpolating this type of unequalized data will not guarantee reconstruction of the time dependent correlation structure in batch data. [79].

An extreme case of multi-rate sampling is simulated to illustrate the performance of interpolation and missing data recovery techniques. In particular, the 10 process variables of the *Saccharomyces Cerevisiae* cultivation process are sampled with different sampling frequencies

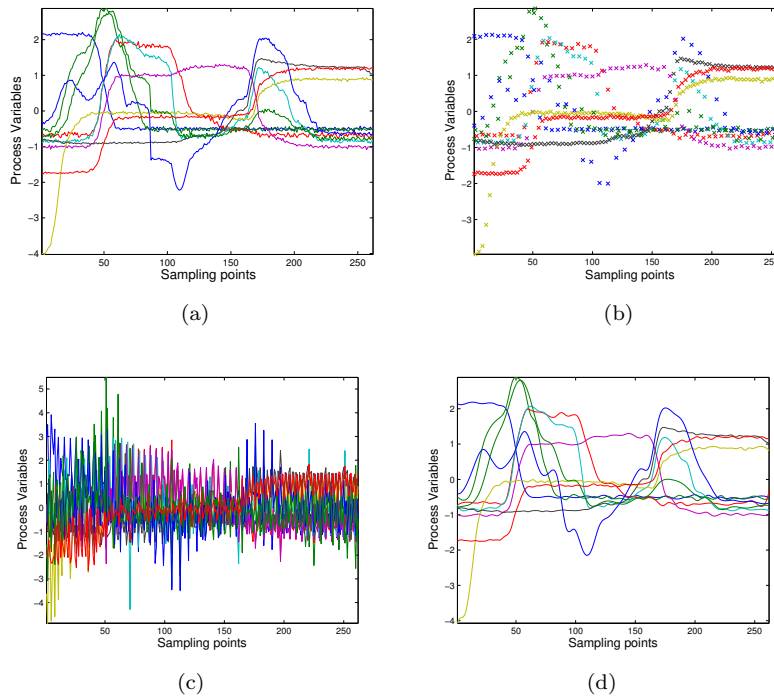


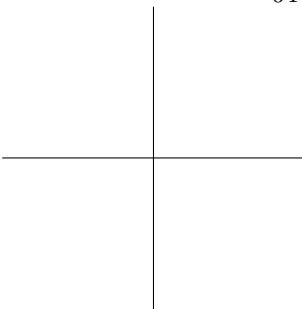
Figure 3.15. Multi-rate example: (a) original data, (b) multi-rate data, (c) data recovered by TSR-IA, and (d) interpolated data.

among sets of variables: the first five variables were collected every 5 sampling time points and at consecutive times; variables from 6 to 8 were collected every 3 sampling time points and at consecutive sampling time points, variable #9 was collected every 7 sampling time points, and finally, variable #10 was collected every 10 sampling time points (see Figure 3.14). As a result, a set of unequalized batch trajectories is generated (see original and re-sampled data for comparison in Figure 3.15(a) and Figure 3.15(b), respectively). The percentage of missing data is excessively high due to the type of multi-rate scenario simulated, as appreciated from Figure 3.14. In this situation, when each observation has more missing than available values, the use of the TSR-IA method is not adequate, as illustrated in 3.15(c). In order to successfully impute missing values there must be a certain degree of redundancy in data and collinearity, which is not provided in the data set of this example. In [137],

a comparative study was conducted among different missing data imputation techniques in scenarios of diverse percentages of missing data without the presence of outliers. One of the conclusions of this study was that the performance of the TSR algorithm considerably degrades when there is a 70%-90% of missing data on measurements, assuming the presence of redundancy and high collinearity. In the simulated multi-rate example, the actual correlation among process variables over time within the batch is not well represented by the updated variance-covariance matrices estimated in each iteration of the IA algorithm. Hence, the missing measurements cannot be accurately imputed from the measured process variables.

In the simulated multi-rate scenario, univariate interpolation improves the results obtained from the missing data imputation at a first glance (see Figure 3.15(d)). However, special caution should be taken with the potential aliasing effect produced by the reconstruction of the trajectory. Comparing the original and re-sampled data (see Figure 3.15(a) and Figure 3.15(d), respectively), there is a loss of information, mainly caused by the sampling frequency at which the data were archived (see Figure 3.15(b)). Though the sampling frequency obeys the Nyquist-Shannon sampling theorem, univariate interpolation might add artificial landmarks, and hence, artificial correlation that can jeopardize the outcomes of subsequent bilinear multivariate statistical modeling. A clear example of this effect can be observed on the oscillations along the flat profiles of some variables affected by white noise (see blue variable trajectories in Figure 3.15(d)). If there is a threat of distorting the covariance matrix of the data across batches, the corresponding equalization should be discarded.

A typical situation where the result of interpolation should be certainly validated is when a high number of consecutive samples is missing due a problem in data collection. Notice that this occurs frequently since it is what happens when a sensor breaks down. An illustrative example is shown in Figure 3.16. A set of 50 consecutive measurements of the ninth variable was lost in the multi-rate system (see missing values between the 80th and 130th sampling time point in the trajectory depicted by black squares in Figure 3.16(b)). The reconstruction using interpolation is not appropriate (see thick black line in Figure 3.16(c)). In practice, practitioners do not know whether interpolation is the best method to apply because the original data is not available. In this example, the type of unequalization is known as well as interpolation is prone to error. A way to improve the estimation is to discard the estimates corresponding to the missing



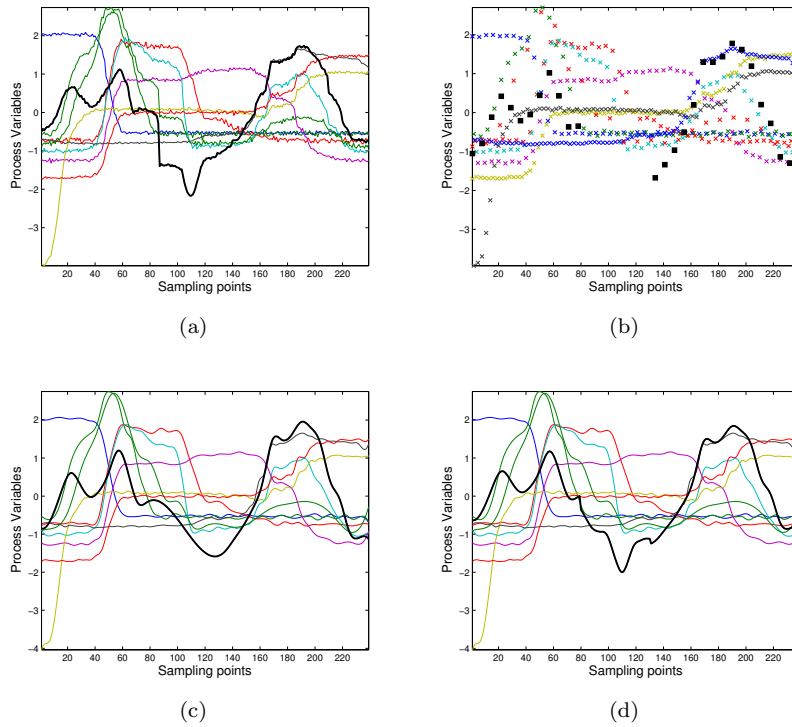


Figure 3.16. Multi-rate example with consecutive missing values due to a sensor failure: (a) original data, (b) multi-rate data with missing values in the time interval $[80,130]$, (c) interpolated data, and (d) interpolated data with estimation of the missing part by the TSR-IA algorithm.

period in the ninth variable and try to recover them using the TSR-IA algorithm and the rest of the interpolated data (see thick black line in Figure 3.16(d)). Although the estimation is not optimum, it outperforms the estimation only considering interpolation. The estimation can be further improved using some of the approaches already discussed in the previous examples, i.e. modeling by phases and incorporating dynamics.

Note that when the sampling rate is different across variables and batches, the unique straightforward solution for data equalization is the interpolation of the samples, as long as the new imposed sampling frequency is compliant with the Nyquist-Shannon sampling theorem. The TSR-IA method cannot be applied since there is

not a common sampling rate in all batches. Hence, the number of samples might notably vary and the actual correlation cannot be exploited for missing data imputation. Even though the interpolation method can be applied, special attention must be taken since there might be missing features that this method might not be able to capture. The best practice is to establish a proper sampling rate in the process that takes into consideration sampling constraints (e.g. cost of sampling for subsequent laboratory analyses or importance of process stages), thereby leading to a similar multi-rate sampling system across batches.

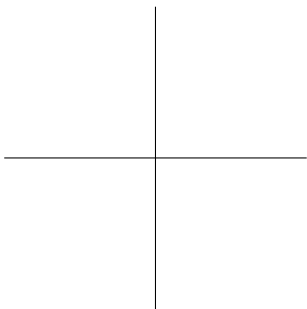
3.5 Conclusions

The problem of the equalization of variables and batches is addressed. To proceed with the alignment of the variable trajectories and the subsequent bilinear modeling of batch data, the variables must be equalized in such a way that they contain values for all the sampling time points across batches. The different types of unequalization scenarios that practitioners might find in batch processes are discussed: different sampling rate per stage and same policy for all variables and batches; multi-rate sampling in process variables and similar policy among batches; and multi-rate sampling both in process variables and batches. Three main solutions are proposed to overcome the problem of equalization: i) discarding intermediate values, ii) estimating missing values and iii) appropriate arrangement of data. The application of these equalization strategies are restricted to the only source of accurate information available at this modeling stage: the within-batch variation. Equalization cannot be performed across batches because there is no guarantee that the batch trajectories are synchronized.

Simulated batch data from the fermentation of the *Saccharomyces cerevisiae* cultivation are used to illustrate the implications of the use of the proposed techniques to overcome the unequalization problem. Only the simplest case of unequalization is considered for illustration, the sampling at lower frequency of the biomass concentration. With this example, we have seen that for this particular case, notable differences were found between the proposed equalization methods. Equalization methods can severely damage the covariance matrix of the available data, even though the approximation between predicted and observed samples is rather acceptable. In contrast, the missing data recovery methods based on the exploitation of the correlation

of process variables clearly outperformed interpolation. Not only the approximation of imputed variables yielded better results than interpolation, but also the multivariate nature of data was preserved. Note that these differences may differ in the type of the process, the degree of unequalization and the number of variables affected by different sampling policies.

Neither discarding intermediate values nor arranging batch data are applicable to the more complicated case of multi-rate systems. In this situation, all process variables may be collected at different sampling time points. Therefore, hardly any complete observation may be found and the complete data set would be discarded according to the first solution. The imputation of the values not sampled in the variables is the most generally applicable method. If a projection model to latent structures is used for data imputation, the data matrix may be enhanced with additional columns with lagged variables. When batch data are affected by non-linear relationships, and these relationships vary over time, the generation of multi-phase multivariate models is required to overcome these common problems in batch processes. However, if batch data are hardly redundant and not collinear, the application of these imputation techniques are inadequate. As an alternative, interpolation might be used at the risk of not reconstructing the time dependent correlation structure in batch data.



Batch synchronization

Part of the contents of this chapter has been included in the following publications:

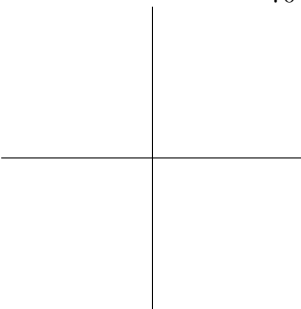
- [2] J.M. González-Martínez, J.A. Westerhuis and A. Ferrer. Using warping information for batch process monitoring and fault classification, *Chemometrics and Intelligent System Laboratory*, 127:210–217, 2013.
- [12] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 3: Batch Process Data. *Batch Processes: Monitoring and Process Understanding*, Wiley-VCH Verlag GmbH, publication due in 2016.
- [22] J.M. González-Martínez and A. Ferrer. Batch synchronization: a paramount step before bilinear modeling in batch multivariate statistical process control. *In proceedings of the Workshop on Chemometrics for Young Researchers*, page 32, A Coruña (Spain), 2011.
- [16] J.M. González-Martínez and A. Ferrer. A comparison of different methods for synhronization of batch trajectories. *In proceedings of 11th Scandinavian Symposium on Chemometrics*, page 83, Loen (Norway), 2009.

4.1 Introduction

In most batch processes, the assumption that all the batch trajectories are synchronized is rarely met. Typically, the recipes for automation are based on triggers that are seldom dependent on time, which causes the batch pace evolution to be different batch to batch [131]. In addition, different sizes of batch charge, modifications of the recipe to release products at lower costs, impurities in the raw materials, and disturbances in the environmental conditions may produce uneven time-length batches [62]. Hence, not only the collected batch trajectories for subsequent multivariate data analysis may have different lengths, but also the key process events do not overlap at the same time in all batches [79].

In this context of asynchronous batches, the application of multivariate projection methods, such as PCA and PLS, is not feasible. One of the strong assumptions of most of these models is that all batch trajectories have equal duration and are synchronized (i.e. similar events happen at the same sampling time points). In order to ensure that all batch trajectories have the same duration and the key process events happen at the same state of evolution, the synchronization of batch trajectories need to be always carried out prior to modeling.

In the literature, several discussions on the different patterns of time-varying batch trajectories and the possible synchronization solution can be found [79]. In case batches have different duration but trajectories overlap in the common time part (see Figure 4.1(a)), a PCA model can be fitted using the information from the long batches while the absent part of the trajectory for the short batches is treated as missing data [134]. For some batches data collection starts earlier than in other batches, so a shift can be applied to synchronize the batches (see Figure 4.1(b)). In more complex cases (see Figure 4.1(c) and 4.1(d)), trajectories of the variables have different shapes, even when batch duration is the same, indicating that the timing for key events during each batch is different. Examples of this occur when, e.g. various decisions during the batch are not automated but left to the discretion of an operator, or the process is divided in several stages based on the occurrence of some phenomena that happen at different sampling time points. In such cases, more complex techniques are required to carry out the synchronization. By and large, the synchronization of batch trajectories is always advisable, no matter the batch duration since process events may not be synchronized.



In this chapter, the context of batch synchronization in process chemometrics is addressed. Section 4.2 provides a state-of-the-art and a detailed description of the most used and cited synchronization methods by the chemometrics community. Section 4.3 introduces the warping information derived from synchronization and its importance in the bilinear process modeling cycle. Section 4.4 presents a comparative study of the selected synchronization methods on accurately capturing the time-varying process dynamics. Finally, some conclusions are provided in Section 4.5.

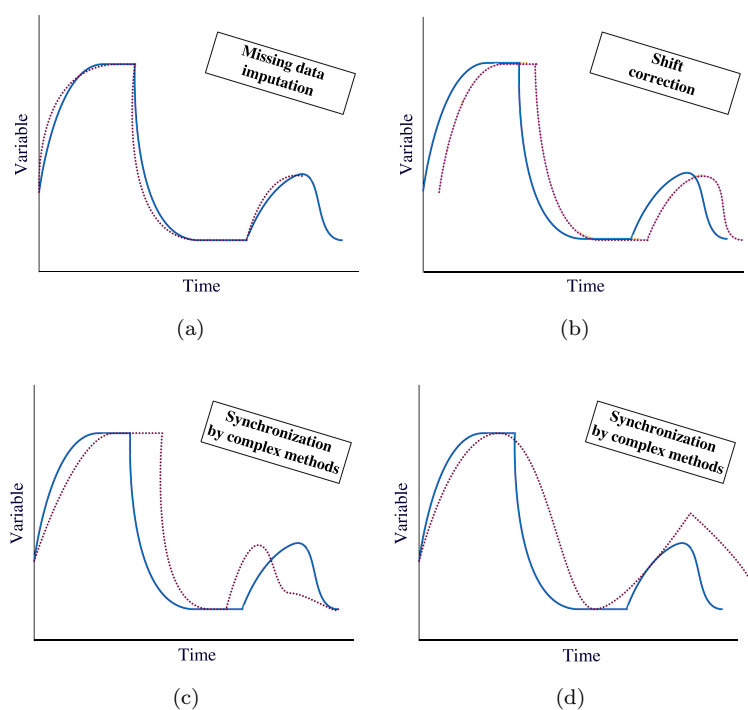


Figure 4.1. Trajectories belonging to one process variable of two batches with different patterns of varying-time trajectories: (a) different batch length with main events occurring at the same time, (b) different starting point yielding a shift in each occurrence, (c) similar batch length with key events do not occur at the same time intervals, and (d) different shape and length.

4.2 Synchronization approaches

A number of proposals for dealing with the most complex synchronization problems can be found in the literature. As mentioned in Chapter 1, the approaches for batch synchronization can be roughly classified into three categories: i) methods based on compressing/-expanding the raw trajectories using linear interpolation either in the batch time dimension or in an indicator variable dimension; ii) methods based on feature extraction; and iii) methods based on stretching, compressing and translating pieces of trajectories.

Within the first category, some authors dealt with the batch alignment issue using simple ideas, such as truncating the trajectories of all batches to the shortest batch length, or compressing/expanding the trajectories using linear time adjustments by dividing each sampling time point along the trajectory by the time at a certain percentage of the end-point [67, 141, 142]. These ideas, although simple, are often inadequate for aligning batch trajectories [80]. In [143], pseudo-batches based on the scarce quality measurements were created, and subsequently aligned to the end-point. These pseudo-batches were cut to the minimum length, yielding this way segments of constant length. Thereafter, a revised version of this methodology was proposed, which aligns batch trajectories by identifying short-window multivariate statistical models at first and then applying these identified models to extend shorter trajectories [144]. However, the latter two methods are not appropriate in a context of complex asynchronisms. An alternative method used for alignment of batch trajectories is the *Indicator Variable (IV)* approach [70]: "*One way to handle varying batch times in online monitoring is to replace time by another measured variable that progresses monotonically in time and has the same starting and ending value for each batch*". Monotonicity and high signal to noise ratio are necessary conditions for an indicator variable, but they are not sufficient. To be useful in practice the indicator variable also needs to represent the mechanism that drives the batch process -this requires process knowledge. The indicator variable can also be a variable computed from some of the measured variables; e.g. extent of reaction [93]. Application of this synchronization approach can be found in [67, 93, 96, 114, 145, 146, 147]. When an indicator variable is not available throughout the batch run, but some process variables can be used as an indicator at different process stages, the batch synchronization can be performed stage-by-stage [66]. If a suitable indicator variable is not available for a given batch process this type of synchronization cannot be

carried out and other approaches are required. PLS models between the variable-wise unfolded batch data matrix and the local batch time were also suggested to predict the batch 'maturity' and align accordingly [83, 96]. However, this strategy has several drawbacks in practice [5].

Procedures based on features extraction were also proposed for batch process synchronization. For instance, a mathematical matched filter to extract key events in batch trajectories in cases where they are not known beforehand was developed in [72]. More sophisticated approaches are curve registration [71, 73, 74] and dynamic locus analysis [148, 149, 150], which identify landmarks or special points that characterize process stages and changes (the so-called singular points) in a set of batch trajectories corresponding to process variables, and then, the test trajectories are warped based on the reference landmarks. In [151], raw batch trajectories are decomposed into approximations and details at different scales using wavelets. Contributions from each scale are collected in separate matrices, and data are synchronized at each level using an algorithm based on stretching, expanding and translating pieces of trajectories. Once synchronized, separate matrices are reconstructed to form new synchronized trajectories.

Other methodologies based on warping techniques, such as DTW and *Correlation Optimized Warping* (COW), have been proposed as methods of pattern matching in speech recognition [152] and methods to correct peak shifts in chromatographic profiles [153, 154, 155, 156, 157]. A good survey on warping methods for spectroscopic and chromatographic signal alignment can be found in [158]. In recent years, these methods have received much attention in process chemometrics to align and synchronize batch trajectories corresponding to process variables [76, 159, 160, 161, 162]. In [76], a pseudo-online version of DTW for batch synchronization was proposed and some guidelines to carry out the real-time synchronization were presented. Nonetheless, this real-time version was shown in a simulation study to be inappropriate for BMSPC due to the high false alarm rate [1]. The RGTW is a solution to overcome this problem [1]. A *Derivative* DTW (DDTW) algorithm was proposed [163] to capture the underlying process behavior fingerprinted in the trajectories using derivatives. Nonetheless, noisy data can severely affect the computation of numerical derivatives [78]. A robust DTW algorithm was proposed in [77] that combines a moving window least squares procedure with Derivative DTW to avoid singularity points and reduce the dependency of the results on the reference trajectory. Later on, the *Robust Derivative* DTW (RDDTW) algorithm was introduced [164] to cope

with singularity point and numerical derivative estimation problems of DTW and DDTW in the presence of noise. Results showed that combining the use of Savitzky-Golay filter and DDTW algorithm significantly reduce the number of singularity points and retain the most important features of the original trajectories. Another proposal to deal with the problem of derivatives computation in noisy data is the *Hybrid Derivative Dynamic Time Warping* (HDDTW) algorithm [78], which combines piece-wise-linear approximations of the unsynchronized trajectories and DDTW. Despite the larger number of synchronization methods proposed in the literature, only the most used and successful synchronization methods applied to process data are reviewed and thoroughly described in the following subsections: IV, TLEC, and DTW.

4.2.1 Indicator Variable

The IV synchronization technique is based on the idea of expressing the evolution of a certain batch as a function of a process variable instead of the batch time [82]. An indicator variable is defined as a process variable, or a variable computed from the measurements, that fulfills some requirements: to be strictly monotonic and smooth (not noisy), to have the same starting and end point over all batches, and to represent the key process events and driving mechanisms. In case such variable is available, a synchronization can be carried out by replacing the batch time as basis by the indicator variable. The indicator variable, for instance, can be the amount of monomer or other feeds added to the reactor, substrate concentrations, product concentration or product yields in bioprocesses. In any case, the choice of the indicator variable depends on the type of process. For a variable to be chosen as an indicator variable, the maturity or percentage of completion of a batch should be represented by such variable. The synchronization using the IV method is performed in the following way. Typically, non-uniform increments or IV levels are selected based on the prior process knowledge to mitigate the problems of univariate interpolation [134]. For instance, these intervals can be defined using polynomial functions of time [146] or the importance of certain process stages in the manufacturing [93]. Nonetheless, equal spaced intervals can also be chosen provided that the process pace is linear. Once the increments are defined, interpolation is used to transform the batch-time dimension into the indicator variable dimension.

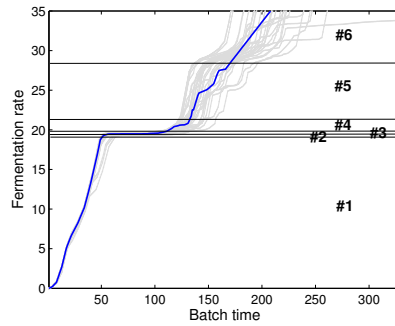


Figure 4.2. Fermentation rate used as an indicator variable for offline batch synchronization of the *Saccharomyces cerevisiae* cultivation process. Non-uniform increments are defined based on the evolution of the fermentation in six different stages of the batch run: #1, 54 increments; #2, 24 increments, #3, 26 increments, #4, 31 increments, #5, 31 increments, and #6: 43 increments. The blue line represents the IV trajectory of a batch whereas grey lines represent the IV trajectories of the remaining batches.

For illustrative purpose, let us qualify the simulation time of the *Saccharomyces cerevisiae* cultivation process as indicator variable (see Figure 4.2). This variable meets all the IV requirements: smooth, continuous, monotonic and representative of the events that drive the fermentation process. In regard to the latter requirement, the profile of the IV defines the fermentation evolution. When a value of 19.3 units is reached, the first exponential growth phase terminates (see Stage #1 in Figure 4.2). During this phase, glucose is in excess, the microorganism cannot digest the glucose at oxide-reductive growth, and consequently ethanol is formed till maximum concentration. Meanwhile, excretion of pyruvate and acetate occurs till maximum concentration. The transition between the first and second exponential growth happens in two different stages. In the first stage, the value of the IV rises until 19.5 units, instant at which the glucose is rapidly consumed and depleted (see Stage #2 in Figure 4.2). In the second stage, pyruvate is completely consumed whereas acetate is partly used during the oxide-reductive growth, instant at which the oxygen uptake rate decreases till minimum and the IV reaches 19.65 units (see Stage #3 in Figure 4.2). At this evolution point, the second exponential growth is triggered and the ethanol accumulated during the first exponential growth is utilized. Prior to this consumption, ethanol concentration rises till reaching the maximum level in the fermentation, at which the IV shows a value of 22 units

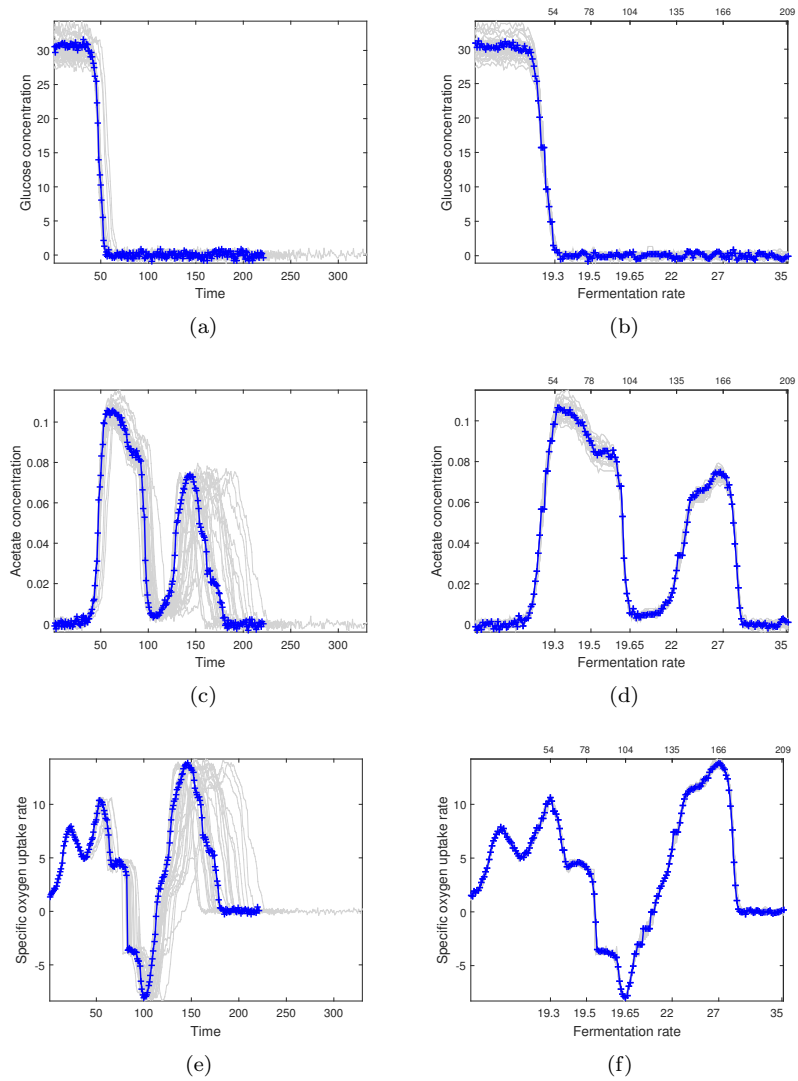


Figure 4.3. Offline synchronization of three variable trajectories belonging to the *Saccharomyces cerevisiae* cultivation process based on the IV technique: (a, b) glucose concentration, (c, d) acetate concentration, and (e, f) oxygen uptake rate. The raw batch trajectories are plotted as a function of time (a, c, and e) and the synchronized batch trajectories as a function of the increments of the indicator variable -fermentation rate- (lower x-axis) and of the number of interpolated samples (upper x-axis) (b, d, and f). Blue crossed lines represent the trajectories of a batch whereas grey lines represent the trajectories of the remaining batches.

(see Stage #4 in Figure 4.2). Afterward, the ethanol is consumed, causing the excretion of acetate till the ethanol concentration and the oxygen uptake rate reach a maximum level, where the former is a local maximum. This is the evolution time at which the second exponential growth finishes, the biomass concentration is maximum in the medium, and the IV rises up to 27 units (see Stage #5 in Figure 4.2). From this point onward, the fermentation enters in the stationary phase until the end, when the IV reaches the value of 35 units (see Stage #6 in Figure 4.2).

The synchronization of the fermentation data is performed by selecting non-uniform increments of the IV. In particular, the following IV levels are defined: 54 equal spaced increments for the IV range at Stage #1, 24 equal spaced increments at Stage #2, 26 equal spaced increments at Stage #3, 31 equal spaced increments at Stage #4, 31 equal spaced increments at Stage #5, and 43 equal spaced increments at Stage #6. Note that for the shortest stages representing the process evolution (Stages #2 and #3), the sampling is considerably higher than in other stages to be compliant with the Nyquist-Shannon sampling theorem (see Chapter 3). To complete the synchronization, the process variables are re-expressed as a function of the IV by interpolating the values at the 209 IV levels (see Figure 4.3).

For online synchronization, the value of the first IV level will not be estimated up to a measurement of the indicator value does not exceed the first increment. When such point is reached, a linear interpolation is carried out and subsequently the data transformed from the original scale to the indicator variable. The IV level time is calculated by interpolating the time belonging to the previous and current measurements. This is repeated as long as new measurements of the ongoing batch are available.

Two aspects of the IV approach worth discussing further are the constraints to nominate a process variable as indicator variable and the re-sampling procedure to transform the batch-time dimension into the indicator-variable dimension. Some authors assume that imposing a starting and ending value in all batches is a correct procedure to ensure meeting this IV requirement [165]. However, this assumption might severely perturb the outcomes of the synchronization. Other authors assume that defining constant intervals on the indicator variable is sufficient to re-sample the batch trajectories for synchronization [66]. However, if the indicator variable does not evolve linearly in time, the imposed frequency sampling would not be compliant with the Nyquist-Shannon sampling theorem, the shape of the resulting trajectories would be harmfully affected, and hence,

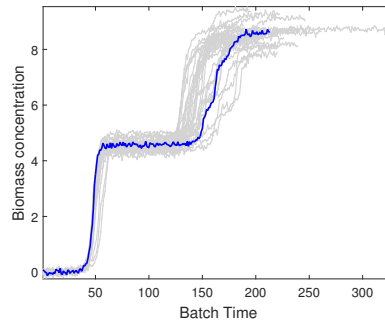


Figure 4.4. Biomass concentration used as an indicator variable for offline batch synchronization of the *Saccharomyces cerevisiae* cultivation process. The blue line represents the IV trajectory of a batch whereas grey lines represent the IV trajectories of the remaining batches.

the synchronized data would not be representative of the mechanism driving the process. In order to illustrate the pernicious effects of these policies, a synchronization is performed, taking the biomass concentration of the *Saccharomyces cerevisiae* cultivation process as the indicator variable, which is the only variable measured in the fermentation process subjected to be used as such (see Figure 4.4). As can be seen, this process variable does not show a common starting and ending value across batches. To proceed with the synchronization, the batch trajectories are cut to the same ending value in all batches, and constant increments are defined along the indicator variable. The resulting trajectories of this synchronization for glucose concentration, acetate concentration and oxygen uptake rate are depicted in Figure 4.5. Comparing the raw and synchronized trajectories, apart from observing a downsampling in all the variables, there is a loss of information caused by interpolating at constant increments on the indicator variable. Even though the resulting variable trajectories are equal in length, the loss of the landmarks defining the main process events over the batch run disables the batch data to be used for modeling.

Despite some authors have claimed in the literature that IV is the easiest approach to implement and apply to batch data, there are some disadvantages that need to be emphasized. The lack of a process variable that meets the requirements to be an appropriate candidate for a single indicator variable for the entire duration of the batch, or the lack of process knowledge to compute an indicator variable

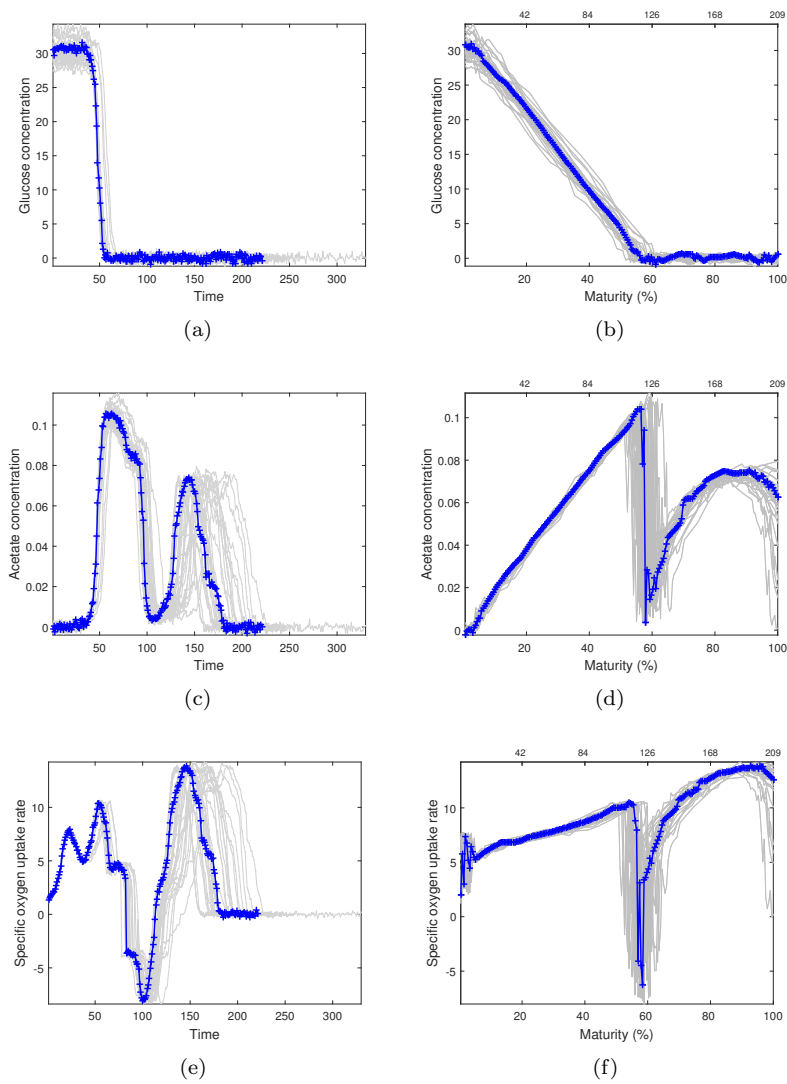


Figure 4.5. Offline synchronization of three variable trajectories belonging to the *Saccharomyces cerevisiae* cultivation process based on the IV technique: (a, b) glucose concentration, (c, d) acetate concentration, and (e, f) oxygen uptake rate. The raw batch trajectories are plotted as a function of time (a, c, and e) and the synchronized batch trajectories as a function of the maturity of the process based on the biomass fermentation (lower x-axis) and as a function of the number of interpolated samples (upper x-axis) (b, d, and f). Blue crossed lines represent the trajectories of a batch whereas grey lines represent the trajectories of the remaining batches.

that represents the driving force of the batch process and contains the key process events, are the main pitfalls. In addition, this synchronization technique requires expert knowledge in the process to define the increments to use in interpolation. If the IV levels are not properly determined, interpolation might produce downsampling and destroy the covariance matrix of the data available. Even though the increments are well designated, interpolation might cause the propagation of outliers and/or the perturbation of the correlation structure [134]. In these cases, the use of more sophisticated synchronization techniques are needed.

4.2.2 Time Linear Expanding/Compressing

The TLEC synchronization [64] is a method integrated in the *Observation Wise Unfolding-T Scores Batch Wise Unfolding* (OWU-TBWU) approach [62] and implemented in SIMCA Release 13.0.3 -Umetrics software- [65] for post-batch and real-time BMSPC. For the sake of thoroughly explanation, the different modeling steps of OWU-TBWU are described.

The OWU-TBWU approach [62] consists of two levels: the Observation-Wise Unfolding (OWU) (also called variable-wise unfolding) and the T-scores Batch Wise Unfolding (TBWU) levels. In the former (OWU level), the main motivation is dimension reduction before the BWU stage, which is desired in a context of a large number of process variables (e.g. spectroscopic or chromatographic data). The TBWU level provides a monitoring scheme for end-of-batch online process monitoring based on the OWU scores, Hotelling's T^2 and DModX (equivalent to Squared Prediction Error, SPE). Concerning the second level, the aim is to analyze the differences among the batches using the information summarized through the matrix scores \mathbf{T} from the observation level. In the following, the observation (OWU) level, which is used in this dissertation, is described in detail. For explanation of the batch (TBWU) level, readers are referred to the original work [62, 83].

Observation (OWU) level and TLEC synchronization approach

In the observation (OWU) level, the three-way array \mathbf{X} is unfolded preserving the variable direction: each variable, measured at different sampling time points for the different batches, is arranged in one

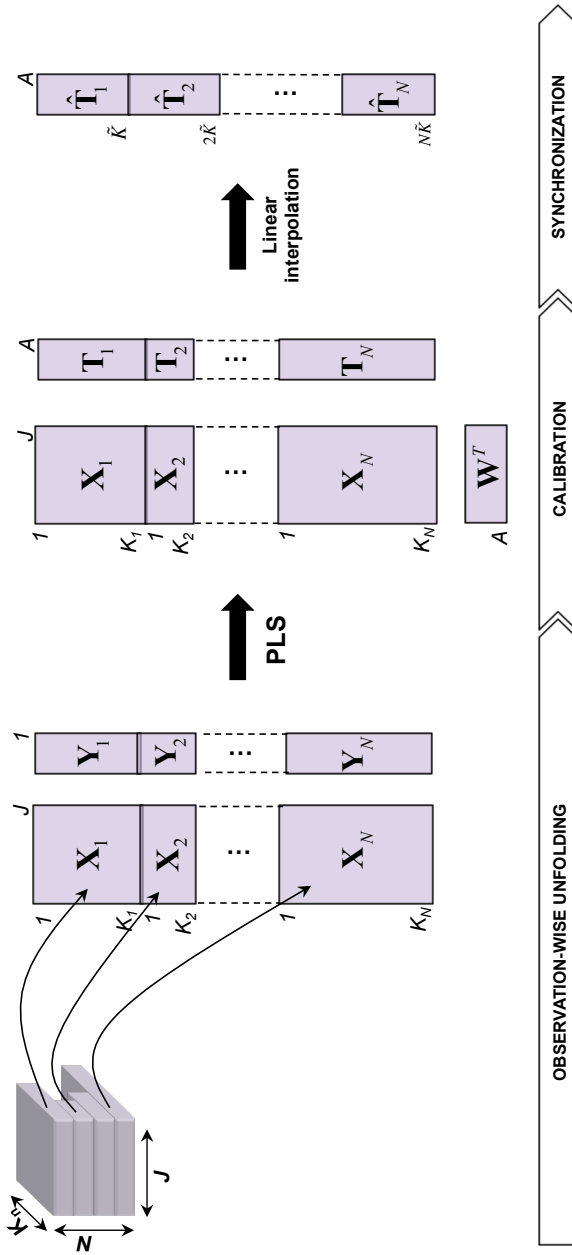


Figure 4.6. Procedure performed in the observation or OWU level.

column of a new two-way array \mathbf{X} ($K_n N \times J$), in a manner that each row corresponds to a single sampling time point at which the measurements are registered (see OWU unfolding step in Figure 4.6). Once the arrangement is performed, each column is autoscaled by subtracting its average value and dividing by its standard deviation (the so-called slab-wise preprocessing or variable centering and scaling). With this normalization, time periods with less variability will be downweighted and periods with more variability will be weighted more in the multivariate analysis. Afterward, a PLS model is built in order to relate all the process variables at every batch sampling time point to a dummy variable y , representing the local batch time (see calibration step in Figure 4.6). This variable is created as follows:

- i. A vector \mathbf{y}_n containing sorted integer values ranging from 0 to the length of i -th batch minus one is built for each single batch.
- ii. Each value of \mathbf{y}_n is transformed as:

$$\mathbf{y}_{n,k}^{new} = \frac{\mathbf{y}_{n,k}}{K_n - 1} \times \tilde{K} \quad (4.1)$$

where K_n is the length of the n -th run and \tilde{K} represents the median of the training batch lengths.

- iii. All the vectors for the different batches are arranged into a single column array, \mathbf{y} , containing the values of the variable y , for each batch at each sampling time point.
- iv. \mathbf{y} is autoscaled.

Note that if all batches have equal duration, the vectors \mathbf{y}_n and \mathbf{y}_n^{new} contain the same number of elements and the same values. In case the batches have unequal duration, all vectors \mathbf{y}_n^{new} have the same starting and ending values, but a different number of intermediate values depending on their batch length K_n . Once the PLS model is fitted, the resulting OWU scores \mathbf{T}_A , Hotelling's T^2 and DModX statistics for each batch are readjusted by linear interpolation (the so-called TLEC-based method) using the \mathbf{y}_n^{new} maturity index vector (see synchronization step in Figure 4.6). This readjustment permits the new OWU scores (denoted as $\hat{\mathbf{T}}_A$) to be used for process monitoring and subsequent multivariate analysis in the batch level.

For real-time process monitoring, the so-called OWU scores control charts (one per each latent variable) are designed. The synchronized OWU scores are firstly batch-wise arranged in such a way that those belonging to a specific batch form one row of a new matrix, $\mathbf{X}_{\hat{\mathbf{T}}_A}$

$(N \times \tilde{K}A)$. Afterward, by computing the average $\tilde{t}_{a,k}$ and standard deviation $s_{a,k}$ of each batch sampling time point for the synchronized scores of all latent variables (columns of $\mathbf{X}_{\hat{\mathbf{T}}^A}$), the Upper and Lower Control Limits (UCL and LCL, respectively) can be straightforwardly calculated:

$$UCL_{a,k}^T = \tilde{t}_{a,k} + z_{\alpha/2} \cdot s_{a,k} \quad (4.2)$$

$$LCL_{a,k}^T = \tilde{t}_{a,k} - z_{\alpha/2} \cdot s_{a,k} \quad (4.3)$$

where $z_{\alpha/2}$ is the $100 \cdot (1 - \alpha/2)\%$ standardized normal percentile.

Hotelling- T^2 -based control charts can also be built from the synchronized OWU scores, but they are rarely taken into account in the OWU-TBWU approach. In contrast, the residual matrix \mathbf{E} derived from the PLS model is used in order to estimate the DModX statistic at every sampling time point for each batch as follows:

$$\text{DModX}_k = \sqrt{\frac{\sum_{j=1}^J e_{j,k}^2}{(J - A)}} \quad (4.4)$$

where J is the number of process variables and A is the number of latent variables extracted. DModX-based control charts are then built and its corresponding Control Limit (CL) is calculated using the F distribution with $J - A$ and $(N - A - 1)(J - A)$ degrees of freedom for in-control observations [64]. Based on the relation between SPE and DModX [47], the control limits for the DModX-based control chart can be estimated by using the approximation proposed by Box [116] or Jackson and Mudholkar [115].

A drawback of the observation level in real-time monitoring of new batches is that the OWU scores and the multivariate statistics cannot be aligned until the completion of the batch. Hence, there is no guarantee that the control charts on the OWU stage show aligned results, and therefore, monitoring may be misleading.

At this point, it is worth commenting that the application of the synchronization procedure integrated in the OWU-TBWU approach (the TLEC-based method) may be completely inappropriate due to the underlying assumptions that are rarely fulfilled in batch processes: i) linear process pace, ii) all batches are completed and all the key process events defining the process evolution throughout the batch run are present in all batches, and iii) batches with equal duration are considered as synchronized. If batch data do not meet these assumptions, the process evolution in the trajectories of the

process variables is different batch-to-batch. Hence, the application of methods based on latent structures for process understanding and monitoring is not recommended.

4.2.3 Dynamic Time Warping

Dynamic Time Warping (DTW) is a technique originating from speech recognition to match similar events between signals. DTW has the ability to synchronize two trajectories by appropriately translating, stretching and shrinking sample points to match with reference points in order to minimize the distance between both trajectories [152].

Kassidas et al. [76] applied the ideas of DTW to batch trajectory synchronization. They adapted the breakthrough work published in [152] and elucidated the crucial role of constraints in the application of DTW to batch trajectory synchronization. This constraint formulation is extended in the RGTW and Multisynchro algorithms proposed in this thesis. In order to introduce the DTW theory, let \mathbf{r} ($K_r \times 1$) and \mathbf{s} ($K_s \times 1$) be the vectors containing the trajectory of one process variable from different batches, measured in K_r and K_s time intervals, respectively. The goal is to make these trajectories equal in length trying to match the key events. For that, a $K_s \times K_r$ grid is firstly built with the trajectory \mathbf{s} in the x-axis and \mathbf{r} in the y-axis (see Figure 4.7(b)). Let us define the warping function or path as

$$\mathbf{f}^T = \{w(1), w(k), \dots, w(K_w)\} \quad \max(K_r, K_s) \leq K_w \leq K_r + K_s \quad (4.5)$$

where each $w(k)$ is a position on the $K_s \times K_r$ grid indicating the elements of trajectories that are matched:

$$w(k) = [i(k), j(k)], k = 1, \dots, K_w \quad (4.6)$$

being k the new index for a common axis that associates the sample with the reference trajectory. Figure 4.7(b) shows an example how $i(k)$ (see Figure 4.7(a)) and $j(k)$ (see Figure 4.7(c)) are express as function of k . At each grid point, a local distance measure between the $i(k)$ -th sampling time point of the \mathbf{s} trajectory vector, $s_{i(k)}$, and the $j(k)$ -th point of the \mathbf{r} trajectory vector, $r_{j(k)}$, which reflects the dissimilarity between them, is assessed as:

$$d(i(k), j(k)) = (s_{i(k)} - r_{j(k)})^2 \quad (4.7)$$

Let us define the cumulative weighted local distance between two trajectories along the path \mathbf{f} as:

$$D(f) = D(i(K_w), j(K_w)) = \sum_{k=1}^{K_w} d(i(k), j(k)) \quad (4.8)$$

The aim of the DTW algorithm is to find an optimal path \mathbf{f}^* among all the possible solutions that optimally matches two trajectories (that connects the start and end grid point) in such a way that the cumulative distance D among them is minimized. This is an optimization problem that can be written as

$$D^*((i(K_w), j(K_w))) = \min_f [D((i(K_w), j(K_w)))] \quad (4.9)$$

A computational demanding way to find the optimal path that satisfies Equation 4.9 would be to calculate all the possible paths. Once cumulative distances belonging to all possible paths are calculated, the path with minimum cumulative distance would be the path chosen as optimal. Obviously, this approach would take too much time to obtain the optimal solution. An alternative more efficient approach is to solve this optimization problem using dynamic programming [152]. This technique is widely used in mathematical optimization and computer programming to solve complex problems by breaking it down in smaller problems. It takes far less time to reach the solution than the above approach. Before commenting in detail the way to find the optimal path by using dynamic programming, some constraints must be defined to reduce the number of possible paths and the large search space.

Warping Function Constraints

Several constraints are imposed in the inner DTW algorithm to obtain the optimal path: constraints on the start and end point of the path, monotonicity conditions to keep the temporal order when events occur, local continuity that defines the allowable slope of the path point to point and global constraints that allow us to reduce the search space.

1. Endpoint constraints

The end point constraint imposes the starting $w(1)$ and last point $w(K_w)$ to be $(1,1)$ and (K_r, K_s) , respectively. When the start and

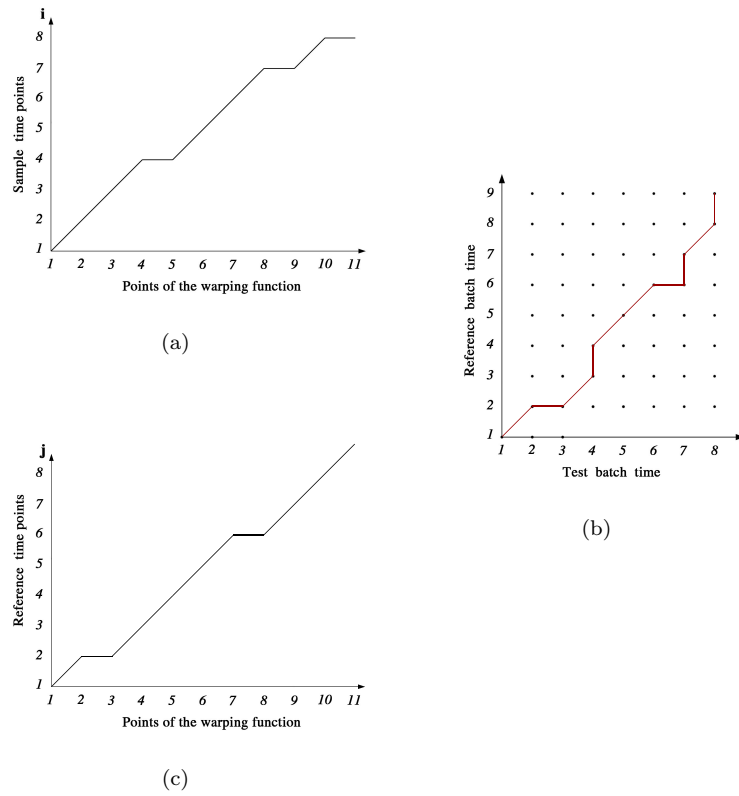


Figure 4.7. Parametric mapping of two different trajectories with different length via the common index k (b). Also, a graphical representation of the index row vector $i(k)$ (a) and $j(k)$ (c) belonging to the test and the reference trajectories, respectively, are shown.

end point are unknown, this constraint can be relaxed in such a way that an allowable region is defined where the best match on the first and last point on the warping function \mathbf{f} is found. This is really useful specially in online synchronization, where the endpoint at each sampling time point is unknown.

2. Monotonicity Conditions.

A paramount requirement in the synchronization is to compare events that occur in the natural temporal order, reducing this way the loss of

information caused by wrong synchronization. Hence, this constraint can be defined by forcing the path to be monotonous and having non-negative slopes:

$$i(k+1) \geq i(k) \quad (4.10)$$

$$j(k+1) \geq j(k) \quad (4.11)$$

3. Local Continuity Constraints.

Furthermore, it is desirable to avoid excessive compressions or expansions of the reference and test time scale. For this purpose, the local slope of the optimal path is constrained to a range, defining a type of allowable predecessors for each grid point.

Sakoe and Chiba [152] proposed a slope constraint based on the idea to constraint the possible relation among several consecutive points on the warping function. In case this warping function moves forward to a certain number of m steps through the same direction (x -axis or y -axis), then the following matched points are not allowed to follow further through the same direction before stepping at least n consecutive points in another direction. In order to evaluate the effective intensity of the slope constraint, an intensity measure $In = \frac{n}{m}$ was proposed in [152]. The larger the In is, the more stiffly the warping function will be constrained. If the intensity measure is equal to 0, no constraints are set. If the slope constraint is not sufficiently strict (small value), the discrimination among trajectories will be degraded. In contrast, if In is large, the DTW algorithm will not work effectively. Sakoe and Chiba [152] reported a study with different In and observed that the best result was for a constraint of 1 step both vertical and horizontal, i.e. a slope constraint In equal to one (see Figure 4.8(a)). Other local continuities used in the literature can be found in Figure 4.8.

4. Global Constraints.

One common constraint widely used in the DTW algorithm is the global constraint. Such constraint speeds up the DTW computation and also prevents wrong alignments by globally constraining the area where the optimal path can lie. The most common global constraint regions used are the Sakoe-Chiba band and the Itakura parallelogram, as depicted in Figure 4.9. Note that these global constraints are automatically defined by the local constraints selected.

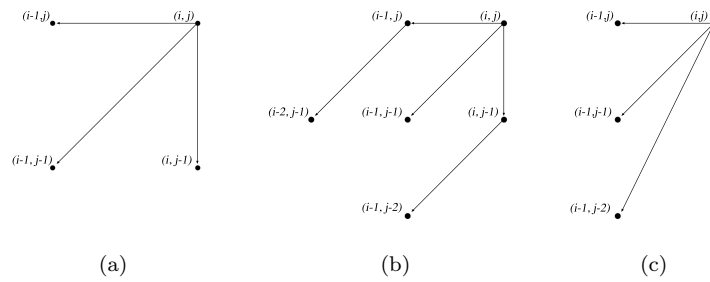


Figure 4.8. Most common local continuity constraints: (a) Sakoe-Chiba local constraint with no constraint on slope, (b) Sakoe-Chiba local constraint with allowable slope in the range $[\frac{1}{2}, 2]$, and (c) Itakura local constraint allowing slope in the range $[\frac{1}{2}, 2]$.

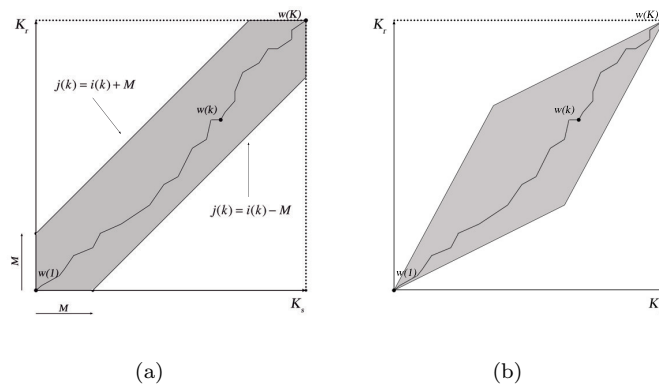


Figure 4.9. Global band constraints: (a) Sakoe-Chiba constraint with width M , and (b) Itakura constraint.

It is due to the imposition of local continuity constraints on the warping function that limit the region where the optimal path can be located.

Figure 4.9(a) illustrates the band global constraint proposed by Sakoe and Chiba [152], which is defined as

$$|i(k) - j(k)| \leq M \tag{4.12}$$

where the parameter M represents the allowable band where the optimal path can go through. This band runs along the main diagonal and constrains the range of warping points to the fixed width of M . In other words, the parameter M defines the length of the vertical vector $j(k)$ from the diagonal path to the upper boundary of the band, and likewise, the length of the horizontal vector $i(k)$ to the lower boundary. This global constraint is usually used together with the local continuity constraint shown in Figure 4.8(a), to match trajectories of process variables avoiding large deviations from the diagonal path.

Regarding the Itakura global constraint (see Figure 4.9(b)), the search space is defined by a parallelogram, i.e. two lines of slope $1/2$ and 2 , starting from the start $(1, 1)$ and the endpoint (K_r, K_s) , constraining the optimal path search. This is directly implied by the Itakura local constraint, which allows the algorithm to exclude some points on the grid to reach the k th point of the warping function \mathbf{f} .

Algorithm

An important aspect to take into consideration to reconstruct the optimal path is the version of the DTW algorithm to use. Namely, there are two versions, the symmetric and asymmetric DTW algorithm. In the symmetric version, time axis belonging to both the reference and sample trajectory will be transformed into a common axis, where the index k will define the matching point among trajectories. The optimal path will go through all the points in both trajectories (see Equation 4.6). Thus, the outcome of the synchronization will not be different if the reference and sample trajectory are swapped on the axis. Regarding the asymmetric version, it will map the time index belonging to the sample trajectory \mathbf{s} -located on the x -axis- onto the time index of the reference trajectory \mathbf{r} -located on the y -axis. In this case, the time index i of the signal \mathbf{s} and the optimal path \mathbf{f}^* will contain exactly the same sampling time points as the sample \mathbf{s} , i.e. K_s . Hence, the optimal path for the asymmetric version is defined as

$$\mathbf{f}^T = \{w(1), w(2), \dots, w(i), \dots, w(K_s)\} \quad (4.13)$$

where each $w(i)$ is a position on the $K_s \times K_r$ grid indicating the elements of trajectories that are matched:

$$w(i) = [i, j(i)] \quad (4.14)$$

This implies that the optimal path will go through all the points of the sample signal \mathbf{s} , but it may skip points of the signal \mathbf{r} .

There are also symmetric and asymmetric versions of synchronization. In the symmetric version, all the points containing the optimal warping function \mathbf{f}^* are used to synchronize the sample trajectory \mathbf{s} . When the optimal path constrains a vertical transition, i.e. from the $(i, j - 1)$ to the (i, j) point, the response of the signal \mathbf{s} is taken twice. In the case of horizontal transition, i.e. from the (i, j) to the $(i - 1, j)$ point, the response of signal \mathbf{r} is taken twice. Hence, the length of the synchronized trajectory will have the same length as the optimal path \mathbf{f}^* ($\max(K_r, K_s) \leq K_w \leq K_r + K_s$). In the case of the asymmetric version, depending on where the reference trajectory is located, either on the x -axis or y -axis, the vertical or horizontal transitions are treated in a different way. For the sake of simplicity, assume that the reference trajectory is located on the y -axis. So, when consecutive points of the sample trajectory \mathbf{s} are synchronized with the same point of the reference \mathbf{r} (i.e. presence of a vertical transition), the average of these points is estimated and synchronized with the corresponding point of the reference trajectory. In the synchronization of multivariate batch trajectories, Kassidas et al. [76] proposed to use the symmetric DTW followed by an asymmetric synchronization step in order that not only all the batch trajectories are synchronized with the reference trajectory, but also with each other.

Optimization Problem

As commented before, the aim of the synchronization of batch trajectories using the DTW algorithm is to find a path that matches a reference and a test trajectory optimally in a such way that the cumulative distance assessed through all the pairwise points $w(k)$ belonging to the optimal path \mathbf{f}^* is minimized (see Equation 4.9). Such problem actually becomes an optimization problem so that the computational cost associated with assessing all the possible paths to obtain the optimal solution is very high, varying exponentially with the batch length.

To reach this solution in a computational efficient way, dynamic programming is used to find the optimal solution by searching on the constrained grid under continuity, local and global constraints, allowing this way a reduction of computational cost. The procedure to obtain the optimal path can be divided into two phases: the forward and backward phases (see Figure 4.10). During the forward

phase, the cumulative distance matrix \mathbf{D} is assessed for each (i, j) grid point by solving the following dynamic programming recursive equation:

$$D(i, j) = d(i, j) + \min \left\{ \begin{array}{l} D(i-1, j) \\ D(i-1, j-1) \\ D(i, j-1) \end{array} \right\} \quad (4.15)$$

At this first part of the iterative procedure, $D(1, 1)$ is firstly initialized to the value $d(1, 1)$. Secondly, the values of the first column and row of matrix \mathbf{D} are calculated. Afterward, the remaining (i, j) grid points are column-wise estimated in function of the set of predecessors -i.e. $(i-1, j)$, $(i-1, j-1)$ and $(i, j-1)$ points- and based on Equation 4.15.

Once the cumulative distance matrix \mathbf{D} is calculated, the optimal path can be reconstructed (backward phase). The warping function can be found by backtracking from the (K_s, K_r) endpoint to the $(1, 1)$ start point, using the allowable predecessors. Firstly, the predecessor of the endpoint is obtained, subsequently, the predecessor of the previous point is derived, and so forth until the start point $(1, 1)$ is reached. In each step k , the coordinates (i, j) of each predecessor are kept in the warping function $w(k)$, thereby building the optimal path \mathbf{f}^{*T} .

End-of-batch DTW synchronization for Batch Monitoring

Let us assume that a set of batch profiles are collected under NOC. Let $\mathbf{B}_n(K_n \times J)$ be the matrix containing the trajectories of J variables measured at K_n time intervals for n -th batch, $n = 1, \dots, N$. DTW works with pairs of patterns, thus, synchronization must be conducted batch to batch and in a multivariate approach. The aim is to synchronize each \mathbf{B}_i with \mathbf{B}_{ref} . The main steps to carry out in this multivariate synchronization using the DTW algorithm [76] are:

- 1) **Reference batch selection.** Choose a reference batch \mathbf{B}_{ref} ($K_{ref} \times J$) whose length is the closest to the median duration of the collected NOC batches. The median is less sensitive to outliers, and thus it is preferred to other choices such as the average.
- 2) **Scaling.** Scale the batches in order to normalize the differences in the units used to record the trajectories of the J variables.

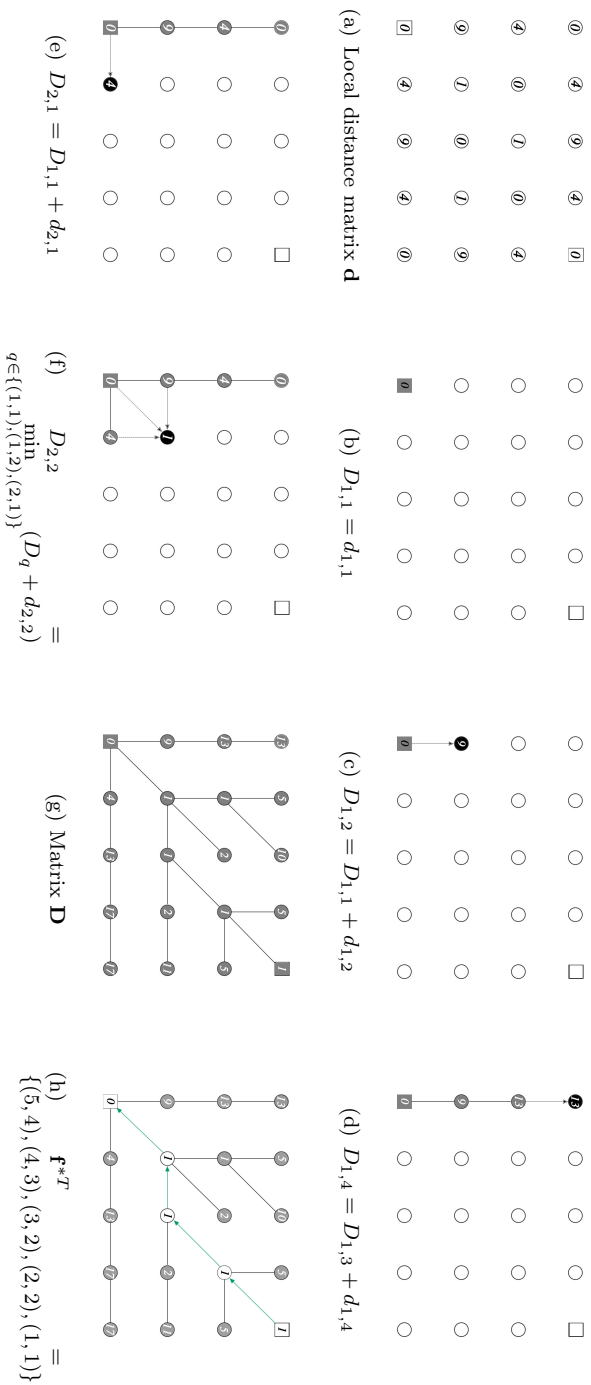


Figure 4.10. Trace of the execution of the symmetric DTW algorithm with Sakoe-Chiba local constraint and no global constraints. Two univariate trajectories, \mathbf{r} (where $\mathbf{r} = \{1, 3, 4, 3\}$) and \mathbf{s} (where $\mathbf{s} = \{1, 3, 4, 3, 1\}$), are taken as reference and test signals, respectively. The dissimilarity among them are expressed by the local distance matrix \mathbf{d} (see Figure 4.10(a)). Figures 4.10(b)-4.10(g) illustrate the forward phase where the cumulative distance matrix \mathbf{D} is obtained. Figure 4.10(h) represents the reconstruction of the optimal path (second phase). A back circle represents the stage that is being evaluated whereas the grey circles denote the states already evaluated. Dotted arrows show the allowable predecessors for a certain state. The points belonging to the warping function \mathbf{f}^{*T} are depicted by white circles in Figure 4.10(h).

Calculate for each variable trajectory its average range \bar{R}_j by averaging the range from each batch as follows:

$$\bar{R}_j = \frac{\sum_{n=1}^N \max(\mathbf{b}_{n,j}^T) - \min(\mathbf{b}_{n,j}^T)}{N} \quad (4.16)$$

where $\mathbf{b}_{n,j}^T (1 \times K_n)$ is the row vector of matrix \mathbf{B}_n .

Then, each variable in all batches is divided by its average range as follows:

$$\tilde{\mathbf{b}}_{n,j}^T = \frac{\mathbf{b}_{n,j}^T}{\bar{R}_j} \quad (4.17)$$

If batch data have the same measurement units across variables, as in spectroscopy data [91] or in data collected from artificial tongues and noses, the scaling step for DTW can be omitted.

3) **Applying the symmetric DTW algorithm with warping function constraints.**

- i) Build a $K_n \times K_{ref}$ grid for all the sampling time points of \mathbf{B}_n and \mathbf{B}_{ref} .
- ii) Estimate the weighted local distance between those points of the $i(k)$ row vector of the \mathbf{B}_n trajectory matrix, $\tilde{\mathbf{b}}_{n,i(k)}^T$, and those points of the $j(k)$ row vector of the \mathbf{B}_{ref} trajectory matrix, $\tilde{\mathbf{b}}_{ref,j(k)}^T$, that are allowed by the upper \mathbf{u} and lower \mathbf{l} boundaries. Such distance is estimated as follows:

$$d(i(k), j(k)) = \left(\tilde{\mathbf{b}}_{n,i(k)} - \tilde{\mathbf{b}}_{ref,j(k)} \right)^T \cdot \mathbf{W} \cdot \left(\tilde{\mathbf{b}}_{n,i(k)} - \tilde{\mathbf{b}}_{ref,j(k)} \right) \quad (4.18)$$

where \mathbf{W} is a diagonal non-negative matrix used to give more weight to certain variables.

- iii) Assess the cumulative distance matrix \mathbf{D} applying the forward DTW algorithm.
 - iv) Obtain the optimal path \mathbf{f}^* by backtracing from the (K_s, K_r) endpoint up to the $(1, 1)$ starting point, taking the optimal predecessor in each intermediate point.
- 4) **Asymmetric synchronization step.** The points of \mathbf{B}_n that are synchronized with the same point of \mathbf{B}_{ref} are replaced with the mean value of the original values of \mathbf{B}_n .
- 5) Repeat steps 3 and 4 for each batch.

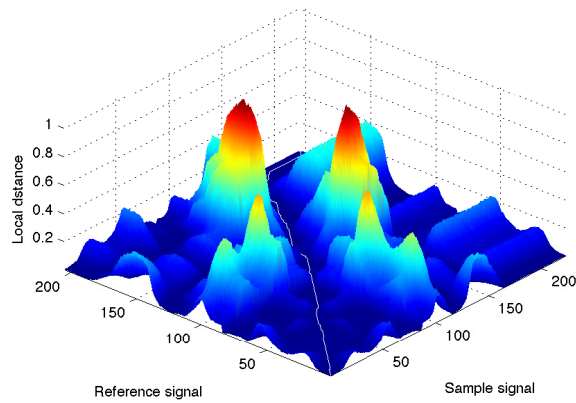


Figure 4.11. Optimal path estimated without global constraints and mapped on the surface plot of local distances for a process variable.

As previously mentioned, the weight matrix \mathbf{W} is used to give more importance to certain variables based on a particular criterion. When process knowledge is available, the weights are set to certain values. However, this fact is not usual in industry. As an alternative, the synchronization of batch trajectories should rely more on variables that are more consistent from batch to batch and less on noisy variables [166]. A procedure was proposed to automatically detect, and increase the weight of consistent variables and decrease the weight of the rest in an iterative manner until convergence [76]. This procedure is carried out as follows. First, the DTW algorithm is applied to all batches at each iteration. Afterward, the sum of the squared deviations from the average trajectory of each variable over all synchronized batches is calculated as an index of its consistency. The inverse of this index will be the weight given to a particular variable. The synchronization of the batches is repeated until the weights of the variables converge. One drawback of this weighting criterion is that consistent horizontal trajectories, with no warping information, might be given more weight to seek the optimal path. Another shortcoming is that process variables with a more dynamic behavior shown as time-varying smooth curves, which are less consistent among batches but important for synchronization, might be given less weight. This fact might affect the quality of synchronization even though a phase-to-phase synchronization is performed, leading to inaccurate alignments of the events that drive the batch process.

In the field of analytical chemistry, elution time variation is a common phenomenon in chromatographic data that must be tackled prior to modeling. Spectra are curved-like variables that can be divided into three constituting parts: the analytical relevant signal in form of peaks of different size and shape, the background or baseline, and the noise. Due to subtle, random and often unavoidable variations in instruments, these signals are usually time-shifted as well. To align the smooth peaks in time, a variant of the DTW algorithm was proposed [160], which focuses on aligning the spectra features giving more weight to those variables (spectra) containing more warping information. This method consists of estimating which variables are more important from a warping perspective in an automatic manner. For this purpose, a weight matrix is initialized to the identity matrix. The first batch is taken and synchronized with the reference one. Local distance matrix for each process variable between the first batch and the reference batch is obtained. Once all local distance matrices are available, a surface plot is depicted for each matrix and the optimal path for each variable is mapped on the surface plot. Important warping information is expected to fall in valleys (see Figure 4.11). In other words, the location of the optimal path must be near a valley and close to the diagonal, characterized by small distances. To quantify the importance a variable should have in the synchronization, the ratio of the mean local distance of the elements not lying on the optimal path to the mean distance of the coordinates of the optimal path was proposed. This weight provides a measure to quantify whether a variable is important or not. If this index is high, it means that the path follows the valley of the surface plot, and hence, the mean local distance of the points lying on the optimal path will be smaller compared to those outside the path. This whole procedure is repeated for each batch to calculate the weights of the J variables. By averaging the weights for each one of the J variables across batches, the diagonal elements of the weight matrix \mathbf{W} are obtained. The complete procedure to calculate \mathbf{W} is repeated until convergence.

Ideally, the variables that are more consistent from batch to batch and less noisy, and that also provide more warping information, should be more weighted to accurately align the main process events. In order to perform a synchronization that takes into account these characteristics, a new weight definition is suggested:

$$w_{j,j} = \sqrt{w_{j,j|Ram} \times w_{j,j|Kass}} \quad (4.19)$$

where $w_{j,j|Ram}$ and $w_{j,j|Kass}$ are the geometric mean of the weights estimated following the Ramaker et al.'s approach and Kassidas et al.'s approach for each variable, respectively. The geometric average is used because it gives a good measure of the central tendency of the set weights that are calculated based on different criteria and might have different range.

Note that in all the above approaches, \mathbf{W} is normalized in such a way that the sum of the diagonal elements is equal to the number of process variables J .

Real-time DTW synchronization for Batch Monitoring

Kassidas et al. [76] proposed an approach to carry out the real-time synchronization for online batch process monitoring. Let \mathbf{B}_{ref} be the trajectory matrix of the reference batch and \mathbf{W} the weight matrix used to carry out the offline synchronization of the NOC batch data. When new measurements are available at the t -th sampling time point, each variable is scaled to the average range estimated from the trajectories of NOC unsynchronized batches. Let \mathbf{B}_{new} ($t \times J$) be the trajectories of J variables measured from the start of the batch until the t current time for an ongoing batch.

The use of the offline DTW algorithm for online synchronization is inadequate because the endpoint is unknown at each sampling time point. Therefore, the endpoint constraint must be relaxed in order to obtain the optimal path that provides the best matching between \mathbf{B}_{ref} and the uncompleted batch \mathbf{B}_{new} .

Following the above idea, a set of cumulative weighted distances $D_{t,j}$ at the t -th sampling time point is calculated, where $j = l_t, \dots, u_t$ are the allowed points of \mathbf{B}_{ref} imposed by the lower l_t and upper u_t bound at the t -th sampling time point. Let us define e_t^* as the sampling time point of the batch reference \mathbf{B}_{ref} where the minimum cumulative weighted distance $D_{t,j}$ occurs at the t -th sampling time point. Hence, the optimal path with the minimal total distance up to the e_t^* point can be determined. After applying the previous procedure, an asymmetric synchronization is required to have a length of e_t^* sampling time points.

Once the ongoing batch is synchronized, it can be monitored online using different approaches [62].

4.3 Warping Information

The warping profiles obtained from synchronization are composed of a set of different transitions at each sampling time point, i.e. vertical, horizontal and diagonal transitions. Based on the number of different transitions the warping path contains, conclusions regarding the performance of the different process stages can be drawn. Let us assume that the test and reference batches are located on the x -axis and y -axis, respectively. An excessive number of vertical transitions in the warping profile means that the test batch needed less time than the reference batch for completion. In contrast, an excessive number of horizontal transitions is related to a slow process pace of the test batch in comparison to the reference batch. To correct these differences in the process pace, the DTW/RGTW algorithm expands and compresses the pieces of trajectories in such a way that the key process events are synchronized across batches. Note that the warping profiles contain valuable information about the duration of the process substages, which may be associated with abnormalities occurring in the process and/or the quality of the final product. Hence, the use of the warping information for process monitoring is highly recommended [1, 2, 66].

4.4 Comparative study of the synchronization techniques

The synchronization of the batch trajectories is the first step in the modeling cycle, jointly with the equalization of the process variables. Incorrect and/or inaccurate synchronization may produce the addition of artifacts, such as spikes or dumps, artificial shifts, and fictitious correlations among process variables. This inaccuracy may seriously affect the outcomes of the subsequent steps in the modeling, and therefore, the understanding of the process, prediction of process variables or properties, and the detection of abnormal situations.

This section is aimed to study the differences between the most used synchronization techniques in the field of process chemometrics, focusing the attention on the change of the correlation structure. Different metrics can be defined for comparison purpose: the total, dynamic partial and instantaneous-dynamic partial maps. The former calculates the variance-covariance matrix of the synchronized, preprocessed and batch-wise unfolded two-way array that contains the batch trajectories. This matrix gives practitioners a picture of the

dynamic relationships in the batch data, not only the instantaneous variance and cross-covariances of the variables at every sampling time point but also the auto-covariances and lagged cross-covariances. Revealing the time-varying dynamics helps practitioners to better understand the relationships among process variables over time. The dynamic partial variance-covariance map is the result of the computation of the partial covariance in time for all the possible LMVs (see Appendix B for details in the calculation). It is useful to observe dynamic relationships without taking into account the instantaneous relationships among process variables. This information should be used when the objective of the modeling is to predict the current value of a variable from past measurements of the process (e.g. to perform one-step ahead predictions). Finally, the instantaneous-dynamic partial variance-covariance map contains the partial covariance taking the instantaneous relationships of the process variables into account (see Appendix B for details in the calculation). The resulting map is useful when the goal is to obtain parsimonious models for process monitoring that properly capture the complex dynamics by adding the optimum number of LMVs.

Simulated data of the fermentation process of the *Saccharomyces cerevisiae* cultivation are used for the comparative study, in particular the three way-array $\tilde{\mathbf{X}}_1$ containing 30 NOC batches (see Chapter 2). Four synchronization techniques are selected to synchronize the batch data: IV, TLEC on the whole trajectories (TLEC) and between defined landmarks (TLEC-events), and DTW. In particular, two synchronizations based on two different indicator variables are performed. The first synchronization uses the fermentation rate as the indicator variable, and non-uniform increments are defined. Regarding the second IV-based synchronization, the biomass concentration is used as the indicator variable with constants increments calculated between the start of the batch and the end value in common in all batches (see Section 4.2.1 for further details). TLEC-based synchronization is carried out by linearly interpolating 209 data points. In regard to TLEC-events, ten key process landmarks are defined based on the evolution of the process variables, and linear interpolation is performed in each defined stage, leading to a total of 209 interpolated data points. For DTW-based synchronization, batch #12 containing 209 sampling time points is selected as the reference batch, and the rest of conditions and constraints are set to [1, 76]. In addition, a second synchronization is applied to the synchronized batch trajectories by IV and TLEC using DTW (IV-DTW and TLEC-DTW). This re-synchronization is executed to check if the application of non-linear

synchronization techniques that take into account the presence of key process events (the so-called SCT-based methods) enhances the synchronization results.

The comparison of the synchronization methods is carried out by taking as a reference the optimum synchronization for this data set. The optimum synchronization of the batch trajectories consists of shrinking, compressing and non-linearly translating segments of data, similarly to the DTW synchronization, in accordance with the evolution of the simulation time. Recall that the latter is the original time of processing warped in the simulation to make all batches different in length. The total variance-covariance matrix \mathbf{S}_j of the each data set synchronized by the selected methods is then estimated for the ten process variables. The dissimilarity between each of these variance-covariance matrices \mathbf{S}_j with the one calculated from the optimum synchronization \mathbf{S}_j^* is assessed by calculating the *Correlation Matrix Distance* (CMD) index [167] as follows:

$$d(\mathbf{S}_j, \mathbf{S}_j^*) = 1 - \frac{\text{Tr}(\mathbf{S}_j \cdot \mathbf{S}_j^*)}{\|\mathbf{S}_j\|_F \cdot \|\mathbf{S}_j^*\|_F} \quad (4.20)$$

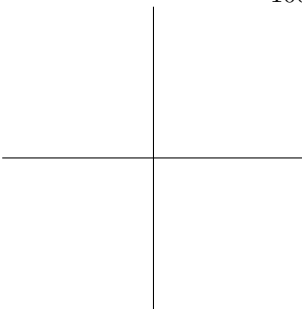
where $\text{Tr}(\cdot)$ denotes the trace of a $(K \times K)$ matrix, $\|\cdot\|_F$ is the Frobenius norm of a $(K \times K)$ matrix and K the number of sampling time points. This dissimilarity index varies from zero to one. It becomes zero if the variance-covariance matrices are equal to a scaling factor and one if they differ to a maximum extent. Hence, the CMD index is a good indicator of how well a specific synchronization technique is able to align the features of the trajectories, minimizing the addition of artifacts, and therefore, reducing the changes in the original correlation structure.

4.4.1 Effect of synchronization on the correlation structure

The total variance-covariance matrices of the batch data synchronized by IV, using the fermentation rate as the indicator variable, and by IV-DTW corresponding to the process variables glucose concentration, acetate concentration and specific oxygen uptake rate are shown in Figure 4.12. At first glance, there exist differences in the relationships over time after the two synchronization approaches were applied, which are more notable in the last two process variables (see Figures 4.12(b) and 4.12(e), and Figures 4.12(c) and 4.12(f) for comparison). In the case of the glucose concentration, a positive correlation is

observed (see red square located on the main diagonal at the first stage of the process -till the 42th sampling time point approximately- in Figures 4.12(a) and 4.12(d)). The variance-covariance matrix of the glucose concentration synchronized by IV shows a change on the process dynamics from the 42th to the 54th sampling time point. In particular, positive lagged cross-covariances of low order are observed (see the fuzzy red diagonals outside the main diagonal from the 42th to the 54th sampling time point in Figure 4.12(a)). Based on these observations, an apparent change of the phase is occurring at this interval. However, the results for IV-DTW reflect a gradual diminution of the correlation of the first phase, which is more an indication of a transition between phases rather than a complete change of phase (see fuzzy red area covering the red square in Figure 4.12(d)). These differences might be caused by a not completely accurate transformation of the time scale into the IV scale using interpolation. To verify this claim, the synchronized trajectories of the three process variables with their respective average trajectories are shown in Figure 4.13. As can be appreciated from Figure 4.13(a) after IV synchronization, the transition between the excess in glucose (maximum values) and the depletion of this substrate (minimum values) -i.e from the 42th to the 54th sampling time point- shows some variability around the average trajectory. If these trajectories are subsequently aligned by using non-linear synchronization steps to overlap the landmarks (IV-DTW synchronization), the variability is substantially reduced (see Figure 4.13(d)). As seen in this example, the increase of the variability due to synchronization might affect the correlation structure of the data. Consequently, it might jeopardize the segregation of the phases and the interpretation of the multivariate models for process understanding, although it might not necessarily impact the detection of abnormalities.

In the first variable under study we have seen slight changes in the process dynamics. However, one of the major effects of misalignment is the creation of non-existent relationships over time. Looking through the variance-covariance matrices estimated from the trajectories synchronized by IV and IV-DTW (see Figures 4.12(b) and 4.12(c), and Figures 4.12(e) and 4.12(f), respectively), some differences are found. In the acetate concentration for IV-based synchronization, there seem to be two main phases, the first phase ranging from the 32th to the 105th sampling time point approximately and the second phase from the 116th to 175th sampling time point approximately (see fuzzy red square at the top-left side and the square at the bottom-right side with positive correlation of diverse magnitude in Figure 4.12(b)).



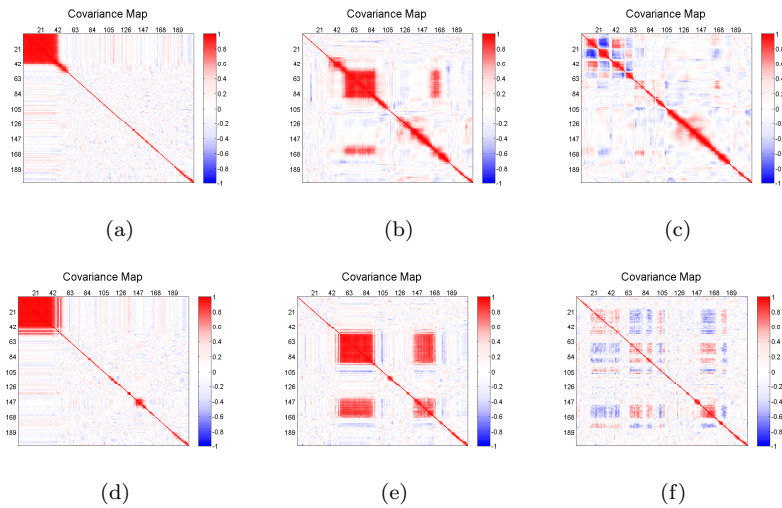


Figure 4.12. Total variance-covariance matrices calculated for the glucose concentration (a,d), acetate concentration (b,e), and specific oxygen uptake rate (c,f). These process variables were previously preprocessed with Trajectory C&S and batch-wise unfolded after synchronizing with IV using the fermentation rate as the indicator variable (a,b,c) and a second synchronization using DTW (IV-DTW) (d,e,f).

Between these intervals there seems to be a transition due to a mild negative correlation. However, the fuzziness of these intervals on the variance-covariance map jointly with the changing correlations over the batch evolution lead us to consider the existence of diverse phases. In the first interval, strong positive instantaneous variances and cross-covariances at every sampling time point are observed (see dark red color in the main diagonal) as well as strong positive auto-covariances and lagged cross-covariances of diverse order (see red squares located on the main diagonal). In particular, three main phases of different duration seem to exist (see red squares at sub-intervals [32, 54], [55, 90], [91, 105] in Figure 4.12(b)). In the second interval, there are strong positive instantaneous variances and cross-covariances at the sub-intervals [116, 130] and [131, 151], with dynamics of low order (see red area at the bottom-right side outside the main diagonal in Figure 4.12(b)). These correlations can be explained and confirmed by looking at the synchronized trajectories depicted in Figure 4.13(b). As can be observed in this figure, the two main intervals coincide with the process stages at which acetate is excreted (see intervals

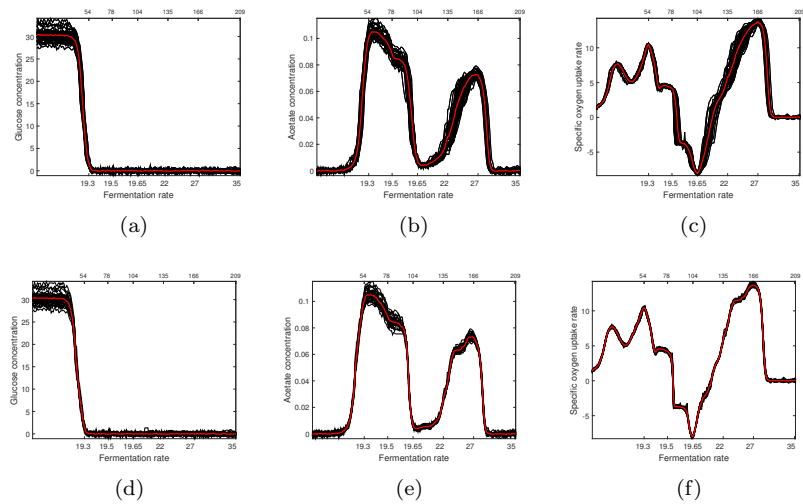


Figure 4.13. Batch trajectories of the glucose concentration (a,d), acetate concentration (b,e), and specific oxygen uptake rate (c,f) synchronized by IV (a,b,c) and IV-DTW (d,e,f), expressed as a function of the IV evolution (lower x-axis) and the number of interpolated samples (upper x-axis). Red lines denote the average trajectory for each process variable.

[32, 105] and [116, 175] in Figure 4.13(b)). However, the change of variability within these intervals produced by synchronization is the main reason why the dynamics vary over time. Even though IV aligns the key process events across batches, the inaccuracy of interpolation does not permit to clearly reveal the actual correlation structure. Improving the results of the IV-based synchronization with a DTW-based synchronization, the two phases previously pinpointed are better reflected in the variance-covariance matrix (see large red squares on the main diagonal in Figure 4.12(e)). Note that a negative correlation is unmasked at the intervals [100,104] and [175, 180], which seems to be the transition among phases, as appreciated from Figure 4.13(e). Thanks to a notable reduction of the variability produced by the IV synchronization, the actual relationships of the process variables are captured.

Another type of change in correlation is found in the specific oxygen uptake rate. At the start of the batch, IV aligns properly the events driving the process although it increases the variability around the average trajectory (see Figure 4.13(c)). As a consequence, the resulting variance-covariance matrix of the IV-synchronized data

shows fictitious correlations over the evolution of the batch. First, alternating positive and negative correlations from the start of the batch until the 63th sampling time point are observed (see top-left area in Figure 4.12(c)). These time-varying correlations are mainly due to alternating intervals at which the batch trajectories evolve above and below the average trajectory (see first 63 sampling time points in Figure 4.13(c)). In addition, two main phases seem to be originated, the first phase ranging from the 64th to the 105th sampling time point, and the second phase from the 106th to 180th sampling time points (see positive correlation depicted as red squares at the aforementioned intervals lying on the main diagonal in Figure 4.12(c)). These intervals are associated with the process stages in which the variability around the average trajectory is higher (see Figure 4.13(c)). After a re-synchronization, this variability is notably reduced, thereby revealing the actual correlations over the batch run for the specific oxygen uptake rate (see Figures 4.12(f) and 4.13(f)).

In Section 4.2.1, the effects of an inappropriate selection of the indicator variable and of the IV levels at which data are interpolated were shown on the resulting trajectories. To study the impact on the correlation structure, the total variance-covariance matrices of the batch data synchronized by IV with the biomass concentration as the indicator variable, and by IV-DTW are provided in Figure 4.14. Comparing these total variance-covariance matrices with those estimated from the data synchronized with the fermentation rate as the indicator variable, there exist significant differences. As expected, when the shape of the batch trajectories is modified, the actual correlation structure is completely perturbed. The relationships found in the synchronized data are not anymore representative of the phenomena occurring in the process. However, it is worth studying the differences found between the synchronizations based on IV and IV-DTW for illustrative purposes.

In the case of the glucose concentration, a positive correlation over time is observed (see red rectangles located on the main diagonal at the first stage of the process -till the 110th sampling time point approximately- in Figures 4.14(a) and 4.14(d)). Comparing the two resulting matrices from the two different synchronizations, we can observe that the duration of these correlations vary. In particular, the duration is longer in those correlations calculated from data synchronized by IV than by IV-DTW. This difference is basically due to inaccuracies in the synchronization caused by the interpolation conducted at equal-distanced intervals, which not only changes the variable trajectories but also increases variability around the average

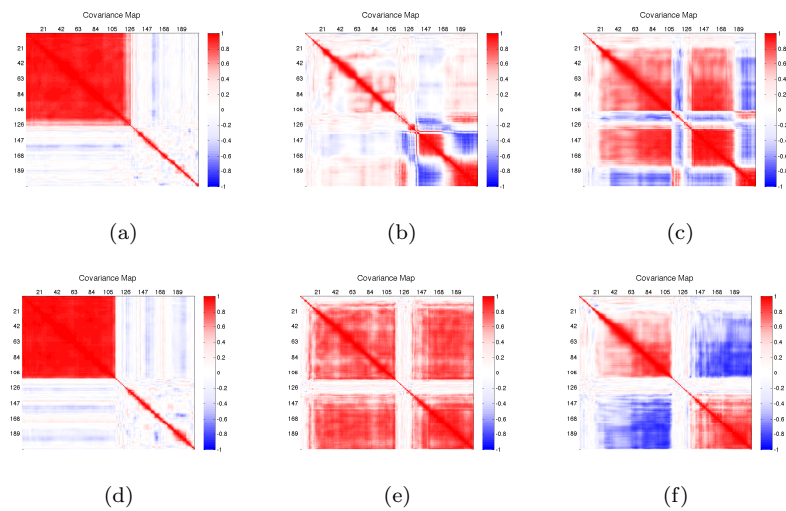


Figure 4.14. Total variance-covariance matrices calculated for the glucose concentration (a,d), acetate concentration (b,e), and specific oxygen uptake rate (c,f). These process variables were previously preprocessed with Trajectory C&S and batch-wise unfolded after synchronizing with IV using the biomass concentration as the indicator variable (a,b,c) and after a second synchronization using DTW (IV-DTW) (d,e,f).

trajectory. To support this claim, the synchronized trajectories of the three process variables with their respective average trajectories are shown in Figure 4.15. As can be appreciated from Figure 4.15(a) after IV synchronization, the sampling time point at which the glucose concentration reaches its minimum value (end of the phase) considerably varies from batch to batch (from the 110th to the 130th sampling time point). If these trajectories are subsequently synchronized taking into account the overlap of the landmarks (IV-DTW synchronization), the variability is reduced (see Figure 4.15(d)).

In the acetate concentration, apparently there are two phases, the first phase ranging from the start of the batch until the 125th sampling time point approximately and the second phase from the latter sampling time point onward (see fuzzy red square at the top-left side and the square at the bottom-right side formed by red and blue sub-squares in Figure 4.14(b)). In the first phase, strong positive instantaneous variances and cross-covariances at every sampling time point are observed (see dark red color in the main diagonal) as well as strong positive auto-covariances and lagged cross-covariances of

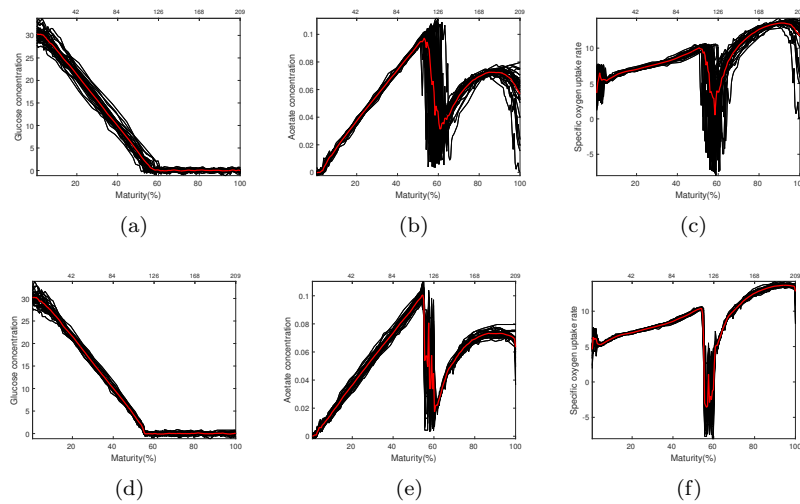


Figure 4.15. Batch trajectories of the glucose concentration (a,d), acetate concentration (b,e), and specific oxygen uptake rate (c,f) synchronized by IV (a,b,c) and IV-DTW (d,e,f), expressed as a function of the process maturity (lower x-axis) and the number of interpolated samples (upper x-axis). Red lines denote the average trajectory for each process variable.

low order (see red rectangles located outside the main diagonal). The latter become milder over time as appreciated from the red fuzzy rectangles outside the main diagonal. In the second phase, there is a strong correlation in the intervals $[140,165]$ and $[180, 209]$, which are negatively correlated to each other (see blue rectangles at the bottom-right side outside the main diagonal in Figure 4.14(b)). Between the phases there seems to exist a transition because of the lack of correlation in the interval $[110, 130]$. These correlations can be explained and confirmed by looking at the synchronized trajectories depicted in Figure 4.15(b). As can be observed in this figure, the trajectories evolve over time following the average trajectory till the 100th sampling time point. At this point, the key process events are not overlapped across batches since all the trajectories are shifted. This misalignment is also observed in the next landmark located around the 110th sampling time point. For thirty sampling time points, there is apparently no clear relationship over time caused by an inaccurate synchronization, thereby producing a transition among phases. In the second stage, from the 140th to the 165th sampling time point most of the batch trajectories are beyond the

average trajectory, whereas from the 180th to the 209th sampling time point it seems that most of the trajectories remain below the average. It explains the negative correlation found among intervals at the second phase in the variance-covariance matrix in Figure 4.14(b). Improving the results of the IV-based synchronization with a second DTW-based synchronization, the two phases previously pinpointed are better reflected in the variance-covariance matrix (see large red squares on the main diagonal in Figure 4.14(e)). However, the correlation of the variable is positive at the second stage in contrast to the IV-based synchronization. Thanks to a notable enhancement of the synchronization, the actual relationships come up.

Let us study the implications of using TLEC-based methods to synchronize batch trajectories. Figure 4.16 provides the variance-covariance matrices of the three process variables trajectories synchronized by TLEC, TLEC-DTW and TLEC-events. Comparing the matrices, we find significant differences between those synchronized by the former (see Figures 4.16(a)-4.16(c)) and the rest (see Figures 4.16(d)-4.16(i)), where in the acetate concentration and specific oxygen uptake rate the differences are to a greater extent. As can be appreciated from Figures 4.16(b) and 4.16(c), there exist positive instantaneous variances and cross-covariances at every sampling time point (see main diagonal) but alternating positive and negative correlations over time (see alternating blue and red areas along the diagonal outside the main diagonal). This is a clear symptom of, first, a lack of synchronization of the landmarks, and second, an excessive number of non-linear shifts in the trajectories that mainly produce this type of relationships among variables and lagged measurements. For confirmation, the respective trajectories synchronized by TLEC are plotted in Figure 4.17. Apart from showing a set of asynchronous trajectories, the non-linear shift produced by the synchronization is greater in the acetate concentration (see Figure 4.17(b)) than in the specific oxygen uptake rate (see Figure 4.17(c)). When a second synchronization is performed on the trajectories already synchronized by TLEC the quality of the synchronization improves (see Figure 4.16(e) and 4.16(f) for TLEC-events, and Figures 4.16(h) and 4.16(i) for TLEC-DTW). The uncommon correlation over time is completely diluted, revealing the actual relationships (see Figures 4.17(e) and 4.17(f) for TLEC-events, and Figures 4.17(h) and 4.17(i) for TLEC-DTW).

The last synchronization method applied to the data set under study is the DTW. The resulting variance-covariance matrices for the three selected variables are plotted in Figure 4.18. As can be appreciated

4.4. Comparative study of the synchronization techniques

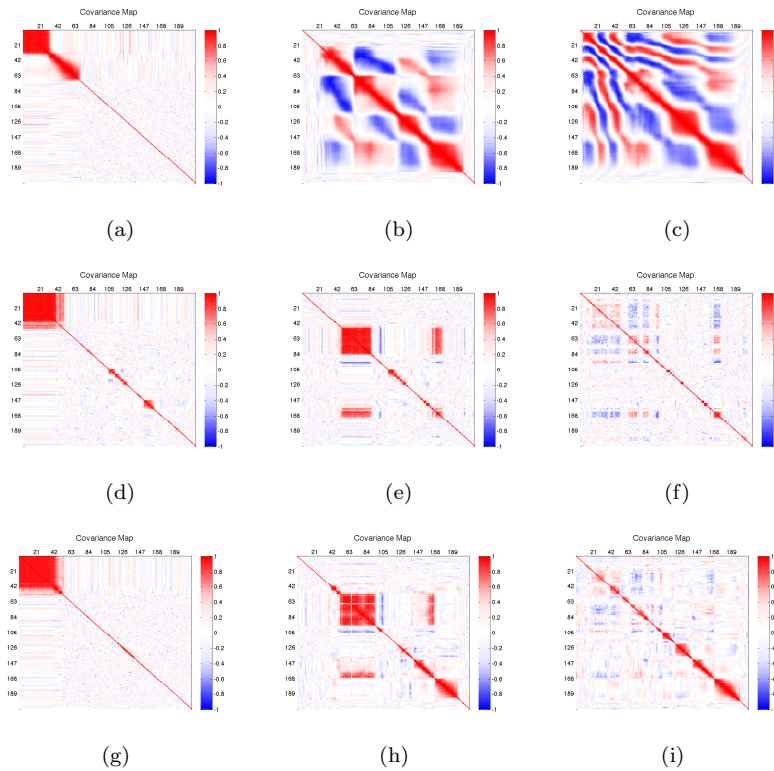


Figure 4.16. Total variance-covariance matrices calculated for the glucose concentration (a,d,g), acetate concentration (b,e,h), and specific oxygen uptake rate (c,f,i). These process variables were previously preprocessed with Trajectory C&S and batch-wise unfolded after synchronizing with TLEC (a,b,c), TLEC based on aligning predefined events (TLEC-events) (d,e,f), and TLEC with a subsequent second synchronization using DTW (TLEC-DTW) (g,h,i).

by comparing Figures 4.12 and 4.16 with Figure 4.18, the results observed for IV-DTW, TLEC-DTW and TLEC-event are similar to those obtained for DTW to a certain extent. However, IV-DTW seems to approximate better the variance-covariance matrix of the data synchronized by DTW. The reason of this enhancement is that the re-synchronization by SCT-based methods of non-optimal synchronized trajectories reduces the variability produced by IV synchronization, and ensures the overlap of key process events in the TLEC synchronization. Hence, the actual relationship of each variable over the evolution of the batch is considerably better captured.

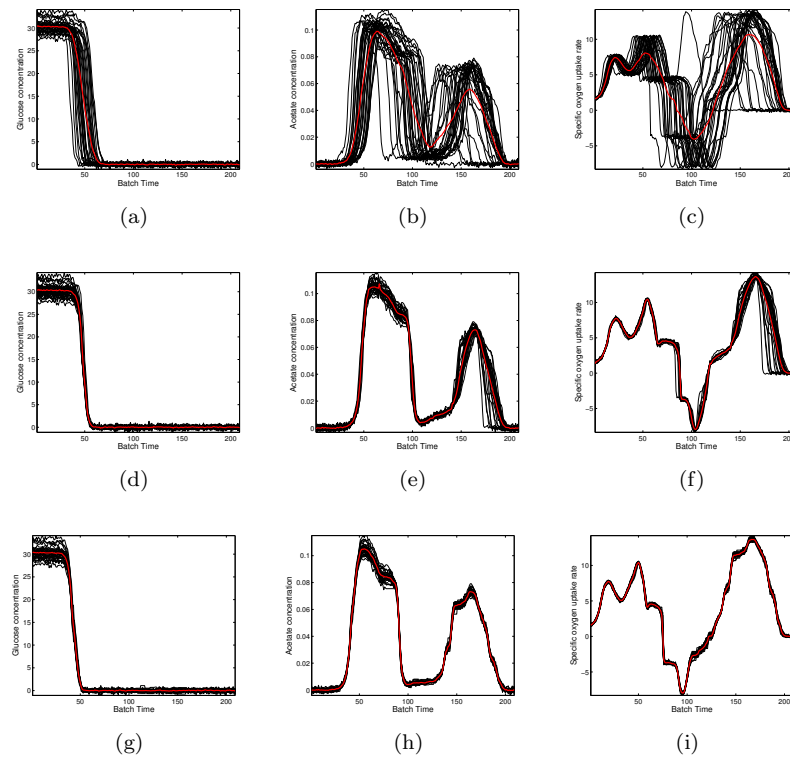


Figure 4.17. Synchronized batches trajectories of the glucose concentration (a,d,g), acetate concentration (b,e,h), and specific oxygen uptake rate (c,f,i) by TLEC (a,b,c), TLEC-events (d,e,f) and TLEC-DTW (g,h,i). Red lines denotes the average trajectory for each process variable.

With the aim of studying whether there are statistically significant differences between the synchronization approaches under study in terms of changes in the correlation structure, an *ANalysis Of Variance* (ANOVA) is carried out. In this case, the variance-covariance matrices of all process variables are calculated for: IV and IV-DTW using as indicator variable the biomass concentration (IV1 and IV1-DTW), and the fermentation rate (IV2 and IV2-DTW), DTW, TLEC, TLEC-events, and TLEC-DTW. The dissimilarity index between the variance-covariance matrices of each process variable for each synchronization method and the variance-covariance matrix obtained from data derived from the optimum synchronization explained in Section 4.4 is used as a response variable in ANOVA. The study

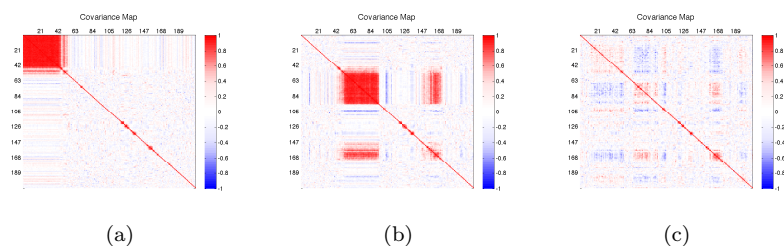


Figure 4.18. Total variance-covariance matrices calculated after synchronization using the DTW algorithm, Trajectory C&S preprocessing and batch-wise unfolding three process variables: (a) glucose concentration, (b) acetate concentration, and (c) specific oxygen uptake rate.

shows statistical differences in the changes of the correlation structure based on the synchronization method employed (p -value < 0.05). The *Least Significant Difference* (LSD) intervals are shown in Figure 4.19. DTW synchronization clearly yielded the best results since the associated dissimilarity index is close to 0 and there are statistically significant differences with the rest of methods. In contrast, no statistical differences are found between IV and IV-DTW, and between TLEC-DTW and TLEC-events (although IV-DTW and TLEC-DTW seem to provide slightly better results than IV and TLEC, respectively). However, these methods clearly outperform the results from the classical TLEC method in terms of changes in the correlation structure. Hence, the better the synchronization of the landmarks, the less change in the correlation structure.

4.5 Conclusions

In this chapter, a review of the synchronization techniques used in process chemometrics is carried out. The basis of the most used techniques (IV, TLEC and DTW) are thoroughly described.

The major objective of the synchronization in the modeling cycle is to ensure that the main features of the variable trajectories are aligned in a multivariate way to make statistical analyses feasible. Due to the nature of the methods and the batch data, the incorrect use of these techniques may produce the modification of the shape of the trajectories, consequently affecting the instantaneous and time-lagged relationships of the variables. To bring more light in this direction, a comparative study of the synchronization techniques in terms of

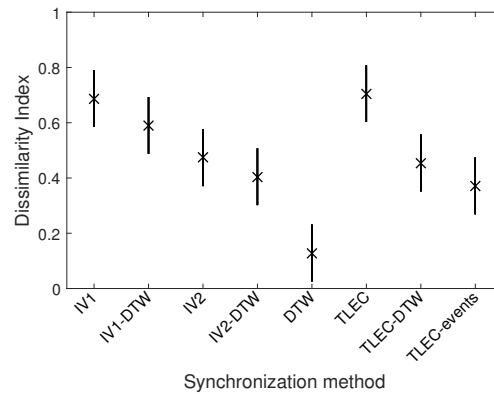
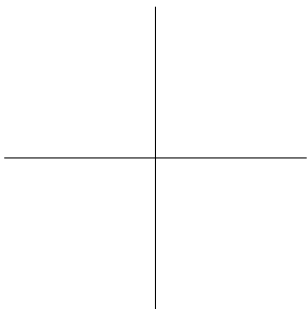


Figure 4.19. LSD intervals for the dissimilarity index among the variance-covariance matrices estimated for each process variable using the optimum synchronization and the synchronization based on IV, IV-DTW, DTW, TLEC, TLEC-DTW and TLEC-events.

changes in the correlation structure is performed by using simulated data of the *Saccharomyces cerevisiae* cultivation process. The analysis yields that those methods focused on linearly or non-linearly synchronize the landmarks of the variable trajectories clearly outperform those that are only aimed at making the trajectories equal in length. These simulated studies indicate that even though the key process events are not fully aligned by IV or TLEC, a second synchronization using SCT-based methods, which are devoted to reduce variability caused by synchronization and overlap the main features, notably enhances the quality of the synchronization. However, this improvement is not sufficient to reach the performance of SCT-based methods applied in a first synchronization. This is namely due to the perturbation of the trajectories caused by TLEC method, and by the increase of variability produced by IV. IV is a special case of TLEC-based method, where the difference lies in the dimension where the synchronization is done; the variable domain in the former and the time domain in the latter. When equal spaced intervals are defined in the indicator variable, the batch process is assumed to evolve linearly over time, as in TLEC-based synchronization, which is an assumption rarely met in this type of processes. In case of non-linear process pace, the definition of non-uniform increments or the selection of different IV per phase is needed to ensure the alignment of the events driving the process. In the simulated study, we have seen that IV clearly

outperforms TLEC when the process evolution is not linear, provided that non-equal spaced intervals are defined by taking into account the process stages. When synchronization is not focused on aligning the process events, the resulting synchronized batch trajectories may have different correlation over time in comparison to the original ones, which may thereafter affect the outcomes of the preprocessing and calibration steps in the modeling cycle. To sum up, it is crucial going over the nature of the asynchronism present in the process data to decide what synchronization method to use in order to get the optimum results without perturbing the actual relationships of the variables.



Relaxed Greedy Time Warping for BMSPC

Part of the content of this chapter has been included in the following publications:

- [1] J.M. González-Martínez, A. Ferrer and J.A. Westerhuis. Real-time synchronization of batch trajectories for online multivariate statistical process control using Dynamic Time Warping, *Chemometrics and Intelligent System Laboratory*, 105:195–206, 2011.
- [2] J.M. González-Martínez, J.A. Westerhuis and A. Ferrer. Using warping information for batch process monitoring and fault classification, *Chemometrics and Intelligent System Laboratory*, 127:210–217, 2013.
- [12] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 3: Batch Process Data. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.

5.1 Introduction

Most of the research done on batch synchronization has been focused on post-batch process monitoring. Until today, there has not been a high interest to face the issue of synchronization in BMSPC for real-time applications. Only Kassidas et al. [76] and Ramaker et al. [160] commented on this topic (see Chapter 4). Nonetheless, there are several open issues deserving research.

One important issue in the real-time synchronization procedure proposed by Kassidas et al. [76] is the search of the optimal path at each sampling time point. Performing a backward search when a new sample is available to find the best matching path from the current sampling time point t to the start point $(1,1)$ may derive a different optimal path from the synchronization carried out from the previous sampling time point $t-1$ to the start point. This allows the procedure to search for the optimum global path as the batch evolves, but since this is being done in real-time, the batch trajectories synchronized at the later stages may differ from those obtained from the synchronization at the earlier stages. The projection of these trajectories onto the latent space will lead to different values in the multivariate statistics that may cause uncertainty and false alarms in the monitoring scheme. Another aspect under discussion is the adaptation of the global constraints in real-time synchronization. Since the duration of incoming batches is unknown and may differ from those batches used to tune up the parameters of the algorithm, the update of the upper and lower boundaries is necessary. If this global constraint is not updated on demand, erroneous synchronizations may be produced with the subsequent increase of the false alarm rate.

Another topic scarcely ever discussed is the use of the warping information provided from batch synchronization for process monitoring and fault classification in BMSPC. As explained in Chapter 4, the warping profiles give information on the performance of the different stages or phases of the process throughout the batch run. This may be related to the appearance of faults during the process and may have a direct effect on product quality. Hence, the analysis of the set of warping profiles obtained from offline and online synchronizations is highly desired. Some authors have emphasized the importance of not discarding the information derived from the synchronization [76] and others have used this warping information as an extra variable in the multivariate analysis [1, 66]. However, there is no sound study on the use of the warping information for: i) unsupervised (i.e. requiring no a priori knowledge about the type of faults) process monitoring,

and ii) supervised (i.e. incorporating prior knowledge from a data base of historical faults) fault classification.

This chapter presents solutions to the synchronization challenges that have not been addressed yet. An adaptation from Kassidas et al.'s approach [76] is introduced to perform the synchronization of batch trajectories using the DTW algorithm in real-time. In the proposal, a new boundary definition is presented for accurate real-time synchronization of ongoing batches as well as a way to adapt mapping boundaries to batch length. A relaxed greedy strategy is introduced to avoid assessing the optimal path each time a new sample is available. For offline applications the DTW algorithm is however preferred because it provides the optimum global solution. The key advantages of the proposed strategy are its accuracy in the context of batch real-time process monitoring and its computational speed. In addition, the use of the warping profiles for process monitoring and fault classification is discussed. Furthermore, guidelines for building unsupervised control charts for fault detection and classifiers for fault classification based on the warping information are provided.

The structure of this chapter is as follows. Section 5.2 presents the new algorithm for real-time batch synchronization. Section 5.3 describes how to use the warping information derived from synchronization for fault detection and classification. Their application to post-batch and real-time process monitoring is illustrated in Sections 5.4 and 5.5. Finally, Section 5.6 gives some concluding remarks.

5.2 Relaxed Greedy Time Warping algorithm

The RGTW algorithm builds up a piecewise solution following a greedy optimization approach, so that at each time the best local synchronization improvement is incorporated into the global synchronization solution. This synchronization procedure is based on the proposal of Kassidas et al. [76] but synchronization in the new proposal is carried out within a sliding window ζ of defined width γ , which is optimized by cross-validation. Hereafter, the core of the algorithm will be presented: i) a new global band constraint definition that takes into account the variability across all batches in each time period, ii) an optimization procedure to estimate the parameters of the algorithm, and iii) a strategy to adapt mapping boundaries to the batch length. Also, the implications of using the sliding window of suboptimal paths in RGTW instead of the globally optimal path of Kassidas et al. [76] is discussed.

5.2.1 Enhanced global constraints

The band global constraint is a paramount issue that has to be taken into account in online applications. The use of this constraint avoids large deviations from the diagonal path, preventing similar features in different time periods from being matched. For a feasible search, the band size M has to be at least equal or greater than the absolute value between the reference and test batch duration. As the duration of a new batch is unknown beforehand, this parameter cannot be straightforwardly set. In this situation, the maximum, median or average batch duration from NOC batches is typically used to impose M . The advantage of this strategy is that a large search space is available to obtain the optimal synchronization solution. However, the actual variability of the process in terms of duration is not considered, which may cause similar features over time to be incorrectly synchronized. This drawback may harm synchronization to a larger extent if notable differences in batch length are found stage-to-stage or phase-to-phase across batches. To overcome these setbacks, the search space should be defined based on the consistency from batch-to-batch, i.e. boundaries ought to be narrowed at time periods with less deviation from the diagonal path.

A band boundary more suitable for batch process monitoring is proposed. The warping window is optimized based on batch process understanding and offline synchronization of NOC batch trajectories. Once a set of synchronized batches is available after aligning without global constraints, boundaries from historical NOC batch data are obtained. Note that the offline synchronization is performed without boundaries in this case because the main goal is to find the optimal path across all batches at the risk of matching similar features belonging to different phases. In case that non-optimal global constraints were used, there would be no guarantee that the optimal path could be derived.

Let \mathbf{F} be a $(N \times K_{w_n})$ matrix with all optimal paths \mathbf{f}^* obtained after applying the DTW algorithm, where N is the number of NOC batches and K_{w_n} is the number of warping indexes for the n -th batch. From all optimal paths, new boundaries are defined as:

$$\begin{aligned} \hat{u}_t &= \max(\mathbf{F}_{1,w(t,j(k_1))}, \dots, \mathbf{F}_{N,w(t,j(k_N))}) \\ \hat{l}_t &= \min(\mathbf{F}_{1,w(t,j(k_1))}, \dots, \mathbf{F}_{N,w(t,j(k_N))}) \\ t &= 1, \dots, \max(K_n) \text{ and } n = 1, \dots, N \end{aligned} \quad (5.1)$$

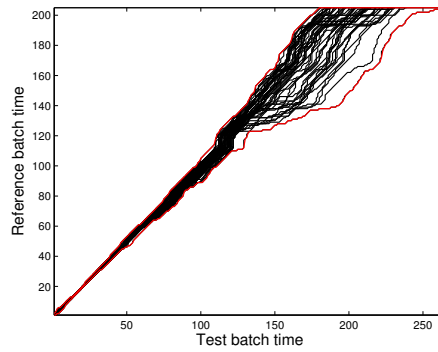


Figure 5.1. Set of optimal paths found by applying the offline DTW algorithm without global constraints. The red lines are the minimum (\hat{l}_t) and maximum (\hat{u}_t) values at each sampling time point, which define an envelope.

where \hat{u}_t and \hat{l}_t are the upper and lower boundaries at the t -th sampling time point, respectively. These boundaries generate an envelope that includes all possible ways that the optimal path may go along to synchronize multivariate batch trajectories based on the synchronization of NOC batches (see Figure 5.1).

Note that the synchronization should be re-evaluated by using a well-defined band global constraint provided that some process events temporally spaced are incorrectly aligned. The warping information from the first synchronization should be used to define a band that enables a more restricted search of the optimal solution in periods with less variability and less restricted in periods with more variability in the process pace. This constraint would allow the synchronization algorithm to find the optimal solution and avoid that similar features belonging to different process phases are aligned. For instance, assume that for the simulated example shown in Figure 5.1, wrong alignments were obtained in the first half of the batch runs, where the differences in pace and length are smaller than in the second half of the batch runs. These misalignments would be reflected in the warping profiles in form of large vertical transitions. To avoid the occurrence of wrong matches, two band global constraints could be defined: a band for the first 100 sampling time points whose band size is $M_1 = 15$, and a second band for the remaining part of the run whose band size is $M_2 = 60$. Note that these bands are not so restrictive as the ones imposed from the warping information shown in Figure 5.1, but

restrictive enough to limit the matching of common features from different periods between the reference and the test batch. Once an optimal synchronization is derived, the upper and lower boundaries can be estimated from the warping profiles as explained above.

5.2.2 The RGTW algorithm

In order to introduce the fundamentals of the RGTW algorithm, let us assume a new vector of measurements of an evolving batch $\tilde{\mathbf{b}}_{new,t}^T$ ($1 \times J$) is available at the t -th sampling time point. The RGTW algorithm can be defined by the following steps:

- 1) **Scaling.** Scale the new vector of measurements $\tilde{\mathbf{b}}_{new,t}^T$ using the average range \bar{R}_j estimated from NOC batches.
- 2) **Applying the symmetric DTW algorithm with constraints.**
 - i) Add new column in the $t - 1 \times K_{ref}$ grid.
 - ii) Estimate the weighted local distance between $\tilde{\mathbf{b}}_{new,t}^T$ and the row vectors of the \mathbf{B}_{ref} trajectory matrix corresponding to the batch reference sampling time points belonging to the interval defined by boundaries \mathbf{u} and \mathbf{l} .
 - iii) Assess the set of cumulative weighted distances $D_{t,j}$ at the t -th sampling time point, where $j = \hat{l}_t, \dots, \hat{u}_t$.
 - iv) Obtain the sampling time point of the batch reference \mathbf{B}_{ref} where the minimum cumulative weighted distance $D_{t,j}$ occurs at the t -th sampling time point, i.e. e_t^* .
 - v) Reconstruct the optimal path.
If $t < \gamma$, the optimal path is derived following the approach of Kassidas et al. [76]: from the starting point $(1, 1)$ to the end point e_t^* .
Otherwise, reconstruct the optimal path $\mathbf{f}_{t-\gamma+1,t}^T$ for the current window $\zeta_{t-\gamma+1,t}$ at the interval $[t - \gamma + 1, t]$, where γ indicates the window width.
- 3) **Saving optimal path from the current window $\zeta_{t-\gamma+1,t}$.**
If $t < \gamma$, any pairwise points of the optimal path from the window $\zeta_{1,t}$ is saved to the final solution $\hat{\mathbf{f}}^T$.
Otherwise, coordinates from the optimal path within the window $\zeta_{t-\gamma+1,t}$ for the $(t-\gamma+2)$ -th sampling time point will be saved in the final solution $\hat{\mathbf{f}}^T$ as

$$\hat{\mathbf{f}}_t^T = \hat{\mathbf{f}}_{t-1}^T \cup \hat{\mathbf{f}}_{t-\gamma+1|w(t-\gamma+2,j(k))}^T \quad (5.2)$$

being $\hat{\mathbf{f}}_{t-\gamma+1|w(t-\gamma+2,j(k))}^T$ every coordinates of the optimal path $\hat{\mathbf{f}}_{t-\gamma+1}$ found within the window ζ belonging to the reference batch points that match with the $(t-\gamma+2)$ -th evolving batch sampling time point. $\hat{\mathbf{f}}_{t-1}^T$ and $\hat{\mathbf{f}}_t^T$ are the final optimal path at the previous and current sampling time points, respectively.

5) **Sliding current window** $\zeta_{t-\gamma+1,t}$.

If $t < \gamma$, the following window at the t -th sampling time point will be the same one, i.e. $\zeta_{1,\gamma}$.

Otherwise, the current window is moved one position forward. Set the start point to the last pairwise point belonging to $w(t-\gamma+2, j(k))$.

6) **Updating boundaries.**

If $t > \max(K_n)$

$$u_t = u_{t-1}, l_t = l_{t-1} \quad (5.3)$$

else if $e_t^* = u_t$

$$u_i = \min(u_i + 2, K_{ref}), i = t + 1, \dots, \max(K_n) \quad (5.4)$$

else if $e_t^* = l_t$

$$l_i = \begin{cases} l_i - 1 & i = \{t + 1, t + 2\} \\ l_i - (l_{t+2} - l_t) & i = t + 3, \dots, \max(K_n) \end{cases} \quad (5.5)$$

being $n = 1, \dots, N$, the number of batches from the calibration data set.

When the duration of the ongoing batch is longer than the longest duration of NOC batches, with every new measurement coming in, the boundaries will be extended by taking the last value of the upper and lower boundaries.

After handling the synchronization step, synchronized samples for the $(t-\gamma+1)$ -th batch time are available. Depending on whether the algorithm compresses or expands the samples, an update of the last monitored sample (an average of previous samples that match with the same reference point is calculated, yielding a new value) or new values (replicates of the previous synchronized sample) will be derived, respectively. As a consequence of the previous step, an incomplete batch in the reference time scale is available to be monitored. Hence, the prediction of the unknown batch part becomes a crucial task to proceed with monitoring. Nomikos and MacGregor [82] proposed three approaches to predict unknown samples up to the

end of the batch. Garcia et al. [168] presented a comparative study of trajectory prediction and process monitoring for online applications that recommended the TSR method [135, 169] for predicting the unknown samples. In this work, the TSR method is used to impute unknown samples, which is a simple and efficient method to estimate missing data. Eventually, once the entire batch is available with the unknown part estimated, it can be monitored in real-time by following the modeling approach of Nomikos and MacGregor [70, 82].

For the sake of understanding, an example of real-time batch synchronization using the RGTW algorithm is provided in Figure 5.2. Let us assume that the boundaries \mathbf{u} and \mathbf{l} , and the synchronization window ζ with width $\gamma = 3$ units are the cross-validated RGTW parameters. In this example, when a new vector of measurements is available at each sampling time point, the end point e^* (i.e. the grid point inside the window where the minimum cumulative weighted distance occurs) is assessed and the optimal path is subsequently calculated (equivalent to Kassidas et al.'s online implementation). However, none of the pairs $[i(k)j(k)]$ from the optimal path \mathbf{f}^T within the first window $\zeta_{1,3}$ is saved to the final solution $\hat{\mathbf{f}}^T$, and consequently, no synchronization is carried out (see Figure 5.2(a)). When the measurements belonging to the 4th sampling time point are collected, the synchronization window must be moved one position forward, leading to the window $\zeta_{2,4}$ (see Figure 5.2(b)). At this point, the pairs $[i(k)j(k)]$ (green line in Figure 5.2(b)) corresponding to the first ongoing batch sampling time point is saved to the final solution $\hat{\mathbf{f}}^T$ and the first synchronized batch samples are then available. Note that, following this approach, the synchronization is lagged $\gamma = 3$ sampling units. Again, the end point e^* is assessed and the optimal path $\mathbf{f}_{2,4}^T$ within the window $\zeta_{2,4}$ is calculated (blue line in Figure 5.2(b)). Afterward, the lower boundary is updated since the end point e^* is just located in one of the points belonging to the mentioned boundary. The procedure to perform the batch synchronization at the 4th sampling time point is repeated in the consecutive sampling time points (Figures 5.2(c)-5.2(f)). Recall that the estimation of the optimal path $\hat{\mathbf{f}}^T$ from the RGTW algorithm is optimized window-to-window, discarding the wrong synchronization steps (magenta solid lines shown in Figures 5.2(c)-5.2(f)).

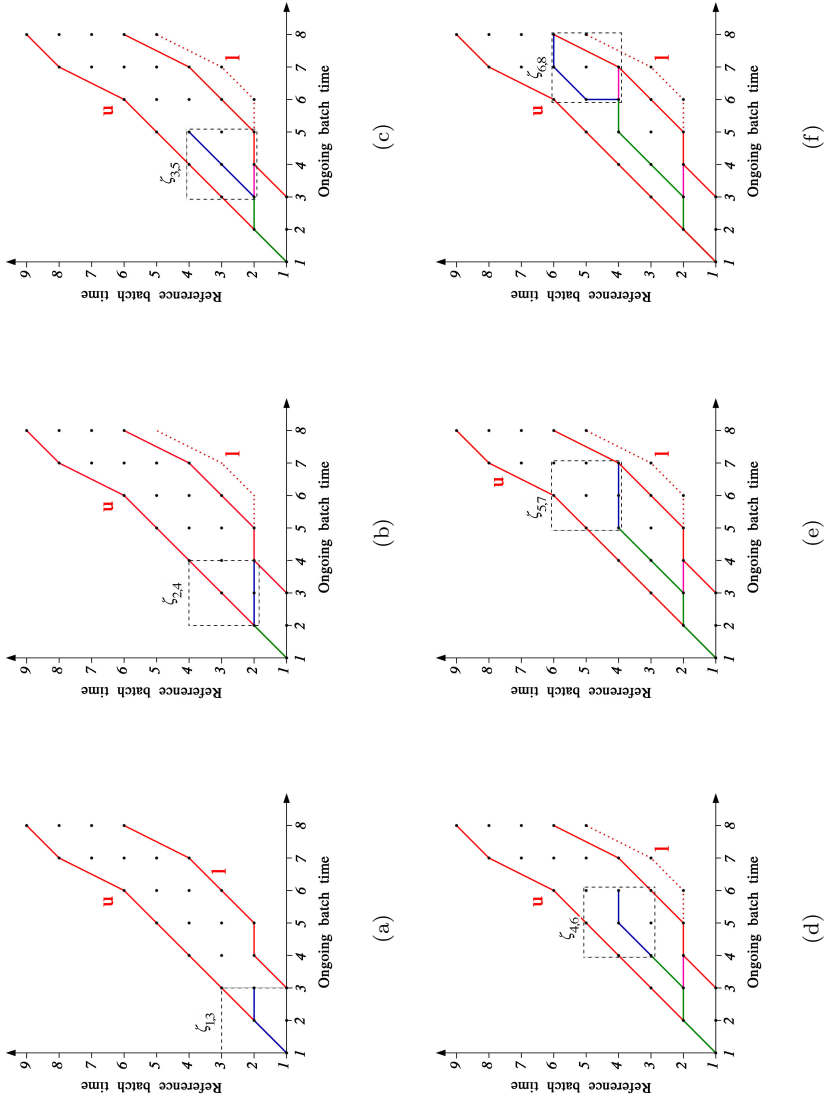


Figure 5.2. Performance of the online synchronization of an ongoing batch from the 1st up to the 8th sampling time points in chronological order (from (a) to (f)), using the RGTW algorithm with a window width γ equal to 3 units. Red solid lines represent the boundaries that restrict the optimal path search and the red dashed line represents the lower boundary update. The final optimal path $\hat{\mathbf{f}}_t^T$ and the optimal path $\hat{\mathbf{f}}_{t-\gamma+1}$ found in the window $\zeta_{t-\gamma+1}$ are plotted by green and blue solid lines, respectively. Magenta solid lines are path segments that are not taken into account on the final solution, and therefore, are considered as wrong synchronization.

5.2.3 Cross-validation for the estimation of the RGTW parameters

In order to apply the RGTW algorithm, the width γ of the synchronization window ζ and the boundaries, must be set up by following a cross-validation procedure [1]. This procedure is performed as follows (see Figure 5.3).

Firstly, the matrix with N optimal paths \mathbf{F}_{DTW} ($N \times K_{ref}$) is achieved after applying the offline DTW algorithm without global constraints to the raw batch data. Later on, the raw batch trajectories are split into a training set (train) and test set (test) N times in the loop, being N the number of batches. This step is carried out following the leave-one-out procedure. Every number of window widths γ that is considered is used to synchronize the train set in each iteration. As a result, a matrix with $N - 1$ suboptimal paths and window width γ $\mathbf{F}_{RGTW,\gamma}$ is obtained. Once the upper and lower boundaries are defined from those $N - 1$ paths, the test batch is synchronized using the online approach, leading to a suboptimal path $\mathbf{f}_{n,RGTW}^T$, being $n = 1, \dots, N$. This procedure is repeated until all N batches are synchronized over all window widths. The Pearson's correlation coefficient ($r(\mathbf{f}_{n,DTW}, \mathbf{f}_{n,RGTW})$) between the suboptimal paths found in the loop for each window width γ and the optimal path from the offline synchronization for all NOC batches is assessed.

Once the above procedure has been executed, a study of the performance of the RGTW-based real-time synchronization as function of the different window widths γ studied is necessary. The most adequate procedure is to perform an ANOVA on the Fisher Z-transformed correlation coefficients¹ to determine whether there are statistically significant differences among the window widths under study or not, following the idea of [1]. When the window width to be used for the RGTW has been selected, the upper and lower boundaries are estimated as the envelope of the warping paths obtained in the cross-validation procedure for such γ .

An assumption in the estimation and optimization of the synchronization parameters for online applications is that the data used is representative of the normal operating conditions of the process under study. However, there must be significant disturbances, such as composition variations, manufacture campaigns, and seasonal changes that were not reflected in the calibration data set, but it might appear

¹Fisher transformation is an elementary transcendental function called the inverse hyperbolic tangent function. This transformation on the Pearson's correlation coefficient is required to approximate it to a normal distribution.

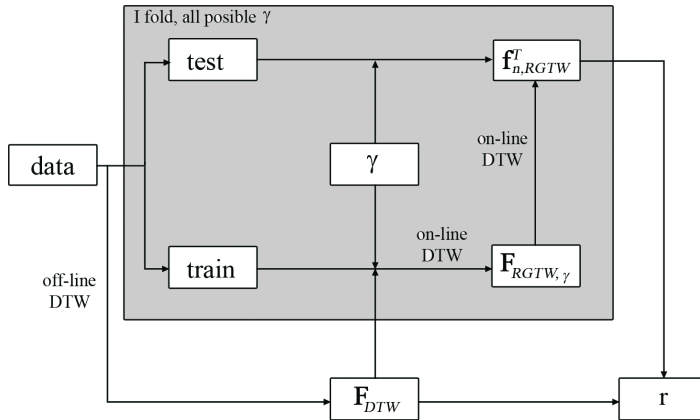


Figure 5.3. Iterative procedure to assess the performance of the RGTW algorithm using different window width γ .

while monitoring the process. In these cases, the meta-parameters of the subsequent multivariate model built for real-time monitoring should be adapted or even be subject to a new modeling, depending on the nature of the disturbances or conditions, and on the changes of the correlation structure. Likewise, an update of the synchronization parameters (weights of the process variables, sliding window and boundaries) should be performed to accommodate new patterns of asynchronisms and process paces. If the adaptation of the synchronization parameters is neglected, it might cause the addition of false features in the trajectories or even the misalignment of key process events. The inaccurate synchronization of the batch trajectories would likely lead to an increase of the false alarms in the control charts. To carry out this adaptation, apart from recalculating the weights given to the process variables to perform the synchronization with new historical batches, the RGTW parameters should be tuned again by re-evaluating the cross-validation procedure. Nonetheless, in the cases where incoming batches are considered as NOC by the statistical model, assuming that the warping information derived from synchronization is taken into consideration as another process variable, the re-estimation of the synchronization parameters would not be required.

5.2.4 Trade-offs of RGTW and comparison with online DTW

The ideal result of the synchronization methods in real-time applications would be to find the global optimal solution that the DTW algorithm finds in offline applications. Nonetheless, the fact that the remaining part of the ongoing batch until completion, i.e. from the current sampling time point until the end of the batch, is unknown disables the DTW algorithm to find the global optimum. When the process evolves, the Kassidas et al.'s online implementation finds the optimal local solution at each sampling time point, which usually diverges from the optimal global solution at the end of the batch. Even though the algorithm makes corrections when new data measurements are available to optimize the synchronization, the optimum is not necessarily achieved until the batch is completed. In contrast, the RGTW algorithm provides a suboptimal solution at each sampling time point, which is to a certain extent similar to the final solution obtained with the Kassidas et al.'s online implementation at the end of the batch. The RGTW algorithm does not make any rectification of the final solution in real-time because the method is based on a greedy search within a window whose width is tuned by a cross-validation procedure. This strategy permits minimizing the number of misalignments caused by corrections and maximize the similarity between the online and offline results derived from RGTW and DTW, respectively. The trade-off of RGTW is that the optimality of the solution that can be achieved is related to how wide the synchronization window ζ is. The larger the window width, the closest the solution to the global optimal solution. However, the disadvantage of imposing a window of certain width γ is that the start of the monitoring is delayed as many sampling time points as the width of the window has. Hence, the selection of the window width should be a compromise between how optimal the solution should be at least and the maximum acceptable delay to start monitoring the process.

The usage of the sliding window ζ jointly with the updated boundary constraints are key factors in the RGTW algorithm to reduce the number of false alarms in the control charts. The synchronization within a window allows the algorithm to find a suboptimal solution making corrections when required during the time interval established by the window itself. This favors the optimization of the synchronization solution without compromising the performance of the control charts. Only when the sliding window moves on to the

next sampling time point is when a new synchronized measurement vector is projected onto the latent structure to assess whether the process evolves under NOC. This is one of the big differences versus the Kassidas et al.'s online implementation, where all the available samples are re-synchronized and re-evaluated by the model again at each sampling time point. However, when an ongoing batch remains NOC but has different features in comparison to the historical NOC batches, the sliding window is not sufficient to reduce the false alarms. As the synchronization is restricted by the global band constraint, it might happen that ongoing NOC batches that are, for instance, shorter or longer are inaccurately synchronized. When this divergence from historical NOC batches occurs, the optimal path lies along the boundaries. The update of the boundaries in this situation permits the algorithm to expand the search of the suboptimal solution, thereby reducing the risk of adding artificial features (subtle peaks) or misalignments that might cause false alarms in the control charts. Hence, the sliding window and the adaptation of the boundaries to the new features of incoming NOC batches allows the RGTW algorithm to reduce the number of false alarms. Note however that if there are significant changes in the process conditions (e.g. seasonal variations, composition variations, changes in concentration or flows that affects the kinetics of biological reactions), the RGTW parameters would not impede the appearance of false alarms even though the incoming batches remain NOC. In these cases, it would be required to re-evaluate the synchronization and the optimization of the parameters.

5.3 Warping information for enhanced BMSPC

Unsupervised and supervised tools for process monitoring and fault classification are proposed in this section: i) an unsupervised control chart based on the warping profiles from NOC batches (*NOC Warping Information-based Control Chart* (NOC-WICC)), ii) supervised classifiers based on warping information-based control charts (faulty WICC) and supervised chemometric tools (*Partial Least Squares Discriminant Analysis* (PLSDA) [170] and *Soft Independent Modeling of Class Analogy* (SIMCA) [171, 172]) for fault classification.

5.3.1 NOC warping information-based control charts

RGTW-based synchronization provides not only the synchronized multivariate measurements, but also the optimal warping functions \mathbf{f}_n derived for each of N batches. For each of the test batches synchronized against the reference batch, a warping function with length equal to K_{w_n} is obtained. The length of the warping information among batches is different. In order to use this information to build the NOC-WICC and design the fault classifiers, all warping profiles need to have the same length. Hence, the warping profiles must be expressed as function of the reference batch to have equal length. This transformation is performed as follows (see Figure 5.4). The test batch sampling time point that matches with each one of the reference batch sampling time points is estimated. In case that a set of consecutive horizontal transitions are present, i.e. n_t test batch sampling time points are matched with the k_{ref} -th reference batch sampling time point, the last sampling time point of this set is taken as the matched point. Note that the interpretation of this consecutive test batch sampling time points matching a certain reference batch sampling time point is different in the inner RGTW algorithm. In the latter case, an average of the values belonging to the multivariate batch trajectories is calculated and matched with a defined reference batch sampling time point. Hence, it is recommended at a first instance to compare the original and transformed warping profiles as well as investigating the synchronized trajectories to ensure that the batch synchronization was correctly performed. In the example in Figure 5.4, 2nd-4th sampling time points of the test batch are matched with the 2nd sampling time point of the reference batch (see warping profile f_1 in Figure 5.4(a), line depicted with asterisks), so the 4th sampling time point of the test batch is matched with the 2nd sampling time point of the reference batch (see warping profile f_1 in Figure 5.4(b), line depicted with asterisks). This procedure is repeated for each one of the K_{ref} reference batch sampling time points over all N_{NOC} test batches. At the end of the execution, a set of warping profiles with equal length is available (see Figure 5.4(b)). Once this transformation has been performed, the matrix \mathbf{F}_{NOC} ($N_{NOC} \times K_{ref}$) containing the N_{NOC} warping profiles expressed as a function of the reference batch is obtained. Data containing this matrix define the consistent and normal processing pace through the batch time. For the sake of interpretation, the monotonic increasing behavior of the warping profiles is removed by subtracting the average values of each one of the reference batch points (columns of matrix

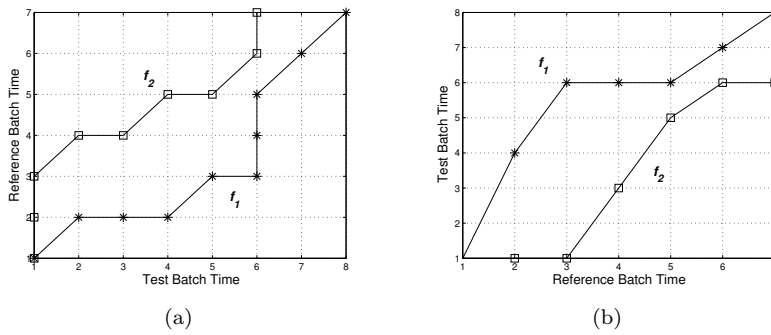


Figure 5.4. Warping profiles f_1 and f_2 belonging to two different trajectories obtained from the RGTW-based synchronization (a) and expressed as a function of the reference time (b).

\mathbf{F}_{NOC}). From this centered \mathbf{F}_{NOC} matrix, we propose to build the NOC-WICC. This is a complementary tool to Hotelling- T^2 and SPE control charts for post-batch process monitoring. The corresponding control limits at 99% confidence level can be assessed by estimating the percentile 0.5 and 99.5 at each reference batch time point (columns of matrix \mathbf{F}_{NOC}).

In real-time applications, NOC-WICC can also be used for unsupervised process monitoring. In this context, a new point from the warping information is available when a set of sampling time points from an ongoing batch are matched with the next k_{ref} -th sampling time point from the reference batch. Hence, the real-time monitoring of the warping profile would have a certain delay from the original batch time. This is necessary in order to ensure that the ongoing warping profile has the same length as those corresponding to NOC and can be monitored with NOC-WICC.

5.3.2 Fault classification procedures

Let us assume that a set of warping profiles derived from the RGTW-based synchronization of batch trajectories belonging to historical faulty batches with different types of faults l ($l = 1, 2, \dots, L$) are collected in the matrices $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_L$. Warping information obtained from the RGTW-based synchronization not only can be used for process monitoring, but also for fault classification. Once the

monitoring system has detected an out-of-control signal, if historical faulty batches are available and the different faults are fingerprinted in their warping information profiles, supervised methods can be used to classify the type of fault that occurred in post-batch applications.

In this chapter, three different supervised techniques are proposed: faulty WICC-, PLSDA- and SIMCA-based classifiers. For the design of these classifiers, each one of the matrices \mathbf{F}_l ($N_l \times K_{ref}$) is split up into two different data sets, a training and a test data set containing warping profiles of $N_{training}$ and N_{test} faulty- l batches, respectively, where $N_{training} + N_{test} = N_l$. Using the training data set a model/classifier is developed and optimized after outlier removal. The test data set is used to estimate a different classification index for each classifier: membership probability to fault l in faulty WICC, predictions in PLSDA and SPE in SIMCA.

In order to assess the quality of classification using a defined threshold, measures derived from the confusion table, which consist of the number of false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN) existing in the classification, are used [173]. From these measures, the sensitivity and specificity are estimated. Sensitivity (true positive rate) measures the proportion of actual positives which are correctly identified as such, i.e. $TP/(TP+FN)$. Specificity (true negative rate) measures the proportion of negatives which are correctly identified as such, i.e. $FP/(FP+TN)$. Both specificity and sensitivity depend on the setting of the classification threshold of the classifier used. By shifting this threshold, a set of sensitivity and specificity values are obtained. Both indexes are between 0 and 1; the closer to 1 the better. To establish the best classification threshold, the *Matthews Correlation Coefficient* (MCC) [174] is estimated as follows:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FN) \cdot (TN + FP)}} \quad (5.6)$$

The Matthew's correlation coefficient ranges from $-1 \leq MCC \leq 1$. A value of $MCC = 1$ indicates the best possible classification (positive correlation between predicted and real case), i.e. every faulty- l batch is correctly classified, and only the true faulty- l batch is classified as fault l class. In contrast, a $MCC = -1$ indicates the worst possible classification (negative correlation between predicted and real case), i.e. none of faulty- l batches is correctly classified, whereas the rest of faulty batches are erroneously classified as fault- l class. Finally, a

value of $MCC = 0$ stands for no correlation between predicted and real case (random distribution). Hence, that threshold whose MCC value is the closest to 1 will be selected as the classification threshold.

For the sake of comparison among classifiers, the *Receiver Operator Characteristic* (ROC) curve will be used. In order to build the ROC curve, the sensitivity of the faulty- l data set versus 1-specificity is plotted for each of the possible thresholds. The closer the curve follows the left-hand border and subsequently the top border of the ROC space, the more accurate the classifier. This accuracy can be obtained by calculating the area under the ROC curve (the so-called AUROC), which can be seen as an index measuring the goodness of the classifier. In case the classifier is capable of perfectly separating the different classes, the area will be equal to 1. In contrast, if the classifier provides random classifications, the points will fall into the main diagonal, yielding to an area of 0.5.

Supervised faulty WICC

The idea is to build a control chart from the warping information contained in each matrix \mathbf{F}_l corresponding to the training data set. First, each warping profile is centered to the NOC average warping profile. Afterward, the control limits at 99% confidence level are assessed by estimating the percentile 0.5 and 99.5 at each reference batch time (columns of the matrices \mathbf{F}_l). The test data set of each of the l types of faults is then plotted onto the faulty- l WICC and the percentage of points falling within the control limits is calculated. This is an index of the membership probability to fault l . This procedure is repeated for each one of the matrices $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_L$, i.e. for every known fault. Finally, the faulty WICC-based classifier is built as explained above. A new complete faulty batch can be classified into fault class l , if its percentage of points falling within the control limits of the fault- l WICC is larger than the corresponding classification threshold.

In online applications, when new RGTW-based synchronized measurements are available, the corresponding warping information is plotted on each of the control charts designed (one for each class under study). This is necessary since the membership of the ongoing batch to one of the defined classes is unknown a priori. A value of the warping function falling outside the control limits or any other non-random set of points in one of the control charts would signal the running batch does not belong to that particular batch class. In order to make the interpretation of the supervised faulty warping

information-based control charts easier, the ratio of the number of points falling within the confidence region and the total number of points plotted in each of this control charts can be calculated. Then, a membership probability of each one of the known faults can be also estimated.

Classifiers based on PLSDA

The following approach is based on the fit of a PLSDA model from the matrix \mathbf{F} , which was obtained after arranging the matrices $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_L$ corresponding to the training data set, one below the other. The response \mathbf{Y} matrix is defined by dummy variables denoting the type of faulty batches (i.e. classes). In this type of analysis, the elements of the column vector \mathbf{y}_c are one for batches belonging to class c (fault-1, fault-2, ..., fault- l , ..., fault- L) or zero the other way around. Both matrices \mathbf{F} and \mathbf{Y} are autoscaled. Once the PLSDA model is fitted, the warping profiles of each type of fault belonging to the test data set are preprocessed and projected onto the PLSDA model, yielding the fault class predictions. After checking that their Euclidean distances to the latent model (SPE) are lower than the 99% confidence level (control limits estimated using the approximation method by Jackson and Mudholkar [115]), the best classification threshold is calculated by following the procedure explained at the beginning of this section. When a new complete warping profile \mathbf{f}_{new} corresponding to a faulty batch is available, it can be classified. For this purpose, \mathbf{f}_{new} is centered and scaled using the mean and variance estimated from the training data matrix \mathbf{F} , and its prediction vector $\hat{\mathbf{y}}_{new}$ is predicted. Provided that the SPE value of the new faulty batch is below the control limit and its prediction $\hat{\mathbf{y}}_{new}$ is above the classification threshold of fault- l classifier, this batch can be classified as fault- l .

In real-time applications, the synchronization of an ongoing batch is achieved at every time a new measurements vector is available. Note that only the warping information belonging to the k_{ref} -th reference batch time matched is available (denoted as $\mathbf{f}_{new,1:k_{ref}}$) whereas the warping function belonging to the remaining $K_{ref} - k_{ref}$ sampling time points are unknown (represented as $\mathbf{f}_{new,k_{ref}+1:K_{ref}}$). In order to project the new measurements onto the latent space, yielding the corresponding vector $\hat{\mathbf{y}}_{new}$, it is necessary to have the values of the complete warping information vector \mathbf{f}_{new} . Here, the use of the TSR method as imputation technique is suggested. For details concerning

its theoretical formulation and application to PLS models, reader are referred to [89].

Classifiers based on SIMCA

Soft Independent Modeling of Class Analogy (SIMCA), is a well-known classification method. This approach consists of the fit of a PCA model from each autoscaled matrix \mathbf{F}_l containing the warping information of the faulty- l batch ($l = 1, \dots, L$), corresponding to the training data set. Once the PCA models have been fitted, the warping profiles of faulty batches belonging to the test data set are preprocessed and projected onto the latent space, obtaining the squared Euclidean distances to the latent model (SPE). In case these values do not exceed the control limits at 99% confidence level (control limits estimated from theoretical results [115]), they are used to estimate the best classification threshold by following the procedure explained early in this Section. When a new complete warping profile \mathbf{f}_{new} corresponding to a faulty batch is available, this can be classified. For this purpose, \mathbf{f}_{new} is centered and scaled using the mean and variance estimated from the training data matrix \mathbf{F} , and then its SPE value is estimated. The SIMCA classification is made based on the proximity of the new faulty batch to the model plane of the l -th SIMCA classifier, which is measured by the SPE statistic. If this value is below the classification threshold of fault- l classifier, the new faulty batch is assigned to fault- l class. Otherwise, the new faulty batch might belong to either one of the other faulty classes under study or to an unknown class if the SPE value exceeds the classification threshold of all the SIMCA classifiers.

For real-time applications, the projection of new samples of an ongoing batch can not be achieved if the warping information vector is not completed. In such context, again the application of imputation techniques to estimate the remaining values is needed. In this case, the PCA-based TSR is suggested to be used given its good performance. For details about its theoretical formulation and application, readers are referred to [136].

5.4 Application of batch synchronization

A comparative study of the performance of the RGTW algorithm and the online synchronization approach based on DTW from Kassidas et al. [76] and the effect on fault detection is conducted. For this

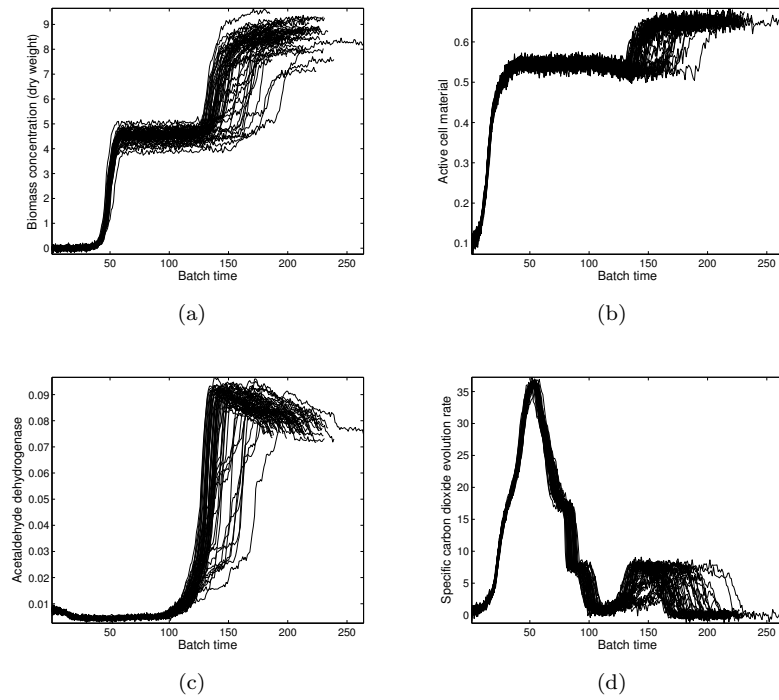


Figure 5.5. Trajectories belonging to four process variables before synchronization: (a) biomass concentration (dry weight), (b) active cell material, (c) acetaldehyde dehydrogenase, and (d) specific carbon dioxide evolution rate.

purpose, the simulated data of Set #2 containing 50 NOC batches ($\tilde{\mathbf{X}}_3$) and 3 faulty batches ($\tilde{\mathbf{X}}_4$) are used (see Chapter 2).

Figure 5.5 shows 4 out of 10 process variables for 50 good quality batches from the simulated process. The variables shown illustrates the lack of synchronization among the trajectories and the dissimilarity in length. Before carrying out the statistic modeling, the raw data need to be synchronized offline. Once the synchronized data have been modeled, the comparison is performed using the online version of the synchronization methods.

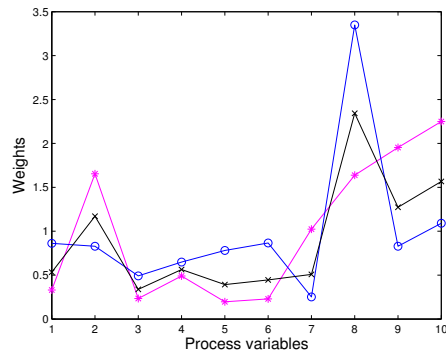


Figure 5.6. Values of the weight matrix \mathbf{W} assessed by following the approach of Kassidas et al.'s (magenta asterisks) and Ramaker et al.'s (blue circles) approaches together with the geometric mean of both weights (black crosses).

5.4.1 Offline synchronization

In the selection of the batch that represents overall typical batch-to-batch variation, the closest batch to median length is taken as reference batch to perform the synchronization based on DTW. The median length is taken instead of the average length because the median is a statistic less sensitive to outliers. In this case, the median length is 207 sampling time points and batch #4 is the closest with length equal to 205 sampling time points. All conditions and constraints described in Section 4.2.3, except for the Sakoe and Chiba band constraint, are fixed. The reason why this global constraint is not established is because the objective is to find an envelope defined for all the optimal paths.

An issue to be taken into account is what variables should be introduced into the DTW algorithm. It may happen that there are on/off variables providing information about different episodes or phases in the process that would be extremely useful in the synchronization step to improve the synchronization. In contrast, these variables would not likely provide important information in PCA-based modeling if they are not correlated with other variables. In this example, all the measured variables are used for synchronization. The approaches of Kassidas et al. [76] and Ramaker et al. [160] (see Section 4.2.3 for a discussion on the weighting criteria in batch synchronization) are applied to calculate the weight matrix \mathbf{W} , giving as a result (at 8th and 5th iteration, respectively) the weights shown in magenta

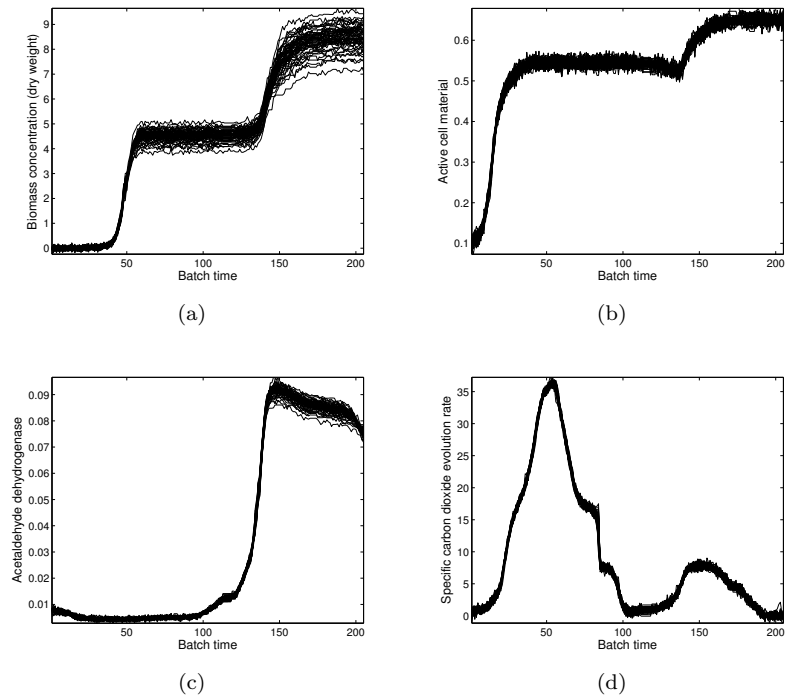


Figure 5.7. Trajectories belonging to four process variables after offline synchronization using DTW based on the weights calculated as the geometric mean of Ramaker et al.’s and Kassidas et al.’s weights: (a) biomass concentration (dry weight), (b) active cell material, (c) acetaldehyde dehydrogenase, and (d) specific carbon dioxide evolution rate.

stars and blue circles, respectively, in Figure 5.6. Later on, the final weights are calculated as the geometric mean of the previous weights (see black crosses in Figure 5.6) to have a proper estimate of the variables that have more batch-to-batch consistency and warping information (see Section 4.2.3). After the DTW algorithm is applied to 50 NOC batches, a set of batches synchronized taking as reference batch #4 is obtained (see Figure 5.7).

As explained in Section 4.2.3, Kassidas et al.’s approach gives more weight to those variables more consistent from batch to batch, i.e. those variables whose trajectories are not notably deviating from the average trajectory. These variables are characterized by a high signal-to-noise ratio, therefore high weights are given. In this example, the

iterative procedure gives less weight to glucose, acetaldehyde, acetate, ethanol and biomass concentration, where the last two variables are the least weighted (see the lowest weights marked as magenta asterisks in Figure 5.6). However, the most weighted variables are the specific oxygen uptake rate and carbon dioxide evolution rate (see the highest weights marked as magenta asterisks in Figure 5.6). This can be understood from Figure 5.7 where the least, intermediately and most weighted variables (biomass concentration, active cell material, acetaldehyde dehydrogenase and specific carbon dioxide evolution rate) are plotted after offline DTW synchronization. The biomass concentration (the 6th process variable depicted in Figure 5.7(a)) has a low signal to noise ratio in certain stages. This fact is caused by the high variation observed throughout the entire batch, yielding a low weight in the synchronization. On the contrary, the specific carbon dioxide evolution rate (the 10th process variable plotted in Figure 5.7(d)) shows little variation around the average trajectory, so a high weight is given. In an intermediate situation are the active cell material and the acetaldehyde dehydrogenase variables (the 7th and 8th process variable shown in Figures 5.7(b) and 5.7(c), respectively), where larger differences are appreciated. In the case of the active cell material, there is not a notable variation between the average trajectory and the training batches from the start to the 40th sampling time point whereas from the 50th sampling time point onward, variation raises. Regarding the 8th variable, acetaldehyde dehydrogenase seems to be consistent from the beginning to the 150th sampling time point but after this sampling time point, a notable variation is appreciated. Hence, these last two variables are given an intermediate weight, being lower on the 7th variable than on the 8th variable, as can be seen in Figure 5.6.

According to Ramaker et al.'s approach, the weights should be given in relation to the amount of warping information contained in the variables. The weights calculated by following this approach are depicted by blue circles in Figure 5.6. This plot shows that variables with less warping information are the acetaldehyde concentration and the active cell material variables (the 3rd and 7th process variable, respectively), whereas the variable acetaldehyde dehydrogenase (the 8th process variable) has a high warping information content. Furthermore, it is worth emphasizing the fact that the 8th process variable receives more weight following the approach of Ramaker et al. than the approach of Kassidas et al. The cause of this difference lies in the shape of the 8th variable trajectory, which contains most important occurrences in the process. Hence, this variable serves as a

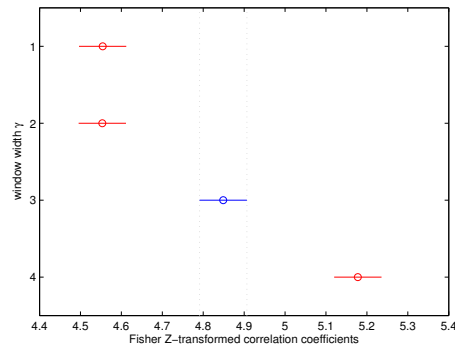


Figure 5.8. LSD intervals for the Fisher Z-transformed correlation coefficients for the different window widths γ .

benchmark to achieve a more accurate synchronization of the rest of process variables. In order to give weights more suitably, the weights of both approaches are geometrically averaged. The idea is to give more weight to those variables that are more consistent from batch to batch and that also incorporate valuable warping information. As can be seen in Figure 5.6, those variables that contain more warping information but are not consistent from batch to batch or vice versa, are weighted to an intermediate value between Ramaker et al.'s and Kassidas et al.'s weights. In case that variables are weighted with similar values following the approach of Ramaker et al. and Kassidas et al., similar weights are given by the geometric mean (see equation 4.19).

Before moving on to the model building and exploitation phase, the width γ of the sliding window ζ to be used by the RGTW algorithm must be set and validated by the cross-validation procedure explained in Section 5.2.3. For this purpose, the Pearson's correlation coefficients between the suboptimal paths found for each window width and the optimal paths from the offline DTW-based synchronization are calculated. This allows us to determine the best window width that reduces the lack of fit caused by the greedy strategy. An ANOVA was performed on the Fisher Z-transformed correlation coefficients for the different window widths, showing statistically significant differences (p -value < 0.05). The LSD intervals displayed in Figure 5.8 show that correlation is statistically higher for window width $\gamma = 4$ (average $r = 0.99992$), whereas no statistically significant differences between window widths $\gamma = 1$ and $\gamma = 2$ are found. As a compromise

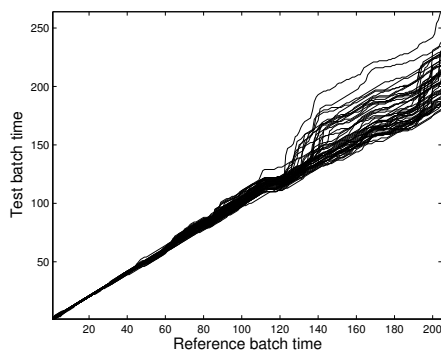


Figure 5.9. Distortion of the batch time over all batches (warping information) after applying the RGTW algorithm.

between computational complexity and performance is desired, a window width $\gamma = 3$ (average $r = 0.99986$) is selected.

Figure 5.9 shows the warping profiles obtained from the validation procedure using $\gamma = 3$. As can be seen, the warping information is smooth for all batches from the start of the batch to the 50th sampling time point. This means that almost no compressions or expansions were necessary to synchronize all batches. In contrast, a greater number of vertical and horizontal transitions were required to align the batches from the 50th to the 120th sampling time point. Specifically, a series of expansions (horizontal paths) can be observed from the 105th to the 120th sampling time point. One reason for this fact may be the phase transition in the process. From this point, an important number of compressions (vertical paths) were needed because many batches have longer duration than the reference batch.

Warping information can be used to improve troubleshooting of problems that may be masked by synchronization. However, caution needs to be exercised when the warping information is taken into account in latent structure-based models. In processes characterized by a large amount of variables, the inclusion of warping information as a new variable may cause this variable to have no influence on the model. This is due to the fact that the warping information may not be correlated with any of the process variables. Consequently, this new variable would have negligible weight on the model. In such cases, this variable should be considerably weighted in order to avoid this problem. In this work, the warping information is introduced as a new variable into the PCA model without applying any special

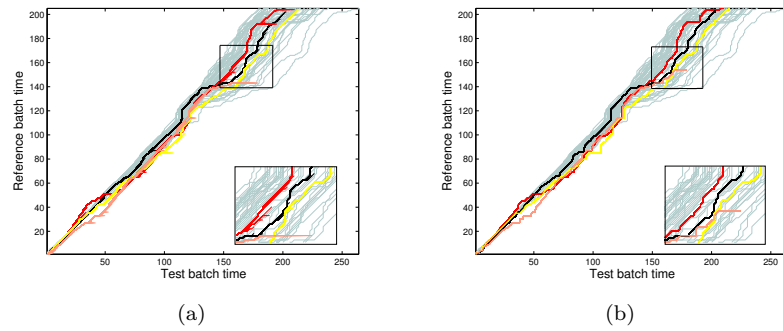


Figure 5.10. Suboptimal paths from a test batch (black path) and three abnormal batches (red, yellow and pink path) calculated by: (a) applying the Kassidas et al.'s online DTW implementation, and (b) applying the RGTW algorithm with a window width γ equal to 3 units.

weight procedure due to the features of this process.

Once all batches are synchronized by using the DTW and RGTW algorithms, they are arranged together with the warping information into two $50 \times (11 \times 205)$ [batches \times (variables \times time)] batch-wise unfolded two-way arrays, one array for each synchronization method. Multivariate analyses based on PCA on both data sets did not show outliers. Three principal components that explain 56% and 57% of the total variation of the data set synchronized by the DTW and RGTW algorithms, respectively, are selected. Hotelling- T^2 and SPE-based control charts for both data sets are designed and their control limits are initially estimated from theoretical results [110, 115]. Afterward, the control limits are adjusted by cross-validation to ensure that both monitoring schemes have the same *Overall Type I* (OTI) risk values for comparative purposes. Limits are raised or lowered based on the Hotelling- T^2 and SPE values obtained in the cross-validation so that the Overall Type I risk is equivalent to the imposed 95% and 99% significance level for both statistics and data sets.

5.4.2 Online synchronization

With the aim to evaluate the effect of the online synchronization approaches under study on process monitoring, the NOC and faulty test batches of the simulated example are synchronized and monitored

in real-time. Suboptimal paths from the NOC batch and the three abnormal batches assessed from the Kassidas et al.'s online approach and the online synchronization applying the RGTW algorithm are shown in Figures 5.10(a) and 5.10(b), respectively. Batches affected by an abnormality on ethanol, active cell material and acetaldehyde are plotted with yellow, pink and red lines, respectively. Kassidas et al.'s online DTW implementation needs to estimate the optimal path for each sampling time point to achieve the synchronization. As a consequence, the algorithm provides different synchronization sequences over time because of the correcting decisions made in the past. These actions can be appreciated from Figure 5.10(a), where many branches come out from the root path. The monitoring scheme then monitors the entire ongoing batch, from the starting up to the current sampling time point, yielding the Hotelling- T^2 and SPE statistics values for each sample. As a result of the alignment rectifications made by the Kassidas et al.'s online implementation, the values of these statistics may be beyond the control limits despite the process is running under normal operating conditions. Hence, the monitoring system would signal a false abnormal situation in the process. An example of this phenomenon can be seen in Figure 5.11(a), where the SPE values calculated over the run of a NOC batch (blue dots) exceed the control limits. Specifically, 21 samples exceed the control limits at 99% confidence level ($OTI = 9\%$), relatively far from the number of false alarms expected by chance ($OTI = 1\%$). This behavior is however not observed in the RGTW algorithm due to the fact that this algorithm uses a greedy strategy that provides a synchronized sample after the first γ steps. All optimal paths calculated for all synchronized batches are smooth, in contrast to Kassidas et al.'s online implementation (see Figures 5.10(a) and 5.10(b)). The effect of this synchronization improvement is the reduction of false alarms in the control charts, as appreciated from Figure 5.11(b), where only 5 NOC samples were detected as abnormal ($OTI=2.4\%$).

In this simulated example the RGTW algorithm outperforms the online DTW algorithm in terms of false alarms. Note also that the suboptimal solution of the online RGTW algorithm found during the evolution of the batch is very similar in terms of false alarms and magnitude of the SPE values to the optimal solution of Kassidas et al.'s online implementation calculated at the end of the batch (see black dots in Figures 5.11(a) and 5.11(b)). Hence, the online RGTW algorithm provides a suboptimal solution very close to the optimal solution that shows fewer false alarms during the evolution of the

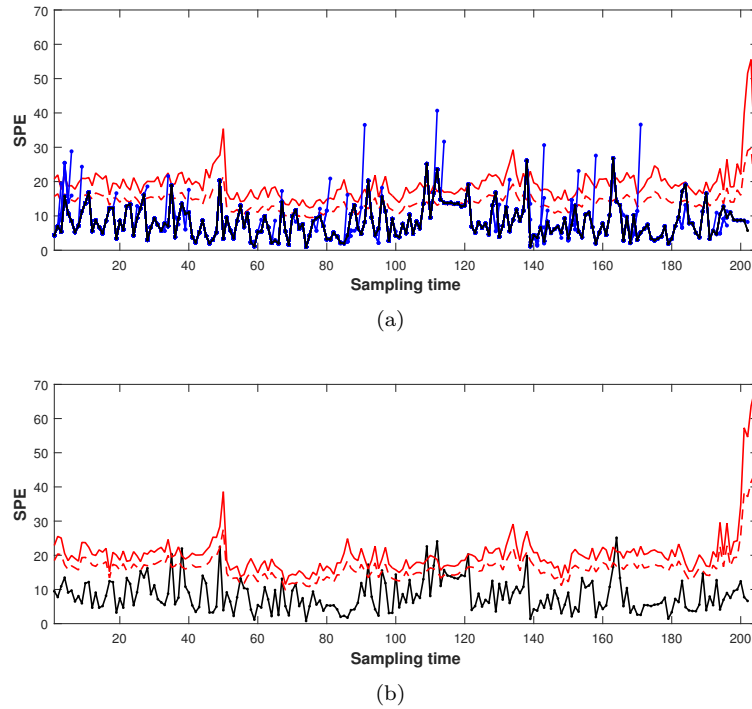


Figure 5.11. SPE control chart monitoring a NOC batch after applying the Kassidas et al.'s online DTW implementation (a) and the RGTW algorithm (b) in each sampling time point. Blue dots represent SPE values estimated at the k -th sampling time point and black dots SPE values calculated at the end of the batch after synchronization in (a). Dash and solid lines denote the control limits adjusted by cross-validation at 95% and 99% confidence level, respectively.

batch.

Another issue to discuss is the role of the warping information in the monitoring scheme. The result of the synchronization on the abnormal batches is different from the one on NOC batches (see Figure 5.10). From the 25th to the 60th and from the 70th to 90th sampling time point a different pattern is fingerprinted in the warping information. During this time interval, there are more expansions (vertical paths) registered in the batch that presents an abnormality caused by a modification on the constant $k11$ (called abnormality $bk11$) (see red lines in Figure 5.10) than in NOC batches (see light

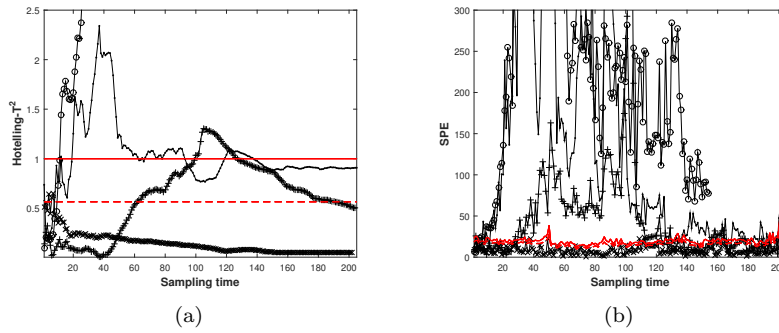


Figure 5.12. Hotelling- T^2 (a) and SPE (b) control chart monitoring a synchronized NOC batch (star line) and three faulty batches (abnormality simulated in the kinetic constants k_6 (cross line), k_{10} (circle line) and k_{11} (dot line)) from the starting point up to the end using the RGTW algorithm in each sampling time point. Cross-validated control limits for a 95% (dashed red line) and 99% (red solid line) confidence level are shown.

green lines in Figure 5.10). However, a different behavior is present in the abnormal batch that represents the degradation rate of the active cell material (see pink lines in Figure 5.10). Unlike the batch with b_{k11} abnormality, more compressions (horizontal paths) are shown from the 25th to the 60th sampling time point than in the NOC batches.

Figure 5.12 shows the result of the online monitoring of the NOC batch and three abnormal batches for the complete batch run. SPE and Hotelling- T^2 control charts are shown for all simulated batches with an abnormal behavior and under NOC. It can be seen that the SPE control charts correctly signals the abnormal batches from the very beginning of the batch run because each simulated fault breaks down the correlation structure from the starting point. In addition, the control charts show fewer false alarms for the batch data synchronized by the RGTW algorithm in comparison to those synchronized by the Kassidas et al.'s online DTW implementation.

In Figure 5.13, the SPE contributions for the three faulty batches at the 90th sampling time point are shown. These charts allow diagnosing the particular fault. For the simulated batch with an abnormal behavior on the formation of ethanol from acetaldehyde, the SPE contribution highlights the variables acetaldehyde (variable

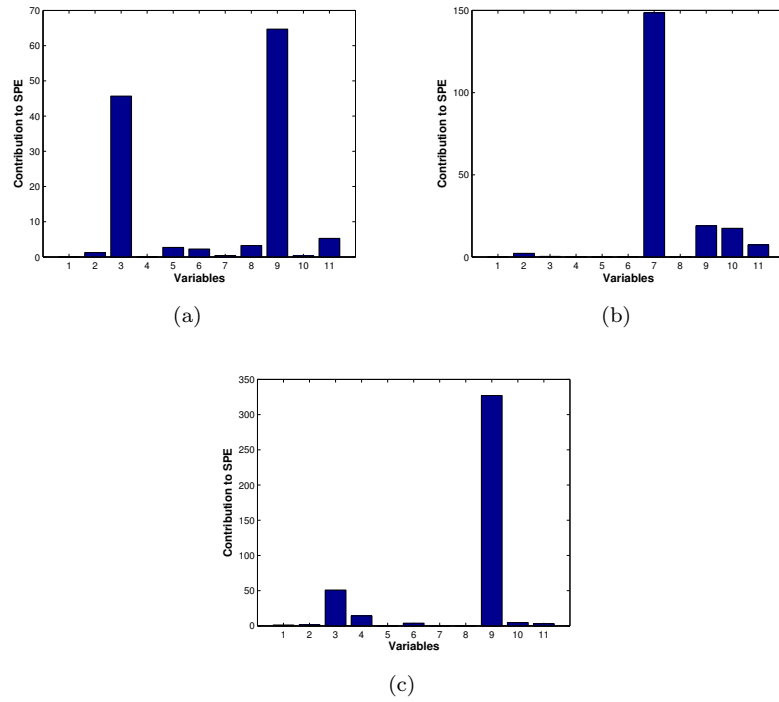


Figure 5.13. Variable contribution to SPE statistic for fault diagnostics of the abnormal batches at the 90th sampling time point: (a) an abnormal behaviour on the formation of ethanol from acetaldehyde, (b) degradation rate of the active compartment (dependent on the glucose and ethanol concentration), and (c) degradation of acetaldehyde dehydrogenase compartment.

#3), and specific oxygen uptake rate (variable #9) as responsible for the abnormality. Figure 5.13(b) represents the root causes of the degradation rate of the active compartment, mainly, the active cell material (variable #7) and CO_2 evolution rate (variable #10). Finally, the monitoring scheme diagnoses the specific oxygen uptake rate (variable #9), and to much less extent acetaldehyde (variable #3) and acetate (variable #4) as the variables involved in the abnormality of the third faulty batch (Figure 5.13(c)). Another aspect worth being stressed is that the contribution plots for the two first faulty batches give some weight to the warping information variable. This supports the claim that adding the warping information as new variable in the data matrix helps to detect an abnormal behavior in

the synchronization as long as the process duration is affected.

The novel RGTW algorithm, apart from reducing false alarms produced by the synchronization, provides computation benefits compared to Kassidas et al.'s online implementation. In order to demonstrate the efficiency of the new approach, both algorithms are applied 1000 times to the simulated NOC batches, measuring the runtime at each iteration. For this purpose, the system clock on an Intel Core Duo with 4GB of RAM memory with minimal background processes running is used. At the end of the computation experiment, the average of runtimes measured is assessed. The RGTW algorithm is run by using three different window widths ($\gamma=1,3,5$) together with the bands defined by the cross-validation procedure. In case of Kassidas et al.'s implementation, Sakoe-Shiba global constraint with a width equal to $M=59$ (the difference between the largest length of NOC batches and the reference batch length) is set. The evaluation yields that the RGTW algorithm is statistically significant faster than Kassidas et al.'s online DTW implementation for all the window widths tested. In particular, the RGTW algorithm requires on average 0.241, 0.251 and 0.262 seconds to synchronize the NOC batch with a window width γ equal to 1, 3 and 5 units, respectively. In contrast, Kassidas et al.'s online implementation takes 1.935 seconds on average. Thus, the maximum speed-up factor for this batch length is 8. An ANOVA is performed on the run times obtained for each approach, showing statistical significant differences (p -value < 0.05) between RGTW and Kassidas et al.'s approach. It is worth noting that the larger the data set in terms of sampling time points, the greater the differences on execution time. The outstanding performance of RGTW relies on the modifications presented. Firstly, using the new global band constraint definition, the synchronization procedure is accelerated by a constant factor because less cells of the weight matrix \mathbf{W} need to be evaluated. Secondly, the relaxed greedy strategy avoids calculating the suboptimal path for each sampling time point, reducing the optimization problem to the window ζ instead of the entire batch length. Thirdly, the relaxed greedy synchronization does not require the complete cumulative weighted local distance \mathbf{D} to be kept in memory, as in case of Kassidas et al.'s approach. It only needs to store the current and $\gamma - 1$ previous columns in memory because the RGTW algorithm finds the suboptimal path within the window ζ instead of over the ongoing batch time.

5.5 Use of warping information

This section is aimed to illustrate i) the use of the unsupervised NOC-WICC as a complementary tool to Hotelling- T^2 and SPE control charts for process monitoring, and ii) the use of classifiers based on PLSDA, SIMCA and faulty-WICC for supervised fault classification. For this purpose, the Set #3 is used, which contains the three-way arrays $\tilde{\mathbf{X}}_5$, $\tilde{\mathbf{X}}_6$ and $\tilde{\mathbf{X}}_7$ with 85 unsynchronized batches simulated under normal conditions, 44 faulty batches simulated by manipulating the nominal value of V_{max} (k_{11} parameter of the first principle-based model associated to the reaction describing the glucose uptake system and the glycolytic pathway), and 44 faulty batches simulated by modifying the kinetic constants k_6 (related to the reaction describing the formation of ethanol from acetaldehyde), respectively. For further information on the data sets, readers are referred to Chapter 2.

Batch data were split up into a calibration and test data set. 60 NOC batches and 20 batches from each of the simulated abnormalities were randomly selected to form the training data set, and the remaining batches were used to arrange the corresponding test data sets (25 NOC, 24 Faulty-1 and 24 Faulty-2 batches). Before the multivariate modeling is carried out, the synchronization of NOC batches must be performed. In order to establish the proper parameters of the RGTW algorithm, the cross-validation procedure explained in Section 5.2.3 was run.

Firstly, the basic parameters of the offline DTW algorithm were assessed. The batch whose duration is the closest to the median length of the historical NOC batches was selected as the reference batch. In this case, batch #30 was chosen with a duration of 193 sampling time points. As discussed in Section 5.4, the weights were assessed as the geometric average of the weights estimated from the approaches of Kassidas et al. [76] and Ramaker et al. [160]. Secondly, the proper width γ of window ζ was estimated. For this purpose, the performance of the RGTW algorithm varying γ between 1 and 5 units were studied. The RGTW-based synchronization using a window width equal to 3 units was finally selected and the bands were calculated based on the warping information derived from the latter synchronization.

Once the set of 60 calibration NOC batches were synchronized, the slices containing information of all process variables at the k -th sampling time points were arranged side by side in a two-way data array \mathbf{X} (60 batches \times (10 variables \times 193 sampling time points)). After

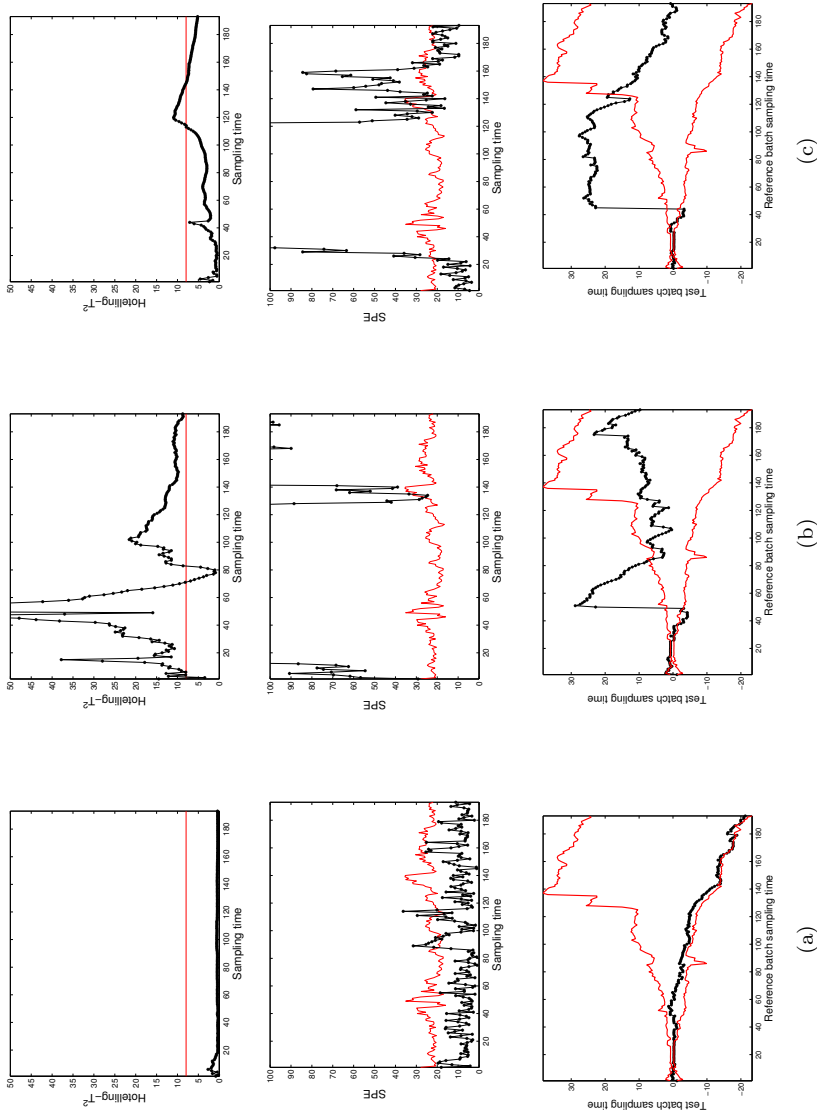


Figure 5.14. Hotelling- T^2 (first row), SPE (second row) and NOC-WICC (third row) monitoring a (a) NOC, (b) Faulty-1, and (c) Faulty-2 test batch. Cross-validated control limits for a 99% confidence level (solid red line) in the Hotelling- T^2 and SPE control charts are shown. Also, the upper and lower control limits of the NOC-WICC established at the percentile 0.5 and 99.5 are denoted by solid red line.

batch data were autoscaled, i.e. the mean trajectory was subtracted and all process variables at every sampling time point were scaled to unit variance, a PCA model was fitted. The selection of the optimum number of PCs was carried out based on the *PREdicted Residual Sum of Squares* (PRESS) function derived from the cross-validation procedure and the results obtained from the study of the performance of the PCA model using the OTI and *Overall Type II* (OTII)² risks [91, 90, 175].

For performance evaluation of the PCA model, the first ten PCs were taken into consideration in the study. For each of the PCs, a monitoring system was built by designing two multivariate Shewhart control charts based on Hotelling- T^2 and SPE statistics. Their control limits were estimated from NOC process data and later readjusted using cross-validation techniques for an *Imposed Significance Level* (ISL). The NOC and faulty test sets composed by 25 and 48 batches (24 batches for each of the two abnormalities), respectively, were projected onto the model and the OTI and OTII values for both statistics were calculated. Once the complete procedure was repeated for each of the PCs considered, the values of both indices as a function of the number of PCs were studied jointly with PRESS. In this example, two principal components were finally extracted since the corresponding model had a better performance in relation to the aforementioned parameters.

5.5.1 NOC-WICC for process monitoring

The warping information obtained from the RGTW-based synchronization of the batch trajectories corresponding to the set of 60 calibration NOC batches were used to build the NOC-WICC introduced in Section 5.3.1, where an upper and lower control limit established at percentile 0.5 and 99.5, respectively.

Three different batches (one NOC, one Faulty-1, and one Faulty-2 batch) were randomly chosen from the test data set to be synchronized and monitored, yielding the warping information, Hotelling- T^2 and SPE statistics over the batch run (see Figure 5.14). In the case of the NOC test batch, no clear out-of-control signal is detected in any of the three control charts (see Figure 5.14(a)). Regarding the pseudo-online monitoring results of the two different faults, the

²The OTII values are calculated by following $OTII = 100 \cdot \frac{nnf}{N_{ab} \cdot k} \%$, where nnf is the number of non-detected faults, N_{ab} is the number of faulty batches and k is the length of the faulty period.

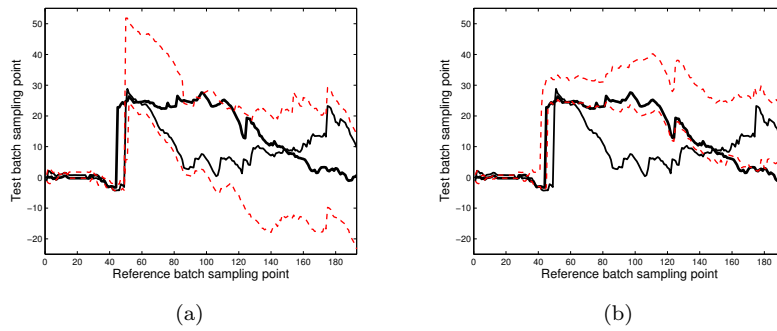


Figure 5.15. Faulty-1(a) and Faulty-2 (b) WICC with control limits defined at percentile 0.5 and 99.5 (dashed lines) are shown. Warping information belonging to Faulty-1 (medium line-width) and Faulty-2 (thick line-width) batches is plotted.

monitoring system has correctly detected the abnormalities through the SPE control charts (see Figures 5.14(b) and 5.14(c)). It is worth noting that the statistics-based control charts detect the fault earlier than the warping information-based control charts, in particular in the case of the Faulty-1 batch. Nonetheless, the use of the latter control chart provides a good insight into the process performance as well as a valuable complementary tool for fault classification. This will be explained in Section 5.5.2. The remaining test batches also showed the same behavior as the three selected batches in Figure 5.14 (results not shown).

5.5.2 Supervised faulty WICC

Using the 20 batches for each abnormality belonging to the training data set, the faulty WICC were built by following the procedure explained in Section 5.3.2. Differences among the various classes can be found by looking at the control limits of the faulty WICCs shown in Figure 5.15. From the control limits of the Faulty-1 WICC (Figure 5.15(a)) and NOC WICC (Figure 5.14, bottom), one can observe that these batches required more time than the NOC batches to reach the stage limited from the 45th to the 50th reference time sampling time point. In particular, a large amount of vertical transitions are shown at this time interval; therefore the batch trajectories were compressed by the RGTW algorithm to synchronize the process events. From the

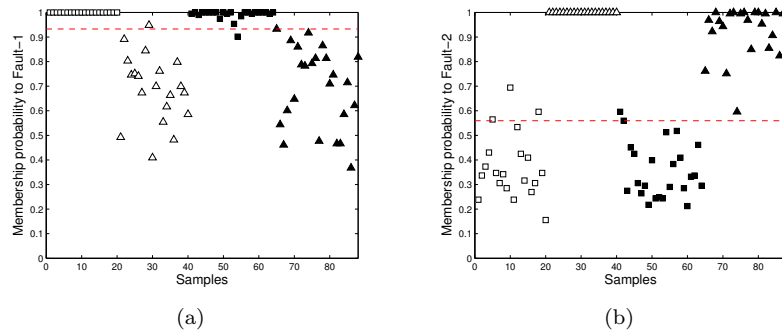


Figure 5.16. Classification of warping profiles for the Fault-1 (a) and Fault-2 (b) classes using faulty WICC. Faulty-1 (empty squares) and Faulty-2 batches (empty triangles) corresponding to the training data set are shown. Filled squares and triangles denote the Faulty-1 and Faulty-2 batches from the test data set, respectively. The dashed lines are the thresholds that yielded the highest MCC on the corresponding faulty classifier. Recall that only the batches corresponding to the test data set (those represented with filled symbols) were used to assess the classification thresholds.

50th to the 120th sampling time point, a larger number of horizontal than vertical transitions are shown. Consequently, Faulty-1 batches needed less time to reach the end of such process stage with respect to NOC batches. Regarding the Faulty-2 batches, the control limits of the Faulty-2 WICC (see Figure 5.15(b)) show that from the 40th to 45th reference sampling time point, the RGTW algorithm compressed the batch trajectories, yielding to vertical transitions. Again, the first process stage lasted longer than NOC batches. In contrast to the Faulty-1 batches, the Faulty-2 batches showed a similar behavior as NOC batches from the 50th reference sampling time point onward.

To illustrate the performance of the control charts proposed for end-of-batch fault classification, only the two faulty batches selected in Section 5.5.1 (one Faulty-1 and one Faulty-2 batch) were plotted in the Faulty-1 and Faulty-2 WICCs, respectively (see Figure 5.15). The *Membership Probability* (MP) of the two selected faulty test batches to fault-1 and fault-2 class was calculated. In the case of the Faulty-1 test batch, the corresponding warping information (medium line-width) falls fully inside the control limits of the Faulty-1 WICC (193 out of 193 points), yielding a $MP_{l=1} = 100\%$ (see Figure 5.15(a)). For the Faulty-2 test batch the $MP_{l=1} = 86.53\%$ (167 out of 193

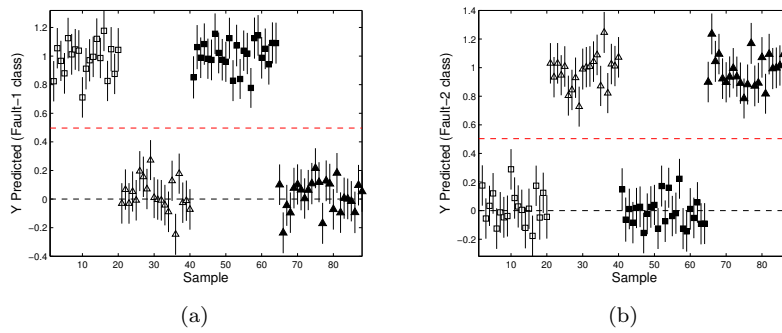


Figure 5.17. PLS-DA two latent variable model predictions for the Fault-1 (a) and Fault-2 (b) class. Faulty-1 (empty squares) and Faulty-2 batches (empty triangles) corresponding to the train data set are shown. Filled squares and triangles denote the Faulty-1 and Faulty-2 batches from the validation data set, respectively. The red dashed lines are the classification thresholds whose MCC value associated is the highest on the corresponding faulty data set. The bar of each batch represents the estimation error for each prediction or classification from the PLS-DA model. These intervals are calculated as the standard deviation of the measurement error in the reference values obtained by cross-validation [176].

points). Concerning the Faulty-2 WICC (see Figure 5.15(b)), 164 out of 193 points of the Faulty-2 batch were inside the control limits ($MP_{l=2} = 84.97\%$) while 100 out of 193 points belonging to the fault-1 batch falls within the control limits of the mentioned control chart ($MP_{l=2} = 51.81\%$) (see Figure 5.15(b)).

Following the above procedure, the warping profiles corresponding to the 24 test batches of each type of fault were used to estimate the membership probability to the known fault. Once these probabilities were obtained (see Figure 5.16), a threshold per faulty WICC was calculated by following the procedure explained in Section 5.3.2 (classification threshold for Fault-1 and Fault-2 class are 0.93 and 0.56, respectively). Note that these thresholds substantially differ from each other, mainly due to the different patterns found in the warping profiles of Faulty-1 and Faulty-2 batches. As can be appreciated in Figure 5.15(a), most of the points corresponding to the warping profile of the selected Faulty-2 batch fall inside the limits in the Faulty-1 WICC, except for the time intervals [8,14], [45,50], [86,98] and [110,117]. The rest of the Faulty-2 batches also showed the

same pattern in the Faulty-1 WICC (results not shown). Hence, the membership probabilities to Fault-1 class estimated for both Faulty-1 and Faulty-2 batches are expected to be high, being for the former slightly higher than for the latter. It causes that the classification threshold has a high value, close to 1. In contrast, a larger amount of points corresponding to the warping profile of the selected Faulty-1 batch falls outside the control limits in the Faulty-2 WICC, in particular, in the time intervals [45,50] and [60,143] (see Figure 5.15(b)). Again, this behavior is also observed in the rest of the Faulty-1 batches (results not shown). Hence, the membership probabilities to Fault-2 for Faulty-1 and Faulty-2 batches differ considerably, yielding to a lower threshold for classifier of Fault-2 class than for classifier of Fault-1 class.

Accuracy for Faulty-1 and Faulty-2 classes is measured by the area under the ROC curve (AUROC), leading to a value of 0.9911 and 0.9951, respectively, which indicates a good performance of the faulty WICC-based classifier. This is also illustrated in Figure 5.16(a) (membership probability to Fault-1 class) and Figure 5.16(b) (membership probability to Fault-2 class), where almost all the test warping profiles were correctly classified. Note that the membership probabilities belonging to the training data set are plotted for visualization purpose as well.

5.5.3 PLSDA-based classifier

Using the 20 batches for each abnormality belonging to the training data set, a PLSDA model was fitted. The resulting PLSDA cross-validated model yielded two latent variables, with R^2X , R^2Y and Q^2 values of 74.9%, 94.4% and 92.8%, respectively. The faulty batches from the test data set (24 batches for each of the faults) were projected onto the PLSDA model as external validation. All SPE values were inside the corresponding 99% confidence limits. The predictions for class fault-1 and class fault-2 models are shown in Figure 5.17. A threshold per each class was estimated by following the approach explained in Section 5.3.2 (classification threshold for Fault-1 and Fault-2 class are 0.49 and 0.50, respectively). At this point it is worth noting that these thresholds are almost equal. This is because in both cases, the prediction distribution generated for Fault-1 and Fault-2 class do not overlap each other, meaning that the classes are well separated. This interpretation can be observed in Figure 5.17, where all the test faulty batches were correctly classified, both in Faulty-1 and Faulty-2 class.

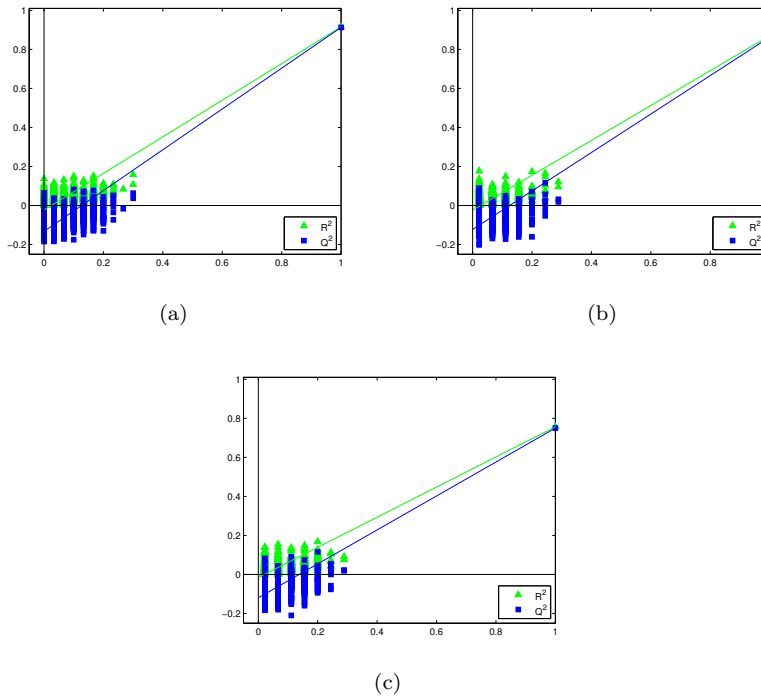


Figure 5.18. Validation plot obtained from the 900 permutation test for (a) NOC, (b) Faulty-1, and (c) Faulty-2 classes. The vertical axis gives the R^2 and Q^2 values for each model. The horizontal axis represents the correlation between the real response vector \mathbf{y}_c and the permuted one.

In order to check for model consistency, a random permutation test was performed to study the model consistency. The permutation test aims to compare both goodness of fit and goodness of prediction of the original model with the values estimated after class randomization [177]. Typically, a distribution of R^2Y and Q^2 values are obtained after performing the permutation test. In Figure 5.18 these distributions are shown jointly with the correlation coefficients between permuted and original response variables (classes). As appreciated, the R^2Y and Q^2 values found by using real class labels were clearly outside the distributions of those statistics found when employing random class labels, which indicates a clear distinction between the permuted-classification and the original classification. Also, the R^2Y

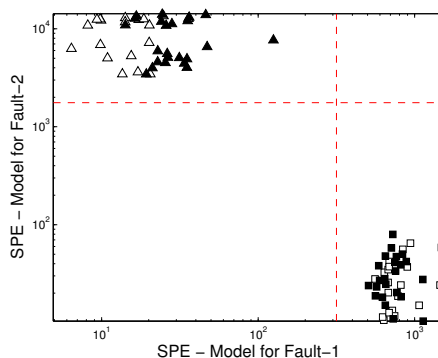


Figure 5.19. Cooman's plot for the Faulty-1 and Faulty-2 batches model. Faulty-1 (empty squares) and Faulty-2 (empty triangles) batches corresponding to the training data set are plotted. Filled squares and triangles denote the Faulty-1 and Faulty-2 batches from the test data set, respectively. The red dashed lines are the classification thresholds selected as those with the highest MCC.

and Q^2 values for permuted classes have low correlation between permuted and original classes. From these results, we can conclude that the PLS models are statistically significant ($P < 0.001$).

In this case, the AUROC value corresponding to the ROC curves for Faulty-1 and Faulty-2 class is 0.9946, showing similar performance than the faulty WICC-based classifier.

5.5.4 SIMCA-based classifier

A cross-validated PCA model was fitted from each of the aforementioned Faulty-1 and Faulty-2 training data sets. The first PCA model yielded 3 PCs, with R^2 and Q^2 values of 84.8% and 72.2%, respectively. The second one was defined by 4 PCs, a goodness of fit equal to 84.8% and a goodness of prediction equal to 67.6%.

The faulty batches from the test data set (24 batches for each fault) were projected onto the PCA models to classify them into their correct model. A Cooman's plot has been done (Figure 5.19), representing the distances of the different training and test sets to the Fault-1 and Fault-2 models. Once these distances were obtained, a threshold per PCA model was calculated by following the procedure explained in Section 5.3.2 (classification threshold for Fault-1 and Fault-2 PCA

model are 316.45 and 1763.3, respectively). Note that as occurred in the PLSDA-based classifier, both models are able to clearly distinguish Faulty-1 and Faulty-2 batches, for the training and test data sets, since their corresponding distances to the model of different fault classes are large.

Again, accuracy was measured estimating the area under the ROC curve designed for each one of classes. The AUROC value of the ROC curves associated to Faulty-1 and Faulty-2 class is 0.9946, denoting a good performance for classification.

5.6 Conclusions

This chapter addresses the problem of batch synchronization in Batch Multivariate Statistical Process Control in both post-batch and real-time applications. The drawbacks of well-known synchronization methods for batch synchronization are pointed out and the need of a new strategy to overcome the underlying setbacks is stressed.

In real-time synchronization there is a key requirement that must be met for the success of the monitoring system: the accuracy of the algorithm for synchronizing the key process events. If the trajectories are not properly aligned as in the offline synchronization, the resulting variables might show either slight or abrupt changes that would not correspond to the actual phenomena occurring in the process. These modifications in the trajectories would most likely cause artificial deviations around the trajectory, and hence, potential false alarms in the control charts. However, the achievement of an optimal solution in real-time becomes very challenging due to the fact that the end of the batch is unknown. The Kassidas et al.'s online implementation finds an optimal solution from the start of the ongoing batch until the current sampling time point, which is eventually suboptimal in comparison to the solution achieved at the end of the batch. The execution of this algorithm might produce slight artificial deviations at each time a new measurement vector is available, which might be enough to produce more false alarms than expected in the control charts, as illustrated in the example created with realistic simulated data.

With the aim of overcoming the shortcomings of the Kassidas et al.'s online implementation, a new time warping algorithm based on a relaxed greedy strategy and the original DTW of Kassidas et al.'s approach, called Relaxed-Greedy Time Warping (RGTW), is proposed. The new proposal avoids assessing the optimal path each

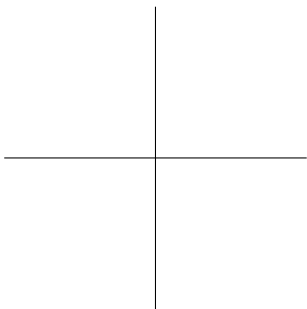
time a new sample is available, increasing the computational speed. Furthermore, it reduces the uncertainty in the monitoring statistics and predictions in such a way that false alarms are reduced, as shown in the simulated example. The main contributions of the RGTW approach are: (1) a new global band constraint definition that takes into account the variability across all batches in each time period; (2) a definition of a cross-validated monitoring window in order to use an optimized window size yielding the minimum local optimal paths (a relaxed greedy strategy); and (3) a way to adapt mapping boundaries to batch length. This last point is highly critical because in the online approach the endpoint of the ongoing batch is unknown, therefore, normal boundaries must be adapted by the current batch run. The disadvantage of this method in comparison to the Kassidas et al.'s online implementation is that the start of the monitoring is delayed as many sampling time points as sampling time points of width the optimized window has. Hence, the selection of the window width should be a trade-off between how optimal at least the solution should be and the maximum delay acceptable to start monitoring the process. In addition, the RGTW algorithm is sensitive to changes in the operating regimes or to external disturbances affecting the duration of the batches. In this situation, the RGTW parameters should be re-estimated by using new batches affected by the disturbances. Otherwise, the algorithm might produce inaccurate results that might affect the performance of the monitoring system.

Additionally, the use of the warping information obtained from the RGTW-based batch synchronization both for batch process monitoring and supervised fault classification is addressed. An unsupervised control chart based on the warping profiles from NOC batches (NOC-WICC) is proposed as a complementary tool to the Hotelling- T^2 and SPE control charts for post-batch and real-time batch process monitoring. In case that process faults are fingerprinted in the warping profiles, this chart can be useful to detect their occurrence in the process. Nevertheless, the NOC-WICC may not considerably improve the performance of the traditional multivariate Shewhart control charts. This improvement is subject to different factors, such as the nature of the process or the influence of the fault in the process phases, among others. For subtle change detection (ramps, small step changes, etc.), memory control charts, such as EWMA or CUSUM, should be used.

When a rich faulty database is available, warping information can be used to build the so-called supervised warping information-based control charts (faulty WICC) or to fit classification models using

supervised chemometric tools. Although in this chapter simple and widely used tools such as PLSDA and SIMCA are used, other classification techniques, such as *Support Vector Machines* (SVM) [178], LDA and QDA [179], and *K-Nearest Neighbors* (KNN) [180], could be taken into consideration, among others. In this chapter, the three approaches studied showed good classification performance in terms of the area under the ROC curve (the so-called AUROC). The use of the faulty-WICC-based classifiers depends much on the type of fault; whether faults have characteristic fingerprints in their corresponding warping profiles at specific time periods that are different from the rest. The more different the warping profiles from faulty batches, the better the accuracy of the classifier. In contrast, PLSDA and SIMCA-based classifiers are more accurate in fault classification when no clear differences among warping profiles are found.

In this study, using the warping profiles derived from the RGTW-based synchronization has been sufficient to design the classifiers with good prediction performance. In cases that the warping information belonging to different faulty batches does not show clear different patterns, it is suggested to use the raw batch trajectories jointly with the warping profiles for fault classification.



Batch synchronization in scenarios of multiple asynchronisms

Part of the content of this chapter has been included in the following publications:

- [4] J.M. González-Martínez, O.E. de Noord and A. Ferrer. Multi-synchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms, *Journal of Chemometrics*, 28(5): 462-475, 2014.
- [12] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 3: Batch Process Data. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [26] O.E. de Noord and J.M. González-Martínez. Recent developments in multivariate data analysis and monitoring of chemical manufacturing processes. *In proceedings of the 2nd African-European Conference on Chemometrics (AFRODATA 2012)*, page , Stellenbosch (South Africa), 2012.
- [28] J.M. González-Martínez, O.E. de Noord and A. Ferrer. A novel approach for batch synchronization in scenarios of multiple asynchronisms. In: *Proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 28, Djurönäset (Sweden), 2013.

6.1 Introduction

The presence of multiple asynchronisms in batch trajectories makes synchronization a challenging modeling step since the inappropriate manipulation of variable trajectories may considerably affect the interpretation of subsequent statistical models and jeopardize the performance of the monitoring scheme. This chapter addresses the multivariate synchronization problem prior to the model building phase with the objective of developing an online monitoring system.

The design of a monitoring scheme based on multivariate statistical methods requires a data set of NOC batches. Typically, principal technologists and experts in charge of the plant production select a time period in which the quality of the released product was on specification. However, this selection does not prevent the data set from containing either inadvertent faulty batches caused by reliability or safety issues, or incomplete batches. Despite these inadvertent abnormalities might be solved later in the batch run, the trajectories would most likely show a different pattern in comparison to NOC trajectories. In this situation, the affected batches should be discarded for synchronization. Otherwise, these batches may affect the accuracy of the parameter estimates and the quality of synchronization, and therefore, the results of the bilinear modeling.

The presence of multiple asynchronisms in the batch trajectories also poses a threat to bilinear modeling. Four different types of asynchronism can be found: i) batches with equal duration but key process events not overlapping at the same sampling time point in all batches (class I asynchronism), ii) batches with different duration and process pace (class II asynchronism), iii) batches with different duration due to incompleteness of the last process stage, irrespective of whether the process pace is the same or not across batches (class III asynchronism); and iv) batches with different duration due to delay in the start but batch trajectories showing the same evolution pace after (class IV asynchronism). Traditional synchronization approaches cope with asynchronism issues by applying a single synchronization procedure to all batches without considering the nature of asynchronism of each batch. In a context of multiple asynchronisms, applying the same synchronization procedure may distort the original trajectories and decrease the signal-to-noise ratio, adversely affecting the original correlations of the process variables over time.

In this chapter, a novel synchronization approach named Multisynchro is proposed to deal with batches affected by inadvertent abnormalities and multiple asynchronisms. The new approach uses the

valuable information on the process pace of each batch derived from DTW/RGTW-based synchronization (the so-called warping information) for two purposes: i) detecting the type of asynchronism of each particular batch, and ii) deploying the appropriate synchronization procedure based on the nature of asynchronisms. The new approach also includes a procedure that performs abnormality detection and batch synchronization on non-shifted and complete batches in an iterative way. In addition, a discussion on the consequences of common practices in synchronization of complex scenarios of asynchronisms is provided.

The outline of the chapter is as follows. In Section 6.2, the core of the novel Multisynchro approach is introduced. An optimization of the DTW and RGTW algorithms that deals with the presence of abnormalities to enhance the synchronization quality is presented in Section 6.3. This enhanced procedure is the core of the Multisynchro algorithm that allows the optimization of the synchronization parameters without them being affected by anomalies. The potential failure modes of this novel synchronization algorithm is rigorously discussed in Section 6.4. Section 6.5 presents the material of the research work. Section 6.6 illustrates i) the performance of the novel Multisynchro approach for batch synchronization in scenarios of multiple asynchronisms and ii) the effect of inappropriate synchronization on the batch trajectories. Furthermore, a discussion on whether the batch synchronization should be a compulsory step in the bilinear modeling of batch processes is provided. Finally, some conclusions are drawn in Section 6.7.

6.2 Multisynchro approach for batch synchronization

The Multisynchro approach is devoted to synchronize the key process events ensuring the same evolution across batches, irrespective of the type of asynchronism present in batch data. The algorithm takes as inputs the three-way array arranging the calibration batches, the technique to weight the process variables and the strategy to select the reference batch. The procedure returns the synchronized batch data array and the warping time profiles that indicate how to warp the batch trajectories to make them synchronized.

The Multisynchro algorithm is composed of a high-level and low-level routine (see Figure 6.1). The high-level routine is aimed at recognizing the different types of asynchronous trajectories for the subsequent

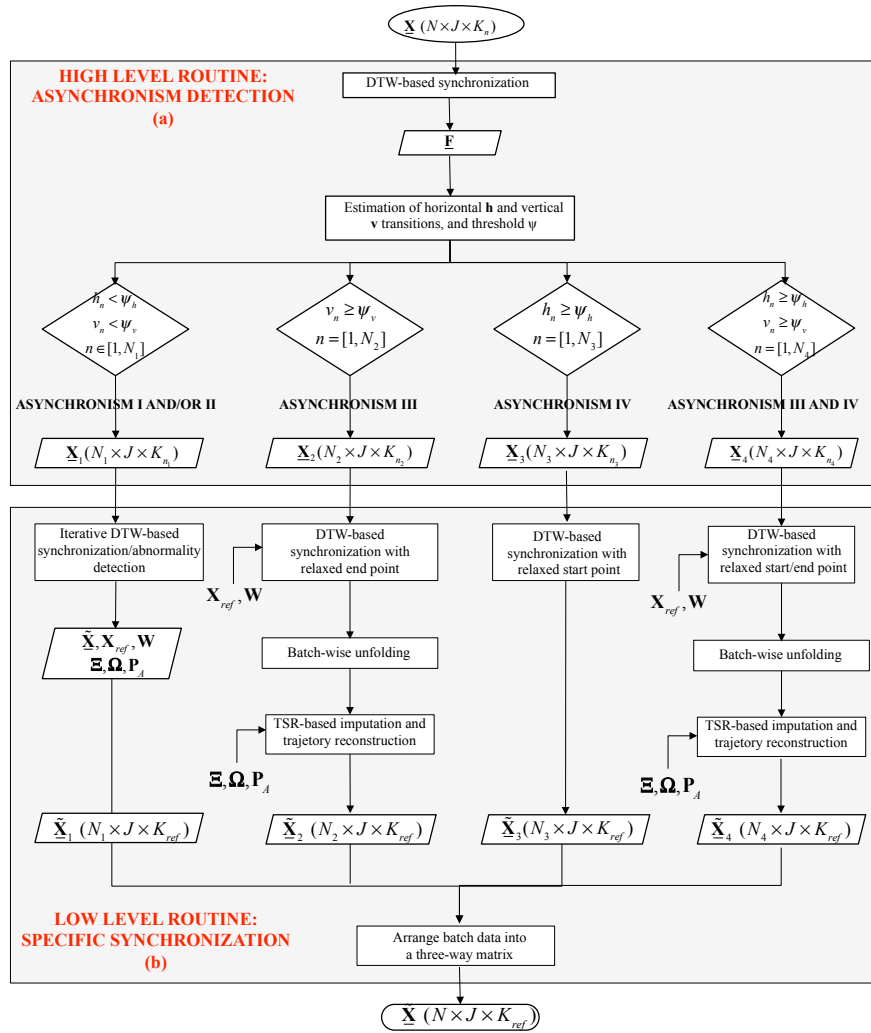


Figure 6.1. Flow diagram of the Multisynchro approach composed of the high-level (a) and low-level (b) routines for batch synchronization in scenarios of multiple asynchronisms.

batch classification as function of the nature of asynchronism (see Figure 6.1(a)). The low-level routine is in charge of synchronizing the variable trajectories of each one of the batches with a specific procedure based on the type of asynchronism (see Figure 6.1(b)). In the following, the algorithm is described.

6.2.1 Asynchronism detection

The high-level routine is divided into two steps (see Figure 6.1(a)). The first step is devoted to recognize the different types of asynchronous trajectories, which is carried out by using the warping time profiles derived from a preliminary synchronization as follows:

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

iii.1 Count the number of consecutive horizontal transitions at the first time period of the n -th batch, which denotes the number of compressions carried out by the synchronization algorithm at the start of the n -th batch:

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

iii.2 Count the number of consecutive vertical transitions at the last time period of the n -th batch, which denotes the number of expansions carried out by the synchronization algorithm at the end of the n -th batch:

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

The parameterization of patterns on the warping time profiles is used to detect the different types of asynchronisms that might be present in data. If there are batches affected by class I and/or class II asynchronism, the resulting warping profiles are expected to show a reasonable combination of horizontal and vertical transitions throughout the batch run. Batches that are influenced by class III asynchronism are associated with warping profiles containing an excessive number of vertical transitions at the last time period of the runs. Batches shifted at the start of the run will show a high number of horizontal transitions at the start of the batch on their corresponding warping profiles, assuming that the reference batch is selected from the set of non-shifted batches (asynchronism IV).

The second step of the high-level routine (see Figure 6.1(a)) is aimed at classifying each batch by the type of asynchronism and arranging them into different data sets. For this purpose, the features of the warping time profiles obtained from the first step of the high level routine are used to distinguish among asynchronisms. To determine whether the number of transitions exceeds the normal limit from which a batch can be qualified being affected by class III and/or class IV asynchronism, and not by class I and/or II asynchronism, thresholds ψ_v and ψ_h are derived. These thresholds are calculated as a fraction κ of the interquartile range of both the vertical transitions at the last time period \mathbf{v} and the horizontal transitions at the first time period \mathbf{h} estimated for all the synchronized batches, respectively. Even though κ is a heuristic value dependent on the distribution of the transitions, it is recommended that κ does not exceed 0.5. Also, note that the interquartile range is used as a dispersion statistic due to its robustness to outliers and extreme values, whose breaking point is 50% of outliers in the data set.

i. Repeat for all batches:

i.1 If the number of compressions h_n and expansions v_n are less than their respective thresholds, arrange the n -th batch into the three-way array \mathbf{X}_1 , which contain batches affected by class I and II asynchronisms (see classes I and II in Table 6.1).

i.2 If only the number of expansions at the end of the batch v_n is greater than or equal to the threshold ψ_v , arrange the n -th raw batch into the three-way array $\underline{\mathbf{X}}_2$, which contain batches affected by class III asynchronism (see classes III in Table 6.1).

i.3 If only the number of compressions at the start of the batch h_n is greater than or equal to threshold ψ_h arrange the n -th raw batch into the three-way array $\underline{\mathbf{X}}_3$ by class IV asynchronism (see class IV in Table 6.1).

i.4 If the number of compressions h_n and expansions v_n are greater than or equal to their respective thresholds, arrange the n -th raw batch into the data matrix $\underline{\mathbf{X}}_4$ by class III and IV asynchronisms (see classes III and IV in Table 6.1).

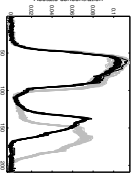
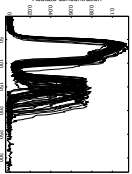
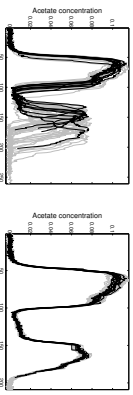
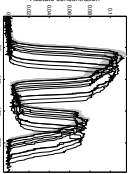
6.2.2 Specific batch synchronization

The Multisynchro approach continues the execution synchronizing the different data sets with different types of asynchronism (low level routine depicted in Figure 6.1(b)):

- i. Synchronize the three-way batch data array $\underline{\mathbf{X}}_1$ containing the batches whose trajectories are neither shifted nor incomplete in such a way that possible abnormalities present in batch data do not affect the synchronization quality. First, the iterative synchronization based on the DTW algorithm is applied to the whole data set. The isolation of the faulty synchronized batches is then conducted with a multi-way PCA model built on the assumed NOC synchronized batches. Once done, the NOC batches identified by the multi-way PCA with their raw variables trajectories are again synchronized with the iterative DTW algorithm. These steps are iteratively repeated until no more outliers are detected. For a rigorous description of this iterative batch synchronization/abnormalities detection procedure, readers are referred to Section 6.3.

The detection of suspicious batches in $\underline{\mathbf{X}}_1$ is crucial to obtain accurate parameters that are not affected by anomalies for the synchronization of the rest of data sets and for the imputation of missing trajectories. Note that if the batches of other data sets were synchronized with this procedure, the asynchronisms with which these batches are affected would harm the synchronization results. The DTW algorithm

Table 6.1. Case-based reasoning of the Multisynchro algorithm

Class	Asynchronism illustration	Description
I		Batches with equal duration but key process events not overlapping at the same sampling time point in all batches are classified as batches affected by class I asynchronism. Batches with data manipulated depicted in black lines will be classified as batches affected by class I asynchronism because no shifts are appreciated at the start of the batches ($h_n < \psi_h$) and completion of the last stage of batches is observed ($v_n < \psi_v$), assuming that the reference batch is selected from those batches that are not manipulated (grey batches in the left-hand figure).
II		Batches with different duration and process pace are classified as batches affected by class II asynchronism. Left hand side figure shows different process pace across batches, and as in the previous case, neither shifts ($h_n < \psi_h$) nor incompleteness ($v_n < \psi_v$) are observed. The difference between class I and II lies on the warping profiles: i) there will be more variation in the number of compressions and expansions over the batch run in class II than in class I, ii) existence of different matching end points in class II due to different length in all batches, which is a feature not observed in class I.
III		Batches with different duration due to partial incompleteness of the last process stage, irrespective of whether the pace is the same or not across batches, are classified as batches affected by class III asynchronism. Manipulated batches shown in black lines in the left and right-hand figures will be classified as class III asynchronism batches because $v_n \geq \psi_v$, as long as the reference batch is selected from the set of not manipulated batches (grey batches in the figures). Note, however, that their warping profiles will show differences since the process pace is different over all batches in the left-hand figure, in contrast to those depicted in the right-hand figure. Regarding the non-manipulated batches depicted by grey lines, those shown in the left-hand figure will be treated as class II due to the difference in pace. Those depicted in the right-hand figure might be treated as either class I or class II asynchronous batches, depending on whether these batches have different duration in comparison to the reference batch.
IV		Batches with different duration due to delay in the start but trajectories showing the same evolution pace after are classified as batches affected by class IV asynchronism. Manipulated batches depicted in black lines will be classified as batches affected by asynchronism IV because $h_n \geq \psi_h$, assuming the reference batch is one of the batches that are not manipulated (grey batches). Batches that are not manipulated (grey batches) will be classified as batches affected by class I and/or II asynchronous because $h_n < \psi_h$ and $v_n < \psi_v$ and the assignment will depend on whether the duration is the same or not in relation to the reference batch.

without relaxed end point constraints would create flat profiles when batches are incomplete at their last stage, or might add incorrect features at the start of the batch when batches are shifted. As a consequence, the weights of the process variables, the reference batch estimated as the average trajectory and ultimately the alignment of the process events would be severely perturbed. The poor synchronization quality would compromise the correct segregation of the batches in NOC and faulty, and eventually, the overall synchronization. This knock-on effect would inevitably cause the estimation of inaccurate and incorrect preprocessing parameters, and most importantly, the modification of the actual correlation structure.

At the end of the synchronization of $\underline{\mathbf{X}}_1$, the procedure returns the matrices of average trajectories $\underline{\Xi}$ and standard deviation trajectories $\underline{\Omega}$, the weighting matrix $\underline{\mathbf{W}}$, the loading vector $\underline{\mathbf{P}}_A$ obtained from the PCA-based modeling, and the three-way array $\underline{\tilde{\mathbf{X}}}$ that arranges the synchronized NOC and faulty batches.

- ii. Synchronize the three-way batch data array $\underline{\mathbf{X}}_2$ containing only incomplete batches in such a way that their last point is well synchronized with the best matching sampling time point of the reference. This type of synchronization is needed to prevent the incomplete batch trajectories from being misaligned and to avoid the addition of flat profiles that break the correlation structure. For this purpose, the DTW algorithm with the relaxed end point constraint using those parameters estimated in the iterative synchronization must be applied. This version of the DTW algorithm synchronizes batches against a segment of the reference batch limited by the first point and the best matching end point e^* instead of the reference as a whole. The algorithm returns the batch trajectories synchronized till the best end point of each batch $\underline{\tilde{\mathbf{X}}}_2$. The missing part of each batch is then imputed using the correlation structure captured by the multivariate model obtained from the iterative DTW-based synchronization/abnormality detection procedure through the application of the TSR method [169]. The procedure returns the three-way array $\underline{\tilde{\mathbf{X}}}_2$ containing the synchronized batch trajectories.
- iii. Synchronize the three-way batch data array $\underline{\mathbf{X}}_3$ using the DTW algorithm with the relaxed starting point constraint

and the parameters calculated in the iterative synchronization. This version of the DTW algorithm synchronizes segments of batches against a reference batch. The segments are limited by the best matching starting point s^* of each batch with the first point of the reference, and their last point. In other words, this procedure truncates the shifted batches to the best matching point with the reference batch to mitigate the risk that the iterative synchronization introduces artifacts when the first samples are shrunk or expanded. The presence of noise or even unexpected landmarks until the first best matching point with the reference batch might cause the addition of undesired features in the synchronized trajectories. The procedure returns the three-way array $\tilde{\mathbf{X}}_3$ containing the synchronized batch trajectories.

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

The application of different non-linear warping functions with different end point constraints might alter the covariance matrix if the asynchronisms are not treated in accordance with their nature. The synchronization of incomplete batches based on Multisynchro does not necessarily imply a modification of the covariance structure because the synchronization obtained is the optimal alignment from the start of the batch till the last sampling time point available. The remaining part is imputed based on the correlation structure of NOC data. If there is a breakage of the correlation structure, it might be most likely caused by an abnormal situation occurred over the batch. However, the inclusion of the warping information as a new variable in the data set might help to assess whether the breakage of the correlation is caused by the alignment itself or by an unexpected event in the process.

After synchronizing batch data using Multisynchro, all the resulting submatrices need to be merged into a three-way array $\tilde{\mathbf{X}}$ ($N \times J \times K_{ref}$) for subsequent bilinear process modeling. In addition, the warping profiles obtained in each one of the specific synchronizations are added as a new variable into the synchronized three-way array $\tilde{\mathbf{X}}$ for monitoring purpose. The inclusion of synchronized faulty batches in the calibration data set

does not mean that these batches will be considered in the final model for monitoring. If these batches are eventually abnormal, the model built in the next step of the modeling cycle will spot them as such, and hence, after investigation the batches will be eventually removed from the data set.

The real-time application of the Multisynchro approach is straightforwardly done by using the RGTW algorithm instead of the DTW algorithm. For offline applications, DTW is preferred since it provides us with the optimum global solution. However, if the main goal is to design a monitoring scheme for real-time application, the RGTW algorithm is required. For further details on its implementation, readers are referred to Chapter 5.

6.3 Iterative batch synchronization/abnormalities detection procedure

Batch synchronization needs to be implemented taking into account the possible presence of abnormalities in batch data. The existence of faulty batches in the calibration data set may yield inappropriate synchronizations since possible artifacts may be introduced due to abnormalities. For instance, batch trajectories that break the correlation structure usually contain different shapes in comparison to batch trajectories run under NOC. It may affect the estimation of the weight matrix \mathbf{W} and the synchronization quality, leading to synchronized batch trajectories with artificial shapes at different time periods. To overcome this problem, an iterative synchronization/abnormalities detection procedure is presented. The aim of this new procedure is to synchronize each batch with a reference batch and with each other in such a way that suspicious batches do not affect the synchronization quality. The main steps of the algorithm are (see Figure 6.2):

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

- ii. Preprocess batch data by trajectory centering and scaling using the estimated matrices of averages Ξ ($K_{ref} \times J$) and standard deviations Ω ($K_{ref} \times J$)¹, yielding the three-way data array $\tilde{\mathbf{X}}$.
- iii. Fit a PCA model on the batch-wise unfolded and preprocessed data matrix $\tilde{\mathbf{X}}$ satisfying the following equation: $\tilde{\mathbf{X}} = \mathbf{T}_A \cdot \mathbf{P}_A^T + \mathbf{E}$, where A is the number of PCs extracted².
- iv. Design a control chart based on the SPE statistic. Its control limit $SPE_{lim,\alpha}$ is estimated from the synchronized calibration batch data at $(1-\alpha)$ confidence limit.
- v. Offline post-batch monitor all the synchronized calibration batches for fault detection.
 - v.1 Compute the SPE statistic for each batch and sort out the corresponding values in ascending order.
 - v.2 Calculate the acceptable number of batches R that can exceed the control limits at $(1-\alpha)$ confidence level by chance as α times the number of calibration batches.
 - v.3 If the number of batches exceeding $SPE_{lim,\alpha}$ I_f is greater than R , the first $B_l = I_f - R$ synchronized batches with the highest SPE values are treated as faulty batches. In any case those batches whose SPE values are beyond λ times $SPE_{lim,\alpha}$ are also considered as faulty. To isolate these faulty batches for subsequent synchronization different from that performed on NOC batches, arrange them into the three-way array $\tilde{\mathbf{X}}_B$ ($B_l \times J \times K_{ref}$), recover their raw trajectories and add them to the three-way array \mathbf{X}_B ($B_L \times J \times K_b$), which contains the rest of raw faulty batches isolated in previous iterations.
 - v.4 The remaining batches are considered as NOC and their trajectories are arranged into the three-way array $\tilde{\mathbf{X}}_C$ ($G \times J \times K_{ref}$).
- vi. If one or more batches were detected as abnormal in the offline post-batch monitoring at the l -th iteration, compute the repeat loop (i)-(v) with the new calibration batch data

¹This preprocessing approach is selected due to suitability for batch process modeling and monitoring [3, 79].

²The interest of building a PCA is to design a monitoring scheme for fault detection. In process monitoring, the interest is in the distributions of latent variables and residuals, which are those used to estimate the control limits for incoming data. This should be taken into consideration to select A [181].

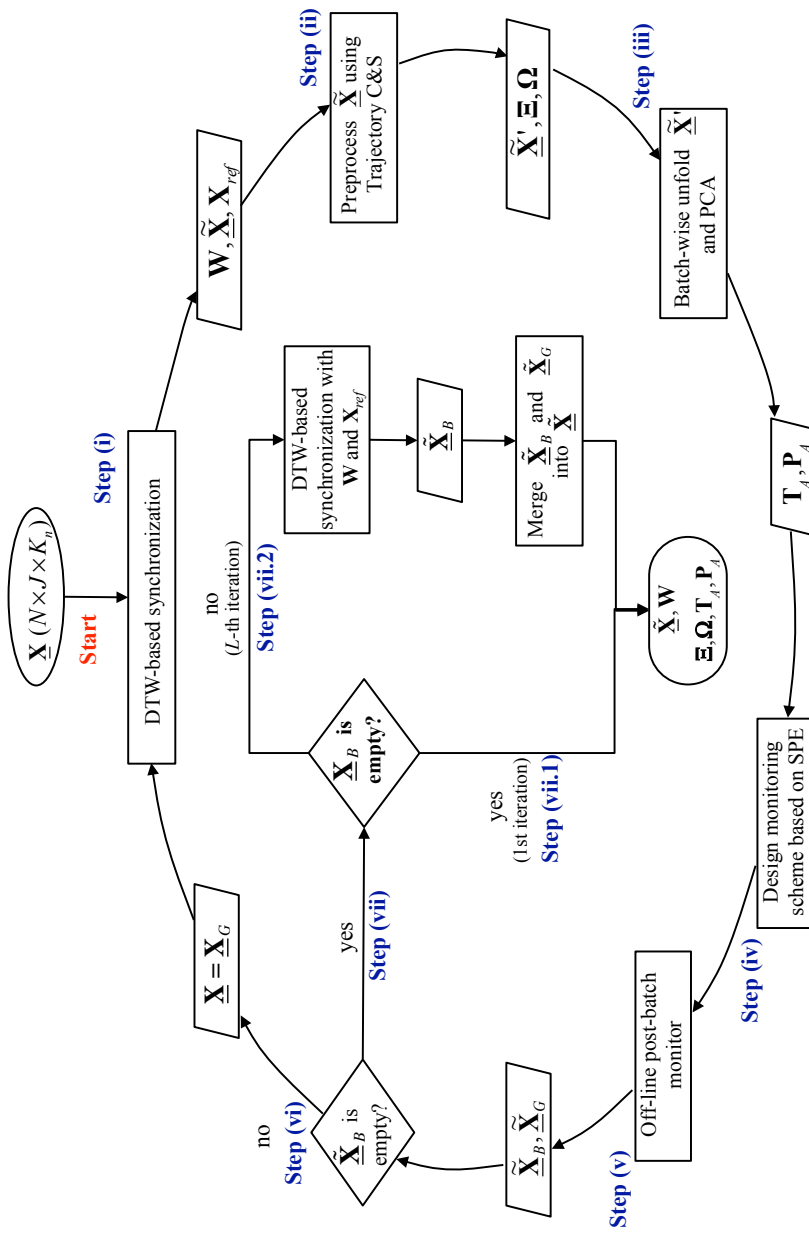


Figure 6.2. Flow diagram of the iterative batch synchronization/abnormalities detection procedure. Note that $\tilde{\mathbf{X}}_B$ is the three-way array containing the synchronized faulty batches isolated at the l -th iteration whereas \mathbf{X}_B is the three-way containing all the raw faulty batches isolated in the L iterations of the iterative procedure.

array $\underline{\mathbf{X}} = \underline{\mathbf{X}}_G$, where $\underline{\mathbf{X}}_G$ is a $(G \times J \times K_g)$ three-way array containing the raw batch trajectories of G NOC batches.

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

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

vii.2 If some batches were detected as abnormal in the offline post-batch monitoring after L iterations, synchronize each faulty batch \mathbf{X}_b from the three-way faulty batch array $\underline{\mathbf{X}}_B$. For this purpose, the DTW algorithm is applied using the reference batch \mathbf{X}_{ref} and the weighting matrix \mathbf{W} that were assessed in the NOC batch synchronization in step (i) at the last iteration. Once the synchronized three-way faulty batch array $\tilde{\underline{\mathbf{X}}}_B$ is available, merge it with the three-way array of NOC batches $\tilde{\underline{\mathbf{X}}}_G$ into the three-way array $\tilde{\underline{\mathbf{X}}}$.

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

As a general rule, the results of the synchronization must be thoroughly examined by a domain expert to ensure that the synchronization algorithm does not contain distorted data. For this purpose, the raw and synchronized NOC batches and the raw and synchronized faulty batches should be visually inspected.

6.4 Failure modes and complex non-linear interactions

The Multisynchro algorithm, as well as the DTW and RGTW algorithms, use non-linear warping functions to synchronize data, occasionally with relaxed constraints to treat specific asynchronisms. Depending on how diverse and distributed the asynchronisms are throughout the batches, and how these non-linear functions are applied to batches, the Multisynchro algorithm

might not perform as expected. In the following, the failure modes of the categorization of the batches based on the type of asynchronism and the failure modes of the iterative batch synchronization/abnormalities detection procedure are discussed. A reflection on the complex and non-linear interaction between DTW, PCA and SPE based segregation of NOC and faulty batches is also provided.

6.4.1 Failure modes of the asynchronism categorization

Caution should be taken when the categorization of the batches is done based on the warping information derived from the unconstrained synchronization. Similar features spaced in time might be matched incorrectly because the search of the solution is not constrained. Upon completion of the high level routine, the practitioners should carefully look through the warping profiles to ensure that this phenomenon did not occur in the preliminary synchronization (an example of a visual exploration of the warping profiles can be found in Section 10.5.2). Otherwise, the categorization of the batches would be at risk as well as the final results of the Multisynchro synchronization. If misalignments have been produced, a new synchronization should be performed by constraining the search of the solution with a band created with the information of the warping profiles. For a discussion on this subject, the reader is referred to Section 5.2.1.

A strong assumption of the Multisynchro algorithm is the presence of batches in the calibration data set that are completed and not shifted, irrespective of whether the batches have a similar or different pace. The absence of this type of batches would prevent the algorithm from continuing because the estimation of the synchronization parameters to deal with the different types of asynchronisms cannot be performed. In this situation, practitioners should revisit the batch trajectories and the warping profiles derived from the preliminary synchronization, with the support of experts in the process if needed, to figure out the reason why all the retrieved batches are shifted, uncompleted, or affected by a combination of these asynchronisms.

When all the batches are affected by a shift at the start of the batch, a new data set should be created where the variable trajectories are truncated at the point that matches with the first sampling time point of the reference batch. To determine

this matching point across batches, the warping profile should be investigated. Once the new data set is built, the higher level routine should be re-evaluated again.

In case that all the batches are incomplete in a relatively small part of the last stage of the process, practitioners should verify whether the missing trajectory forms part of the actual processing of the raw materials to release the product, or whether it is just a simple transition period to switch to the next batch. If the missing part is critical for manufacturing the product, and assessing its quality and the process performance, the modeling should stop at this point because there are no good batches for analysis. If the truncation of the trajectories is only due to variation in the switch among batches, the trajectories might be truncated to the common point that can be obtained from the warping profiles of the preliminary synchronization. Note that this alternative can be used as long as truncation does not imply a loss of valuable information, which must be verified and confirmed by experts in the process. As a next step, the higher level routine should be re-evaluated again by using the data set containing the treated batches. Note that if all the batches are shifted and incomplete, the guidelines described above to separately tackle these two asynchronism cases apply to this scenario.

Another topic subjected to discussion is whether the thresholds ψ_v and ψ_h used for the segregation of batches by asynchronism should be re-calculated when the iterative batch synchronization/abnormalities detection procedure isolates the faulty from the NOC batches. First, the batches used in this synchronization iterative procedure are not affected by shifts at the start of the batch or by incompleteness of part of the last stage of the process. The high-level routine of Multisynchro ensures that the set of batches in which the iterative batch synchronization/abnormalities detection procedure is applied to is not affected by any of these asynchronisms. Hence, the compression and shrink of the trajectories at early and late stages of the process is not expected. Second, the presence of abnormalities in the batches does not necessarily affect the distribution of the number of horizontal and vertical transitions observed in the warping profiles at the start and end of the batches. Even though it might affect the distribution, the algorithm uses the interquartile range as a metric to estimate the threshold, which is robust to the presence of outliers in comparison to considering the whole distribution

created from all the transitions. Thus, the isolation of faulty batches in the iterative procedure does not create the need to re-estimate the thresholds.

6.4.2 Failure modes of the iterative batch synchronization/abnormalities detection procedure

The procedure to synchronize batch trajectories without abnormal batches affecting its result contains an iterative and non-linear warping procedure for synchronization (DTW). This procedure is coupled with an iterative and linear modeling procedure (PCA) that together drive another iterative procedure that is segregating the batches into NOC and faulty categories based on the SPE statistic. The presence of anomalies in the data set together with the complex and non-linear interaction between DTW, PCA and SPE might lead to an incorrect synchronization, and hence, to an inaccurate segregation of the batches undertaken by the multivariate model.

The existence of historical batches that exhibit operating problems during their progress may negatively influence the quality of synchronization. If a faulty batch is automatically selected as a reference to synchronize each batch with it, and all the batches with each other, the resulting trajectories would be incorrectly synchronized. The misalignment would be namely caused by the addition of non-linear features from the non-linear warping procedure that diverge from NOC landmarks. This effect would be even more harmful from a modeling perspective when the fault is characterized by non-linear abnormal variation that differs from the common systematic variation observed in NOC data. In this case, those batches ran under NOC would be recklessly converted into faulty batches. To avoid the occurrence of undesirable misalignments, the selection of the reference batch should rely on prior knowledge of the operating conditions in which the historical batches released products of good quality. If such information is not available beforehand, it is highly advisable to trace the actions that the iterative batch synchronization/abnormalities detection procedure takes. In the event of mistaken synchronization and segregation, the faulty batches should be removed from the calibration data set prior to execute again the iterative synchronization procedure.

Despite the calibration batches are synchronized with a NOC reference, the resulting warping profiles added to the calibration data matrix might also influence the PCA model, and hence, the SPE statistic used to segregate batches. It might happen that a certain batch took longer than normal to release an on-spec product, producing a warping profile that differs from normal warping profiles. If the warping variable is not properly scaled in the model, it might cause that this NOC batch is flagged as faulty, though the batch did not show any operating issue in the rest of process variables throughout the run. Hence, the variable containing the warping profiles should be weighted in such a way that the synchronization effects do not override variation diverging from common variation for fault detection.

Another issue to take into account when synchronizing is the common and unusual variation between phases across batches. When there is a systematic abnormal occurrence in a certain phase, the PCA model might pinpoint the whole batch as abnormal. However, the information of previous phases might be of major importance to optimally tune the synchronization parameters, and hence, to accurately synchronize the batch trajectories. In this case, the synchronization should be performed phase to phase to avoid the excessive loss of batches for the optimization of the synchronization parameters. Note that a phase-to-phase modeling should be performed in order to ensure consistency in the whole modeling cycle.

The batch-wise multi-way PCA modeling relies upon the assumption that data preprocessed by trajectory centering and scaling, and later on batch-wise unfolded, contain only linear relationships. However, the process data might show non-linear relationships due to reasons such as incorrect synchronization of batch data, shifting operation phases, various process conditions, different types of process faults, etc. Hence, the linearity assumption after synchronization and processing should be checked, otherwise the modeling based on the linear PCA approach would not be adequate. To overcome the problem of non-linearity, non-linear PCA [98] or the transformation of data using kernels should be conducted [182].

6.5 Material and methods

The three-way arrays $\tilde{\mathbf{X}}_8$ and $\tilde{\mathbf{X}}_9$ defined in Chapter 2 are used as a baseline. All the batches contained in these data sets are characterized by having class II asynchronous trajectories whose process events do not overlap and whose length differs across batches. Two extra data sets, affected by class I asynchronism that contain trajectories of equal duration but with pace slightly differing across batches, complete the baseline. To generate these two data sets, the three-way arrays $\tilde{\mathbf{X}}_8$ and $\tilde{\mathbf{X}}_9$ are suboptimally synchronized by using the DTW algorithm with relaxed constraints. Specifically, the process variables are equally weighted to get a non-optimized synchronization, where the key process events are not entirely aligned. The rest of conditions and constraints are set according to [76]. Batch #12 and #10 with $K_{ref} = 209$ sampling time points and different pace are selected as reference batches from the three-way arrays $\tilde{\mathbf{X}}_8$ and $\tilde{\mathbf{X}}_9$, respectively. As a result of this suboptimal synchronization, the three-way arrays $\tilde{\mathbf{X}}_8$ ($N_8 \times J \times K_{ref}$) and $\tilde{\mathbf{X}}_9$ ($N_9 \times J \times K_{ref}$) are derived. The four three-way arrays from the baseline are combined, and partially manipulated in some cases, to generate four different scenarios of multiple asynchronisms.

The first scenario is associated with one type of class III asynchronism and consists of a set of batch trajectories of different length caused by a partial incompleteness of the last stage of the batch run with key process events closely overlapping across batches (see Figure 6.3(a)). For the generation of this type of asynchronism, $N_8^{(1)} = 10$ suboptimally synchronized batches randomly selected from $\tilde{\mathbf{X}}_8$ are manipulated to have different length. Ten different end points are randomly generated and the batch trajectories corresponding to the $N_8^{(1)}$ batches are subsequently truncated to these points (see case #1 in Table 6.2). The remaining $N_8^{(2)} = 30$ suboptimally synchronized batches are arranged jointly with the $N_8^{(1)}$ incomplete batches into the three-way array $\mathbf{X}_{c\#1}$ ($N_{c1} \times J \times K_{n_{c1}}$).

In the second case of asynchronism corresponding to class IV, batches have different length due to delay in the measurement collection but their trajectories show the same evolution pace over all batches (see Figure 6.3(b)). To generate this type of asynchronism, the $N_8^{(1)}$ suboptimally synchronized batch

Table 6.2. Batch data composing the four data sets with different asynchronisms.

Case	Asynchronism	Batch data	Explanatory text
#1	Class I	$\tilde{\mathbf{X}}^{(2)} \subseteq \tilde{\mathbf{X}}_s, N_s^{(2)} = 30$	No data manipulation
	Class III	$\tilde{\mathbf{X}}^{(1)} \subseteq \tilde{\mathbf{X}}_s, N_s^{(1)} = 10$	Random shrinkage of batch duration: #184, #193, #183, #173, #168, #141, #156, #185, #192 and #184.
#2	Class I	$\tilde{\mathbf{X}}^{(2)} \subseteq \tilde{\mathbf{X}}_s, N_s^{(2)} = 30$	No data manipulation
	Class IV	$\tilde{\mathbf{X}}^{(1)} \subseteq \tilde{\mathbf{X}}_s, N_s^{(1)} = 10$	Random shift length for batches: #2, #5, #7, #11, #13, #14, #22, #23, #28 and #37. $\tilde{\mathbf{X}}^{(2)} \in N(\mu^{(2)}, \sigma^{(2)}),$ $\mu^{(2)} = \mu(\tilde{\mathbf{X}}_{-\xi,j}^{(2)}), \sigma^{(2)} = \sigma(\tilde{\mathbf{X}}_{-\xi,j}^{(2)}), \xi = 5$
#3	Class II	$\mathbf{X}^{(2)} \subseteq \mathbf{X}_s, N_s^{(2)} = 30$	No data manipulation
	Class III	$\mathbf{X}^{(1)} \subseteq \mathbf{X}_s, N_s^{(1)} = 10$	Random shrinkage of batch duration: #174, #171, #161, #158, #173, #128, #156, #169, #202 and #151.
#4	Class I	$\tilde{\mathbf{X}}^{(2)} \subseteq \tilde{\mathbf{X}}_9, N_9^{(1)} = 10$	No data manipulation
		$\tilde{\mathbf{X}}^{(1)} \subseteq \tilde{\mathbf{X}}_s, N_s^{(1)} = 30$	No data manipulation

trajectories are manipulated in the following way. Firstly, the duration of the delay for each batch $\phi_{n_s^{(1)}}$ is randomly generated from $k_{\phi_{n_s^{(1)}}} = 1$ to $k_{\phi_{n_s^{(1)}}} = 50$ sampling time points. Secondly, data for each one of the J process variables are generated by following a normal distribution with $\mu = \mu(\tilde{\mathbf{X}}_{-\xi,j}^{(2)})$ and $\sigma^{(2)} = \sigma(\tilde{\mathbf{X}}_{-\xi,j}^{(2)})$, where $\mu(\tilde{\mathbf{X}}_{-\xi,j}^{(2)})$ and $\sigma(\tilde{\mathbf{X}}_{-\xi,j}^{(2)})$ represent the average and the standard deviation calculated for the j -th process variable at the window of length $\xi = 5$ (*i.e.* the first sampling time points of the batch), respectively (see case #2 in Table 6.2). Finally, these measurements are added to each process variable and the resulting batch trajectories are arranged with the $N_s^{(2)}$ suboptimally synchronized batches into the three-way array $\underline{\mathbf{X}}_{c\#2} (N_{c2} \times J \times K_{n_{c2}})$.

In case #3, the variable trajectories show not only different duration due to incompleteness of the last stage but also the key process events do not overlap at the same batch time across batches, which are classified as batches affected by class III asynchronism (see Figure 6.3(c)). For the generation of these asynchronism patterns, the $N_s^{(1)}$ raw batch trajectories are manipulated. The cut points generated in case #1 in the domain

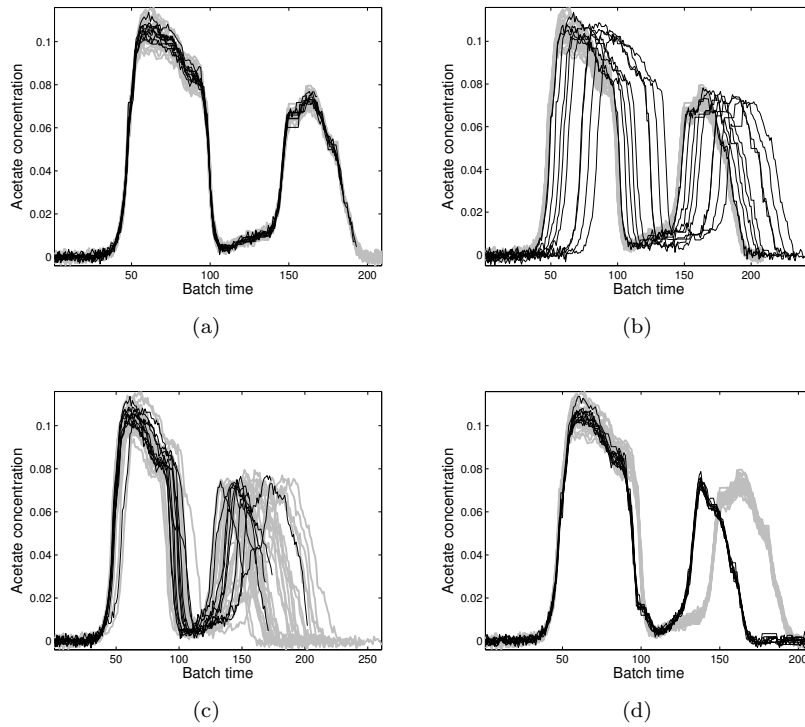


Figure 6.3. Trajectories of the acetate concentration corresponding to 40 NOC batches in four different scenarios of asynchronisms: (a) case #1: class I (grey lines) and class III (black lines) asynchronisms; (b) case #2: class I (grey lines) and class IV (black lines) asynchronisms; (c) case #3: class II (grey lines) and class III (black lines) asynchronisms; and (d) case #4: class I asynchronism (grey and black lines). The grey lines represent the 30 batches without data manipulation and the black lines represents the data manipulation described in Table 6.2 for each scenario.

of the synchronized time are chosen and their corresponding matching point in the actual batch time is reconstructed by using the warping information (see case #3 in Table 6.2). Afterwards, the $N_8^{(1)}$ raw batch trajectories are cut to these points. Finally, these batch data are arranged with the remaining $N_8^{(2)}$ raw batches into the three-way array $\underline{\mathbf{X}}_{c\#3}$ ($N_{c3} \times J \times K_{n_{c3}}$).

Concerning the fourth case of asynchronism categorized as class I, the batch trajectories have the same length but the evolution

pace is different among batches (see Figure 6.3(d)). For this case, the suboptimally synchronized batch trajectories from $\tilde{\mathbf{X}}_8$ and $\tilde{\mathbf{X}}_9$ are arranged into the three-way array $\mathbf{X}_{c\#4}$ ($N_{c4} \times J \times K_{ref}$)³. The resulting asynchronous batches for each case from Table 6.3 are depicted in the acetate concentration variable in Figure 6.3, where the grey lines are the 30 batches with no data manipulation and, the black lines are the 10 batches with the data manipulation.

6.6 Results

The objective of this section is to illustrate i) the performance of the novel Multisynchro approach for batch synchronization in scenarios of multiple asynchronisms and ii) the effect of inappropriate synchronization on the batch trajectories by comparing DTW and Multisynchro, and iii) discuss whether the synchronization of the batch trajectories should be always a mandatory step in the bilinear modeling of batch processes.

6.6.1 Performance of synchronization based on DTW and Multisynchro

Batches with four different types of asynchronism (see Table 6.2) are synchronized by using the Multisynchro approach. The high-level routine is executed for asynchronism detection. As a result of this step, a set of 40 warping profiles for each scenario of asynchronism is derived (see Figure 6.4). Looking at these profiles, in which every action taken by the synchronization algorithm is fingerprinted, insight into the nature of asynchronism present in batch data can be obtained.

In cases #1, #2 and #4, the warping profiles belonging to the 30 out of 40 batches (see grey lines in Figure 6.4(a), Figure 6.4(b) and Figure 6.4(d), respectively) almost follow the main diagonal. Note that these batches have equal duration and apparently their key process events overlap at the same sampling time points across batches (see grey lines in Figures 6.3(a), 6.3(b) and 6.3(d), respectively). Nonetheless, the slight deviations observed from the diagonal profile denote that even though most

³The simulated data manipulated to generate the asynchronisms under study are available in MVBatch software (at request) and on the website of Industrial & Engineering Chemistry Research, journal in where an extended version of this data set was published.

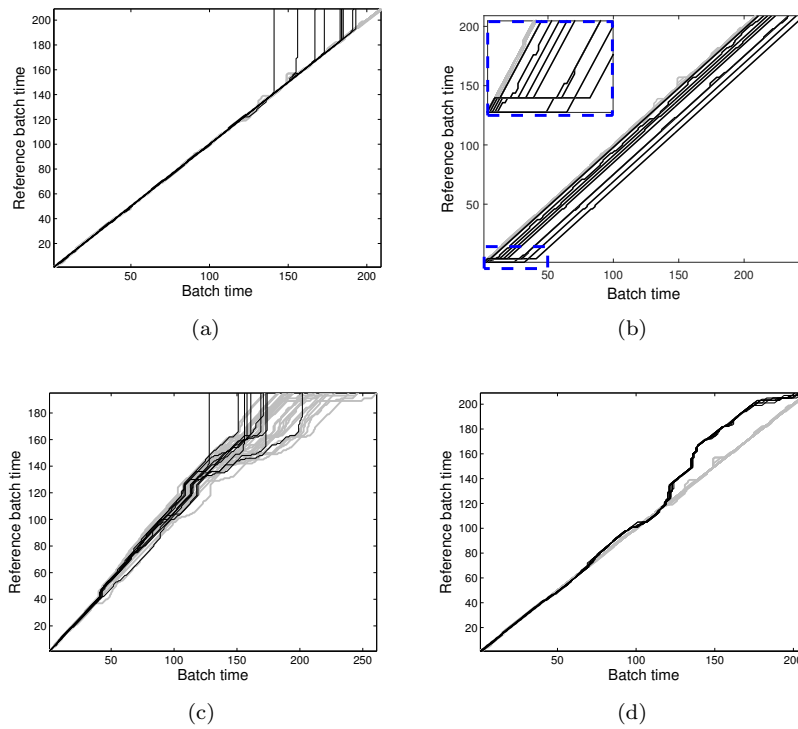


Figure 6.4. Warping information derived from the DTW-based synchronization of the raw batch trajectories for each of the asynchronism scenarios: (a) case #1: class I (grey lines) and class III (black lines) asynchronisms; (b) case #2: class I (grey lines) and class IV (black lines) asynchronisms; (c) case #3: class II (grey lines) and class III (black lines) asynchronisms; and (d) case #4: class I asynchronism (grey and black lines). The grey lines represent the warping profiles of 30 batches without data manipulation and the black lines represents the warping profiles of those batches with the data manipulation described in Table 6.2 for each scenario.

of the batches have equal duration, the main events are not perfectly synchronized. This supports the claim that the batch synchronization is required even when the variable trajectories show the same evolution pace. Regarding case #3, the warping profiles of the first 30 batches (see grey lines in Figure 6.4(c)) shows a clear variation in the duration of the batches, which is more prominent at late stages than early stages of the process. This type of pattern was expected due to the nature of the

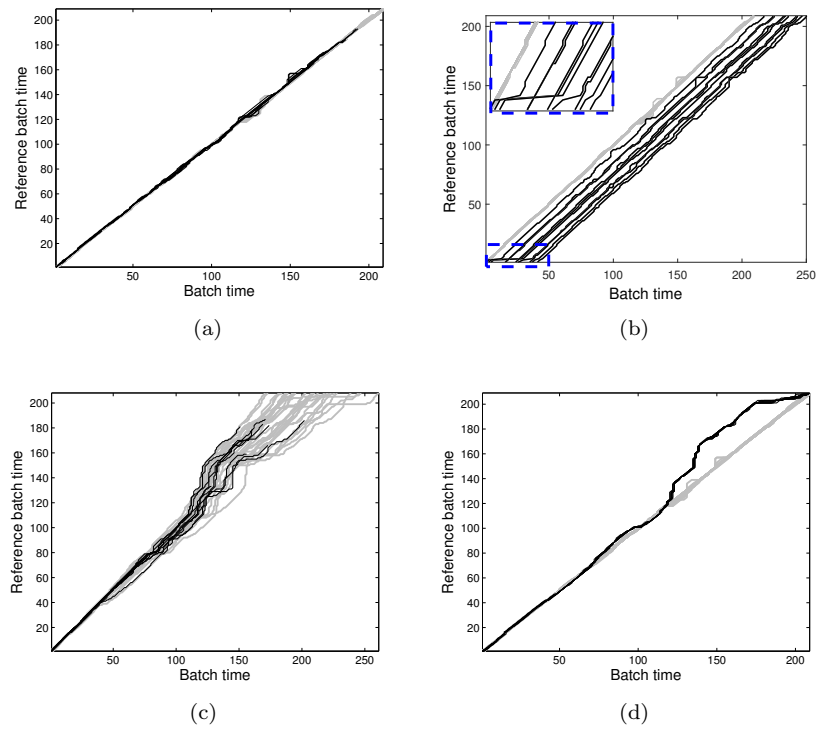


Figure 6.5. Warping information derived from the Multisynchro-based synchronization of the raw batch trajectories for each of the asynchronism scenarios: (a) case #1: class I (grey lines) and class III (black lines) asynchronisms; (b) case #2: class I (grey lines) and class IV (black lines) asynchronisms; (c) case #3: class II (grey lines) and class III (black lines) asynchronisms; and (d) case #4: class I asynchronism (grey and black lines). The grey lines represent the warping profiles of 30 batches without data manipulation and the black lines represents the warping profiles of those batches with the data manipulation described in Table 6.2 for each scenario.

asynchronism with which these batches are affected. Concerning the warping profiles corresponding to the rest of batches (see black lines in Figure 6.4), a different asynchronism pattern is recognized in each case.

In Figure 6.4(a) and Figure 6.4(c), 10 out of the 40 warping profiles (black lines) show an excessive number of vertical transitions in comparison to the rest (grey lines). The difference between the two cases is that the raw trajectories in the latter

are completely unsynchronized from the start of the batches (see the warping profiles notably deviating from the main diagonal in Figure 6.4(c)). This pattern is directly related to the presence of batches that were not entirely completed. In common practice, these incomplete batches are inadvertently taken into consideration for batch synchronization, causing severe and undesirable changes in the profiles of the process variables. In case #2, 10 out of 40 batches (those at which a shift-type asynchronism was introduced) show a diagonal warping profile that is parallel to the main diagonal (see Figure 6.4(b)). This pattern is characteristic in the cases where similar values of the J process variables are registered at the start of the batch. It leads to an excessive number of horizontal transitions in the synchronization, as can be seen in the blue dashed rectangle in Figure 6.4(b). The higher the duration of the shift from the start, the higher the number of horizontal transitions in the warping profile. In this case, the DTW algorithm shrinks the corresponding batches at that time interval by averaging the measurements of the J process variables. Finally, the warping profiles related to case #4 show a divergence from the diagonal, confirming a different process pace at the second half of the run across batches (see Figure 6.4(d)).

Figure 6.5 shows the resulting warping profiles derived from the low-level Multisynchro routine for the four cases of asynchronisms under study. In the cases where the batches are partially incomplete at the end of the batch runs, irrespective of the process pace, the Multisynchro algorithm finds the best matching between their end point and the sampling time point of the reference that minimizes the weighted multivariate distance. The relaxation of the end point can be appreciated in the incomplete warping profiles in cases #1 and #3 (see black lines in Figure 6.5(a) and (c)). The remaining trajectory of the process variables, including the warping profiles, are imputed by using the variance-covariance matrix of the in-control statistical model obtained from the NOC batches only affected by class I and II asynchronisms. As regards to class IV asynchronism, the resulting warping profiles still show the shift over the main diagonal, although the starting point is the last sampling time point of the series of observations that matches the first sampling time point of the reference (see black lines in Figure 6.5(b)). Finally, the warping profiles associated with case #4 are very similar to the warping profiles from DTW at a first glance, although

subtle differences are observed at certain phase of the process (see Figure 6.5(d)).

To illustrate whether there is an effect of the type of synchronization on the synchronized trajectories, the batch data affected by the four cases of asynchronisms are compared with the outcomes of the synchronization based on the DTW and Multisynchro algorithms in Figure 6.6. The first remarkable finding for the NOC data sets simulated in this study is the similarity of the results in cases #2 and #4 (see Figures 6.6(d)-(f) and Figures 6.6(j)-(l), respectively). Only subtle differences are observed at certain phases of the batch runs. Note however that if the batch trajectories are perturbed by an abnormality (e.g. spike or subtle drift) in the shifted part of the trajectory, the DTW algorithm would shrink the corresponding batches at that time interval by averaging the measurements of the J process variables. The most likely, the abnormal feature not belonging to the batch processing might remain in the trajectories, which might produce a false alarm. In contrast, the Multisynchro algorithm would automatically discard the first sampling time points, thereby averting the addition of non-process features.

Regarding cases #1 and #3, there are major differences in the final result due to the presence of batches affected by class III asynchronism (see Figures 6.6(a)-(c) and Figures 6.6(g)-(i), respectively). When the batches are not completed, the DTW algorithm correctly synchronizes the batch trajectories from the initial point $(1, 1)$ to the optimum last matching point (k_{ref}, K_i) (the last closest point of the black lines to the diagonal profile in Figure 6.4(c)). From the $(k_{ref} + 1)$ -th to the K_{ref} -th sampling time point of the reference batch, the last point of the n -th batch is matched, leading to the vertical transitions observed (see Figure 6.4(c)). This would lead to expansions of the batch trajectories, i.e. the addition of replicated values of the K_i -th sampling time point in the n -th batch. Consequently, flat profiles in the process variables (i.e. replicated values of the last actual value) are introduced (see Figure 6.6(b) and Figure 6.6(h)). This is an artifact since the batches were not actually finished and the remaining trajectory till completion is computed in an inappropriate way. In addition, these inaccuracies are inherited in the synchronization of that stage. Note that when a batch is finished earlier than the historical batches, the addition of artifacts in data may be higher. This would cause a possible change of the trajectory profile. In the batch data simulated,

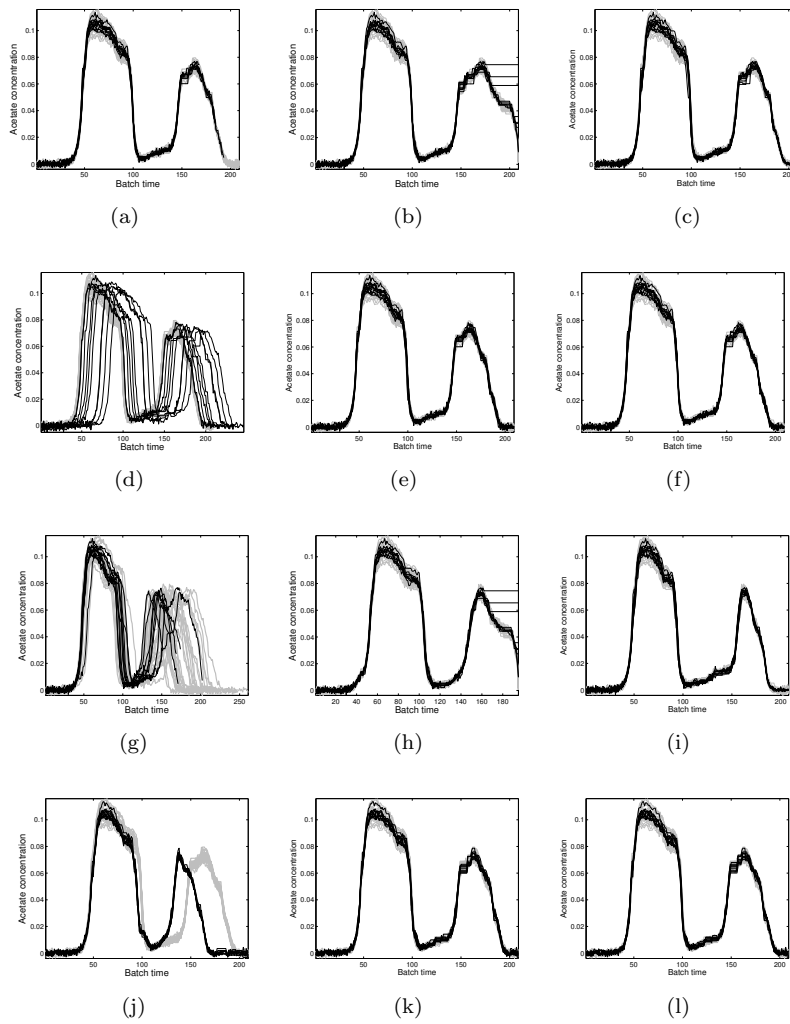


Figure 6.6. Comparison of the acetate concentration of 40 NOC raw batches (a, d, g, and j) with their respective trajectories synchronized by DTW (b, e, h, and k) and by Multisynchro (c, f, i and l) in four different asynchronism scenarios: (a, b, and c) case #1: class I (grey lines) and class III (black lines) asynchronisms; (d, e and f) case #2: class I (grey lines) and class IV (black lines) asynchronisms; (g, h and i) case #3: class II (grey lines) and class III (black lines) asynchronisms; and (j, k and l) case #4: class I asynchronism (grey and black lines).

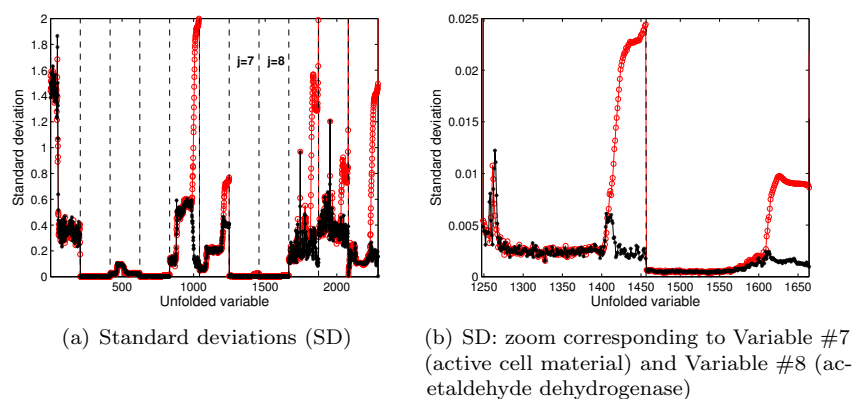


Figure 6.7. Comparison of the standard deviation vectors obtained from batch data synchronized by using the DTW algorithm without taking into consideration the asynchronous patterns from case #3 (red empty circles line) and by using the Multisynchro approach (black stars line).

the largest cut was approximately 50 sampling time points. As can be observed in Figure 6.6(b) and Figure 6.6(h), it produces changes in the shape of the profiles in the second half of the batch runs, and consequently, in the normal process pace.

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

In order to study the improvement reached by the application of the Multisynchro approach versus DTW-based synchronization applied to all batches in case #3, the standard deviation vectors of the corresponding synchronized batch trajectories are computed and shown in Figure 6.7. Figure 6.7(a) reveals that when the incomplete batches are treated separately from the rest in the batch synchronization, the resulting standard deviation val-

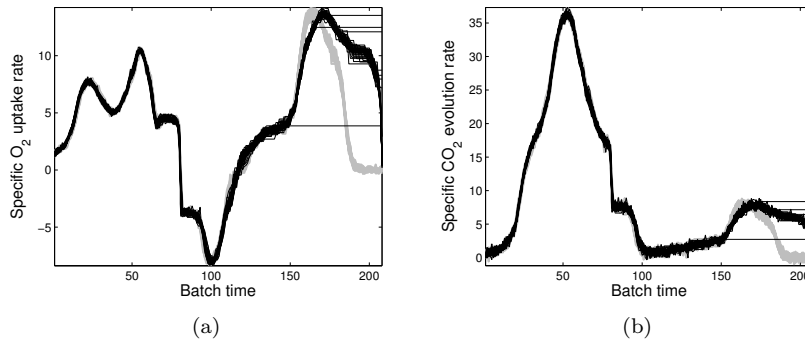


Figure 6.8. Batch trajectories belonging to the process variables specific oxygen uptake rate (a) and specific carbon dioxide evolution rate (b) after synchronization. Black lines represent trajectories synchronized by using the DTW algorithm without taking into account the type of asynchronism and grey lines represent those batch trajectories synchronized by the Multisynchro approach.

ues are lower (black stars lines) than for the classical approach (red empty circles lines). These differences are more prominent in Variables #5, #6, #9 and #10 (see trajectories of Variables #9 and #10 in Figure 6.8), in particular at the last stage of the process (last 50 sampling time points from the batch runs), where some batches are incomplete or shrunk. Even though the standard deviation values seem to be similar for the rest of variables, these differences are also observed at the same batch time period but to less extent (see Figure 6.7(b) for Variables #7 and #8). This is a clear indicator that synchronizing the batch trajectories without taking into consideration this type of asynchronism seriously affects the resulting trajectories, decreasing the signal-to-noise ratio.

6.6.2 Synchronization of batches with equal duration

Several publications in the literature and commercial software packages express the belief that only batches with different duration are subject of synchronization. However, when some key process events are not fully aligned, regardless of the batch duration, the batch trajectories must be synchronized. This is clearly shown in case #4 represented in Figure 6.3(d). As

can be appreciated from this figure, the acetate concentration trajectory shows that the second half of the batch run (from the 120th sampling time point onward) has a different pace for the two groups of batches, denoted as black and grey lines. From the start to the 90th sampling time point, the main process phenomena apparently occur at the same sampling time point across batches. In order to ensure that the key process events are actually synchronized, batch synchronization should be applied to batch data. In Figure 6.4(d), the resulting warping profiles from the synchronization at the high-level step are depicted. As can be seen, there are two groups of profiles clearly distinguished, those corresponding to the 10 batches where the different process pace was forced (black profiles) and the rest of batches (grey lines). Looking at the profiles corresponding to the 10 batches with asynchronism, one can observe two main time periods of large deviation from the main diagonal: from the 75th to the 100th sampling time point and from the 120th to the end of the batch, the deviation being smaller in the former than in the latter. In both time periods, these warping profiles have a higher number of vertical transitions than horizontal transitions. This is the reason why these warping profiles are beyond the main diagonal. It indicates that the batches with asynchronism had slower process pace than the rest. Hence, the synchronization algorithm needs to expand the corresponding batch trajectories at the aforementioned batch time periods.

In case #4, batch synchronization is rarely applied because batches have already the same duration. There is commercial software for batch process monitoring, e.g. SIMCA Release 13.0.3 [65], that only demand the synchronization of the batch trajectories when they have different length⁴. In order to emphasize the importance of this step, the raw batch trajectories are again compared with those obtained from the batch synchronization based on the Multisynchro algorithm, whose results are very similar to those obtained with DTW. For comparative purposes two different process variables are illustrated in Figure 6.9. As can be observed, the raw trajectories of the specific oxygen uptake (see Figure 6.9(a)) and carbon dioxide evolution (see Figure 6.9(c)) rates belonging to 10 out of the 40

⁴The main synchronization procedure used in SIMCA Release 13.0.3 is the so-called time linear expanding/compressing (TLEC)-based method, which is based on linearly expanding and/or compressing pieces of variable trajectories in the local batch time dimension [64]. In case the differences in batch length is greater than 20%, a maturity variable is used as the basis of batch synchronization instead of the local batch time [64].

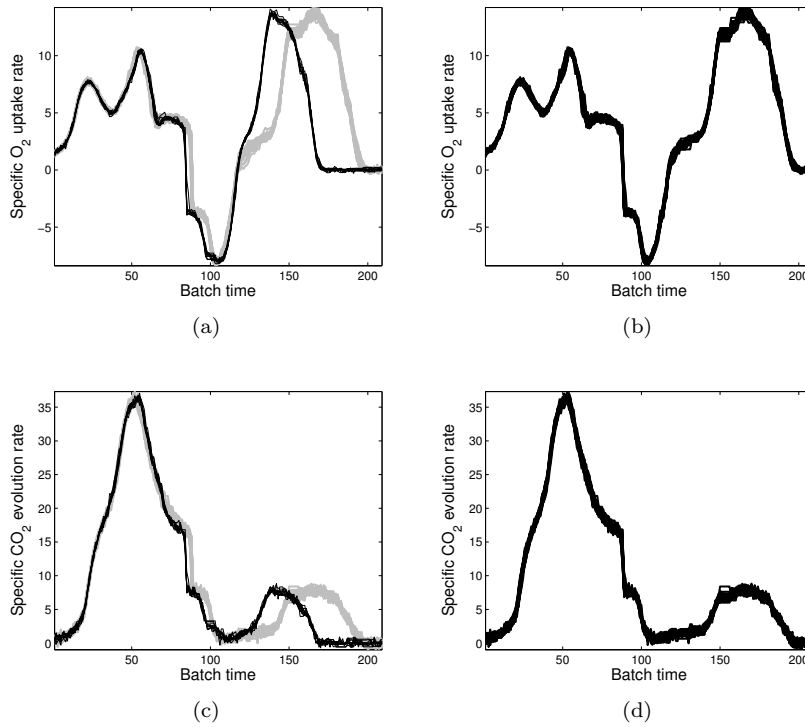


Figure 6.9. Batch trajectories belonging to the process variables specific oxygen uptake rate (a, b) and specific carbon dioxide evolution rate (c, d) without applying any synchronization (a and c, respectively) and applying the Multisynchro approach (b and d, respectively). The black lines in (a) and (c) represent the raw trajectories belonging to 10 out of 40 batches with the case #4 asynchronism embedded.

raw batches (black lines) differ with those corresponding to the rest of batches (grey lines). Mainly, these differences are shown at the last stage of the process, from the 120th sampling time point onwards. This reflects that the fermentation at the second half of the process took less time than in the rest of the batch trajectories. Hence, the synchronization of this stage is needed for subsequent analysis. Once the Multisynchro approach is applied to batch data, the resulting 40 profiles not only have equal length but also the segments of the profile corresponding to the last process stage overlap across batches (see the synchronized

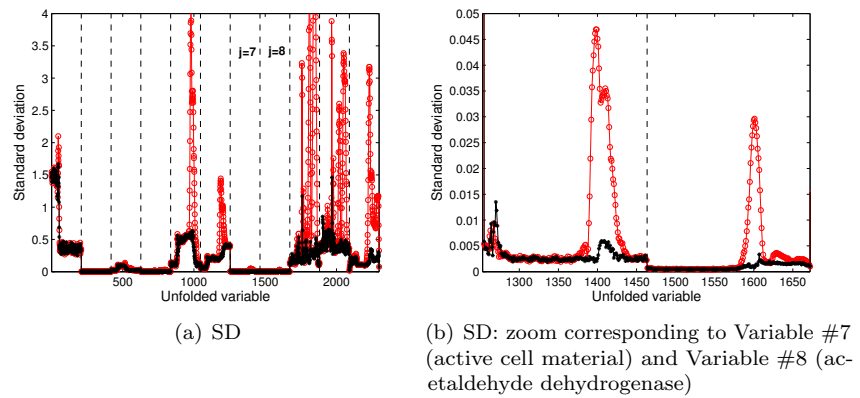


Figure 6.10. Comparison of the standard deviation vectors obtained from raw batch data from case#4 (red empty circles line) and from synchronized batch data derived from the Multisynchro-based synchronization (black stars line).

trajectories of the specific oxygen uptake and carbon dioxide evolution rates in Figure 6.9(b) and Figure 6.9(d), respectively). Again, the standard deviation vector is estimated from the raw and synchronized batch data to study the improvement achieved when the Multi-Synchro approach is applied in comparison to take no action for synchronization (see Figure 6.10). If the batch trajectories are not synchronized, the standard deviation vector derived contains more variability (see red empty circles line in Figure 6.10(a)) in comparison to that derived from the data synchronized with the Multisynchro approach (see black stars line in Figure 6.10(a)). These differences are more prominent in Variables #5, #6, #9 and #10 (see trajectories of Variables #9 and #10 in Figure 6.9), but also existent in the rest of process variables (see Figure 6.10(b)). In this case, these differences are mainly found between the 120th onwards, the time period at which the batch profiles are clearly not synchronized. Note that the variation from the main trajectory is approximately eight times higher when the key process events are not aligned in comparison to when batch data are synchronized with the Multisynchro approach. This again supports the idea that the type of asynchronism needs to be taken into consideration in batch synchronization, not only to focus the multivariate statistical analysis on the same point of process evolution but

also to reach better synchronization quality.

6.7 Conclusions

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

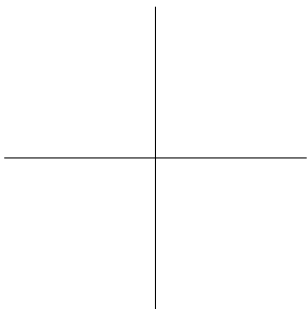
The application of the same synchronization procedure to batches with asynchronisms of different nature may cause the addition of extreme artifacts, affecting severely the synchronization quality. Based on the original DTW and RGTW algorithms, a novel synchronization approach called Multisynchro that takes into consideration the multiple asynchronisms present in batch data is proposed. The new proposal is composed of two routines. The first one (high-level routine) is devoted to detect the different patterns of asynchronism of each particular batch based on the warping information derived from the RGTW or DTW algorithm. The second one (low-level routine) performs the batch synchronization using specific procedures based on the nature of the asynchronism. The new approach also includes a procedure that performs abnormality detection and batch synchronization in an iterative way. This avoids batch abnormalities to affect synchronization quality. The simulated example has shown that the Multisynchro approach might outperform the standard approach of applying the same synchronization procedure, in particular, in the cases where incomplete batches are mixed with batches affected by other types of asynchronisms. The main disadvantage of the Multisynchro algorithm is its complexity and non-linear interaction among the different techniques used. In particular, the complex interaction between the iterative and non-linear synchronization procedure with the iterative and linear modeling approach disables its automatic application. To ensure the best synchronization, the visualization and track of the algorithm actions are required.

Inappropriate synchronization affects not only the quality of batch synchronization, but also the subsequent steps of bilinear modeling. When the key process events do not overlap at the

same point of process evolution ensuring the same process pace in all batches, the capability of monitoring schemes for fault detection may be dramatically reduced. In Chapter 7, the impact of different synchronization methods in the model parameter stability is studied. In addition, the consequences of inappropriately synchronizing batch data with multiple asynchronisms in process monitoring will be investigated in Chapter 8. The novel Multisynchro algorithm is a promising synchronization technique that mitigates the influence of multiple asynchronisms on the batch modeling cycle.

Part III

**Modeling of Batch
Process Data**



Parameters stability on bilinear process modeling

Part of the content of this chapter has been included in the following publications:

- [3] J.M. González-Martínez, J. Camacho and A. Ferrer. Bilinear modeling of batch processes. Part III: Parameter Stability, *Journal of Chemometrics*, 28(1): 10–27, 2014. **Awarded with the 4th Siemens Process Analytics Prize for an outstanding publication in the field of Process Analytics by the German Working Group "Prozessanalytik" in cooperation with Siemens.**
- [13] J. Camacho, J.M. González-Martínez and A. Ferrer. Chapter 4: Bilinear Modeling of Batch Process Data. Batch Processes: Monitoring and Process Understanding, *Wiley-VCH Verlag GmbH*, publication due in 2016.
- [27] J.M. González-Martínez, J. Camacho and A. Ferrer. Enhancement of batch process understanding and monitoring: a matter of parameters stability. *In proceedings of the 13th Conference on Chemometrics in Analytical Chemistry*, page 39, Budapest (Hungary), 2012.

7.1 Introduction

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.

For the design of monitoring schemes, the measurements of J process variables collected at K different sampling time points over N batches run under NOC are used. Setting a BMSPC system becomes a challenging task due the nature of batch data [45, 71]: 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 PCA and BMSPC, combined with the adequate preprocessing methods are frequently used for the generation of empirical models [71, 183]. Using this type of methods, process understanding can be gained and process operating problems can be troubleshooted in a timely manner. From this offline 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) [62].

A principal concern when designing BMSPC systems based on PCA should be to ensure that the system is fit for purpose in the context in which it will be used. Issues such as acceptance by the operational personnel, robustness during typical process operation and appropriate maintenance procedures are factors that need to be considered. Another factor relating to the methodologies employed in the system concerns 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 [184].

The application of bilinear models like PCA to batch data requires the rearranging of the three-way data array in a number of two-way arrays. This transformation can be performed following a number of different approaches. To date, there is no sound study on the effect of the bilinear modeling approaches on parameter stability in BMSPC. This chapter is devoted to investigate the parameter stability associated to the most used synchronization and PCA-based BMSPC methods. The synchronization methods included in this study are: IV, DTW, RGTW and TLEC-based methods. In addition, different arrangements of the three-way batch data array into two-way arrays are considered, namely: single-model approaches, K -models approaches and hierarchical approaches. Results are discussed in connection with previous research work [87, 89].

This chapter is organized as follows. Section 7.2 presents the materials and methods used in the comparative study. Section 7.3 illustrates the effect of the batch synchronization on the parameter stability. Section 7.4 is devoted to present and discuss the results of the comparison of the different rearranging methods under study. Finally, conclusions are drawn in Section 7.5.

7.2 Material and methods

The different modeling approaches under study are compared in terms of parameter stability using data corresponding to Set #1 (three-way arrays $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$) from realistic simulations of a fermentation process of the *S. cerevisiae* cultivation (see Chapter 2). Two data sets were generated based on the biological model of the aerobic growth of *Saccharomyces Cerevisiae* on a glucose limited medium [118].

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 7.1). For the DTW-based synchronization on raw batch data, the reference batch selected in both data sets 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 [1, 76]. 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 extent 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 time points #23, #38, #54, #65, #89, #96, #104, #119, #140 and #166 in the reference batch are extracted by examining its variable trajectories. Afterward, 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 time points. Concerning IV-based synchronization, the simulation time denoting the fermentation evolution (the so-called fermentation rate) is selected as indicator variable given its monotonic and increasing behavior. Non-uniform increments were defined in six different process stages, and a total of 209 data points are obtained by linear interpolation (see Section 4.2.1 for a detailed description of the synchronization based on IV). Let us name this synchronization as IV1. To illustrate the effect of an incorrect decision on the selection of the indicator variable and the increments for interpolation on the parameter stability, a second IV-based synchronization is performed, taking the biomass concentration as the indicator variable (IV2). The biomass concentration is the only measured process variable that might potentially be

selected as indicator variable due to its monotonicity and increasing value over time. To fulfill the remaining requirements of an IV, a start and end point in the biomass concentration variable is selected across batches. Equal-distanced intervals are defined, leading to a total of 209 data points obtained by linear interpolation. In addition, a second synchronization on the IV-based synchronized trajectories using the DTW algorithm (IV1-DTW and IV2-DTW) and the RGTW algorithm (IV1-RGTW and IV2-RGTW) with the parameters specified above is carried out. The purpose of this re-synchronization is again to check to what extent IV properly synchronizes the key process events.

The comparison of the PCA-based BMSPC approaches in terms of the parameter stability is organized in three categories: single-model, K -models and hierarchical-model approaches (see Table 7.2). Among the single-models, VW, BD and BW models are studied. The approaches VW-TCS and VW-VCS represent a VW unfolding where Trajectory C&S and Variable C&S¹ 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 BW 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 point 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.

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

¹The application of Variable C&S is only meaningful in VW. Hence, this preprocessing approach is not taken into consideration for the rest of the BMSPC approaches in this study.

Table 7.1. 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 RGTW	Reference: batch #12 (209 sampling time points), constraints according to [76] Reference: batch #12 (209 sampling time points), constraints according to [1]
	Time Linear Expanding/Compressing (TLEEC)-based method	TLEEC TLEEC-events	209 interpolation points 209 interpolation points, key processes events at sampling time points: #23, #38, #54, #65, #89, #96, #104, #119, #140 and #166
Variable	(TLEEC & SCT)-based method	TLEEC-DTW TLEEC-RGTW	Parameters from TLEEC and DTW models Parameters from TLEEC and RGTW models
	IV-based method	IV1 and IV2 IV1-DTW and IV2-DTW	IV: fermentation rate (IV1) and biomass concentration (IV2) Parameters from IV1 and IV2, and DTW models
	(IV & SCT)-based method	IV1-RGTW and IV2-RGTW	Parameters from IV1 and IV2, and RGTW models

Table 7.2. 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-TGS	Variable-wise	Trajectory C&S	J
	VW-VCS	Variable-wise	Variable C&S	J
K -model ($M = K$)	BD1	Batch-dynamic with ILMV	Trajectory C&S	$J \cdot (1 + LMV)$
	LM	Local K -model	Trajectory C&S	J
	UWMMW ILMV-var	Uniformly Weighted Moving Window	Trajectory C&S	n_k
	UWMMW ILMV-obs	K -model with ILMV in the variables	Trajectory C&S	J
	EWEMW-var	Uniformly Weighted Moving Window	Trajectory C&S	J
	EWEMW-obs	K -model with ILMV in the observations	Trajectory C&S	$k \cdot J$, for k from 1 to K
Hierarchical-model ($M = K$)	AHKM	Adaptive hierarchical K -model with $d = 0.2$ and $d = 50$	Trajectory C&S	J

Normalized Squared Difference (NSD) between the different parameter vectors (averages, standard deviations, sum of squares and loadings) is computed as follows:

$$NSD_{\theta} = \sum_{j=1}^J \left(\frac{\theta_j^{(1)}}{\|\boldsymbol{\theta}^{(1)}\|} - \frac{\theta_j^{(2)}}{\|\boldsymbol{\theta}^{(2)}\|} \right)^2 \quad (7.1)$$

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 7.2, $M = 1$ and $M = K$ different models are obtained from PCA-based bilinear modeling in the single-model and K -models approach, respectively. The size of the loading vectors in each model depends on the number of LMVs 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 time point k (i.e. $NSD_{\theta} = \sum_{k=1}^K NSD_{\theta_k} / K$, where NSD_{θ_k} is assessed by following Equation 7.1). When including LMVs, exception made for BW models, data from a specific sampling time point 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 values. 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/AHKM), 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 1LMV-obs/EWEW-obs) and as variables (UWMW 1LMV-var/EWEW-var).

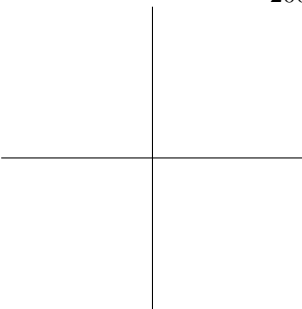
Batch data synchronized by all the synchronization approaches under study (see Table 7.1) are employed to study the effect of batch synchronization on parameter stability in Section 7.3. To proceed with the comparison of the rearranging methods in terms of parameter stability in Section 7.4, for the sake of easy

understanding only the two data sets synchronized by using the RGTW algorithm are used.

7.3 Effects of batch synchronization

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, provided that the batches are all NOC and do not inadvertently include batches with abnormal behavior.

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 7.1 and Figure 7.2. These figures reveal 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 7.1) 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, #3, #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 7.1(a)) than those synchronized with TLEC-events (magenta empty circles lines in Figure 7.1(a)). Hence, the latter synchronizes the batch trajectories with more accuracy, reducing the variability in comparison with the former. In regard to the IV-based methods, when the indicator variable is accurately defined jointly with the corresponding IV levels for synchronization, the key process events are aligned although the variability might be affected to a certain extent (see red empty circles in Figure 7.1(c)). However, when the parameters of the synchronization based on IV are not well selected and tuned, the shape of the process variables might be seriously affected. An illustrative example of this effect can be seen on the standard deviation vectors in Figure 7.1(d), which notably differ from those obtained from data properly synchronized with



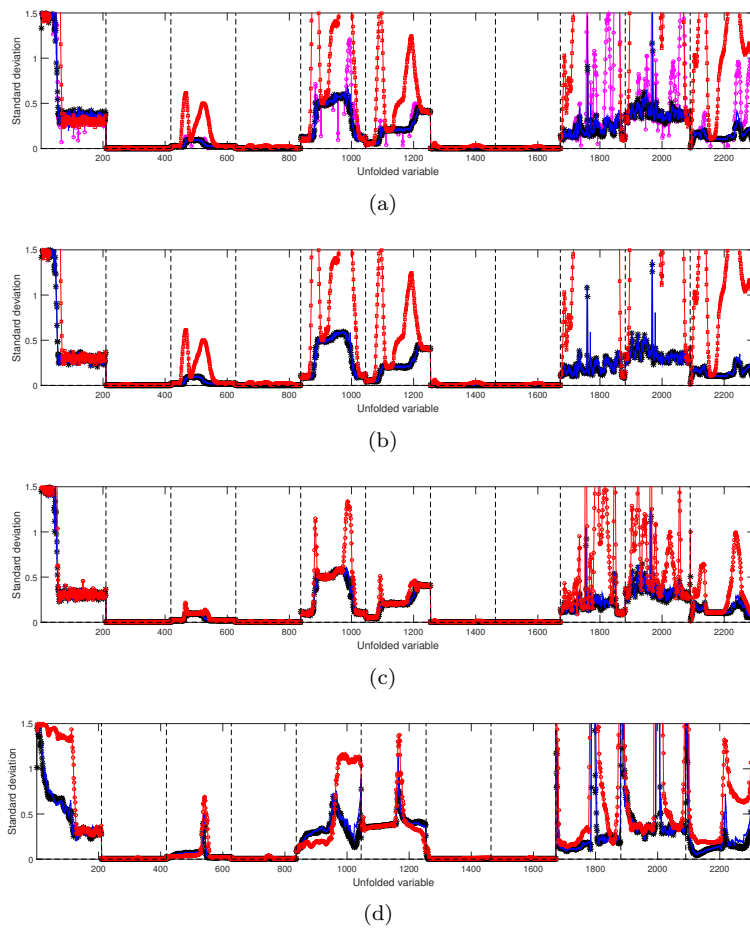


Figure 7.1. Standard deviation vector of the synchronized trajectories (average from both data sets): (a) comparison between the SCT-based and TLEC-based methods: blue dots (RGTW), black asterisks (DTW), magenta empty circles (TLEC-events) and red empty squares lines (TLEC); (b) comparison between the (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 using the fermentation rate as the indicator variable: blue dots (IV1-RGTW), black asterisks (IV1-DTW) and red empty circles (IV1) lines; and (d) comparison between the (IV & SCT)-based and IV-based methods using the biomass concentration as the indicator variable: blue dots (IV2-RGTW), black asterisks (IV2-DTW) and red empty circles (IV2) lines.

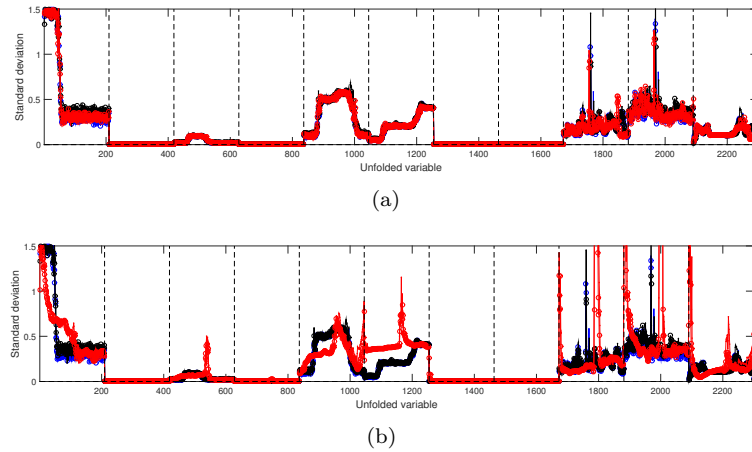


Figure 7.2. Standard deviation vector of the synchronized trajectories (average from both data sets): (a) comparison between SCT-based methods and (TLEC/IV & SCT)-based methods using the fermentation rate as the indicator variable: black empty circles (DTW), black dots (RGTW), blue empty circles (TLEC-DTW), blue dots (TLEC-RGTW), red empty circles (IV1-DTW) and red dots (IV1-RGTW); and (b) comparison between SCT-based methods and (TLEC/IV & SCT)-based methods using the biomass fermentation as the indicator variable: black empty circles (DTW), black dots (RGTW), blue empty circles (TLEC-DTW), blue dots (TLEC-RGTW), red empty circles (IV2-DTW) and red dots (IV2-RGTW).

IV (see Figure 7.1(c)). This shows how critic the selection of the indicator variable and the increments are.

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 in Figure 7.2(a) (compare black, blue and red empty circles -i.e. DTW, TLEC-DTW, and IV1-DTW- with black, blue and red dots -i.e. RGTW, TLEC-RGTW, and IV1-RGTW-, respectively). Nonetheless, when the IV-based synchronization is not correctly performed (IV2), differences in terms of deviation around the mean trajectory appear among methods because the synchronization as defined distorts the variable trajectories (compare black, blue and red empty circles -i.e. DTW, TLEC-DTW, and IV2-DTW- with black, blue and red dots -i.e. RGTW, TLEC-RGTW, and IV2-RGTW-, respectively, in Figure 7.2(b)). This denotes a similar perfor-

mance among synchronization methods (RGTW and DTW) in terms of synchronization quality provided that the different synchronization are correctly conducted, ensuring the alignment of the landmarks. 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, the addition of artifacts, and consequently, the amount of noise, is reduced. Hence, the variability of the resulting batch trajectories is lower than those where the key process events are not properly synchronized (synchronization based on TLEC and on IV with the biomass concentration as the indicator variable and constants increments). Another phenomenon worth emphasizing is the risk of interpolating data when an indicator variable is used for synchronization. Though IV aligns the key process events, it might cause an increase of the variability around the average trajectory. Note that variability might rise if the increments in which the process data are interpolated are not properly defined. 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 7.1(b)) and by IV (see blue dots and black asterisks lines in Figure 7.1(c) and Figure 7.1(d)) contain lower values than those derived from the synchronized trajectories by TLEC and IV (see red empty squares line in Figure 7.1(b), and see red empty circles line in Figure 7.1(c) and Figure 7.1(d), 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.

Comparing SCT-based synchronization and re-synchronization, some findings are worth being highlighted (see Figure 7.2). 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 in Figure 7.2(a)) and those derived after applying an SCT-based synchronization in trajectories already synchronized by TLEC (see blue empty circles and dots in Figure 7.2(a)) and IV using the fermentation rate as the indicator variable (see red empty circles and dots in Figure 7.2(a)). Only slight differences in terms of variability can be found at a late stage of the process evolution between the standard vectors of the trajectories synchronized by IV and

the rest of methods (see second half of standard deviations in variables #10, #11 and #12 in Figure 7.2(a)). Note, however, that when the batch data are incorrectly synchronized by an indicator variable where the increments were not precisely defined, the resulting standard deviations differ much from the rest (compare red empty circles and dots with the rest symbols in Figure 7.2(b)).

An ANOVA was performed on the NSD values of each preprocessing parameter -i.e. mean and standard deviation- (see Table 7.3(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 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 ($NSD_{std,VCS} = 7.691e-05$) than those from Trajectory C&S ($NSD_{std,TCS} = 2.564e-02$), 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 7.3(b)). This will be discussed in detail in the next section.

In order to check if there are statistical 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 7.3(b)). This yielded that 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 LSD intervals are computed (see Figure 7.3). The NSD values corresponding to batch data synchronized by the group of SCT-based methods are statistically lower on average ($NSD_{DTW} = 1.525e-01$ 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, $NSD_{TLEC-events} = 1.910e-01$). Re-synchronization with SCT-based methods provides statistically significant improvements for TLEC (statistically lower NSD values on average: $NSD_{TLEC-DTW} = 1.587e-01$ and $NSD_{TLEC-RGTW} = 1.700e-01$ in comparison with $NSD_{TLEC} = 2.743e-01$). Regarding IV-based methods, there is a statistically significant enhancement of the parameter stability when the trajectories synchronized by the biomass concentration

Table 7.3. 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		
		Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
Synchronization method							
IV1-RGTW		9.390e-03	3.520e-02	1.632e-03	3.051e-04	3.051e-04	3.051e-04
IV1-DTW		8.354e-03	3.434e-02	1.527e-03	2.450e-04	2.450e-04	2.450e-04
IV1		1.364e-04	2.856e-02	3.276e-06	2.017e-05	2.017e-05	2.017e-05
IV2-RGTW		3.804e-04	1.989e-02	8.404e-06	2.196e-05	2.196e-05	2.196e-05
IV2-DTW		1.739e-03	2.546e-02	2.613e-05	1.697e-04	1.697e-04	1.697e-04
IV2		3.383e-04	1.695e-02	7.508e-06	3.022e-05	3.022e-05	3.022e-05
TLEC-RGTW		2.925e-05	1.856e-02	8.181e-06	8.363e-06	8.363e-06	8.363e-06
TLEC-DTW		4.184e-05	2.127e-02	7.630e-06	9.321e-06	9.321e-06	9.321e-06
TLEC		1.600e-03	3.162e-02	3.065e-04	8.980e-05	8.980e-05	8.980e-05
TLEC-events		1.308e-04	3.493e-02	9.388e-06	1.493e-05	1.493e-05	1.493e-05
RGTW		3.010e-05	2.077e-02	7.412e-06	5.007e-06	5.007e-06	5.007e-06
DTW		7.012e-05	2.000e-02	6.958e-06	3.022e-06	3.022e-06	3.022e-06

(a)

		K-Model						Hierarchical-Model				
		Single Model			K-Model			Hierarchical-Model				
		BW	VW-TCS	VW-VCS	BD1	LM	UWMW ILMV-var	UWMW ILMV-obs	EWEW-var	EWEW-obs	AHKM $d = 0.2$	AHKM $d = 50$
IV1-RGTW		1.998e-01	4.325e-03	1.363e-02	6.108e-03	4.181e-01	3.669e-01	4.137e-01	9.817e-02	2.241e-01	2.211e-01	4.180e-01
IV1-DTW		1.615e-01	5.376e-03	1.140e-02	8.398e-03	3.648e-01	2.816e-01	2.911e-01	6.450e-02	1.936e-01	1.889e-01	3.646e-01
IV1		3.255e-01	4.880e-03	2.852e-05	1.093e-02	2.428e-01	1.924e-01	2.123e-01	1.147e-01	4.573e-01	6.683e-01	2.427e-01
IV2-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
IV2-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
IV2		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	3.337e-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	3.197e-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)

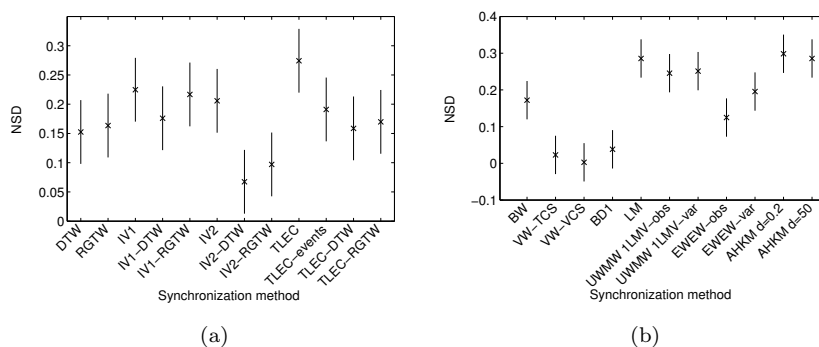


Figure 7.3. LSD intervals (95 % confidence) for the NSD values estimated from the first loading vector of both data sets for (a) the synchronization method and (b) the modeling approach.

at constant intervals are re-synchronized with SCT-based methods (statistically lower NSD values on average: $NSD_{IV2-DTW} = 6.737e-02$ and $NSD_{IV2-RGTW} = 9.701e-02$ in comparison with $NSD_{IV2} = 2.058e-01$). It supports the claim that the incorrect definition of the indicator variable and the IV levels leads to an inaccurate synchronization and modification of the shape of process variables. As a conclusion, the better the key process events are synchronized, the higher stability in the 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 post-batch process monitoring in terms of parameter stability. From these results, the application of other SCT-based synchronization methods (e.g. [77, 78]) may deserve further research.

In regard to the differences among modeling approaches, the NSD values corresponding to the VW models (VW-TCS and VW-VCS) and BD1 are statistically lower on average than those corresponding to the other models. Despite there are not statistically significant differences among evolving models of the same class (UWMW and EWEW), the evolving models that contain less variables show lower NSD values on average than those that contain more variables ($NSD_{UWMW-obs} = 2.456e-01$ versus $NSD_{UWMW-var} = 2.513e-01$, and $NSD_{EWEW-obs} = 1.247e-01$ versus $NSD_{EWEW-var} = 1.955e-01$). These results seem to indicate that the more parsimonious the model, the

better the parameter stability. A more meaningful insight into how the parameter stability is affected by the type of rearranging method used for modeling is provided in the next section using data synchronized by RGTW.

7.4 Effect of the rearranging methods

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 7.4.1, 7.4.2 and 7.4.3, respectively. In addition, the study on the K -model approaches -i.e. LM, UWMW and EWEW- is presented in Subsection 7.4.4. Finally, the parameter stability of the hierarchical-model approach -i.e. AHKM- is studied in Section 7.4.5. For the sake of simplicity, the two data sets synchronized by using the RGTW algorithm are used.

7.4.1 Batch-wise unfolding

As stated in Introduction section, parameter stability depends on two main factors. Firstly, precise identification relies on an informative data set and 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 [185].

The first question is the amount of calibration data which is enough to identify the parameters accurately. In Figure 7.4, 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 7.4(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 N 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 $N = 30$ 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

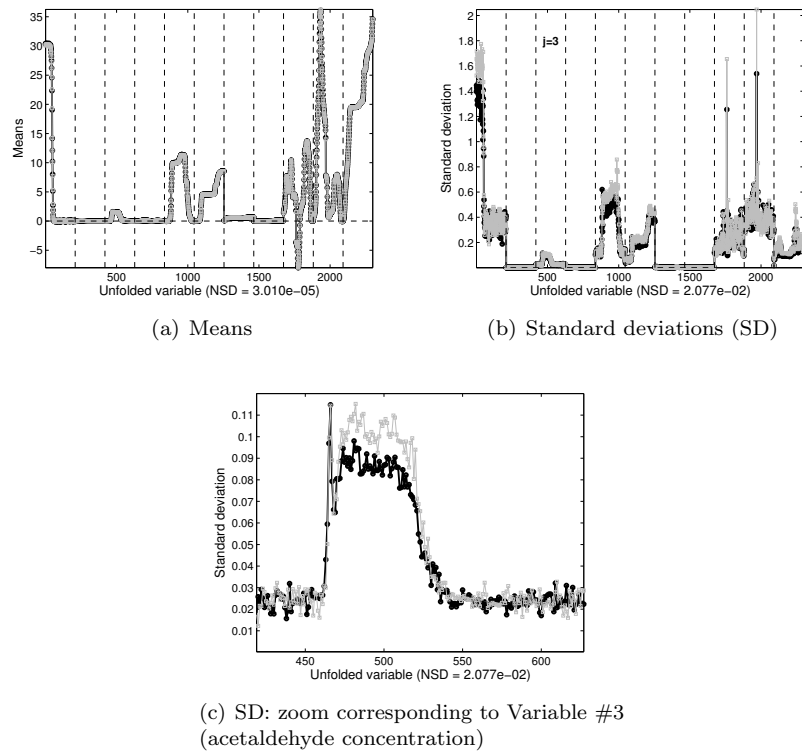


Figure 7.4. BW unfolding and Trajectory C&S. Comparison of the preprocessing parameters (means and standard deviations) obtained from the two simulated data sets BW unfolded, after applying Trajectory C&S. *NSD*: normalized squared differences between the average and standard deviations vectors of the two simulated data sets.

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

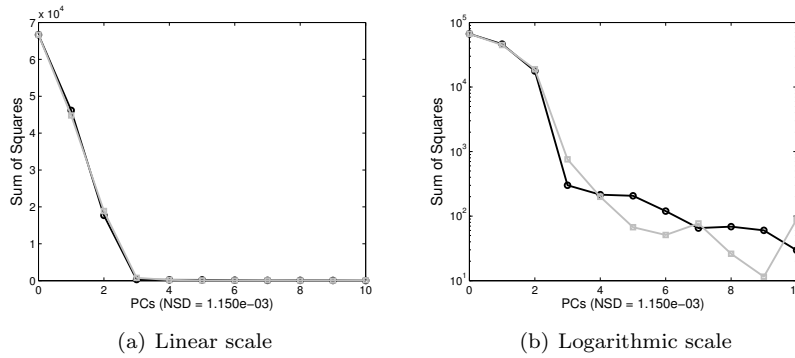


Figure 7.5. BW unfolding and Trajectory C&S. Explained Sum of Squares (SS) vs #PCs extracted in BW unfolding from the two simulated data sets. Note that PC #0 corresponds to the sum of squares remained after applying Trajectory C&S on batch data. *NSD*: normalized squared differences between the sum of squares vector captured by each PC of the two simulated datasets.

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 7.5 shows the plot of the explained sum-of-squares vs #PCs extracted for the current example assuming BW unfolding and Trajectory C&S. For the sake of visualization, both the linear and logarithmic scales are presented. As can be seen, the sum of squares captured by PC#1 ($SS_1 \approx 4.500e + 04$) explains a high percentage of the sum of squares remained after applying Trajectory C&S on batch data, i.e. SS_0 , (approximately 70%) and differs enough from that captured by PC#2 ($SS_2 \approx 1.800e + 04$). Consequently, the loadings of the first PC are expected to be stable. Note that the sum of squares captured from PC#3 onward 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. Hence, 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

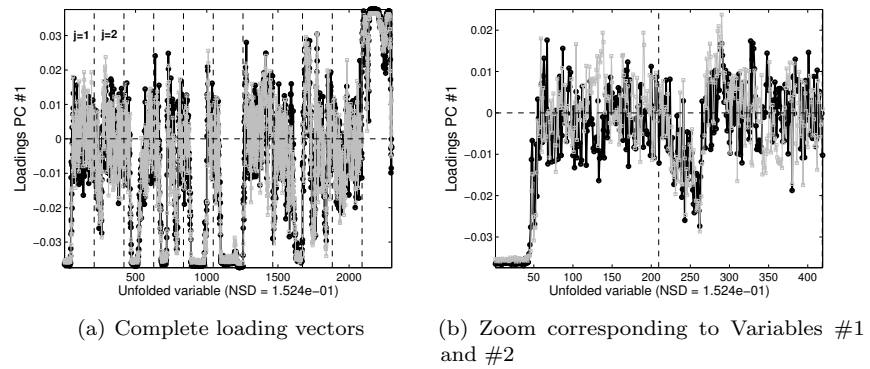


Figure 7.6. BW unfolding and Trajectory C&S. Comparison of the loading vector corresponding to the first PC obtained from the two simulated data sets BW unfolded, after applying Trajectory C&S. *NSD*: normalized squared differences between the first loading vector of the two simulated data sets.

the BMSPC method of choice.

There is a comment necessary regarding the use of a plot like the one in Figure 7.5 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 time points 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 7.6, 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 N observations each. It is clear that there is a parallelism between Trajectory C&S, and BW unfolding from the point of view of uncertainty estimation. In the zoom of Figure 7.6(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

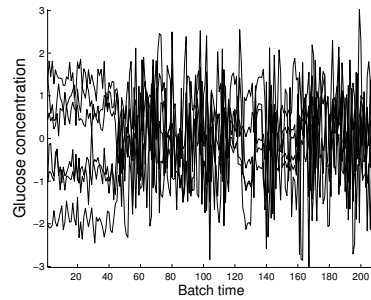


Figure 7.7. BW unfolding and Trajectory C&S. Trajectory of Glucose concentration (variable #1) after trajectory centering and scaling in some of the batches of the second simulated data set.

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 7.6(b)). Despite the fact that with BW unfolding a very complete modeling structure can be estimated [90], the noisy loadings shown in Figure 7.6 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 7.7).

7.4.2 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 $N \cdot K$ observations. In this example, $J = 10$ and $N \cdot K = 6270$. Due to the fact the number of parameters-to-the number of observations ratio in Trajectory C&S ($R_{TCS} = \frac{J \cdot K}{N} = \frac{2090}{30}$) is much higher than in Variable C&S ($R_{VCS} = \frac{J}{N \cdot K} = \frac{10}{6270}$), 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 7.8, the explained sum-of-squares vs #PCs extracted for the current example assuming VW unfolding and Trajectory

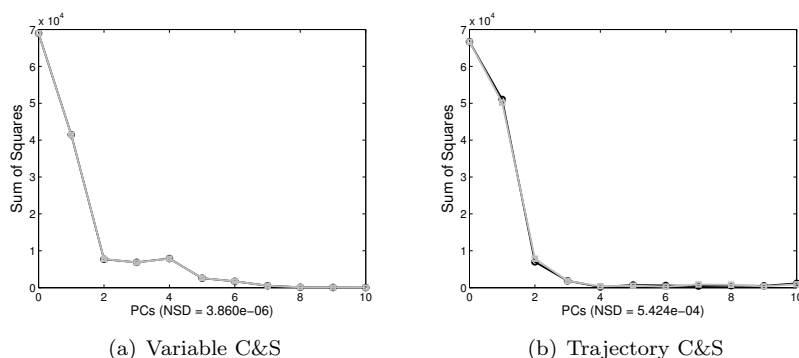


Figure 7.8. Variable-wise unfolding. Explained Sum of Squares (SS) vs #PCs extracted in VW unfolding from the two simulated data sets. Note that PC #0 corresponds to the sum of squares remained after applying Variable C&S (a) and Trajectory C&S (b). *NSD*: normalized squared differences between the sum of squares vector captured by each PC of the two simulated datasets.

C&S (see Figure 7.8(a)) and Variable C&S (see Figure 7.8(b)) are shown. Note that the sum-of-squares at PC#0 remaining after Variable C&S for both data sets ($SS_0 = 6.896 \times 10^4$) is slightly higher than after Trajectory C&S ($SS_0 = 6.667 \times 10^4$). 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 $SS_0 = (N \cdot K - 1) \cdot J = (30 \cdot 209 - 1) \cdot 11 = 6.896 \times 10^4$ units whereas in the latter is equal to $SS_0 = (N - 1) \cdot K \cdot J = (30 - 1) \cdot 209 \cdot 11 = 6.667 \times 10^4$.

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 7.8(a)), the sum of squares captured by PC#1 ($SS_1 \approx 4.145 \times 10^4$) 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 ($SS_2 \approx 7.700 \times 10^3$). 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 7.8(b)), the sum of squares captured by PC#1 ($SS_1 \approx 5.100 \times 10^4$)

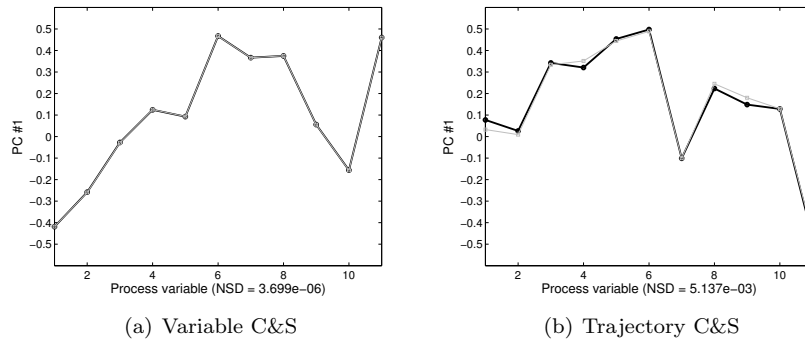


Figure 7.9. VW unfolding. Comparison of the loading vector corresponding to the first PC obtained from the two simulated data sets. NSD : normalized squared differences between the first loading vector of the two simulated data sets.

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 ($SS_2 \approx 7.500e + 03$). As a consequence, the loadings of the first PC are expected to be stable. Note that the sum of squares captured from PC#2 onward 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-VCS and for VW-TCS, respectively) confirms that Variable C&S outperforms Trajectory C&S in terms of parameter stability.

In order to compare the stability of the first PC, the corresponding loadings for both preprocessing methods are shown in Figure 7.9. 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 [87, 89]. It should be

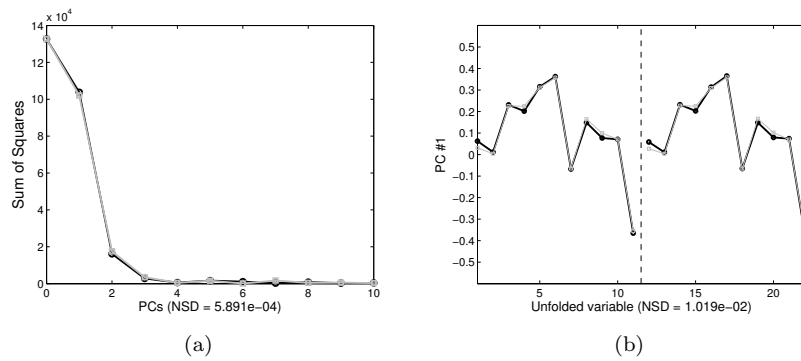


Figure 7.10. BD unfolding and Trajectory C&S. Explained Sum of Squares (SS) vs #PCs extracted (a) and first loading vector (b) in BD unfolding with 1 LMV from the two simulated data sets. Note that PC #0 corresponds to the sum of squares remained after applying Trajectory C&S on batch data. *NSD*: normalized 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.

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.

7.4.3 Batch-dynamic unfolding

Figure 7.10 shows the explained sum of squares vs #PCs and the loading vectors corresponding to the first PC for the two data sets after BD unfolding with 1 LMV and Trajectory C&S. These results are quite similar to those obtained for VW unfolding and Trajectory C&S. Hence, Figures 7.10(a) and 7.8(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 7.10(b) and 7.9(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 VW and BW unfolding. In particular, variability in BD is lower ($NSD=1.019e-02$) than in BW after Trajectory C&S ($NSD=1.524e-01$), and higher than in VW after Trajectory C&S ($NSD=5.137e-03$) and after Variable C&S ($NSD=3.699e-06$). This result is expected since BD is a generalization of VW and BW (its number of parameter-to-number of observation ratio is higher than VW, but lower than BW). Figure 7.10(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.

7.4.4 *K*-models

Figure 7.11 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 point $k = 10$ 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 7.11, the NSD values 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 point) instead of all the loadings (like in the single-model approaches). This is done to make comparison between *K*-models approach and the rest of approaches under study.

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 7.11(b) and Figure 7.11(d) in comparison with Figure

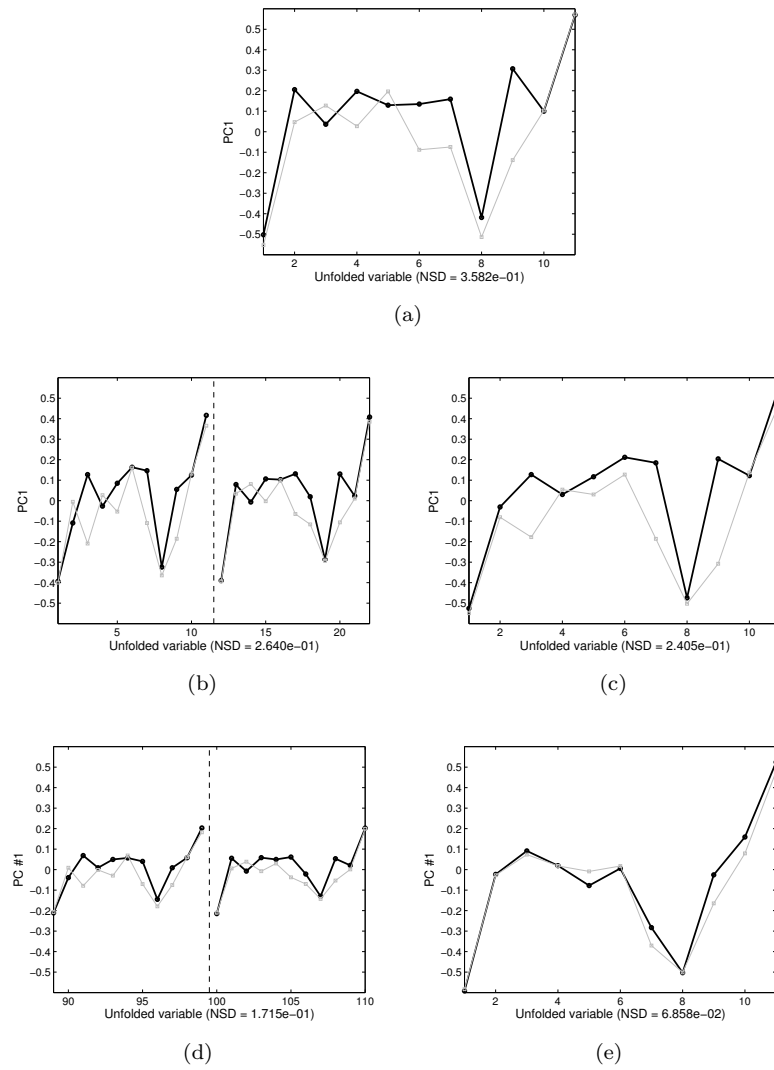


Figure 7.11. K -models and Trajectory C&S. First loading vector for the two data sets at the k -th sampling time point: (a) local model, (b) UWMW model with 1 LMV in the variables, (c) UWMW model with 1 LMV in the observations, (d) EWEW model with LMVs in the variables and $\lambda = 0.97$ (only the loadings corresponding to the k - and $(k - 1)$ -th sampling time point are shown for the sake of comparison) and (e) EWEW model with LMVs in the observations and $\lambda = 0.97$. In this approach, the NSD value is estimated as the average of the NSD values calculated at each k sampling time point.

7.11(c) and Figure 7.11(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 time point). On the other hand, adding LMV as new observations has 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 value than the EWEW-var and UWMW 1LMV-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 value of the loadings corresponding to the first PC for BW ($NSD = 1.524e - 01$, see Figure 7.6) and local models. In both cases, a total of $J \cdot K$ parameters are estimated from the data of N 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 latter. 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 three-way into two-way array should be selected depending on the data at hand [94].

7.4.5 Adaptive hierarchical K -models

Firstly, the identification of the PCA model parameters is studied through the sum of squares captured for each PC. Figure 7.12 shows the explained sum-of-squares (SS) vs #PCs for an AHKM approach by using weighting factors $d = 0.2$ (see Figure 7.12(a)) and $d = 50$ (see Figure 7.12(b)). Weighting factor d is used to give less or more importance to the information collected

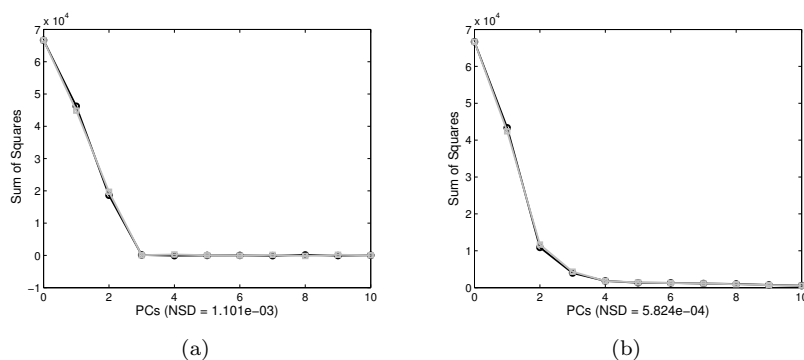


Figure 7.12. Adaptive hierarchical K -models and Trajectory C&S. Explained Sum of Squares (SS) vs #PCs extracted in the adaptive hierarchical K -models (AHKM) with (a) $d = 0.2$, and (b) $d = 50$. Note that PC #0 corresponds to the sum of squares remained after applying Trajectory C&S on batch data. NSD : normalized squared differences between the sum of squares vector captured by each PC of the two simulated datasets

at the current sampling time point with regard to the past information. This factor plays the same role as the exponential weighting factor in an EWMA model [103]. 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 -models approach uses memory of the past information, and therefore, this approach becomes similar to BW unfolding. As d grows further than one, the AHKM 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 7.12(a)), the sum-of-squares captured by PC#1 ($SS_1 \approx 4.550e + 04$) explains roughly 70% of the sum of squares remained after applying Trajectory C&S on batch data, i.e. SS_0 , and differs enough to that captured by PC#2 ($SS_2 \approx 1.920e + 04$). Hence, the corresponding loadings are expected to be stable, like in BW unfolding (see Figure 7.5 for comparison). Again, from PC#3 onward, 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 7.12(b)). The sums-of-squares captured by the first 2 PCs

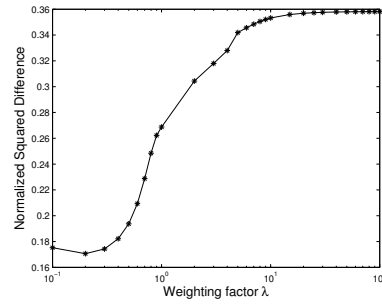


Figure 7.13. Adaptive hierarchical K -models and Trajectory C&S. Normalized squared differences (NSD) between the first loading vector of the two data sets as a function of the weighting factor d .

($SS_1 \approx 4.290e + 04$ and $SS_2 \approx 1.130e + 04$) differ each other enough to consider the loadings of the corresponding PCs stable. In contrast, the sum of squares captured from PC#3 onward 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 values computed for the loadings of the first PC are shown in Figure 7.13. 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 ($NSD = 1.706e-01$). 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 $d \geq 20$ (e.g. $NSD = 3.580e-01$ for $d = 50$), due to the adaptive hierarchical-model approach converges to the classical local K -models approach (the curve of Figure 7.13 converges to the NSD value of Figure 7.11(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.

7.5 Conclusions

In this chapter, 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 desirable, but not the only attribute. 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 three-way into two-way array. 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 $J \cdot K$ averages and $J \cdot K$ standard deviations are computed from N 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 $N \cdot K$ 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, decreas-

ing their stability. Depending on the type of rearranging method performed on the three-way batch data array, 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 chapter 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 time 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. Also, the number of PCs in a model can be made larger to obtain robustness to sensor failures or missing data. Hence, the modeling approach must not be selected from the consideration of the parameter stability alone. The findings of this chapter need to be combined with those from the complementary research works [87, 89] for a proper choice.

As a conclusion, three are the 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 online prediction.

Implications of batch synchronization in process monitoring

Part of the content of this chapter has been included in the following publications:

- [5] J.M. González-Martínez, R. Vitale, O. E. de Noord and A. Ferrer. Effect of synchronization on bilinear batch process modeling, *Industrial & Engineering Chemistry Research*, 53(11): 4339–4351, 2014.
- [30] J.M. González-Martínez, R. Vitale, O.E. de Noord and A. Ferrer. Does synchronization matter in Batch Multivariate Statistical Process Control? *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 58, Djurönäset (Sweden), 2013.
- [29] A. Ferrer, J.M. González-Martínez and J. Camacho. Practical implications of synchronization, preprocessing and bilinear modeling of batch processes for MSPC. *In proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 26, Djurönäset (Sweden), 2013.

8.1 Introduction

In the scientific community, there is a widespread assumption that batch synchronization is only required if the batch trajectories have different duration [186, 187]. However, equal duration is not a sufficient condition to consider batch trajectories to be synchronized. Some authors proposed methods that only address the problem of the different duration among batches without considering the overlap of the key process events, such as the OWU-TBWU approach (also known as variable-wise unfolding) [62, 83], implemented in the SIMCA software package by Umetrics. Wold and coworkers [in reference to [83]] *use stretched and contracted values of local time, and thus re-express each batch in terms of interpolated data so that in the end each batch has the same number of rows for the same sampling time points*. Martin et al. [188] stated that the observation level—the first step of the OWU-TBWU approach—is an alternative approach to the analysis of batches with unequal lengths. Ündey et al. [189, 96] ensured that OWU-TBWU *provides solutions to data synchronization* and in [190] that *equalization and alignment of trajectories are required if batches in this reference set (X) are of different lengths*. Simoglou et al. [88] reported a comparative evaluation of four multivariate statistical process control techniques for online monitoring, pointing out that Wold et al.’s proposal is able *to overcome the problem with unequal batch lengths*. Fransson et al. [161] indicated that *there exists a number of methods including [...] using local batch time as the response vector in a PLS model when unfolding the three-way array in the variable direction*, which the OWU-TBWU is based on, *to deal with varying batch-to-batch process time*. In case these methods do not work out, these authors suggested using time-alignment algorithms like DTW [76] and COW [161]. Eriksson et al. [64] also stressed *One assumption of most methods used for analysis of batch data is that batches have equal duration and are synchronized, i.e. measurements are made at the same sampling time points. If this is not the case, the batch data need to be aligned*. Facco et al. [191] emphasized that *an advantage of variable-wise unfolding is that the uneven batch duration problem in process modeling is spontaneously solved without trajectory synchronization*. Mingxing et al. [192] pointed out that variable-wise unfolding *has the advantage of being very simple to carry out, because it can be applied in a straightforward way to sets of batches which have*

different time duration, without the need of synchronizing the batch length". Lee et al. [186], Zhao et al. [187], Yao and Gao [95], and Huang and Qu [193] claimed that OWU-TBWU does not require that all batches be of equal length. Recently, Wold et al. [62] stated that "for the OWU-TBWU approach, the data do not need alignment before the OWU modeling, but certainly the resulting OWU scores, \mathbf{T} , need alignment before the subsequent batch modeling using the unfolded scores". In addition, there are commercial software packages for batch process monitoring, e.g. SIMCA Release 13.0.3 [65], which only demand synchronization of batch trajectories when they have different length.

The main aim of the investigation in this chapter is: i) to demonstrate that batch synchronization is a crucial and necessary step prior to batch process modeling, no matter whether batches have equal duration or not, ii) to show that not all batches may need the same synchronization method to be aligned, and iii) to show the effect of inadequate synchronization that is not capable of tackling scenarios of multiple asynchronism on process monitoring. Two different synchronization approaches are evaluated under scenarios of multiple asynchronisms: the Multisynchro approach [4] and the method based on linearly expanding and/or compressing pieces of variable trajectories in the local batch time dimension [64], which is referred as the TLEC method¹. The selection of the synchronization techniques relies on both their capacity to differently handle asynchronous batch trajectories and their widespread use in the chemometrics community. The method based on an indicator variable [70] is not taken into consideration in this study because one of the major requirements for the usage of IV is not met. Specifically, one of the data sets under study is affected by class III asynchronism², therefore, none of the process variables represents the complete evolution of the batch process for synchronization. To proceed with the comparative study, five experimental cases with different types of asynchronism are designed using data from realistic simulations of a fermentation process of the *Saccharomyces cerevisiae* cultivation. Batches that produced on-spec product

¹TLEC is the default synchronization procedure implemented in SIMCA Release 13.0.3. In case the differences in batch length is greater than 20%, a maturity variable is used as the basis of batch synchronization instead of the local batch time [64]. Also, TLEC is one of the synchronization techniques provided in ProMV Batch Edition Release 13.02.

²Batches with different duration mainly caused by a partial incompleteness of the last process stage, irrespective of whether the process pace is the same or not across batches.

while operating under NOC and faulty batches containing the different types of asynchronism introduced in next section are simulated. These batches are used to evaluate the performance of both approaches in terms of synchronization quality and their influence in the monitoring schemes to accurately detect faults. For this purpose, the OWU-TBWU approach, which integrates the TLEC method for batch synchronization, is used for bilinear batch process modeling.

The chapter³ is split in four sections. Section 8.2 introduces the materials used in the comparison. Section 8.3 discusses the results of the comparison of the two synchronization methods in terms of synchronization quality. In Section 8.4, the impact of inappropriate synchronization in fault detection is addressed. Finally, some conclusions are drawn in Section 8.5.

8.2 Material

Five simulated cases with different types of asynchronism are generated for the calibration and test data sets: a) case #1: batches with equal duration and different evolution pace in the last stage of the batch run -i.e. class I asynchronism (see black and grey lines in Figure 8.1(a)); b) case #2: batches with different duration produced by natural process variability and key process events not overlapping across batches -i.e. class II asynchronism (see black lines in Figure 8.1(b)); c) case #3: a combination of batches with different duration due to partial incompleteness of the latest process stage and key process events overlapping -i.e. class III asynchronism (see black lines in Figure 8.1(c))- and batches of equal length with slightly different process pace -i.e. class I asynchronism (see grey lines in Figure 8.1(c)); d) case #4: a combination of batches with different duration produced by a shift in the start of the batch and the same evolution pace across batches -i.e. class IV asynchronism (see black lines in Figure 8.1(d)), and batches of equal length with slightly different process pace -i.e. class I asynchronism (see grey lines in Figure 8.1(d)); and e) case #5: incomplete batches at the latest process stage with events not overlapping across batches -i.e. class III asynchronism (see black lines in Figure

³The figures of this chapter have been reprinted with permission from "Effect of Synchronization on Bilinear Batch Process Modeling. J. M. González-Martínez, R. Vitale, O. E. de Noord, and A. Ferrer. *Industrial & Engineering Chemistry Research* 2014, 53 (11), 4339-4351". Copyright 2014 American Chemical Society

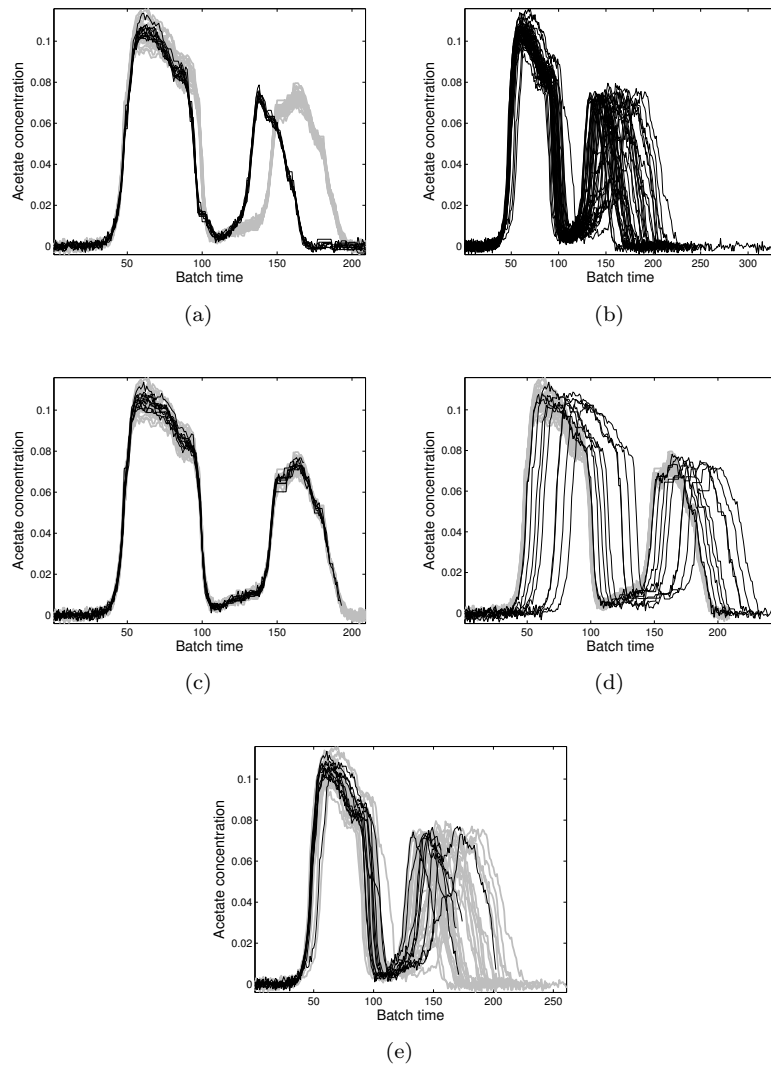


Figure 8.1. Trajectories of the acetate concentration corresponding to 40 NOC batches in four different scenarios of asynchronisms: (a) case #1: class I asynchronism (black and grey lines); (b) case #2: class II asynchronism (black lines); (c) case #3: class III (black lines) and class I (grey lines) asynchronisms; (d) case #4: class IV (black lines) and class I (grey lines) asynchronisms; and (e) case #5: class III (black lines) and class II (grey lines) asynchronisms.

8.1(e)) and complete batches with different duration caused by natural variability -i.e. class II asynchronism (see grey lines in Figure 8.1(e)).

For the generation of these asynchronisms, the three-way arrays belonging to Set #5 are used. The batches of the three-way array $\tilde{\mathbf{X}}_{10}$ are split up into NOC calibration and test data sets composed of 40 and 20 batches, respectively. These batches jointly with the 10 test faulty batches for each fault ($\tilde{\mathbf{X}}_{11}$, $\tilde{\mathbf{X}}_{12}$ and $\tilde{\mathbf{X}}_{13}$) are manipulated following the procedure explained in Section 6.5 to generate the different types of asynchronisms mentioned above⁴. As a result, the following structures were created per each case: (a) case #1: class I asynchronism (40 NOC calibration batches, 20 NOC test batches, and 10 faulty batches for each fault); (b) case #2: class II asynchronism (40 NOC calibration batches, 20 NOC test batches, and 10 faulty batches for each fault); (c) case #3: class III (10 NOC calibration batches, 5 NOC test batches, and 5 faulty batches for each fault) and class I (30 NOC calibration batches, 15 NOC test batches, and 5 faulty batches for each fault) asynchronisms; (d) case #4: class IV (10 NOC calibration batches, 5 NOC test batches, and 5 faulty batches for each fault) and class I (30 NOC calibration batches, 15 NOC test batches, and 5 faulty batches for each fault) asynchronisms; and (e) case #5: class III (10 NOC calibration batches, 5 NOC test batches, and 5 faulty batches for each fault) and class II (30 NOC calibration batches, 15 NOC test batches, and 5 faulty batches for each fault) asynchronisms. In total, 5 data sets containing 60 NOC batches and 30 faulty batches are generated.

Comparison of synchronization methods

The two synchronization approaches under study are evaluated in scenarios of multiple asynchronisms from two different perspectives: i) synchronization quality and ii) accuracy of the monitoring schemes designed from synchronized batch data to efficiently detect abnormal situations. Different metrics are defined for comparison purpose.

⁴The simulated data manipulated to generate the asynchronisms under study are available in MVBatch software (at request) and on the website of Industrial & Engineering Chemistry Research, journal in which these data sets were published.

The synchronization quality is defined as the accuracy of a synchronization procedure to make the key process events overlapped throughout the batch run, ensuring the same process pace in all batches. To assess this factor in the two synchronization techniques under study, the variability of the resulting synchronized batch trajectories around their mean trajectory is obtained. This can be measured by the standard deviation vector after the mean trajectory has been subtracted from batch data. The lower the difference among standard deviation vectors, the higher the synchronization quality.

Data corresponding to the experimental cases simulated are used to build the monitoring schemes based on the OWU-TBWU approach explained in Chapter 4. The control limits of the control charts designed at the OWU level are initially estimated from theoretical results and subsequently adjusted for an imposed significance level α . The aim of this readjustment is to ensure that the OWU scores and *Distance to model* (DMoDX) control charts have the same percentage of faults detected by chance for a batch under normal operating conditions.

The performance of the monitoring schemes based on the OWU approach for the different synchronization approaches will be compared using two indices [91, 90, 175]. To evaluate the proper adjustment of the control limits, the OTI risk from the batches under NOC belonging to the test data set is computed. This value can be understood as the actual percentage of faults in the NOC batches or the false alarms rate and is estimated as follows:

$$OTI = 100 \cdot \frac{nf}{N_{NOC} \cdot K} \quad (8.1)$$

where nf denotes the total number of faulty sampling time points and N_{NOC} the number of NOC batches considered. Note that the adjustment of the control limits can be considered appropriate when the OTI value is close to the imposed significance level α .

To assess the accuracy of the control chart in terms of fault detection, the OTII risk is calculated as:

$$OTII = 100 \cdot \frac{nnf}{N_{faulty} \cdot l} \quad (8.2)$$

where nnf represents the number of non-signaled faulty sample points, N_{faulty} the number of faulty batches and l the length

Table 8.1. PLS models results in the different cases of asynchronism for the two synchronization procedures under study. LVs, R^2 , and Q^2 stand for the latent variables extracted, the goodness of fit and prediction, respectively.

CASE	APPROACH	no. LVs	R^2	Q^2 (%)
#1	TLEC	4	89.5%	98.4%
	Multisynchro	4	90.1%	99.3%
#2	TLEC	4	91.1%	95.5%
	Multisynchro	4	90.4%	99.2%
#3	TLEC	4	87.6%	93.3%
	Multisynchro	4	90.8%	99.3%
#4	TLEC	4	91.4%	98.6%
	Multisynchro	4	89.8%	99.3%
#5	TLEC	4	89.5%	94.5%
	Multisynchro	4	91.8%	98.8%

of the faulty time period. To consider the monitoring system has a good performance, the OTII value should be close to 0 as much as possible.

As a first step of this study, the calibration batch data set of each type of asynchronism under study are synchronized using the Multisynchro approach. In this case, the DTW algorithm is used for batch synchronization⁵. The reference batch selected in each scenario of asynchronism was the closest one to median length from the batches arranged for the iterative synchronization. The rest of conditions and constraints are set according to the specifications in [1]. Secondly, a cross-validated PLS model for each of the five synchronized calibration data sets are fitted (see results in Table 8.1). For the sake of comparison, the TLEC method was applied for batch synchronization as well. First, a PLS model is fitted in each of the calibration raw batch data sets. Second, the corresponding OWU scores and DModX statistic derived are synchronized by using TLEC following the steps in Chapter 4.2.2.

⁵If the aim is to design monitoring schemes for real-time applications, the RGTW algorithm should be chosen.

8.3 Effects of asynchronisms in synchronization quality

The standard deviation vector of the corresponding synchronized OWU scores for each scenario of asynchronism are computed for comparison purpose. Note that the length of these vectors differs among synchronization approaches since batch duration is different. To make them equal in length, the standard deviation vectors derived from data synchronized by TLEC are linearly interpolated to the same number of values as those obtained from data synchronized by Multisynchro. The resulting vectors for all components are shown in Figure 8.2. This figure reveals that when data are synchronized using the Multisynchro approach, the standard deviation values are lower (black stars lines) than those obtained from data synchronized by the TLEC-method (red empty circle lines). It implies that the Multisynchro approach clearly outperforms the TLEC method in terms of synchronization quality. Also, the standard deviations from TLEC show that this synchronization approach is less vulnerable to class I and class IV asynchronisms (lower values in cases #1 and #4, respectively, in Figure 8.2) than to class II and class III asynchronisms (higher values in cases #2, #3 and #5 in Figure 8.2).

In the cases with different evolution pace, cases #2 and #5 show higher standard deviations for all the scores than in case #1. This high variability is basically produced by the type of asynchronism and the way how the TLEC method addresses the batch synchronization. Case #2 and case #5 have in common that the corresponding raw batch trajectories are different in length due to normal variability of the process. It has two main effects: i) the key process events do not overlap in all batches from the early stage of the process, causing differences in the evolution pace across, and ii) the duration among batches differs much more than in the other cases of asynchronism. In contrast, batch trajectories belonging to case #1 have equal length and different evolution pace only at the last stage of the process. As the TLEC method linearly interpolates data without considering the overlapping of the key process events, the asynchronism present in the raw batch data is inherited in the resulting OWU scores (see OWU scores and DModX control charts for the NOC test batches affected by cases #1, #2 and #5 of asynchronism in Appendix C). Hence, the normal

process variation is dramatically affected by the inheritance of the asynchronism. The more different the evolution pace and the duration among batches, the higher the variability.

Comparing the standard deviation vectors of cases #2 and #5, higher values are observed at the last stage of the process (last 50 sampling time points) in the latter than in the former. These differences are caused by incompleteness of some batches in case #5 (combination of class II and class III asynchronisms). In this scenario, TLEC-based synchronization worsens asynchronism by introducing misaligned points and flat profiles, which produce artificial variability not related to normal process variation. The same phenomenon occurs with batches from case #3 only affected by class III asynchronism. The difference lies in the variability associated with the batches of case #3, which is apparently lower in comparison to the batches of case #5 (see Figure 8.2(c) and Figure 8.2(e) for comparison). As a conclusion, the larger the incompleteness of the batches, the higher the variability.

These results show that the accuracy of the synchronization approach to make the key process events overlap in all batches is crucial in bilinear process modeling. TLEC-based synchronization is not focused on ensuring the same process pace in all batches, but on duration equality (TLEC linearly interpolates data without considering the overlapping of the key process events). This means that the different types of asynchronism present in raw batch data are inherited in the latent structure. Hence, the derived OWU scores and DModX statistic have undesired variability that may seriously affect the performance of the monitoring schemes.

8.4 Effects of synchronization in process monitoring

To illustrate the effect of the propagation of asynchronisms in the performance of monitoring schemes, the OWU control charts for the first latent variable for each type of asynchronism after TLEC-based synchronization are shown in Figure 8.3. The resulting synchronized OWU scores belonging to the 10 out of 40 batches in which specific types of asynchronism were simulated (see black lines in Figure 8.3) clearly show the propagation of trajectory variability from raw batch data to the latent structure. In case #1, the difference of the process evolution at the last stage of the

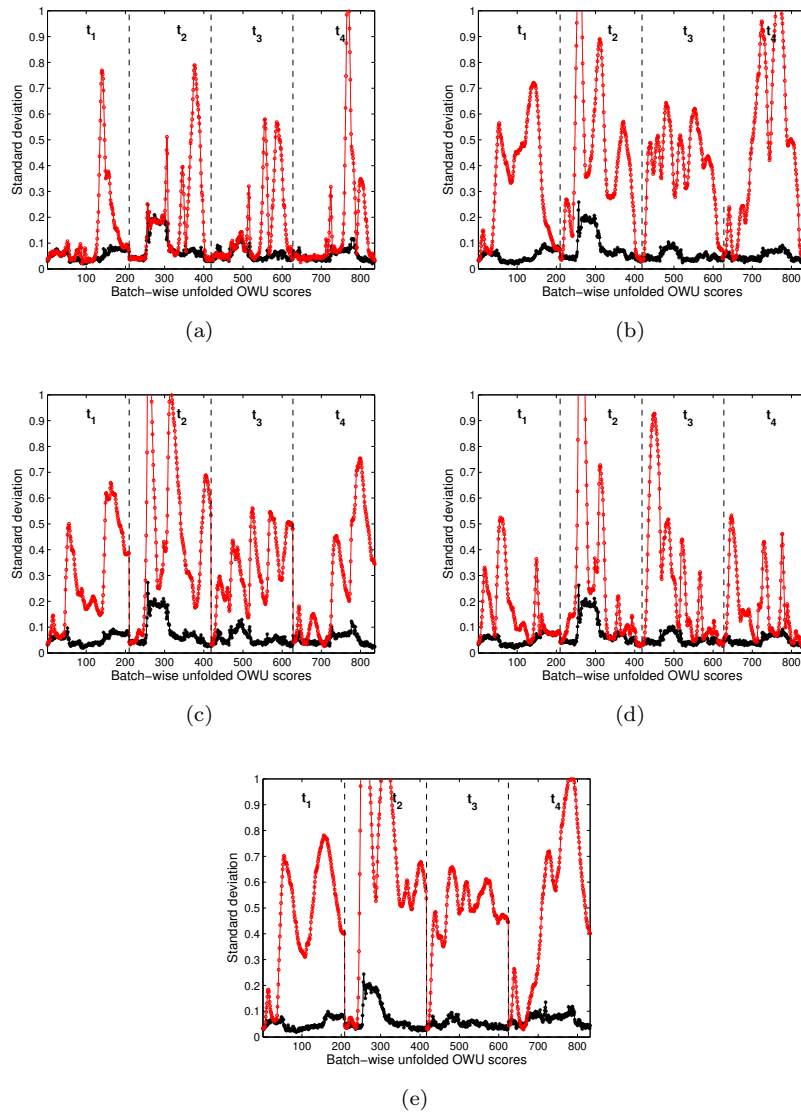


Figure 8.2. Comparison of the standard deviation vectors obtained from the four OWU scores (separated by dashed lines) for the Multisynchro approach (black stars lines) and by the TLEC method (red empty circles lines) for all the scenarios of asynchronism: (a) case #1, (b) case #2, (c) case #3, (d) case #4, (e) case #5.

process remains in the OWU scores since TLEC does not take any action (see black and grey lines in Figure 8.3(a)). The OWU scores belonging to case #2 show the same type of asynchronism than raw batch data, which is more prominent from the 50th sampling time point onward. These plots confirm that the asynchronism present in batch data is inherited in the OWU scores due to inappropriate synchronization. In case #3 (see Figure 8.3(c)), the OWU scores show a gradual asynchronism from the first sampling time points onward, where the last stage of the process becomes more severe. This is produced by the incompleteness of the batch trajectories -i.e. by the missing trajectory belonging to the last period of the process. Regarding case #4, the shift at early stage of the process stays in the OWU scores. Nevertheless, this effect is corrected gradually by the execution of a high number of linear interpolations at the end of the runs (see black lines in Figure 8.3(d)). Finally, in case #5 (see Figure 8.3(e)), a similar phenomenon to case #3 is observed given the similarity between the class of asynchronism. The difference lies in the 30 out of 40 batches (grey lines), whose key process events do not coincide through the batch run in case #5. This asynchronism can be also observed in the OWU scores.

The existence of asynchronism in the OWU scores produces a high trajectory variability between batches, as discussed before (Figure 8.2). Hence, the control limits estimated from data need to be wide enough to meet the ISL requirement (5%). The higher the variability, the wider the control limits. When the batch trajectories (and therefore the OWU scores when TLEC-based synchronization is applied) have different process evolution and lack of overlapping in the key process events (i.e. cases #2, #3 and #5), the control limits will be wider (see Figures 8.3(b), 8.3(c) and 8.3(e)) than for the rest of types of asynchronism (see Figures 8.3(a) and 8.3(d)). Unnecessary wide control limits may cause that some types of faults cannot be properly detected and diagnosed, putting safety and reliability of the process at risk. When the raw batch trajectories are synchronized by taking into consideration the different types of asynchronism (i.e. applying the Multisynchro approach), the same process evolution and occurrence of the process events in time are ensured. This yields synchronized OWU scores with narrower control limits (see Figure 8.4) than those obtained when OWU scores are synchronized by TLEC (see Figure 8.3). To study the risk of applying an inappropriate synchronization

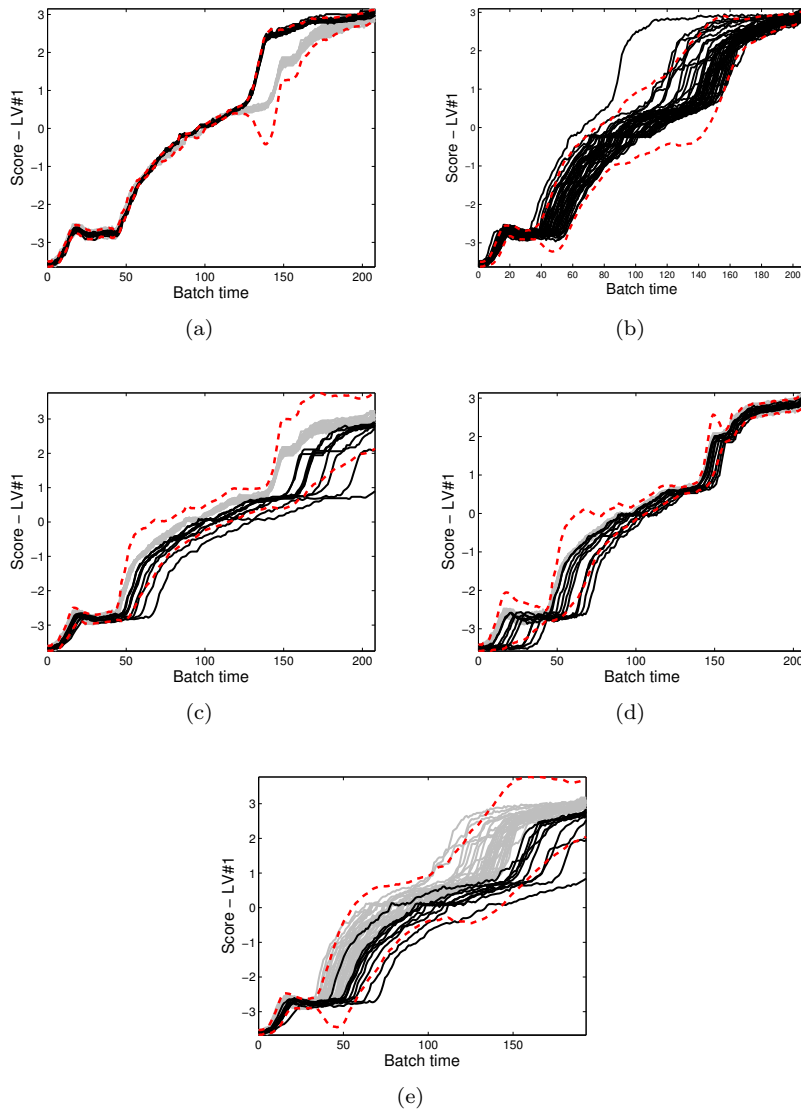


Figure 8.3. OWU scores control charts of the first LV monitoring from the NOC calibration data set with different asynchronism patterns after TLEC-based synchronization: (a) case #1, (b) case #2, (c) case #3, (d) case #4 and (e) case #5. Red dashed lines represent the control limits at 95% confidence level. Batches with simulated asynchronisms in black lines.

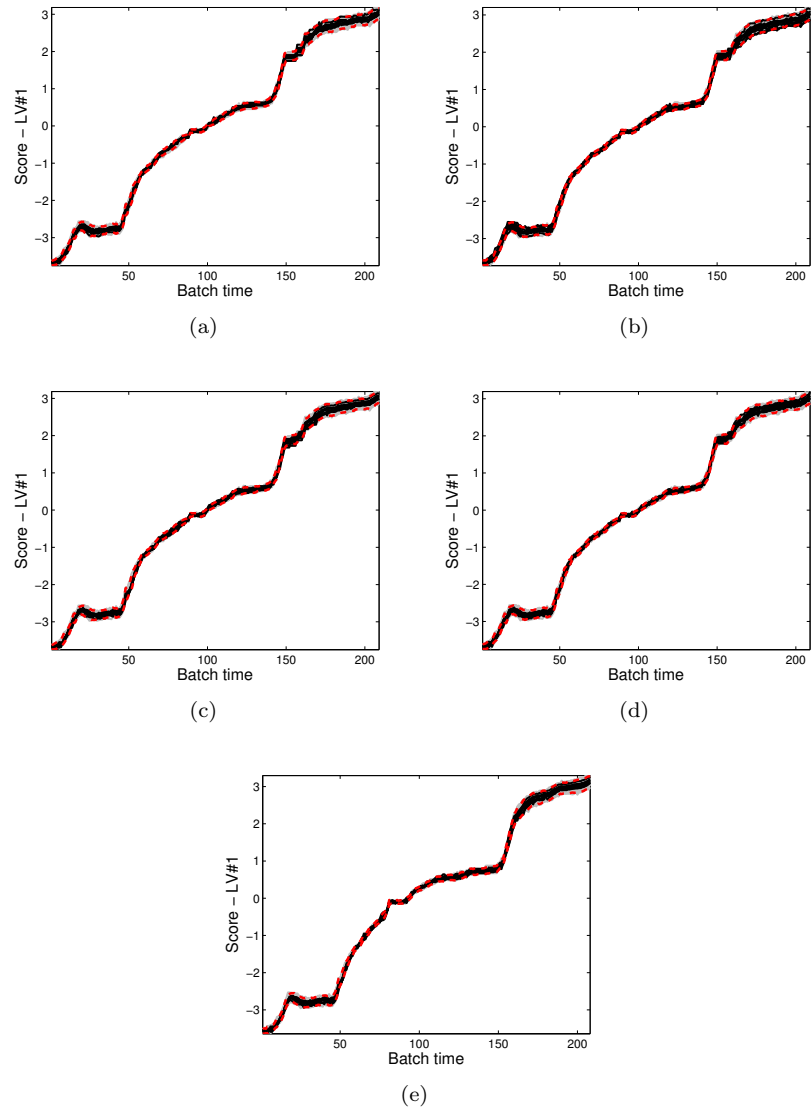


Figure 8.4. OWU scores control charts of the first LV from the NOC calibration data set with different asynchronism patterns after Multisynchro-based synchronization: (a) case #1, (b) case #2, (c) case #3, (d) case #4 and (e) case #5. Red dashed lines represent the control limits at 95% confidence level.

Table 8.2. OTI values for the control charts based on the DModX statistic and the OWU scores (t_a) for the two synchronization procedures under study. ISL=5%.

CASE	APPROACH	Test-NOC				
		OTI				
		DModX	t_1	t_2	t_3	t_4
#1	TLEC	4.6%	2.9%	3.4%	4.7%	3.1%
	Multisynchro	4.2%	5.3%	5.0%	7.4%	4.7%
#2	TLEC	5.5%	1.9%	2.8%	3.0%	3.2%
	Multisynchro	4.8%	6.8%	6.0%	7.3%	5.9%
#3	TLEC	5.0%	6.6%	6.2%	7.2%	6.7%
	Multisynchro	3.5%	3.6%	2.6%	5.1%	3.3%
#4	TLEC	4.3%	2.7%	2.0%	3.9%	2.5%
	Multisynchro	4.8%	8.1%	6.3%	10.6%	6.8%
#5	TLEC	4.1%	6.7%	6.4%	7.2%	6.6%
	Multisynchro	2.3%	3.7%	3.6%	5.4%	3.2%

in fault detection, the monitoring performance of control charts obtained from batch data synchronized by Multisynchro and TLEC on the raw OWU scores and DModX are compared. To carry out this comparative study, the OTI and OTII values are calculated (see Tables 8.2 and 8.3, respectively). For a fair comparison, the control charts of the two synchronization approaches applied to the five different types of asynchronism should present similar OTI values. Otherwise, the OTII results are not comparable. This can be achieved by re-adjusting the theoretical control limits estimated using the calibration data sets. The OTI values shown in Table 8.2 are computed using the independent test sets. As can be appreciated, the OTI values are quite similar for both synchronization approaches in all the types of asynchronism, being close to 5% - which is the ISL of the limits.

To carry out the comparison of OTII values, readers should not focus the attention on the specific percentage shown in Table 8.3 - since it is dependent on the magnitude of the process fault- but in the difference of OTII values between the two approaches in each asynchronism case. At first glance, the Multisynchro approach seems to outperform the TLEC method

Table 8.3. OTII values for the control charts based on the DMoDX statistic and the OWU scores (t_a) computed from test batches containing three different types of faults and five different types of asynchronism. These asynchronous faulty batches are synchronized using the Multisynchro approach and the TLEEC method. Lowest OTII values in each case, approach and type of fault in bold.

CASE	APPROACH	Test type I-fault				Test type II-fault				Test type III-fault						
		OTII				OTII				OTII						
	DMoDX	t_1	t_2	t_3	t_4	DMoDX	t_1	t_2	t_3	t_4	DMoDX	t_1	t_2	t_3	t_4	
#1	TLEEC	55.5%	45.9%	71.1%	68.8%	70.1%	66.8%	57.3%	61.6%	56.4%	52.6%	79.4%	31.3%	91.1%	89.0%	89.9%
	Multisynchro	43.8%	35.8%	66.1%	62.3%	53.6%	55.2%	49.9%	43.9%	38.8%	39.2%	75.6%	23.5%	89.8%	85.4%	93.6%
#2	TLEEC	61.8%	82.3%	90.6%	84.6%	88.7%	60.4%	65.0%	67.5%	81.9%	66.8%	93.2%	99.0%	97.2%	98.6%	98.8%
	Multisynchro	42.9%	38.0%	66.6%	55.5%	54.1%	58.5%	54.3%	43.2%	36.9%	43.3%	74.7%	20.3%	91.9%	86.4%	92.8%
#3	TLEEC	67.7%	77.0%	90.8%	88.8%	70.4%	69.0%	78.9%	72.4%	71.7%	61.1%	87.1%	90.7%	85.8%	84.6%	85.5%
	Multisynchro	44.0%	37.2%	69.5%	62.2%	60.2%	57.3%	50.7%	47.7%	44.7%	47.8%	73.4%	21.5%	92.8%	89.4%	95.2%
#4	TLEEC	63.8%	65.6%	81.1%	73.3%	69.0%	65.1%	62.7%	55.1%	62.1%	48.0%	85.7%	74.1%	70.8%	76.3%	72.5%
	Multisynchro	42.0%	35.9%	66.4%	56.4%	48.3%	57.0%	54.1%	43.3%	37.0%	41.1%	72.7%	21.0%	88.2%	78.4%	89.5%
#5	TLEEC	66.5%	83.0%	88.1%	82.4%	81.7%	59.9%	73.0%	64.8%	78.7%	61.3%	92.3%	89.1%	87.1%	88.3%	94.2%
	Multisynchro	51.5%	37.1%	62.8%	53.8%	73.4%	60.9%	50.8%	53.6%	50.3%	46.3%	65.4%	26.2%	90.2%	87.7%	85.9%

in terms of accurate fault detection using the OWU-TBWU approach in the first level, irrespective of the type of fault and asynchronism added in batch data. For the first two types of faults, the OTII values derived from batch data synchronized by the Multisynchro approach are lower than those obtained from batch data synchronized by the TLEC method. This can be observed both for all the scores and the DmodX statistic. Note that the OTII values belonging to the first scores are remarkably lower than for the rest of scores. This is because the set of OWU PLS scores captures the average trajectories of the highly correlated variables in different ways at different sampling time points throughout the batch. Remember that these two types of faults illustrate different operating conditions induced from the start of the batch run. Hence, the first score most likely captures the average trajectory of the variables at an early stage of the batch process, when the fault is signaled by the control chart (see OWU scores control charts for type I and type II faults shown in Appendix C). Concerning the type III fault, the OTII values of the scores of the first latent variable and the DmodX statistic are lower for the Multisynchro approach than for the TLEC method. Note that these differences are more prominent in the scores than in the DmodX statistic. This is caused by the type of fault, which produces a different process performance with a break of the data correlation structure after some minutes the fault started. Hence, more samples beyond the control limits are expected in scores than in DmodX. Nonetheless, these differences are not equally important in the Multisynchro approach and in the TLEC method. In the former, the differences of the OTII values between the scores belonging to the first latent variable and the DmodX are considerably higher than in the latter. Thanks to a better synchronization considering the types of asynchronism carried out by the Multi-synchro approach, the variability is reduced and the control limits are better fitted to the actual process variability. This enables the monitoring scheme to detect the faults with more accuracy, reducing the number of faulty samples not signaled (see OWU scores control charts for type III fault shown in Appendix C). Another issue worth being emphasized is the clear differences observed in the OTII values between the different types of asynchronism for all the faults when the TLEC-based synchronization is performed. The OTII values belonging to the scores pointed out above and the DmodX for case #1 and case #4 are lower than for the rest of the cases. This leads to suspect that the degree of

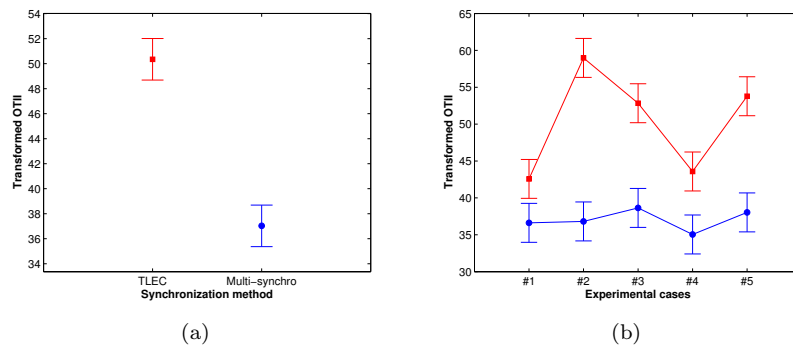


Figure 8.5. LSD intervals (95% confidence) for the arc sin of the OTII values for (a) the simple effect of the synchronization method and (b) the interaction between the experimental cases containing different asynchronisms and the synchronization method. Red and blue LSD intervals represent the TLEC and Multisynchro methods, respectively.

asynchronism may affect the accuracy of the monitoring schemes when the TLEC method is used for batch synchronization.

With the aim of determining whether there exist statistical significant differences in the OTII values of the monitoring schemes among the synchronization methods and the types of asynchronism, an ANOVA is performed on the OTII values (arcsin square root transformation is used). The outcomes of this analysis determined that the simple effect of the synchronization method and the interaction between synchronization method and type of asynchronism are statistically significant (p -value < 0.05). In order to find out in what synchronization approach and type of asynchronism the differences lie in, the 95% confidence LSD intervals are computed (see Figure 8.5). When the Multisynchro approach is used for batch synchronization, the percentage of faults detected as NOC are statistically lower on average (OTII=37%) in comparison to when the TLEC method is applied (OTII=51%)(see Figure 8.5(a)). Unlike TLEC-based synchronization, Multisynchro-base synchronization is robust to the presence of the different types of asynchronism simulated in the process variables since no statistical significant differences are found for the different types of asynchronisms (see LSD intervals in Figure 8.5(b)). Depending on the nature of the asynchronism, the OTII values of the monitoring scheme

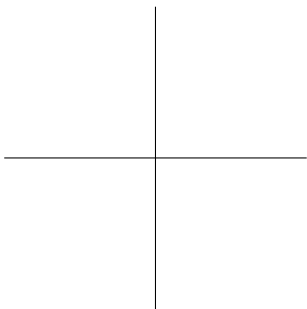
when the OWU scores are synchronized by TLEC are affected in lesser or greater extent. In particular, in cases #2, #3 and #5 the OTII values are, on average, statistically higher than those obtained in cases #1 and #4. In the former scenarios, batches show high variability in the evolution pace and batch duration, and there are in addition incomplete batches. An interesting result is observed in Figure 8.5(b) with case #1. In this case (see Figure 8.1(a)) batches have the same length, and therefore, by applying TLEC trajectories, do not change. Nevertheless, as shown in Figure 8.5(b), LSD intervals between TLEC and Multisynchro do not overlap yielding OTII values, that are on average, significantly higher for TLEC. This proves that equal length does not necessarily assure synchronized data, and by choosing a good synchronization method the performance of the monitoring scheme may improve. These differences in terms of capability to detect faults as a function of the type of asynchronism for TLEC are in accordance with the differences observed in terms of quality of synchronization summarized in Figure 8.2. The higher the variability in the synchronized trajectories, the lower the performance of the monitoring schemes in fault detection.

8.5 Conclusions

The main conclusion drawn from the discussion and results of the simulation study is that synchronization is a critical and necessary preliminary step prior to bilinear batch process modeling. Quality of batch synchronization is one of the critical factors that affects the performance of the monitoring schemes in fault detection. When the key process events do not overlap at the same point of process evolution ensuring the same process pace in all batches, the capability of the monitoring schemes for fault detection is dramatically reduced. Contrary to what is often assumed in practice (as well as in commercial software as e.g. SIMCA Release 13.0.3 by Umetrics), equal length does not guarantee synchronized batches. Simple methods like TLEC (implemented in SIMCA) linearly interpolate data without considering the overlap of the key process events. Hence, the asynchronism present in the raw data is inherited in the resulting OWU scores and DModX statistics. The increase of the variability in batch trajectories due to inappropriate synchronization has an important negative effect. The higher the variability, the

lower the performance of the control charts in fault detection. Multisynchro is a promising approach to perform reliable batch synchronizations with different classes of asynchronisms, and can be used for both post-batch and real-time applications.

Part IV
Application



Modeling process dynamics through multivariate models and control charts.

Part of the content of this chapter has been included in the following publications:

- [9] J.M. González-Martínez, J. Camacho and A. Ferrer. Modeling Time-varying Process Dynamics through Latent Models and Control Charts. In elaboration.
- [25] J.M. González-Martínez, A. Ferrer, F. Arteaga, D. Aguado and J. Ribes. Multivariate Statistical Process Control of a Continuous Biological Removal Process: Designing efficient monitoring schemes robust to sensor malfunctioning. *In proceedings of the 2nd European Conference on Process Analytics and Control technology (EUROPACT)*, page 149, Glasgow (UK), 2011.

9.1 Introduction

In recent years, most *Waste Water Treatment Plants* (WWTPs) have been upgraded to *Biological Nutrient Removal* (BNR) processes in order to be compliant with European legislation (Directive 91/271/EEC) [1]. The objective is to ensure safe and stable operation while meeting increasingly stringent effluent criteria. With these new regulations for quality monitoring of WWTPs, there has been a growing interest in the last decades to develop process monitoring methods for automatic detection and identification of process and instrument faults [194].

The fault detection, isolation and diagnosis involves the usage of either models whose structure is based on fundamentals and whose parameters are estimated from data (mechanistic models) [195, 196], or models whose structure and parameters are empirically identified from plant data techniques (data-driven models) [197]. Traditional monitoring methods based on mechanistic models have been extensively used [41, 198, 199, 200, 201]. However, these casual models impose a structure that relies on many assumptions, some of which cannot be entirely justified [202, 203]. In contrast, data-driven methods are based on non-casual models, which capture the correlation structure existing among the process variables during normal operation conditions, where only "common cause" variation is present in data. This feature makes these models suitable for process monitoring, where the interest is on detecting non-common variation that affects the normal behavior of the process [204, 205, 206]. Incipient work on PCA and PLS [46, 114, 207, 208, 209, 210] enabled the rapid development of the area of MSPC in different industrial sectors, such as the semiconductor manufacturing [211, 212, 213], the steel industry [214, 215, 216], and the chemical industry [217], among others. The application of data-driven statistical methods is relatively more recent in the wastewater treatment process discipline. PCA was applied to WWTP data to classify the operation regimes with the aim of obtaining a decision support system [218]. In [219], PLS was used to model an activated sludge plant, and the authors reported that PLS is a promising tool for detection of shifts in variables. PCA and PLS were later successfully applied for disturbance detection and prediction of wastewater treatment operation [220], leading to the conclusion that MSPC techniques can be successfully applied to WWTPs. Later, PCA was used as a tool to reduce

the dimensionality of the problem and extract meaningful components that explain the biological behavior of activated sludge wastewater treatment data [221]. Recently, critical process faults simulated by using the COST benchmark [222] under different water conditions were well detected by a multi-mode PCA-based monitoring scheme [223].

MSPC techniques suffer from setbacks in WWTP that hampers their successful application for process monitoring [224, 225, 226]: i) changing conditions (e.g. significant seasonal, daily and weekly fluctuations in flow rate and composition of waste water, discrete events that may occur occasionally such as episodes of heavy rain, toxic spills, tips organic load), ii) dynamic relationships between variables with a wide range of time constants, iii) non-linear relationships between variables (e.g. temperature-dependent kinetics), and iv) poor data quality and reliability of sensors. In the light of seeking the acceptance from plant operators of the monitoring schemes based on multivariate methods, which should be robust to the severe operating conditions of WWTPs, these challenges must be tackled.

The non-stationary, non-linear and time-varying relationships in a WWTP are namely caused by recirculation streams, generally by the internal recirculation and sludge recirculation [224, 227]. Hence, the phenomena in the first reactor of the plant is not only dependent on what happens in the influent flow (variances and instantaneous cross-covariances -the instantaneous relationships of the process variables) but also on what happens in the other reactors and the settler flow (auto-covariances and lagged cross-variances of certain order -the dynamic relationship of the variables). The application of multivariate classical methods like PCA or PLS without modeling the dynamic behavior of the process may jeopardize the performance of monitoring schemes due to the distortion caused by autocorrelation in data. Some authors proposed to introduce ideas from time series modeling to tackle the autocorrelation problem and capture process dynamics. Specifically, modeling a data matrix augmented by time-lagged variables by PCA showed good results in terms of detecting the occurrence of small disturbances in dynamic processes [84, 228]. However, the estimation of a variance-covariance matrix from a lagged version of the original data matrix implies the identification of many parameters [224], and hence it might impact the parameter stability of the models irrespective of the type of process [3]. To solve this problem, a

multivariate time series analysis framework using PCA and PLS was proposed [229, 230]. This framework consists of reducing the dimension of the problem by extracting a few PCs that are suitable for time series modeling. The identification of the type and order of the times series is accomplished by lagging each PC score, fitting a PLS model on this new time-lagged score matrix and the inspection of the resulting PLS regression coefficients, which reflects the auto-correlations of the scores. Finally, the appropriate time series model is fitted for each score. Another alternative is fit a canonical-variate state-space model equivalent to an autoregressive moving-average time-series model to reduce the dimensionality and capture the dynamics [231]. A dynamic monitoring system for multiscale fault detection based on dynamic PCA [84], and the monitoring of individual eigenvalues of generic dissimilarity measure was proposed for wastewater treatment processes [232]. The conclusion of this proposal was that the modeling of the auto-correlations and cross-correlations enhanced the detection of faults in dynamic processes such as WWTPs.

The monitoring schemes based on MSPC methods typically rely on the Hotelling's T^2 [113] and SPE [115] statistics for the detection of abnormal situations by using Shewhart control charts, which characterize the variability in the latent and residual space, respectively. However, other statistical distances have been proposed as alternative fault detection indices [117, 233, 234]. If an out-of-control operation is signaled by the monitoring system, the next step is to find the source cause of the deviation. The data-driven methods for fault diagnosis can be roughly classified into two categories [203]: i) methods that associate process behavior patterns to specific faults and ii) methods that relate the process variables that significantly contribute to the deviation of the multivariate statistics. Those belonging to the first category have in common that a set of faulty trends is required, and the association between process data and faults is performed by using pattern recognition techniques such as DTW [166], heuristic rules [235], statistics pattern analysis [236], black-box models such as ANNs [237, 238, 239] and Hidden Markov Models [240], fault identification indices based on fault reconstruction [241, 242, 243], or by using statistical discriminatory distance [244] or classification techniques, such as Fisher Discriminant Analysis [245, 246, 247], Discriminant PLS [248], Support Vector Machines [249], and Correspondence Analysis [250]. The draw-

backs of most of these methods are their dependence on the type of process, the supervised nature of the methods that require a high computational (training) cost when additional source causes need to be added to the fault set or multiple simultaneous faults need to be diagnosed [203, 251]. Fault diagnosis can be straightforwardly conducted by the use of contribution plots [114, 252, 253] or their extension on the fingerprints contribution plots [254], which belongs to the second category. These plots identify the process variables that have significantly contributed to the inflation of Hotelling's T^2 and SPE statistics. The pitfall of this method is that plant personnel is needed to associate those variables that most contribute to the out-of-control signal with process equipment or external disturbances in order to elucidate the physical root causes of the abnormality.

In this chapter, the importance of the modeling approach and its influence on online fault detection, isolation and diagnosis in WWTP using projection methods to latent structures is discussed. Special emphasis is given to the modeling of nonlinearities and to the fact that the process dynamics can affect the performance of the monitoring scheme. First, the nature of the process data under study is described in Section 9.2. The convenience of using either multivariate models or control charts with memory to capture the actual time-varying process dynamics for fault detection enhancement is discussed in Section 9.3. A comparative study between Shewhart and EWMA control charts in terms of the capabilities for the fast detection of abnormalities in the online nutrient sensors measurements is also carried out. In addition, the design of a soft-sensor using missing data imputation techniques to replace the erroneous measurements is presented. Finally, some conclusions will be provided in Section 9.4.

9.2 Nature of data in continuous phosphorus removal processes

In-control data have been simulated, assuming a typical daily variation along four weeks. Furthermore, several faults in the probes installed in the aerobic reactor were simulated, obtaining a data set for each one of the faults. For further details about the process, see Chapter 2.

In Figure 9.1, a representation of the current process behavior is depicted by plotting the measurements belonging to the NO_3

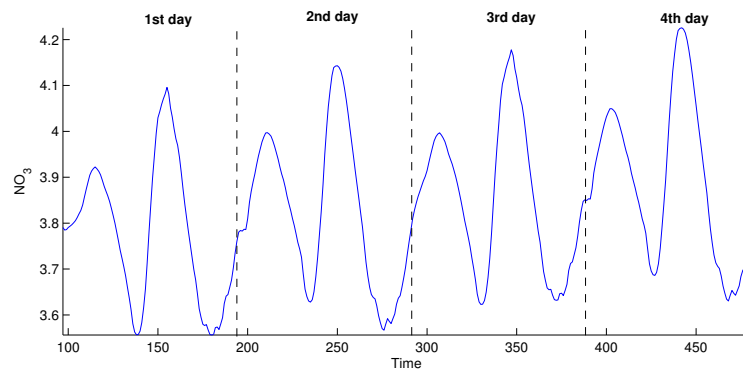


Figure 9.1. Measurements of the NO₃ variable of the simulated continuous biological phosphorus removal process along the four first days. Measurements belonging to one day are limited by dashed black lines.

variable. A daily pattern inherent to the process can be observed. This depends on many factors including seasonal, the population size of the area, the amount of industries closed to the waste water treatment plant, among others. For instance, the majority of industries have nearly a constant wastewater flow rate during the production time, however, this flow rate may drastically change when a batch is finished and a cleanup of the reactors is achieved. Hence, industrial wastewater flow rate varies in function of the time of the day, such as work and lunch time, producing peaks in the wastewater production. On the other hand, the human activity and habits have an important influence on the municipal wastewater flow rate, producing a daily pattern with two peaks located twice per day: in early hours of the morning and the evening, when the human usage of water increases drastically.

In MSPC of continuous processes, data are usually arranged in a two-way array (time \times variables). Nevertheless, given the daily pattern shown in Figure 9.1 that represents the simulated WWTP data, this process may be modeled as a batch process by arranging the data in a three-way array (days \times variables \times time). Care should be taken in the type of bilinear modeling chosen since the use of an inappropriate modeling structure may cause negative consequences in the performance of the monitoring scheme. It may occur that the fault detection is not

accurate over time or even unusual behaviors are not detected. Hence, it is worth emphasizing the importance of modeling the process dynamic without paying attention to the type of process (continuous or batch) but to the process features.

9.3 Batch modeling for continuous processes

In order to show the consequences of modeling a process without capturing its dynamic, the current process was modeled by following two different approaches: arranging the data in a two-way array \mathbf{X}^c ($NK \times J$) (equivalent to variable-wise unfolding of the batch arrangement) and in a three-way matrix \mathbf{X}^b ($N \times J \times K$), K being the sampling time points at which J process variables are measured in each of the N days. Later on, such data matrix was arranged in a two-way array \mathbf{X}^b ($N \times JK$) by batch-wise unfolding the three-way \mathbf{X}^b . Note that the days were considered as batches. Recall that in batch-wise unfolding the relationships among all the process variables at different sampling time points are modeled, whereas in variable-wise unfolding only the instantaneous relationships among the variables are taken into account.

Data should be preprocessed prior to modeling with PCA. The matrix \mathbf{X}^c ($NK \times J$) was mean-centered and scaled to unit variance. Hence, the main non-linear behavior of the process is not removed. In the case of the batch-wise unfolded two-way array \mathbf{X}^b ($N \times JK$), the average trajectory of each process variable is removed. Consequently, PCA is focused on the variation of variables around the average trajectory. Additionally, the data are slab-scaled by scaling all the process variables at all times to unit variance in order to give them the same weight in the PCA model.

The preprocessed two-way arrays, $\tilde{\mathbf{X}}^c$ and $\tilde{\mathbf{X}}^b$, representing the process variability from different points of view, were modeled by PCA, yielding the CM-AD (continuous mode -all days) and BM-AD models (batch mode -all days), respectively. Five PCs were selected in the CM-AD model based on cross-validation results, explaining 95.4% of total variation (R^2) with a goodness of prediction (Q^2) equal to 74.8%. Two PCs were extracted in the BM-AD model by cross-validation, which captured around the 82.6% of the process variability (R^2) with a goodness of prediction (Q^2) equal to 65.8%. Note that the 11 multicollinear

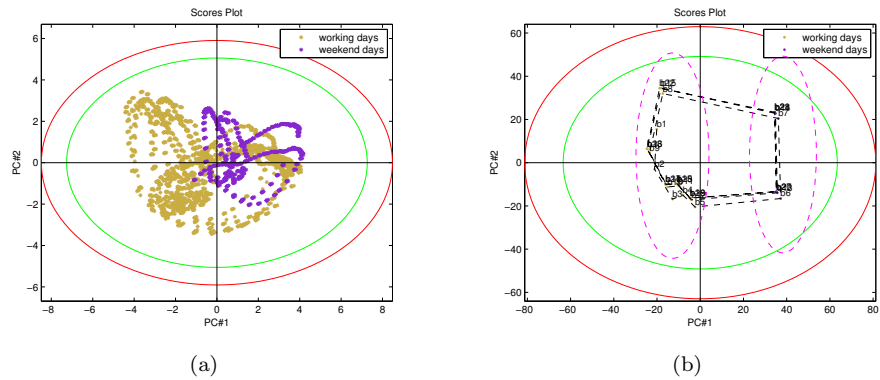


Figure 9.2. t_1/t_2 score plot with the 95% and 99% confidence limits (green and red ellipsoids, respectively) for the CM-AD model (a) and the BM-AD model (b). None of the samples or batches exhibits unusual behavior.

process variables were replaced with five and two latent variables (linear combination of the original variables) for the models CM-AD and BM-AD, respectively.

Figure 9.2 shows the scatter score plots for the first two latent variables with the 95% and 99% confidence ellipsoids for the CM-AD model (Figure 9.2(a)) and the BM-AD model (Figure 9.2(b)). By examining the behaviour of the process data in the latent space defined by the principal components, abnormalities or different operating conditions can be detected. Each star point of the scatter plots represents a measurement registered on the j -th variable at the k -th sampling time point for the CM-AD model (see Figure 9.2(a)) or in a particular day for the BM-AD model (see Figure 9.2(b)). If some samples or batches are consistent with the remaining, then clustering is expected to appear in the score plots. In case outliers exist, it is expected that points lie outside of the in-control region limited by the ellipsoids. As can be seen in Figure 9.2(a) and Figure 9.2(b), samples and batches, respectively, are found within the in-control region limited by the 95% and 99% confidence limits (green and red ellipsoids, respectively). Hence, no outlier or abnormal process behavior is suspected by looking at the scores. Additionally, the squared prediction error was also calculated for each one of the models to detect abnormalities related to breakage of the correlation structure. The results yielded all

Table 9.1. Statistical features of each one of the fitted models: continuous (CM) and batch (BM) models for all days (AD), working days (WO), and weekend days (WE). The selection of the number of principal components was based on the goodness of prediction (Q^2) and the variability explained by the eigenvalues.

Model	A	R^2	Q^2
CM-AD	5	95.4%	74.8%
CM-WO	5	96.1%	76.1%
CM-WE	5	96.7%	77.0%
BM-AD	2	82.6%	65.8%
BM-WO	3	95.7%	60.9%
BM-WE	1	88.7%	32.3%

samples were statistically under control (plots not shown).

By examining the score plot of the CM-AD model (see Figure 9.2(a)) in depth, it is worth noting that most of the measurements belonging to weekend days lie on the right side of the first latent variable (positive values on t_1 score), except for some few samples. Those with a negative score are the most likely due to non-linearities still present in data that produce inaccuracies to describe the actual process variation. These non-linearities are observed by the elliptic trajectory scores depict on the scatter plots. This suggests that different models for working and weekend days may be built in order to capture better the process variability, yielding potentially better performance of the monitoring system to detect faults. Considering the scatter plot for the first two principal components of the BM-AD model (see Figure 9.2(b)), a clear clustering of the batches defining the days (dashed ellipsoids) is identified. As can be observed, the weekend days (purple points) have positive values in the first score t_1 (right side of the scatter plot) whereas the working days (yellow points) have negative values in the aforementioned score (left side of the scatter score plot). This fact confirms that process data may be modeled by two PCA models to better explain the variability associated to the two different operating conditions.

The elliptic trajectory that the score values draw on the scatter plots is a consequence of the non-linearities still present in data.

At this point, raw process data were split into two different

groups: working (WO) and weekend days (WE), and arranged in two-way arrays \mathbf{X}_{wo}^c ($N_{wo}K \times J$) and \mathbf{X}_{we}^c ($N_{we}K \times J$), respectively, and in three-way arrays $\underline{\mathbf{X}}_{wo}^b$ ($N_{wo} \times J \times K$) and $\underline{\mathbf{X}}_{we}^b$ ($N_{we} \times J \times K$), respectively. Note that K are the different sampling time points where the measurements of the J variables were recorded in each one of working (N_{wo} batches) and weekend (N_{we} batches) days. Later on, the three-way arrays, $\underline{\mathbf{X}}_{wo}^b$ and $\underline{\mathbf{X}}_{we}^b$ were batch-wise unfolded, leading to the two-way arrays $\mathbf{X}_{wo}^{b_{we}}$ (working days \times variables at different sampling time points) and $\mathbf{X}_{we}^{b_{we}}$ (weekend days \times variables at different time points), respectively. Once the grand mean was extracted from the two-way arrays \mathbf{X}_{wo}^c and \mathbf{X}_{we}^c and each of its columns were scaled to unit variance, a PCA model was fitted for each one, yielding the CM-WO and CM-WE models, respectively. Two additional PCA models (BM-WO and BM-WE) were fitted on the two-way arrays $\mathbf{X}_{wo}^{b_{we}}$ and $\mathbf{X}_{we}^{b_{we}}$ after trajectory centering and scaling to unit variance (slab-scaling). The characteristics of the different models fitted are summarized in Table 9.1. No outlier was detected in model validation.

Once data were modeled by the different approaches and possible outliers were detected and isolated, a monitoring system was built by designing two multivariate Shewhart control charts based on Hotelling T^2 and SPE statistics. Their control limits (thresholds) were estimated from NOC process data and later readjusted using cross-validation techniques for an ISL. In this procedure, some of the samples¹ are arranged in a test data set while the remaining ones are used as a training data set to build the PCA models. The number of samples containing the test data set depends on the approach chosen to model the process data. For each one of these models, a monitoring system is developed by constructing the aforementioned multivariate control charts and subsequently calculating their control limits. Normally, such control limits are estimated by imposing a significance level equal to 1% and 5%. Later on, the test data set is projected onto the low dimensional space of the PCA model, yielding the SPE and Hotelling T^2 statistics. This procedure is repeated until all samples of the original data set have been in the test set once and only once. Once the statistics belonging

¹In this work, days treated as batches of the two-way batch-wise unfolded arrays and the measurements belonging a process day were selected as samples to build both the test as the training data set.

all the samples are obtained, the percentage of faults in the NOC data or the OTI risk is assessed². In order to obtain an accurate and proper monitoring system, such percentage should be close to the expected percentage of false alarms, i.e. NOC data detected as abnormal. If the OTI estimated is greater or lesser than ISL value, such control limits are raised or lowered. The complete iterative procedure is repeated until the OTI and ISL values are similar.

After readjusting the control limits of the control charts, a set of simulated faults were projected onto each one of the PCA models. SPE control charts resulting from these projections by using the PCA models BM-AD, CM-AD, BM-WO, BM-WE, CM-WO and CM-WE are displayed in Figures 9.3 and 9.4.

When a drift in the ammonium and nitrate probes are involved, the Shewhart SPE control chart from the model CM-AD approximately takes four days to clearly signal the faults (see Figure 9.3(b)). However, when the process is modeled as a batch process and data are batch-wise unfolded, the Shewhart SPE control chart takes two and three days to detect the slow drifts in nitrate and ammonium probes, respectively (see Figure 9.3(a)). A slight improvement in the drift detection is observed when two models are used to capture the process variability associated to the working and weekend days (see Figures 9.4(a) and 9.4(b)). The Shewhart SPE control charts for the CM-WO and CM-WE models detect the fault four days after the drift was originated, like the Shewhart SPE control chart from the model CM-AD. The Shewhart SPE control charts of BM-WO and BM-WE models only take around one day and a half to detect the out-of-control signal (half day and one day earlier for the nitrate and the ammonium probes, respectively, with regard to the model BM-AD). According to the previous results, a better detection of the slow drift faults was reached by using a batch-wise approach in comparison to the approaches equivalent to variable-wise unfolding. In order to improve the interpretation of the results and to accurately evaluate the four approaches in terms of capabilities to detect abnormalities, the percentage of abnormal samples detected as NOC data (so-called OTII) were assessed³ for each one of the approaches under study

²The OTI values are estimated by following $OTI = 100 \cdot \frac{nf}{N_{NOC} \cdot K} \%$, where nf is the number of faults detected, N_{NOC} is the number batches and K the number of sampling time points

³The OTII values are calculated by following $OTII = 100 \cdot \frac{nnf}{N_{ab} \cdot k} \%$, where nnf is the

9. MODELING PROCESS DYNAMICS THROUGH MULTIVARIATE MODELS AND CONTROL CHARTS.

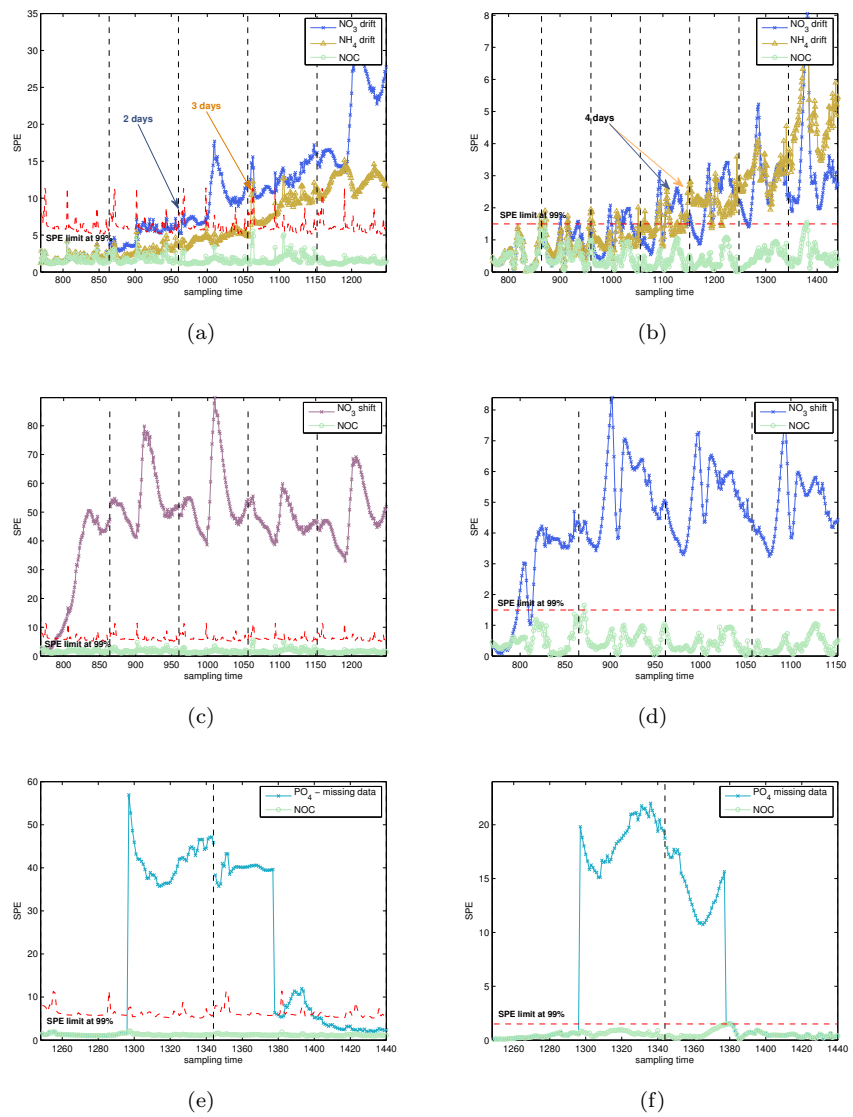


Figure 9.3. Shewhart SPE control chart showing different faults: drifts in the nitrate and ammonium probes (a and b), shift in the nitrate probe (c and d), and lack of measurements in the phosphorous probe (e and f) in the model BM-AD (a,c and e) and in the model CM-AD (b, d and f), respectively.

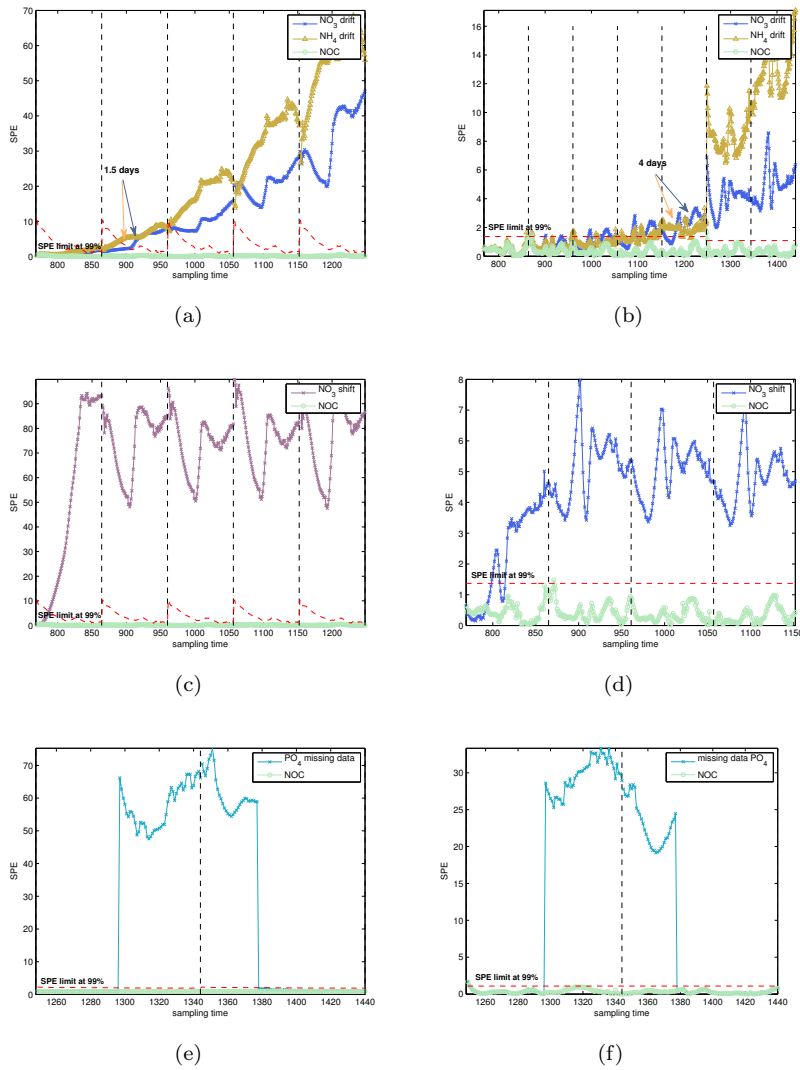


Figure 9.4. Shewhart SPE control chart showing different faults: drifts in the nitrate and ammonium probes (a and b), shift in the nitrate probe (c and d) and lack of measurements in the phosphorous probe (e and f) in the models BM-WO and BM-WE (a,c and e) and in the models CM-WO and CM-WE (b, d and f), respectively. Note that both models take into account the variability of the working and weekend days separately.

(see Table 9.2). Note that the closer to 0 is the OTII value for a certain approach, the better performance of the monitoring system in terms of fault detection will show. By looking at the OTII values obtained for the slow drift in the nitrate and ammonium probes by using the Shewhart SPE control chart (see Table 9.2), it can be concluded that the best approaches to detect this type of fault are the batch-wise models, providing better results the approach in which the different operational conditions throughout the week is modeled (BM-WO and BM-WE with an OTII value equal to 6.1% and 4.9% for the slow drift in the nitrate and ammonium probes, respectively).

Differences found in the capability for slow drift detection are caused by the preprocessing and modeling approach used. Firstly, the subtraction of the mean trajectory permits the main non-linear behavior to be removed from data. Thus, the cyclical pattern found in the data over all days is removed. In contrast, autoscaling allows the reduction of the auto-correlation on the multivariate statistics. Secondly, the approaches equivalent to variable-wise only take into account the instantaneous relationships whereas the batch-wise approaches exploit the process dynamics, considering all the past information till the current sampling time point. Consequently, when slight deviations are present in a batch, such abnormalities can be better detected. Hence, better results on fault detection are expected by using latent models that incorporate the process dynamics, such as the models BM-AD, BM-WO and BM-WE.

Regarding the shift in the nitrate probe, all approaches were capable of detecting this type of fault. In the cases of batch-wise approaches, both the model BM-AD, and the BM-WO and BM-WE model (see Figures 9.3(c) and 9.4(c)), accurately detected the shift fault without any appreciable differences in time detection. This fact can be also seen in all the OTII values of the aforementioned models for the Shewhart control chart (see Table 9.2), since such values are equal to 1 and 0.9% for the BM-AD model, and the BM-WO and BM-WE models, respectively. This means that the monitoring systems detect the fault in time with a very low non-detected faulty batch rate. Regarding the approaches equivalent to variable-wise, no notable difference were found. This can be confirmed by looking at the corresponding OTII values in Table 9.2. For the models CM-AD, and CM-WO and CM-WE, similar values of OTII were reached (1.6% and 1.8%, respectively).

Table 9.2. Comparison of the six different monitoring approaches in terms of the accuracy of fault detection by using the OTII values. In the case of the EWMA control chart, the OTII was only estimated for drift faults since such control chart does not provide any improvement in the detection of other type of faults.

Control chart	Model	Drift - NO_3	Drift - NH_4	Shift - NO_3	PO_4 fault
		OTII	OTII	OTII	OTII
Shewhart	CM-AD	15.5%	16.4%	1.6%	0%
	CM-WO & CM-WE	15.5%	16.6%	1.8%	0%
	BM-AD	8.0 %	14.4%	1.0%	0%
	BM-WO & BM-WE	6.1%	4.9%	0.9%	0%
EWMA	BM-AD	4.7%	4.7%	-	-
	BM-WO & BM-WE	3.8%	3.1%	-	-

When a technical fault occurs in the phosphorous probe, values equal to 0 are recorded by the sensing system. In this type of faults, the four approaches correctly detected the out-of-control signal just in time (see Figures 9.3(e) and 9.3(f), and Figures 9.4(e) and 9.4(f)). In addition, all the monitoring systems were capable of detecting all the faulty samples as abnormal, yielding OTII values equal to 0% (see Table 9.2). Nevertheless, there is a fact that is worth being commented. When the batch-wise approach is selected to monitor the faults, the monitoring system may tend to detect the faults longer than they really took (see Figure 9.3(e)). This fact is due to the auto-correlation inherent in the monitoring statistics caused by the batch-wise modeling. Hence, the Shewhart SPE control chart based on batch-wise approaches will respond slowly (to a greater or lesser extent) to changes in the process.

It was checked that drift faults are better detected by those models that take into account the relationships among the process variables at all sampling time points rather than those only capturing the instantaneous relationships. However, the improvement gained in the capability of detecting drift is not substantial due to the monitoring charts take excessive time to signal the abnormality. In order to improve the fast detection of slow drifts in the batch-wise approaches, an EWMA control chart is proposed to be used on the SPE statistic. The EWMA control charts with smoothing factor $\lambda \in [0.05, 0.20]$ are very effective in detecting small drifts in process parameters since the past information about the process is taken into account when a new sample is available. The smaller the lambda parameter, the faster detection of slow drifts. Consequently, the

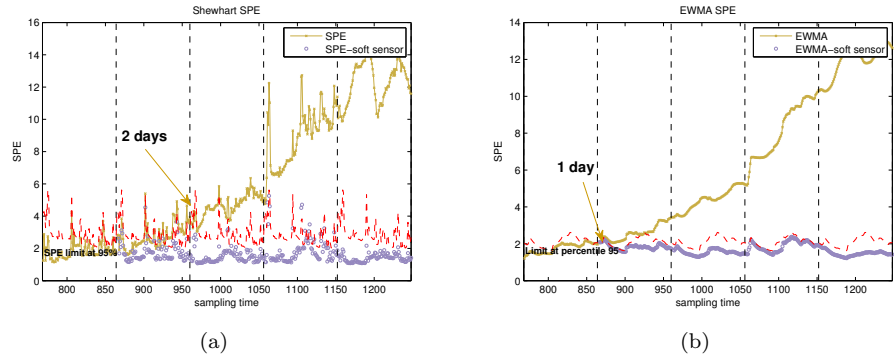


Figure 9.5. Shewhart SPE (a) and EWMA SPE (b) control charts under a simulated slow drift in the ammonium probe by using the BM-AD latent model. Dark yellow points represent the SPE statistic at each k sampling time point whereas the purple circles the SPE statistic obtained from the estimated values of the erroneous measurements and the ones belonging the remaining process variables.

smoothing factor was set to 0.07 in this case study, which was obtained after cross-validation. The EWMA control limits were established at 95 percentile of the SPE_{EWMA} values estimated at all K sampling time points in all N NOC days as follows:

$$SPE_{EWMA_k} = \lambda SPE_k + (1 - \lambda) SPE_{EWMA, k-1}.$$

Faulty batches with a slow drift in the ammonium probe were selected to evaluate the performance of the EWMA versus Shewhart SPE control chart in accordance with their fault detection capabilities using the batch-wise approaches (the best approaches in this case study). The diagnosis capability of the responsible variable/s through the use of contribution plots was also studied. The monitoring results based on the models BM-AD, and BM-WO and BM-WE are shown in Figure 9.5 and 9.6, respectively.

Once again, the approaches, in which the variability associated with the working and weekend days was independently modeled (see Figure 9.6), provided better results in terms of accuracy in fault detection than approaches that modeled such variability through an unique model (see Figure 9.5). By using the model BM-AD with an EWMA control chart needed half the time to detect the drift (see Figure 9.5(b)) in comparison to the Shewhart control chart (see Figure 9.5(a)). In the case of the models BM-

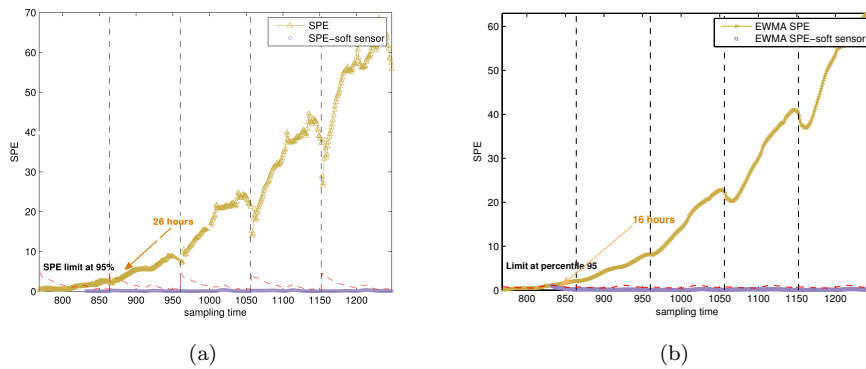


Figure 9.6. Shewhart SPE (a) and EWMA SPE (b) control charts showing a simulated slow drift in the ammonium probe by using the BM-WO and BM-WE latent models. Dark yellow points represent the SPE statistic at each k sampling time point, whereas the purple circles represent the SPE statistic obtained from the estimated values of the erroneous measurements and the values belonging to the remaining process variables. Note that these charts represent the monitoring system built from the separate modeling of working and weekend days.

WO and BM-WE, both the EWMA and the Shewhart SPE control charts (see Figure 9.6(b)) were capable of detecting the fault after 26 hours and 16 hours the drift was originated, respectively. Nonetheless, the Shewhart SPE control chart, after detecting the drift fault, did not detect the fault for 6 hours approximately. Hence, in this case the use of EWMA SPE control chart is preferred to avoid detecting faulty samples as NOC.

Following the MSPC ideas, when the EWMA SPE control chart detected an out-of-control signal, a contribution plot was built to identify the assignable cause of it. Figure 9.7(a) shows the contribution plot for the first abnormality detected by the monitoring system built from the model BM-AD (see Figure 9.5(b)). The interpretation of such plot is that both the first, fourth, sixth and ninth process variable (influent flow rate, pH measured in the anaerobic reactor, ammonium measured in aerobic reactor and pH measured in the return sludge, respectively) are the responsible variables of the abnormality detected. Hence, a wrong diagnosis of the responsible variables was obtained due to the same reason why the Shewhart SPE control chart was not able to detect the fault at this sampling time point.

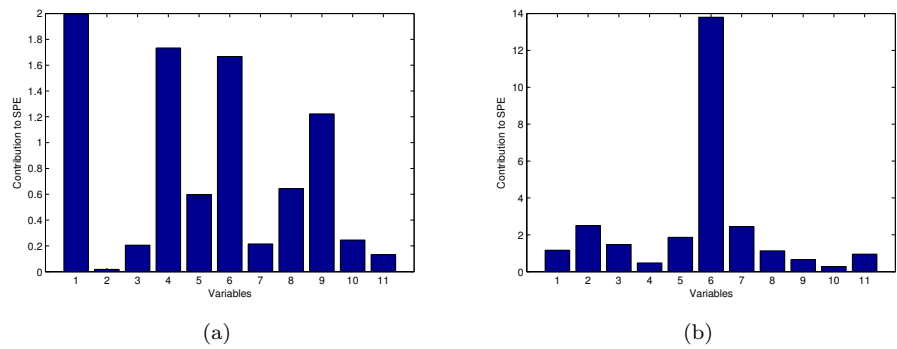


Figure 9.7. Classical contribution plot (a) and EWMA contribution plot (b) for the out-of-control signal detected by the EWMA control chart at the first fault detection sampling time point.

Since the batch-wise approach considers the batch as a whole, slow drifts throughout the complete batch are cumulated in the statistics over time, for this reason, the Shewhart SPE control chart takes longer to signal the abnormality. Consequently, the contribution of the variables to the SPE statistic need time to correctly identify the root causes of unusual process behaviors.

In order to solve this problem, an exponentially weighted moving average of the contributions $c_{j,k}^{SPE}$ to the SPE statistic for each variable j at each sampling time point k is proposed to improve the diagnosis accuracy. Based on the EWMA idea, this procedure allows to take into account the contributions belonging to previous sampling time points, downweighing importance over time. The EWMA-based contribution plot is estimated as $c_{j,k}^{SPE} = \lambda c_{j,k}^{SPE} + (1 - \lambda)c_{j,k-1}^{SPE}$, where $\lambda \in (0, 1)$.

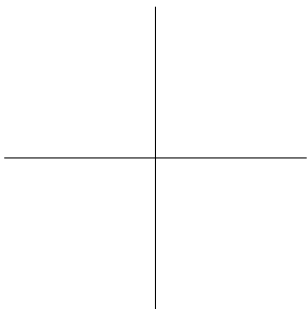
In Figure 9.7(b), the EWMA-based contribution plot of the first out-of-control signal detected by the EWMA SPE control chart is shown. As can be observed, the contribution plot clearly identifies the ammonium variable as the unique responsible variable of the drift in the ammonium probe, which is the ammonium variable. Such tool can be very useful in processes where the high auto-correlation of data makes the MSPC techniques difficult to be successfully applied.

Once the monitoring system diagnoses the root causes of the abnormalities, as the simulated faults based on the malfunctioning of some probes, it is desirable to replace these erroneous

measurements. It would allow the monitoring system to keep running while the faulty sensor is being repaired. For this purpose, a TSR-based soft sensor [169] was developed to replace the erroneous measurements of the responsible variables of the abnormal situation with missing data imputation. Later on, the SPE statistic was calculated and plotted both on the Shewhart and EWMA SPE control charts (depicted as purple circles in Figures 9.5 and 9.6), showing that the process can be considered operating under control, except for the drift in the probe.

9.4 Conclusions

This study has been a preliminary research work to understand how the process dynamics should be modeled for WWTP continuous processes, either through multivariate models or control charts. The unfolding direction and preprocessing techniques have a notable influence in online fault detection. Care should be taken in the type of bilinear modeling chosen since the use of an inappropriate modeling structure of a process may cause negative consequences in the performance of the monitoring scheme. Modeling the WWTP by taking into account the dynamics yielded better results in slow drift fault detection, isolation and diagnosis than the model that only incorporates the variances and instantaneous cross-covariances of the variables. Modeling WWTPs (continuous process) with cyclical patterns as a batch process and unfolding batch-wise provides: (1) the main non-linear behavior is removed from data, and (2) auto-covariances and lagged cross-covariances are captured. In this chapter, an EWMA approach for control charts and contribution plots was also introduced into the monitoring scheme, yielding a clear improvement in slow drift detection and fault diagnosis. Also, a soft-sensor was developed to replace the erroneous measurements allowing the monitoring system to be operative while the involved sensors are being repaired.



MVBatch Toolbox for bilinear batch process modeling

Part of the content of this chapter has been included in the following publications:

- [6] J.M. González-Martínez, J. Camacho, and A. Ferrer. MV-
Batch Toolbox: a MATLAB graphical interface for bilinear
batch process modeling. In elaboration.
- [12] J. Camacho, J.M. González-Martínez and A. Ferrer. Chap-
ter 3: Batch Process Data. Batch Processes: Monitoring
and Process Understanding, *Wiley-VCH Verlag GmbH*, pub-
lication due in 2016.
- [13] J. Camacho, J.M. González-Martínez and A. Ferrer. Chap-
ter 4: Bilinear Modeling of Batch Process Data. Batch
Processes: Monitoring and Process Understanding, *Wiley-
VCH Verlag GmbH*, publication due in 2016.

10.1 Introduction

In the design of monitoring schemes, two phases are involved [47]: model building (exploratory data analysis and post-batch process monitoring) and model exploitation (real-time process monitoring). In the former, understanding the nature of the effects of varying initial conditions and process operating trajectories on the performance of the batches, and on the final product quality is pursued [62]. Thereafter, the understanding gained and the statistical models are used to isolate and diagnose past poor operating conditions and to set up statistical process control schemes for monitoring purpose in the second phase. 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 three-way array to one or several two-way arrays for the subsequent iv) bilinear batch modeling (see Figure 10.1). These steps are iteratively repeated provided that outliers are detected and isolated from the calibration data set.

A novel graphical user-friendly interface for process understanding, troubleshooting and monitoring has been developed as a freely available Matlab toolbox. The main contribution of this software package is the integration of the recent developments in Batch Multivariate Statistical Process Control including the methods developed in this thesis that overcome problems such as the different sampling policy among variables and batches, complex asynchronisms, and time-varying correlation structures. Apart from providing the algorithms in Matlab format for its manipulation in the light of future research by the scientific community, an interface that guides users to handle batch data through the main modeling steps (data alignment, preprocessing, transformation of the three-way array into a two-way array, and calibration) is provided. In addition, a simulator of the fermentation process of the *Saccharomyces cerevisiae* cultivation is available to generate realistic batch data under normal and abnormal operating conditions.

This chapter is structured as follows. In Section 10.2, the modeling cycle of batch processes, which is the main layout of the software package, is briefly discussed. A short discussion on the methods implemented in the MVBatch Toolbox for data alignment and modeling is presented, stressing the difference with other commercial software packages available in the market. Sec-

tion 10.3 presents the software specifications and requirements. The data set used in the software demonstration is explained in Section 10.4. Afterward, in Section 10.5, the graphical user-friendly interface MVBatch Toolbox is presented and its use is described using simulated data. Finally, some conclusions are drawn in Section 10.6.

10.2 Modeling cycle of batch processes

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

10.2.1 Data alignment

The most used commercial software by the chemometrics community for bilinear modeling of batch processes are SIMCA Release 13.0.3 -Umetrics software- [65] and ProMV Batch Edition Release 13.02 -ProSensus software [68]. The former uses TLEC family to make all batches equal in length, but recently it has been proved that this strategy is not appropriate in cases of multiple asynchronisms [5]. The latter, apart from offering synchronization algorithms based on TLEC also includes ordinary SCT-based methods, which are ineffective in the presence of complex asynchronisms. To overcome these problems, the Multisynchro algorithm that successfully tackles these challenging scenarios of asynchronisms is implemented (see Figure 10.1(c.1)). Other approaches available are the Indicator Variable (IV) [70] -TLEC-based method in the domain of the variables- (see Figure 10.1(c.2)), the different versions of the Dynamic Time Warping (DTW) for batch synchronization [76, 160] (see Figure 10.1(c.3)) and the Relaxed Greedy Time Warping algorithm (RGTW) [1] (see Figure 10.1(c.4)).

10.2.2 Data modeling

Once batch data are aligned, the calibration of the model is carried out. Prior to the fit of a multivariate model, batch data need to be preprocessed. The preprocessing strategies available

in the software are: 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 preprocessing approach may be different [39]. Since PCA is a bilinear tool, the aligned and preprocessed three-way array needs to be conveniently rearranged in a number of two-way arrays to apply PCA. As explained in Chapter 1, there are at least three methods to transform the data structure, which are implemented in the MVBatch Toolbox. First, unfolding the three-way matrix of data in a single two-way matrix: variable-wise [255] -method implemented in SIMCA Release 13.0.3 [65], batch-dynamic [255], and batch-wise [70] -method implemented in ProMV Batch Edition Release 13.02 [68]- (see Figures 10.1(d.1), (d.2) and (d.3), respectively). Second, using an adaptive approach where current and past information are combined, e.g. by using hierarchical models [103]. Finally, fitting K PCA local models [91], each one modeling exclusively the information corresponding to a sampling time point. Additionally, both the unfolding and splitting in K models can be combined in approaches such as the evolving modeling [91, 256] or the moving window approach [92] (see Figure 10.1(d.4)). Within this category, we find the multi-stage approach, which is based on the calibration of independent models for different stages of a batch process [96]. These models use a specific and fixed modeling structure, no matter the dynamic nature of the process [87]. It is not surprising that many studies in the literature arrive to 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 [90]. Hence, experiments on different processes may lead to very diverse conclusions. Supporting this idea, recent investigations performed [3, 90] (see Chapter 7 for a rigorous discussion) 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, the MVBatch Toolbox implements the Multi-Phase Framework (MPF) [94], 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 (according to a specific definition

of the optimization function) as depicted in Figure 10.1(d.5). For the estimation of missing trajectories and missing values, the TSR method is implemented because of its outperformance in comparison to other methods [136]. The feature that makes the MVBatch Toolbox different from commercial software packages for bilinear modeling is its flexibility to fit PCA models in the whole spectrum of possibilities: from variable-wise to batch-wise passing through intermediate (batch dynamics) and specific (multi-phase) models.

10.2.3 Design of the monitoring scheme

From the PCA model(s), two Shewhart monitoring charts can be developed: the D-statistic or Hotelling T^2 chart, and the SPE chart [47, 117]. The MVBatch Toolbox estimates their control limits (thresholds) from NOC process data and later adjusts them using cross-validation techniques for a given imposed significance level (ISL) [90] (see Figure 10.1(e.1)). Additionally, an unsupervised control chart based on the warping profiles from NOC batches (NOC-WICC) [2] is designed as a complementary tool to the aforementioned charts for post-batch and real-time batch process monitoring (see Chapter 5 for further details on this control chart).

Note that the statistics for the new batch on the charts are computed online, so that the statistics for a specific sampling time point are available right after the measurements have been collected. If the batch remains under NOC -that is, the statistics remain below the control limits except for punctual cases- we assume that the quality will be within the specifications, and therefore the batch processing will continue. If otherwise consecutive points are beyond the control limits, it means that the batch is behaving in an abnormal way and that the quality may be seriously affected (see Figure 10.1(e.2)). In these situations, the fault has to be diagnosed (see Figure 10.1(e.3)). Overall and instantaneous contribution plots to the D-statistic and SPE [82] are available in the MVBatch Toolbox for fault diagnosis. If the batch can be operationally corrected by taking a control action, then this should be carried out. Otherwise, the batch may be discarded, saving time and resources.

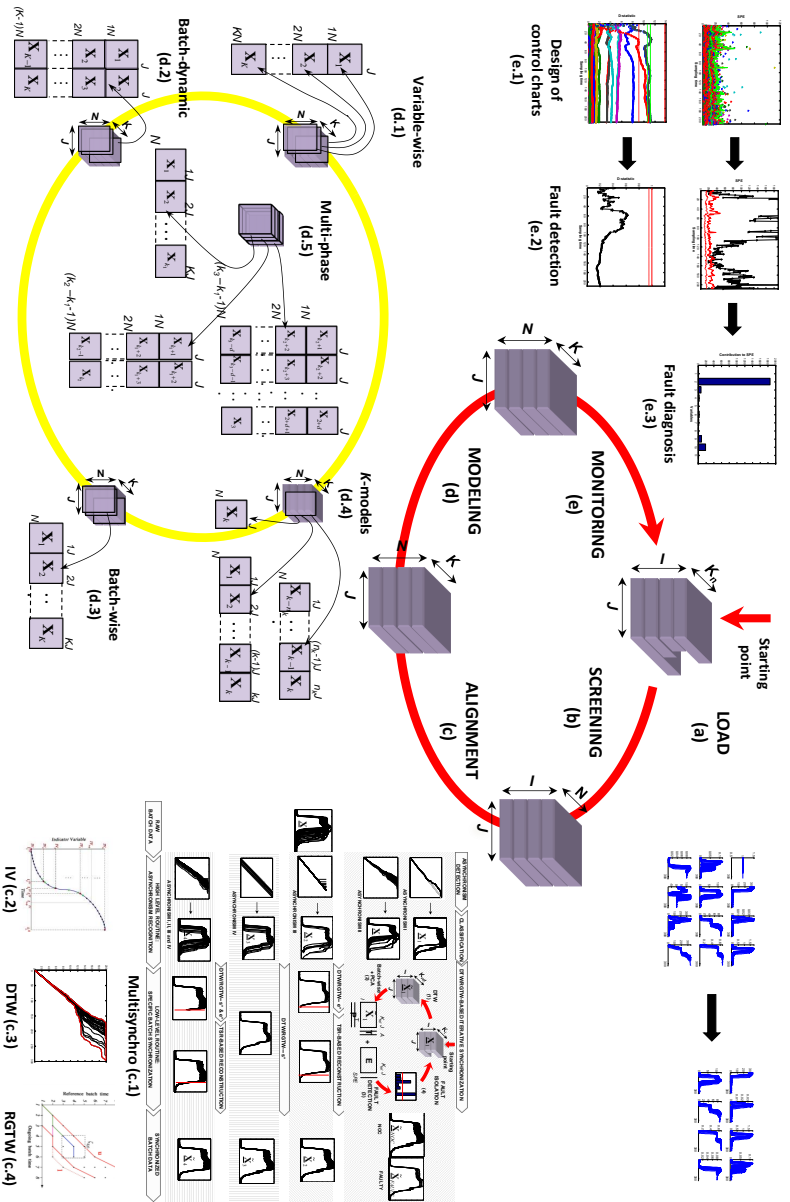


Figure 10.1. Modeling scheme in BMSPC systems based on PCA and modified version of the MVBatch Toolbox. First, the three-way array containing the raw batch trajectories is loaded (a). Once variable and batch screening is performed via data visualization (c), batch data are equalized (if needed) and synchronized (c) prior to bilinear modeling (d). Finally, a monitoring system is designed for fault detection and diagnosis (d).

10.3 Software specification and requirements

The MVBatch Toolbox is a free software based on Matlab release 2013a (The MathWorks, Inc.). The toolbox includes 37 self-contained Matlab p-files (pre-parsed Matlab m-file) (including functions with the algorithms for data equalization and synchronization, preprocessing, modeling and monitoring, and scripts for the implementation of the logic of the user interface), 6 main user interfaces (one for the modeling cycle that integrates all functionalities of the toolbox, and five for each of the main modeling steps, i.e. data visualization and screening, equalization and synchronization, modeling, fault detection and diagnosis) and one batch process data simulator of the fermentation process of the *Saccharomyces cerevisiae* cultivation.

There are two ways to work with the MVBatch Toolbox: using the graphical user interface (GUI) (starting users) and using the commands (expert users). The GUI is self-explanatory and follows the modeling cycle of the batch process explained in [5]. Information, warning and error messages are incorporated into the tools to straightforwardly guide its use. In order to launch the user interface, either the user types 'batchTools' in the Matlab command line or clicks on the batchTools user-friendly interface icon available in the folder of the toolbox, which should be declared in the Matlab path. Note that the latter is a compiled version of the Matlab user interface. For expert users who want to have more interaction between the tool and other software packages, it is recommendable to start up the tool using Matlab. The toolbox is compatible with Matlab release versions R2012 and R2013, and with Microsoft Windows operating systems (Windows Vista/7) and Macintosh OS X (from Leopard to Mavericks) with no requirements of any other third party's utilities beyond the standard Matlab installation with the Statistics toolbox. Both the MVBatch Toolbox and the data set used in this tutorial for illustration are available at the Multivariate Statistical Engineering Research Group web page <http://mseg.web.upv.es/software>.

10.4 Data set

The batch data used for this software demonstration is formed by a combination of the data sets used in Chapter 8, which contains NOC and faulty batches with five different types of

asynchronisms. In particular, 45 NOC batches (the first 25 batches with different duration produced by natural variability with key process events not overlapping at the same sampling time point across batches -case #1 asynchronism-, the next 5 batches with equal duration but different process pace -case #2 asynchronism, the next 5 batches with incomplete trajectories but with most of the key process events overlapping over time -case #3 asynchronism, the next 5 batches with a shift at the start of the batch but with the same process pace -case #4 asynchronism, and the last 5 ones are incomplete batches with different process pace -case #5 asynchronism), and 3 different faulty batches with each of the five aforementioned cases of asynchronism. In total, the three-way array for calibration is composed of 45 NOC batches and the test set of 15 faulty batches.

10.5 Description of the MVBatch Toolbox

The main interface of the software package comprises a top row with the pull-down menus 'File' and 'About' and four main modules, which are titled 'Screening', 'Alignment', 'Modeling', and 'Monitoring'. Users can save the screening of the variables and batches, alignment, and modeling already performed by clicking on the option 'save' of the menu 'File'. Existent analysis can also be loaded through the option 'Open' of the same menu. To initiate the MVBatch Toolbox, a data set must be loaded through the module 'Load'. This software package reads batch data contained in a Matlab structure that must be named *calibration*. This structure contains three fields: 'batch_data', 'batch_names' and 'var_names' that are cell arrays storing the batch data, the identifiers of the batches and the names of the process variables for each of the units, respectively. The field 'batch_data' has as many cells as there are batches (N batches). For each batch, a data structure named *data* is assigned, which contains one matrix per sampling policy (M different sampling rates). In each of these matrices, the sampling time point, the stage identifier as well as the measurements of the process variables registered at a specific sampling frequency are stored. To access the batch data of the n -th batch via the command line of Matlab, users must type 'calibration.batch_data{i}.data{m}', where m is the m -th sampling policy.' Note that this complex way of arranging the batch data is required to take into con-

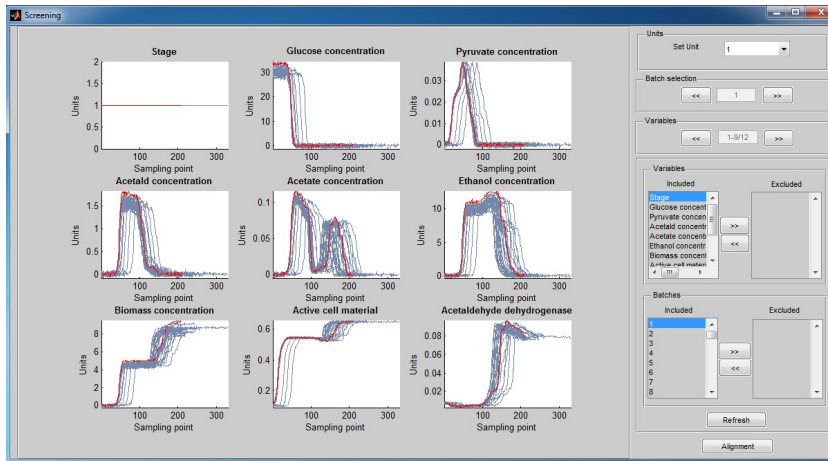


Figure 10.2. Interface for screening batches and variables for subsequent alignment and modeling.

sideration the cases where the process variables are sampled differently batch to batch. Furthermore, all the data sets subject to being analyzed by the MVBatch Toolbox must have the aforementioned structure.

The modules implemented in this software are described in the following sections using the simulated data as an example. To load the data set, users must click on the module 'Load' of the main window (see Figure 10.1(a)) and select the corresponding file. Once the system checks that the data structure complies with the aforementioned requirements, the module 'Screening' (see Figure 10.1(b)) will be enabled.

10.5.1 Screening

In the module 'Screening' users can firstly visualize the raw process variables trajectories for all the batches (see Figure 10.2). The MVBatch Toolbox plots 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, users can highlight those batches of interest by moving forward or backward the batch index in the panel 'Batch selection'. If there are more variables than subplots depicted, the panel 'Variables' is enabled to give users the option of visualizing those not represented. Based on a prior knowledge or process understand-

ing reached through previous analysis, users may be interested in including and/or excluding process variables and batches for subsequent steps in the modeling cycle. For this purpose, users can select them in the variables and batches panels. In the example, the last variable of the data set contains the actual warping introduced in the original data by the simulator to produce the asynchronisms. As it is not of interest for the modeling, that variable is removed from the data set. To update the graphs of the batch trajectories, users must press the button 'Refresh'. Once batch data have been screened, users can move on to the alignment of the batches by pressing the button 'Alignment'.

10.5.2 Alignment

In this interface, equalization and synchronization of batch trajectories are carried out interacting with the 2D and 3D arrangement panels, respectively (see Figure 10.3). The aim of the 2D panel is to equalize the variables to a single sampling rate or common sampling period using interpolation. The interval period can be selected in the popmenu 'interval' where four different choices are available: greatest common divisor, shortest sampling interval, longest sampling interval and least common multiplier. The sampling interval will be a multiple of the unit specified in the edit box 'units'. The interpolation method (nearest neighbor, linear, spline, cubic, and V5cubic) to be used can be selected in the popmenu 'interpolation'¹. Once all the required parameters have been set, the button 'Equalize' can be clicked on to perform the equalization. When this procedure has been carried out, the 3D arrangement is automatically enabled for synchronization.

The panel '3D arrangement' provides users with a set of tools to synchronize the batch trajectories, ensuring not only that the resulting batches have the same 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) [82]), and SCT-based methods (Dynamic Time Warping

¹If there are missing values in data -specified in Matlab as "NaN" (Not-a-Number)-linear interpolation is suggested to be employed. However, if there is suspicion that the correlation structure was destroyed, the values should be imputed by exploiting the correlation structure of the within-batch information available with TSR. In addition, when there are missing values, Matlab will provide warning messages informing of such problem.

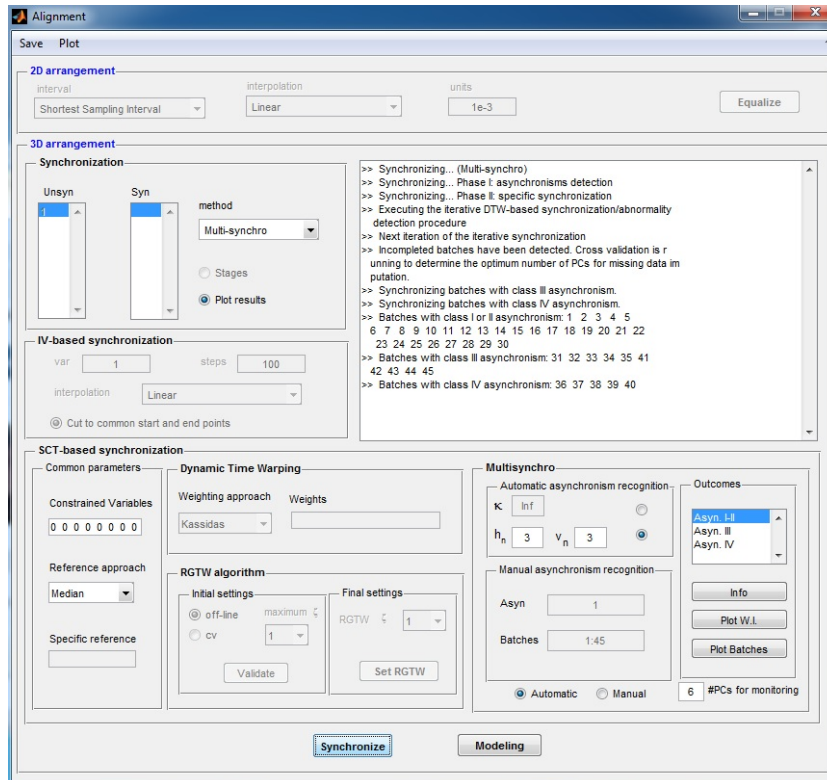


Figure 10.3. Interface for data equalization and alignment.

(DTW) following Kassidas et al.'s [76] and Ramaker et al.'s [160] approaches, the Relaxed Greedy Time Warping (RGTW) [1] and Multisynchro [4]).

For the application of IV, it is required to have a process variable that is a good indicator of the evolution of the batch. This variable must be strictly monotonic and smooth, with the same start and end point in all batches. To proceed with this type of synchronization, first, users must select this method in the popmenu 'method' located in the synchronization panel. Also, the indicator variable and the number of sampling time points must be indicated in the edit boxes 'var' and 'steps', respectively, as well as the type of interpolation in the popmenu 'interpolation' (see IV-based synchronization panel at the center-left side in Figure 10.3). In case that the selected variable does not have

the same start and end point in all batches, the algorithm can discard preceding the initial common point or succeeding 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 [5].

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 10.3). First, a batch of the calibration data set needs to be selected to synchronize the remaining batches. If prior knowledge on the process is available, users can select a suitable reference batch for synchronization. If not, the algorithm can select that batch whose length is the closest to the median or the average duration of the collected batches. For this purpose, the popmenu 'reference' provides the previous options: select, median and average. To select a certain batch, the edit box 'reference' is enabled to type the number of the batch. Second, the constraints of the variables for synchronization is needed. Typically, there are process variables containing too little or no features for synchronization (flat profiles), excessive amount of missing data or noise that might negatively affect the quality of synchronization. In this situation, it is recommendable to constrain these variables in the algorithm by typing 1's logical values (0's logical values stand for non-constrained variables) in the edit box 'constraints'. The next parameters to initialize depends on the SCT method selected in the popmenu 'method'. For DTW, the weights to give more importance to certain process variables in the synchronization are required (see DTW panel at the bottom-center side in Figure 10.3). Users can calculate these weights in order to give more importance to those variables that are more consistent batch-to-batch (Kassidas et al.'s approach [76]), those containing more warping information (Ramaker et al.'s approach [160]), or those that satisfy both features (Geometric average of Kassidas et al. and Ramaker et al.'s weights). Another option is to indicate explicitly the weights in the case of prior knowledge of the process. All these options can be selected in the popmenu 'weights'. If the option 'Select' from the latter menu is chosen, the edit box 'weights' will be enabled

in order to type 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.

The MVBatch Toolbox gives the possibility to synchronize batch data in such a way that it can be used in the design of a monitoring scheme for real-time applications. For this purpose, the optimization procedure based on the RGTW algorithm proposed in [1] needs to be executed. When the synchronization based on DTW has finished, the system will show a message requesting the user to confirm the purpose. Later, the parameters of the RGTW-based synchronization will be enabled (see RGTW algorithm panel at the bottom-center side in Figure 10.3). These parameters are: the width of the sliding warping window ζ and the bands. On the one hand, users can set the bands using the warping information obtained from the offline DTW synchronization for a specific window width ζ . For this option, the radiobutton 'offline' must be marked. Furthermore, the window width can be manually set by selecting the value in the popmenu 'RGTW ζ '. On the other hand, the aforementioned parameters can be optimized by running the cross-validation procedure proposed in [1]. To run this procedure, users must activate the radiobutton 'cross-validation', select the maximum window width to be validated from the popmenu 'maximum ζ ', and finally, press the button 'Validate'. Once the execution has been finished, an Analysis of Variance (ANOVA) is performed on the Fisher Z-transformed² correlation coefficients for the different window widths ζ to determine whether there are statistically significant differences among them, provided that the number of window widths under study are greater than two. This software package subsequently displays figures containing a standard one-way ANOVA table, a box-and-whisker plot and LSD intervals for the Fisher Z-transformed correlation coefficients for each of the specified window widths. Based on these results, users can choose the 'optimal' window width from the popmenu 'RGTW ζ '. As a final step, the button 'Set RGTW' must be pressed to confirm the parameters settings.

As an example of synchronization based on the previous SCT-based methods for the subsequent design of a real-time monitoring system, let us synchronize the calibration data set. At first instance, the DTW algorithm with weights calculated as the

²The Fisher Z transformation is usually performed to approximate the distribution of the Pearson's correlation coefficients to a normal distribution.

geometric average of the Kassidas et al. and Ramaker et al.'s weights is selected (option 'Geometric' in the popmenu 'Weighting approach') with the rest of parameters set to the default values. Once the synchronization has finished, the MVBatch Toolbox will ask the user through a dialogue window whether a second synchronization based on the RGTW algorithm is needed for subsequent steps. After pressing the button 'Yes', the parameters of the RGTW algorithm are enabled. To proceed with the study of the RGTW parameters, we activate the radiobutton 'cv', select a maximum width of the window ζ equal to 4 units from the popmenu 'maximum ζ ' (this cross-validation procedure is time consuming, therefore, no window widths larger than 6 units is recommended) and press the button 'Validate'. In order to determine whether one of the selected window widths implies a statistical significant improvement in terms of synchronization quality without considerably delaying the monitoring of future samples (the larger the window width, the longer a new measurement vector will be available for monitoring at the start of the batch, and the longer the detection of faults), the LSD plots are calculated (see Figure 10.4). As can be appreciated from this figure, the window with width equal to 3 units is preferred, therefore, this value is selected in the popmenu 'RGTW ζ ' and set by pressing the button 'Set RGTW'. The resulting synchronized batch trajectories of this synchronization are plotted in Figure 10.5. Most trajectories are well synchronized except for some batches where flat profiles are added. This is mainly caused by the presence of asynchronisms that neither the DTW algorithm in its different versions nor the RGTW algorithm are capable of tackling.

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 by selecting the option 'Multisynchro' from the popmenu 'method' located in the synchronization panel. Automatically, the software package will enable the Multisynchro panel (see panel at the bottom-right side in Figure 10.3). This synchronization algorithm is composed of a high-level and low-level routine. The high-level routine is aimed at recognizing the different types of asynchronous trajectories for the subsequent

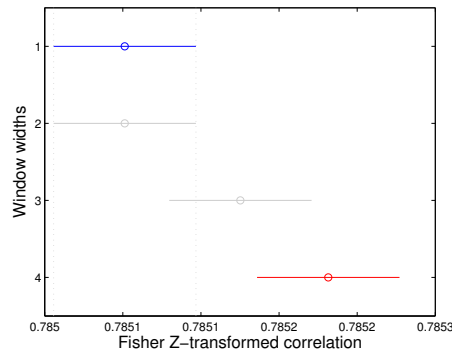


Figure 10.4. LSD intervals for the Fisher Z-transformed correlation coefficients for four window widths.

batch classification as function of the nature of asynchronism. The low-level routine is in charge of synchronizing the variable trajectories of each one of the batches with a specific procedure based on the type of asynchronism. To initiate the algorithm, there are two ways: the automatic (see panel for automatic asynchronism recognition at the bottom-right side in Figure

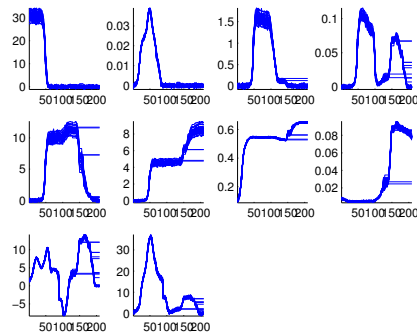


Figure 10.5. Resulting variable trajectories after synchronizing the trajectories using the RGTW algorithm with a window width equal to 3 units. The variables shown in order from top to bottom, and from left to right are: concentrations (glucose, pyruvate, acetaldehyde, acetate, ethanol, and biomass), active cell material, acetaldehyde dehydrogenase, specific oxygen uptake rate, and specific carbon dioxide evolution rate.

10.3) and manual (see panel for manual asynchronism recognition at the bottom-right side in Figure 10.3) asynchronism detection. In the first option, users can decide the thresholds for the number of consecutive horizontal and vertical transitions (see parameters h_n and v_n in the automatic asynchronism recognition panel, respectively) in the warping information from which the algorithm determines what type of asynchronisms batches have. By default, these parameters are set to 3 units each since they are sensible thresholds to distinguish among asynchronisms. In case that users prefer that the algorithm assesses the thresholds based on the distribution formed by the number of horizontal and vertical transitions in the warping information at the start and end of each batch, the parameter κ must be set. This value must be ranged in the interval $]0, 1[$. In the second option, if users know the types of asynchronisms either by expert knowledge or by the outcomes of preliminary synchronizations, the manual version can be used by activating the radiobutton 'Manual' located at the bottom side of the Multisynchro panel in Figure 10.3. In this option, the types of asynchronisms [1, 4] must be specified in the edit box 'Asyn': 1) batches with different or equal duration and different process pace (class I and II asynchronism); 2) batches with different duration due to incompleteness of some batches and key process events overlapping (class III asynchronism); 3) batches with different duration due to delay in the start but batch trajectories showing the same evolution pace after (class IV asynchronism); and 4) the combination of the two last types of asynchronisms. In addition, the set of batches containing different asynchronisms needs to be indicated as well. It is done in a vectorial manner following the Matlab syntax by introducing the batch indices in the edit box 'Batches' located at the bottom side of the Multisynchro panel in Figure 10.3. Another parameter required is the number of principal components (PCs) to extract with the aim of detecting abnormal batches of the calibration data set. Internally, the algorithm iteratively synchronizes trajectories provided that there are abnormal batches still in the data set. For more details on the Multisynchro algorithm, users are referred to the original research work [4].

Due to the fact that the experimental data under study contain different types of asynchronisms, let us synchronize these batches with Multisynchro by selecting the corresponding option in the popmenu 'Method' of the panel 'Synchronization' 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 depicted at the top-right side of Figure 10.3. In case that users want to proceed with a manual synchronization based on the outcomes of the Multisynchro synchronization shown in the console, users should type in the edit box 'Asyn' "1, 2, 3", "1:30, [31:35,41:45], 36:40" in the edit box 'Batches' and 6 in the edit box for the #PCs for monitoring. As a result of the synchronization, the types of asynchronisms are also indicated in the listbox of the outcomes panel located at the top-left side of the Multisynchro panel in Figure 10.3. To visualize either the information of the synchronization performed or the warping profiles of the variable trajectories or the synchronized variable trajectories of the batches classified into one of the types of asynchronism, users must first select the type of asynchronism and then click on the buttons 'Info', 'Plot W.I.' or 'Plot Batches', respectively. For the example data set, the warping information for each class of asynchronism is shown in Figure 10.6. As can be seen, there exist batches with different or similar pace but with no overlap of the key process events (smooth warping profiles over time depicted in red color in Figure 10.6(a)), incomplete batches (see red warping profiles with horizontal transitions in Figure 10.6(b)), and batches shifted at early stage of the process (see red warping profiles with vertical transitions in the first sampling time points in 10.6(c)). In Figure 10.7, the synchronized trajectories by Multisynchro are shown. Comparing these trajectories with those obtained from the synchronization based on RGTW (see Figure 10.5), there is a notable improvement in the quality of the synchronization. Not only misalignment disappears in the former, but also the artificial variability of the variables is reduced and the overlap of the features notably enhanced.

As a result of the synchronization performed by using any of the methods available, the warping profiles (except for IV-based synchronization) and the synchronized trajectories of each process variable are depicted. Afterward, the system enables the button 'Modeling' located at the bottom side of the interface to proceed with the modeling of the synchronized data. Note that at any moment users can re-synchronize the raw batch data, but synchronizations previously performed will be pruned out.

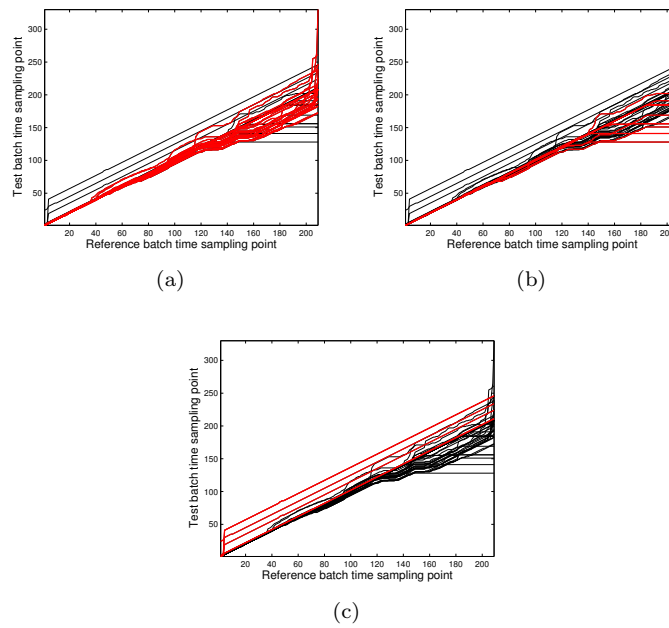


Figure 10.6. Warping information derived from the Multisynchro-based synchronization for the three types of asynchronism found in the data set: (a) class I and II asynchronism, (b) class III asynchronism and (c) class IV asynchronism.

10.5.3 Modeling

Once batch data have been synchronized, the module named 'Modeling' of the main interface is activated. When users click on this option, the Modeling interface pops up (see Figure 10.8). The interface comprises a top row with 'File', 'Preprocessing', 'Cross-validation' pull-down menus. In the first option, users can save, load and print results of the multivariate analysis. In the second option, the preprocessing methods described in Section 10.2.2 are listed for selection. Finally, the third option provides users with different cross-validation methods: row-wise, sample-wise (by default), iterative sample, and cross-corrected sample-wise k-fold [257]. If users are not knowledgeable about these methods, it is recommended to use the default method.

The panel named 'Covariance maps' (see panel located at the top side in Figure 10.8) is devoted to depict the covariance maps of the process under study. Covariance maps are useful tools to

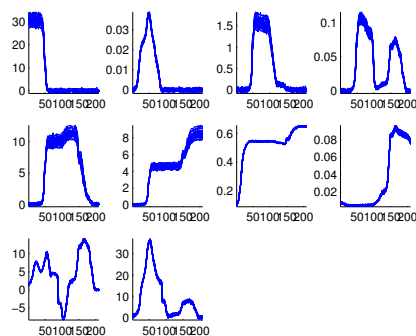


Figure 10.7. Resulting variable trajectories after synchronizing the trajectories using the Multisynchro algorithm. The variables shown in order from top to bottom, and from left to right are: concentrations (glucose, pyruvate, acetaldehyde, acetate, ethanol, and biomass), active cell material, acetaldehyde dehydrogenase, specific oxygen uptake rate, and specific carbon dioxide evolution rate.

reveal the time-varying process dynamics. Prior to modeling, three types of variance-covariance maps are available: total, dynamic partial and instantaneous-dynamic partial. The former calculates the variance-covariance matrix of the synchronized, preprocessed and batch-wise unfolded two-way array that contains the batch trajectories. This matrix gives practitioners a picture of the dynamic relationships in the batch data, not only the instantaneous variance and cross-covariances of the variables at every sampling time point but also the auto-covariances and lagged cross-covariances. Revealing the time-varying dynamics helps practitioners to better understand the relationships among process variables over time. The dynamic variance-covariance matrix is the result of the computation of the partial covariance in time for all possible LMVs. It is useful to observe dynamic relationships without taking into account the instantaneous relationships among process variables. This information should be used when the objective of the modeling is to predict the current value of a variable from past measurements of the process (e.g. to perform one-step ahead predictions). Finally, the instantaneous-dynamic partial map contains the partial covariance taking the instantaneous relationships of the process variables into account. The resulting map is useful when the goal is to obtain parsimonious models for process monitoring that

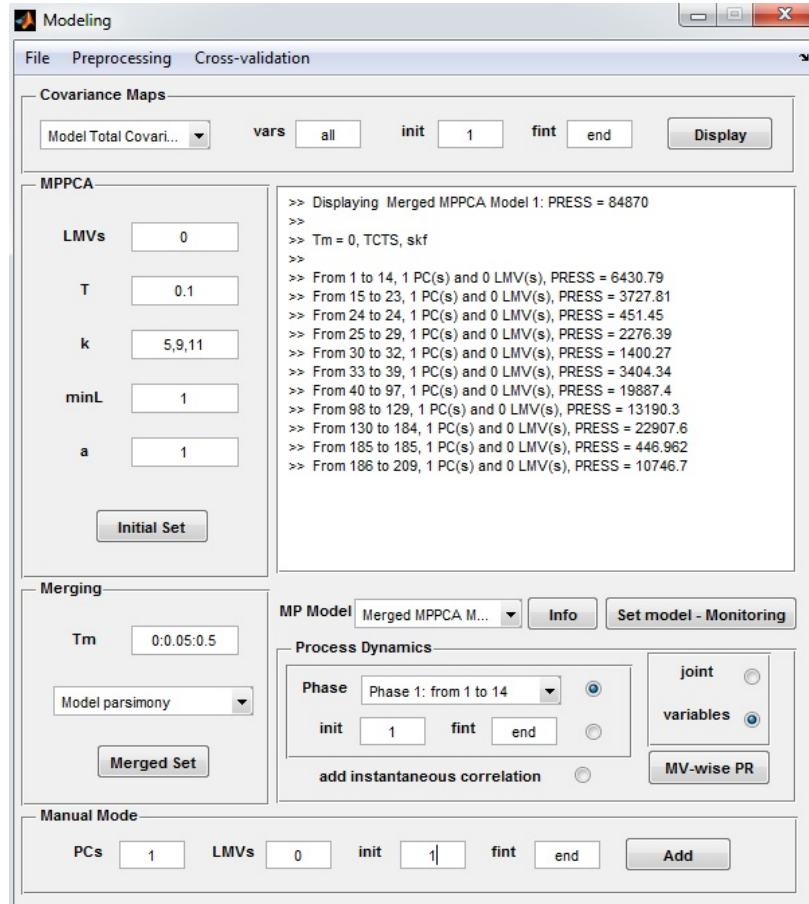


Figure 10.8. Interface for data modeling using the multi-phase framework.

properly capture the complex dynamics by adding the optimum number of lagged measurements vectors. For further details on the covariance maps, readers are referred to Appendix B. To select one of these covariance maps, users must select the corresponding option pulling down the popmenu. In case that users require the calculation of the variance-covariance matrix for a set of variables, these must be indicated in the edit box 'vars'. If the map is desired to be calculated in a time interval, the start and end point must be provided in the edit boxes 'init' and 'fint', respectively. Note that when the number of sampling

time points or process variables is very large, covariance maps are very memory demanding. If Matlab is getting out of memory, users should try to reduce the number of sampling time points in the map by reducing the sampling frequency in the command line of Matlab. For an optimum reduction of number of samples, the procedure explained in [258] is recommended.

Taking back the example of the process fermentation, the first step is to visualize the total covariance map to get insight into the correlation among process variables. Figure 10.9 represents the map of covariances among all the process variables at all sampling time points. As can be seen, the measurements of the process variables collected in the time intervals [45,85], [95,120], [125,175], among others, are highly correlated (high positive covariances depicted in red color). The appearance of rectangles of different sizes in the covariance map is a clear evidence that phases are present in batch data. In this example, there are several phases by the nature of the process -there may be more phases but they are visually indistinguishable - which are strongly correlated to each other due to the high covariance observed (area outside the main diagonal in Figure 10.9). In this context, the decision on whether the process must be divided into phases or not depends on if the correlation among phases is indirect or direct. An indirect correlation between process variables at two different sampling time points k and $k + n$ indicates that these process variables are sequentially correlated. This means that the process variables at sampling time point k are correlated at sampling time point $k + 1$, and those at sampling time point $k + 1$ are correlated at sampling time point $k + 2$, and so forth. In contrast, a direct correlation denotes only a correlation between process variables at sampling time points k and $k + n$, without intermediate correlations. When indirect correlation among phases is observed on batch data, the process should be divided into phases. Otherwise, batch data should not be split up. In order to confirm the existence of direct or indirect correlations, the dynamic partial covariance coefficients must be computed.

Figure 10.10 shows the dynamic partial covariance map calculated for the example of the fermentation process. The points with important information (highlighted with dark colors) are arranged close to the main diagonal, which have a high correlation coefficient. Outside the main diagonal, low correlation coefficients are shown, which indicates that the relationship

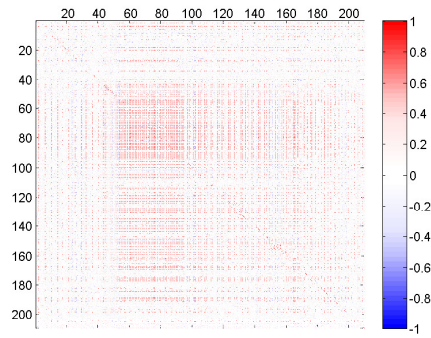


Figure 10.9. Total covariance map showing the covariances among process variables at all sampling time points.

between phases is indirect. The fact that the correlation coefficients are located along the main diagonal also denotes that the process can be properly captured with a parsimonious (low order) PCA model. Let us zoom in the dynamic partial covariance map shown in Figure 10.10(a), from the 30th to 60th sampling time point (see Figure 10.10(b)). This plot shows that there is a sampling time point interval where the relationships among variables look quite different to the other sampling time points. This phenomenon can be appreciated around the 15th sampling time point (actual 45th sampling time point). This time period belongs to process phase at which the glucose is consumed by the yeast as a energy source for the fermentation.

At this point of the modeling, the high total covariance found among phases might suggest that the separate modeling of these phases is not appropriate. Nonetheless, the dynamic partial covariance map points out that the relationship between phases is indirect and then a more parsimonious PCA model (without adding so many lagged measurements vectors as in batch-wise unfolding) should be fitted since the dynamics of the process are of low order. Also, a change of dynamics observed in the main diagonal of the dynamic partial covariance matrix suggests splitting up the process into phases. Hence, a multi-phase modeling is advised in this case.

In the panel 'MPPCA', the parameters required to automatically recognize the phases present in the process using the multi-phase framework proposed in [94] are listed (see panel MPPCA located

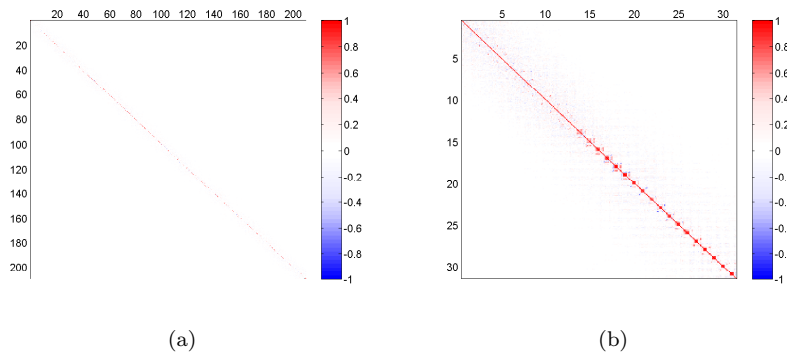


Figure 10.10. Dynamic partial covariance maps for the whole process (a) and for the time interval [30,60] (b).

at the left side in Figure 10.8): the unfolding approach (LMVs), the improvement threshold (T), the gain parameters that control the trend of the algorithm to divide data into more or less phases (k), the minimum length of the phases (minL) and the initial number of principal components (a). The number of Lagged Measurement Vectors (edit box 'LMVs') should be set according to the order of the dynamics. With 0 LMVs a static model -variable-wise- is computed, with 1 LMV a dynamic model of order 1 is computed and so forth. The MPPCA can handle several LMVs at the same time by typing several values for example '0, 1, 2' or '0:2' in the edit box 'LMV' to compute models from order 0 to 2. The number of LMVs recommended is from 0 to the maximum order with important information observed in the partial covariance map. Note that the features of a model depend very much on the number of LMVs [87]. The rest of parameters may set to the following values: 'T' to either 0.1 or 0.05, 'k' values greater than 1, 'minL' to 1, and 'a' to different values greater than 0. Once all the parameters have accordingly been set, the automatic detection can be run by clicking on the 'Initial Set' button. Users should note that the detection of the phases is a time-consuming task, increasing the computation time required with the number of parameters to explore. During this execution, messages about the tasks being performed by the algorithm (adding phases or principal components) are displayed in the console located at the top-right side of the interface (see Figure 10.8).

Let us execute the multi-phase algorithm with the following settings: LMVs = 0, T = 0.1, k = 5,9,11, minL = 1, and a=1. At the end of the execution, three different multi-phase models are listed in the listbox called 'MP Model'. The complete information of each of the models can be displayed in the console by selecting the model in the popmenu 'MP model' and pressing the button 'Info'. For instance, if users select the second multi-phase model named 'MPPCA Model 2', they will check that a total of 11 different phases with 1 PC each were found in the following intervals: [1,14], [15,23], [24,24], [25,29], [30,32], [33,39], [40,97], [98,129], [130,184], [185,185] and [186,209]. The console also displays the PRESS and the parameters used in the calibration, including the preprocessing and cross-validation method. For each phase, apart from the interval corresponding to the phase, the number of PCs, the number of LMVs and the PRESS are provided.

Once the algorithm has generated the multi-phases model, the section named 'Merging' (see bottom-left side in Figure 10.8) is enabled. This section is aimed at post-processing and merging the information obtained in the previous step in an optimum manner [87]. For a reasonable search of the optimum multi-phase model, the parameter 'Tm' should be ranged between 0 and 0.5. For instance, users can specify this range in intervals of 0.05 units by typing '0:0.05:0.5' in the edit box named 'Tm'. A different merging criterion can be also selected in the popmenu located in the same panel: model parsimony, covariance matrix parsimony, minimize LMVs, minimize phases and minimize PCs. To proceed with the merging procedure, users must press the button named 'Merged Set', ending up this task with n multi-phase merged models listed in the popmenu 'MP Model'. In the case of the fermentation process, the values of the parameter 'Tm' are set to the default values, yielding two merged multi-phase models. The software package additionally performs an ANOVA and computes the LSD plot to help users to select the best model that does not produce a significant loss of prediction power (see Figure 10.11). As can be appreciated in Figure 10.11, the second merged multi-phase model is more parsimonious than the first model. However, this simplification does imply a statistically significant loss of prediction capability since the intervals for the two models do not overlap. According to this analysis, the best multi-phase model in terms of parsimony is the first merged model (see features of the model in the console

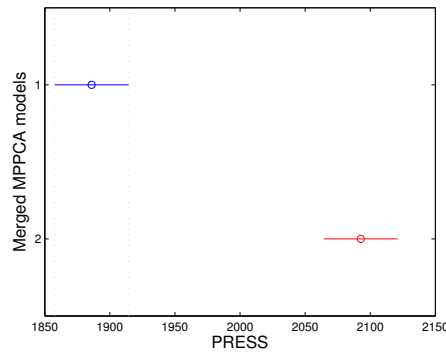


Figure 10.11. LSD intervals estimated for the merging procedure applied to the models found by the multi-phase algorithm.

displayed in Figure 10.8). Special caution should be taken in the interpretation of these results since ideally it should be computed with an independent test set and not with the calibration data set.

At this point, it is strongly suggested comparing the features of the multi-phase models generated with the visual information in the covariance maps and thereafter drawing conclusions by combining the different sources of information. In addition, it is worth going over the covariance maps of the part of the data modeled and of the residuals. For this purpose, users should select the optimum multi-phase model from the list 'MP Model' and the new options added in the popmenu of the covariance maps: 'Model Total Covariance' and 'Residuals Total Covariance'.

From the model and residuals total covariance maps shown in Figure 10.12, we can state that there is still some structure to model (observed with some correlation remaining in the residuals in Figure 10.12(b)). Users can investigate further the dynamics of each of the phases or a specific interval with the tools offered in the panel named 'Process Dynamics' (see panel located at the bottom-right side in Figure 10.8). Through partial regression on the batch data, users can get insight into the actual process dynamics of the model. To run this procedure, users must select a phase from the popmenu 'Phase' or a time interval by indicating the start and end point in the edit boxes 'init' and 'fint' in the panel 'Process Dynamics', respectively, as well as the

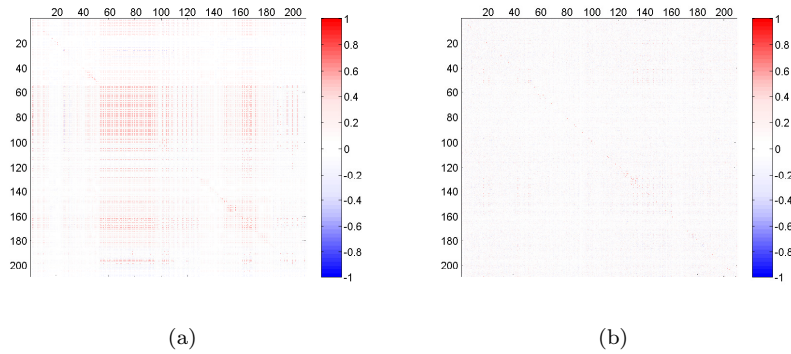


Figure 10.12. Covariance maps of the modeled (a) and residual (b) part.

mode of the regression, either accumulated ('joint' radiobutton) or separated variables ('variables' radiobutton). To take into account the instantaneous correlation, the radiobutton named 'add instantaneous correlation' should be marked. Once the parameters are selected, the regressions are executed by pressing the button named 'MV-wise PR'.

As most of the remaining structure in the example corresponds to the last stage of the process, users should start looking at this phase (Phase #9). The outcomes of the analysis are bar plots showing the normalized mean sum of squares as a function of the LMVs for the accumulate option (see Figure 10.13(a)), and the sum of squares as a function of the LMVs and process variables for the separated variables (see Figure 10.13(b)). From these plots, users can appreciate how important the dynamics of order 1 are in the time interval [130,184]. Hence, the addition of a LMVs would be highly recommended to further optimize the multi-phase model. After looking through all the phases, a new proposal of multi-phase model comes up. To manually fit the new multi-phase model, users should type the aforementioned parameters in the panel 'Manual Mode' (see panel 'Manual Model' located at the bottom side of the interface shown in Figure 10.8). The number of PCs for each phase must be indicated in the edit box 'PCs' (1, 1, 1, 1, 1, 1, 1, 1, 1), the number of LMVs of each phase in the edit box 'LMVs' (e.g. 0, 0, 0, 0, 0, 0, 1, 0), and the start and end points of each phase in the edit boxes 'init' and 'fint' (1, 15, 25, 30, 33, 40, 98, 130, and 186 in 'init' and 14, 24, 29, 32, 39, 97, 129, 185 and 209 in

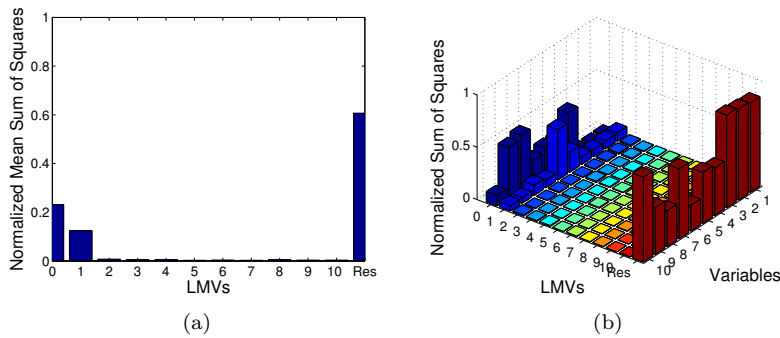


Figure 10.13. Partial regression function for accumulated (a) and separated variables (b) for Phase #9.

'fint').

In case that it is the first iteration of the modeling, it is recommended to also fit a batch-wise model for preliminary exploratory analysis. This type of models is used to first understand the correlation among variables, detect the most severe faulty batches and diagnose their main causes. Again, a manual batch-wise PCA model must be set in panel 'Manual Mode' with the following parameters: PCs = 3, LMVs = 208 ($K - 1$ sampling time points), init = 1, fint = end).

Once all the multi-phase models have been fitted, users can design the monitoring scheme by using the calibration data set and a selected model fitted in the next step of the modeling cycle by pressing the button 'Set model - Monitoring'.

10.5.4 Monitoring

The interface for the design of a monitoring scheme is composed of a menu bar containing the submenu 'File' where different options to save the results and import data are available, four panels devoted to the design of the monitoring schemes and the monitoring of batches named 'Model', 'Monitoring scheme', 'Post-batch offline process monitoring' and 'Post-batch online process monitoring', and one panel aimed at the selection of the statistics and batches for fault diagnosis (see Figure 10.14).

The first step in the design of the monitoring scheme is to re-adjust the theoretical control limits using a leave-one-out cross-

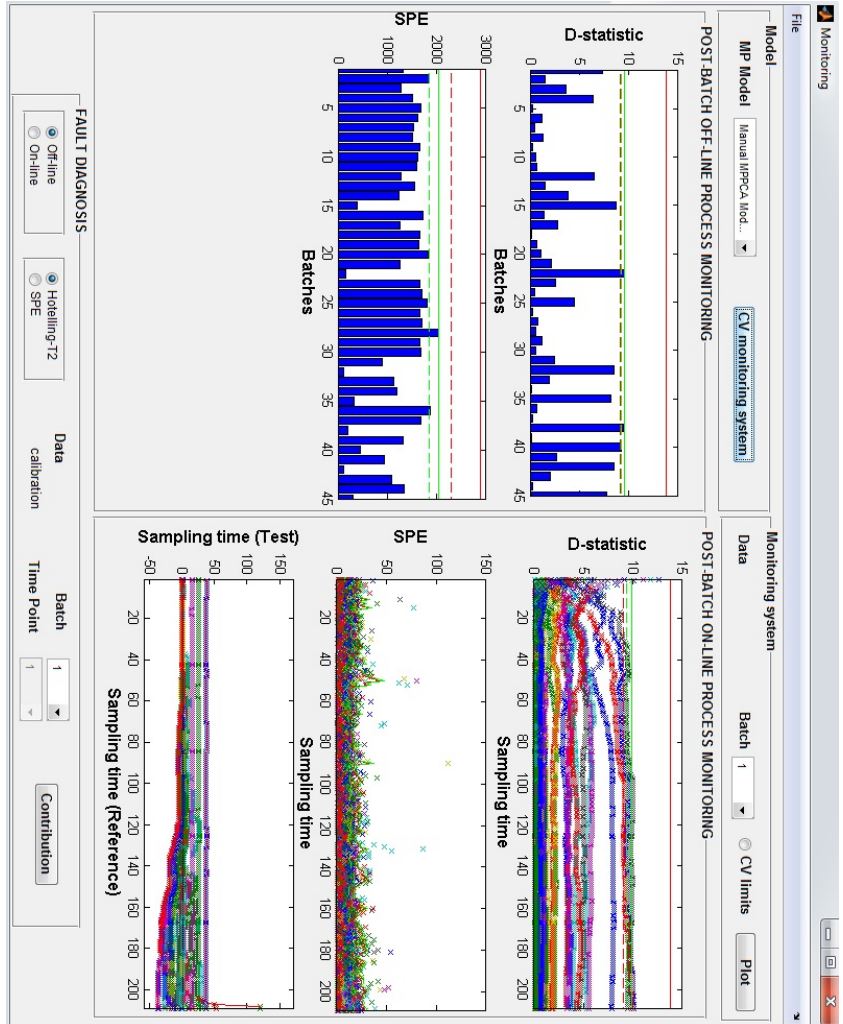


Figure 10.14. Interface for the design of a monitoring scheme using the calibration data set.

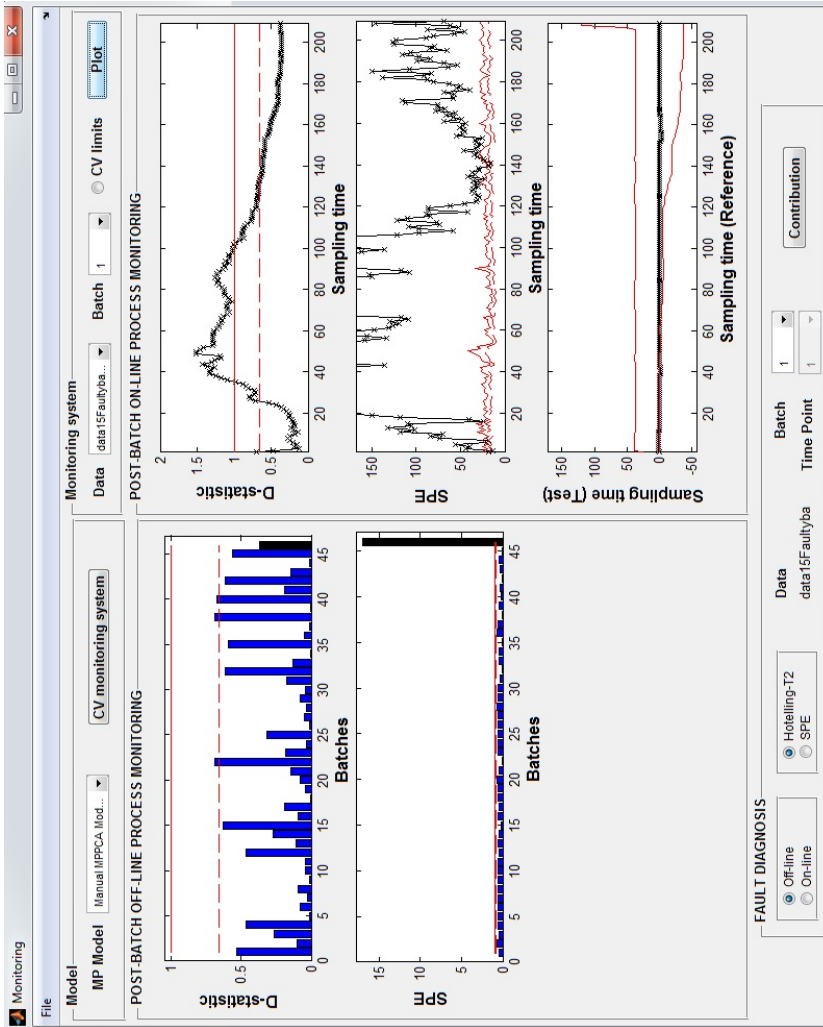


Figure 10.15. Monitoring of the first faulty batch of the test set.

validation approach for a specific model selected in the popmenu 'MP model' (see panel 'Model' at the top-left side of Figure 10.14). To run this procedure, users must press the button 'CV limits'. Depending on the number of sample 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-statistics, SPE and the warping information [2] are plotted jointly with their control limits in the panel 'Post-batch online process monitoring'. In case that a batch-wise PCA model is fitted, the overall D-statistic and SPE control chart are also shown in the panel 'Post-batch offline process monitoring', as is the case of the monitoring scheme designed in Figure 10.14, where the model 'Manual MPPCA Model 1' (batch-wise PCA model with three PCs) is selected. The red control limits are computed using theoretical approximations [82] and the green control limits are corrected by cross-validation. When all the graphs are shown in the interface, the panel 'Monitoring system' is enabled for the monitoring of the calibration data set. To use the corrected control limits in the monitoring of new batches, mark the radiobutton 'Use CV limits'. In case that new batches are desired to be monitored, users can import new data sets using the option 'Import Data' of the menu 'File'. In the popmenus 'Data' and 'Batch' of the panel 'Monitoring system', the data set and the batch to monitor can be selected. Let us load the test set containing 15 faulty batches for exploratory data analysis. To proceed with the monitoring of, for instance, batch #1 of the imported data set, users must select the corresponding data set and batch number, and press the button 'Plot'. In Figure 10.15, the results of the monitoring are shown. As can be seen in the SPE control charts both for post-batch offline and online both in Figure 10.15, the statistics are beyond the control limits, denoting an abnormal situation for that batch.

Upon the detection of out-of-control signals in any of the control charts, the MVBatch Toolbox diagnoses the root causes of these abnormalities via the panel 'Fault diagnosis' (see bottom side of the interface in Figure 10.15). Overall and instantaneous contributions to the D-statistic and SPE are available by marking the radiobuttons 'offline' (overall contributions) or 'online' (instantaneous contributions) and 'D-statistic' or 'SPE'. Both for overall and instantaneous contributions, the selection of a batch is required via the selection of the corresponding batch index

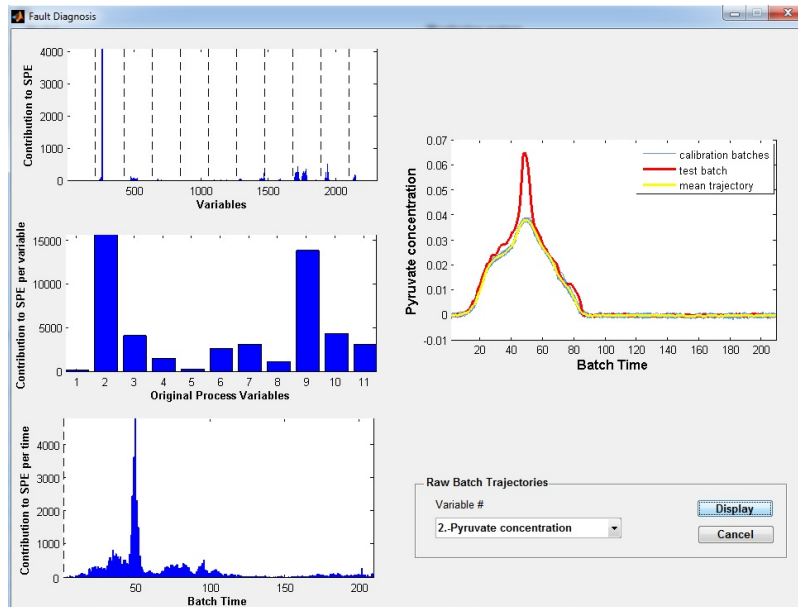


Figure 10.16. Interface for the diagnosis of faulty batches.

in the popmenu 'Batch'. For the instantaneous contributions, the sampling time point at which the contribution plot to the selected statistic will be calculated is also needed (sampling time points of the selected batch available in the popmenu 'Time Point'). To visualize the contributions, users must press the button 'Contribution'. Note that the overall contributions are only available if the multi-phase model fitted corresponds to a batch-wise PCA model.

Figure 10.16 shows the overall contribution plots to the SPE statistic for the first batch of the faulty data set monitored in Figure 10.15. 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 limited by black dashed lines. The second graph at the center-left side represents the accumulated contribution to the statistic of each variable through the batch run. Finally, the third graph at the bottom-left side depicts the contribution to the statistic of all the variables per sampling time point. With these tools, users are able to get insight into the causes of these abnormalities, not only what variables are affected, but also at which time interval

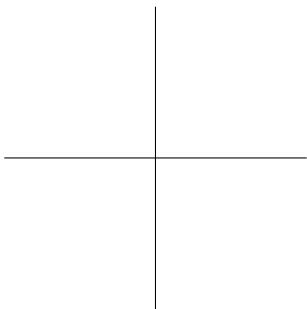
the process behaved differently than normal. To confirm the findings, users can represent the trajectories of a certain process variable of the calibration batches and the faulty batch, and the average trajectory by selecting the variable of interest in the popmenu 'Variable #' and pressing the button 'Display' (see right side of Figure 10.16). Users can always go back to the monitoring interface to interrogate the multi-phase model on other test batches by clicking the button 'Cancel' of the fault diagnosis interface.

10.6 Conclusions

The bilinear modeling of batch processes compromises the application of different steps to end up with a monitoring scheme that is able to detect and diagnose anomalies in the process. The graphical user interface named MVBatch Toolbox is a user-friendly tool designed to carry out these modeling steps: data screening, alignment, modeling and monitoring. In contrast to commercial software packages, the MVBatch Toolbox offers not only the conventional methods in process chemometrics, but also the latest advances proposed in the literature that clearly outperform the former in particular cases, such as Relaxed Greedy Time Warping (RGTW) and Multisynchro for batch synchronization, and multi-phase modeling for batch data calibration. In addition, a simulator of the fermentation process of the *Saccharomyces cerevisiae* cultivation is provided to generate realistic batch data under normal and abnormal operating conditions.

Part V

Conclusions



Conclusions

This thesis is devoted to study the implications of the statistical modeling approaches proposed for batch processes, develop new techniques to overcome the problems not yet solved and apply the developments to simulated data of biochemical processes for illustrative and comparative purposes. The work was initiated by providing a survey on the methods used for the bilinear modeling of batch processes, which were described in the sequence of application in the modeling cycle: alignment, calibration and monitoring. The main drawbacks of the state-of-the-art methods were presented and discussed. Special emphasis was given to batch synchronization and its effect from different aspects of the modeling phases: synchronization quality, changes in the correlation structure, capture of the process dynamics, parameter stability and accuracy to detect abnormal situations. New algorithms were designed to overcome some of the challenges and problems not yet solved in bilinear modeling of batch processes. These new advances were applied to simulated batch data with the purpose of illustrating their performance in comparison to state-of-the-art approaches. These new developments were integrated with the well-known methods for bilinear modeling of batch processes in a Matlab user interface.

Hereafter, the main conclusions of the thesis are summarized, and organized according to the objectives presented at the beginning of this document.

- I. Identify the limitations of the different preprocessing and bilinear process modeling approaches.**

The entire modeling cycle of batch process was thoroughly reviewed, from alignment (including equalization and synchronization) passing through the preprocessing and transformation of the three-way array to some two arrays for bilinear modeling, till process monitoring. Some final conclusions on this goal are drawn per each modeling step below.

Equalization

Three main solutions were investigated to overcome the equalization problem: i) discarding intermediate values, ii) estimating missing values and iii) appropriate arrangement of data. Since there is no certainty that the batch trajectories are synchronized, the application of these equalization strategies are constrained to the within-batch information, which is the only source of information that can be exploited at this step of the modeling.

Equalization methods can severely damage the covariance matrix of the available data, even though the approximation between predicted and observed samples is rather acceptable. Discarding intermediate values can only be carried out if the resulting re-sampling is compliant with the Nyquist-Shannon theorem. Otherwise this equalization procedure is not amenable to apply for further bilinear modeling. When univariate interpolation is performed without caution, the actual relationship of the process variables captured by the covariance matrix might be distorted. There are three main threats to modeling originated from univariate interpolation: i) oscillations generated by the application of polynomials of high degree, the propagation of outliers, and the neglectfulness of the multivariate structure for interpolating values. A solution to these drawbacks is the use of the missing data recovery methods based on the exploitation of the correlation among process variables. In the simulated example it has been shown that these missing data imputation techniques clearly outperform interpolation. Not only the approximation of imputed variables yielded better results than interpolation, but also the multivariate nature of data was preserved.

Regarding the most complex scenario of lack of synchronization, the multi-rate systems, neither discarding intermediate values nor arranging batch data are applicable. In this sit-

uation, all process variables may be collected at different sampling times. Therefore, hardly any complete observation may be found and the complete data set would be discarded according to the first solution. The imputation of the values not sampled in the variables is the most generally applicable method. If a projection model to latent structures is used for data imputation, the data matrix may be enhanced with additional columns with lagged variables. When batch data are affected by non-linear relationships, and these relationships vary over time, the generation of multi-phase multivariate models is required to overcome these common problems in batch processes. However, if batch data are hardly redundant and not collinear, the application of these imputation techniques are inadequate. As an alternative, interpolation might be used at the risk of not reconstructing the time dependent correlation structure in batch data.

Synchronization

Asynchronism in batch processes is a consequence of the misalignment of their features, which define the triggers of events driving the process. The discussion on synchronization revolved around two topics: 1) whether the same synchronization approach must be applied to batch data in presence of different types of asynchronisms, and 2) whether synchronization is always required even though the length of the variable trajectories are constant across batches. Despite the assumption stated in several publications that batch synchronization is only required if the batch trajectories have different duration, equal duration is actually not a sufficient condition to consider batch trajectories to be synchronized. Hence, synchronization is a compulsory modeling step that must be run prior to modeling.

A comparative study in terms of changes in the correlation structure was performed among the most used synchronization methods (IV, DTW, RGTW, TLEC, TLEC-events and their combinations) to determine the implications of inaccurate synchronization. The analysis revealed that those methods focused on linearly or non-linearly synchronizing landmarks of the variable trajectories (SCT-based methods) clearly outperform those that are only aimed at making the trajectories equal in length (TLEC-based methods). The simulated studies indicate that even though the key process

events are not fully aligned by IV or TLEC, a second synchronization using SCT-based methods, which are devoted to reduce variability caused by synchronization and overlap the main features, notably enhances the quality of the synchronization. However, this improvement is not sufficient to reach the performance of SCT-based methods applied in a first synchronization. This is namely due to the perturbation of the trajectories caused by TLEC method, and by the increase of variability produced by IV. IV is a special case of the TLEC-based method, where the difference lies in the dimension where the synchronization is done; the variable domain in the former and the time domain in the latter. When equal spaced intervals are defined in the indicator variable, the batch process is assumed to evolve linearly over time, as in TLEC-based synchronization, which is an assumption rarely met in this type of processes. In case of non-linear process pace, the definition of non-uniform increments or the selection of different IV per phase is needed to ensure the alignment of the events driving the process. In the simulated study, we have seen that IV clearly outperforms TLEC when the process evolution is not linear, provided that non-equal spaced intervals are defined by taking into account the process stages. When synchronization is not focused on aligning the process events, the resulting synchronized batch trajectories may have different correlation over time in comparison to the original ones, which may thereafter affect the outcomes of the preprocessing and calibration steps in the modeling cycle. To sum up, it is crucial going over the nature of the asynchronism present in the process data to decide what synchronization method to use in order to get the optimum results without perturbing the actual relationships of the variables.

With the aim of studying the effects of the synchronization of batch trajectories in bilinear modeling, the parameter stability associated with the most used synchronization methods were investigated. The results of this study confirmed that accuracy in batch synchronization has a profound impact on the PCA model parameter stability. The group of SCT-based methods (e.g. DTW and RGTW) outperforms the group of TLEC-based methods 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 load-

ings. Hence, the better the synchronization of key process events, the better the model parameter stability.

Calibration

A research work was done on the parameter stability associated with the most used principal component analysis-based BMSPC methods (namely single-model, K -models, and hierarchical model approaches). To obtain accurate PCA models for process monitoring, low variability on the model parameters is desired. The existence of uncertainty in both the preprocessing statistics and the latent variables yields to a considerable amount of noise in the model that may affect the performance of the monitoring systems in terms of fault detection and diagnosis. In this study, a notable interaction between the parameter stability in the preprocessing step and in the unfolded model was statistically confirmed.

As conclusion of this research work, we can state that the parameter stability is closely related to the type of preprocessing performed in batch data, and the type of model and unfolding used to transform the three-way data structure to two-way. More accurate conclusions in these issues are drawn as follows:

- * 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 $J \times K$ averages and $J \times K$ 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 $I \times K$ 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 three-way batch data array, this uncertainty is considerably changed.

Those methods that introduce more variables in the model (batch-wise, batch-dynamic, uniformly weighted moving window and exponentially weighted evolving window in its variable-wise version, and adaptive hierarchical K -models, where the latter is a particular case due to its adaptive nature) showed less stability in comparison to those methods that introduce more observations (variable-wise, uniformly weighted moving window and exponentially weighted evolving window in its observation-wise version). 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.

After studying the theoretical implications of the use of different modeling approaches in batch data, we can conclude that three are the 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 online prediction.

Monitoring

At this point of the modeling cycle, the synchronization quality has been shown to have a clear impact on the correlation structure, the process dynamics and the parameter stability. To find out whether these effects are inherited in the monitoring systems producing inaccuracies in fault detection, a comparative study between TLEC-based and SCT-based method was performed. Through this study, it is statistically confirmed that the quality of batch synchronization is one of the critical factors that affects the performance of the monitoring schemes in fault detection. When the key process events do not overlap at the same point of process evolution ensuring the same process pace in all batches, the capability of the monitoring schemes for fault detection is dramatically reduced. Contrary to what is often assumed in practice (as well as in commercial software as e.g. SIMCA Release 13.0.3 by Umetrics), equal length

does not guarantee synchronized batches. Simple methods like TLEC (implemented in SIMCA) linearly interpolate data without considering the overlapping of the key process events. Hence, the asynchronism present in the raw data is inherited in the resulting multivariate statistics. The increase of the variability in batch trajectories due to inappropriate synchronization has an important negative effect. The higher the variability, the lower the performance of the control charts in fault detection.

A second study was performed to analyze the influence of unfolding and preprocessing methods in online fault detection. For this purpose, a MSPC system of a complex biological nutrient removal (BNR) process was designed. The outcomes suggested that the selection of the bilinear modeling is crucial since an inappropriate modeling structure of the process causes negative consequences in the performance of the monitoring scheme. Modeling the BNR process by taking into account the dynamics yielded better results in slow drift fault detection, isolation and diagnosis than the model that only incorporates the variances and instantaneous cross-covariances of the variables. By modeling the BNR process (a continuous process) with cyclical patterns as a batch process and unfolding batch-wise provides: (1) the main non-linear behavior is removed from data and, (2) auto-covariances and lagged cross-covariances are captured. An EWMA approach for control charts and contribution plots was also introduced into the monitoring scheme, yielding a clear improvement in slow drift detection and fault diagnosis. Also, a soft sensor was developed to replace the responsible variables of a probe fault with missing data imputations, taking advantage of the capture of the process dynamics. This type of soft-sensors enables carrying on the monitoring while the involved probes are being repaired.

From the exhaustive study of the bilinear modeling cycle, we can conclude that it is crucial looking through the nature of the asynchronism present in data to decide what synchronization method to use in order to get the optimum results without risking the synchronization quality that directly affects the actual relationships of the variables, the capture of the time-varying process dynamics, the model parameter stability and the accuracy of the monitoring schemes to detect and diagnose faulty situations.

II. Propose new techniques that overcome the limitations pinpointed.

A new time warping algorithm for real-time batch synchronization based on relaxed greedy strategy and the original dynamic time warping (DTW) of Kassidas et al.'s approach, called Relaxed-Greedy Time Warping (RGTW), was proposed. The new proposal avoids assessing the optimal path each time a new sample is available (as in Kassidas et al.'s approach), reducing the uncertainty in the monitoring statistics and predictions in such a way that false alarms are notably reduced in comparison to other SCT-based methods. The main contributions of the RGTW approach are: (1) a new global band constraint definition that takes into account the variability across all batches in each time period; (2) a definition of a cross-validated monitoring window in order to use an optimized window size yielding the minimum local optimal paths (a relaxed greedy strategy); and (3) a way to adapt mapping boundaries to batch length. This last point is highly critical because in the real-time approach the endpoint of the ongoing batch is unknown, therefore, normal boundaries must be adapted by the current batch run. The disadvantage of this method in comparison to the Kassidas et al.'s online implementation is that the start of the monitoring is delayed as many sampling time points as time points of width the optimized window has. Hence, the selection of the window width should be a trade-off between how optimal at least the solution should be and the maximum delay acceptable to start monitoring the process. In addition, the RGTW algorithm is sensible to changes in the operating regimes or to external disturbances affecting the duration of the batches. In this situation, the RGTW parameters should be re-estimated by using new batches affected by the disturbances. Otherwise, the algorithm might produce inaccurate results that might affect the performance of the monitoring system.

Furthermore, the use of the warping information obtained from the RGTW-based batch synchronization both for batch process monitoring and supervised fault classification was addressed. An unsupervised control chart based on the warping profiles from batches ran under normal operating

conditions (NOC) (NOC-WICC) was proposed as a complementary tool to the multivariate statistical control charts for post-batch and real-time batch process monitoring. In case that process faults are fingerprinted in the warping profiles, this chart can be useful to detect their occurrence in the process. Nevertheless, the NOC-WICC may not considerably improve the performance of the traditional multivariate Shewhart-type control charts. This improvement is subject to different factors, such as the nature of the process or the influence of the fault in the process phases, among others. For subtle change detection (ramps, small step changes, etc.), memory control charts, such as EWMA or Cumulative SUM (CUSUM), should be used.

When a rich faulty database is available, warping information can be also used to build the so-called supervised warping information-based control charts (faulty WICC) or to fit classification models using supervised chemometric tools. Although in this thesis simple and widely used tools such as PLSDA and SIMCA are used, other classification techniques could be taken into consideration. In this thesis, the three approaches studied (faulty WICC-, PLSDA, SIMCA-based classifiers) showed good classification performance in terms of the area under the ROC curve (the so-called AUROC). The use of the faulty-WICC-based classifiers depends much on the type of fault; if faults have characteristic fingerprints in their corresponding warping profiles at specific time periods that are different from the rest. The more different the warping profiles from faulty batches, the better the accuracy of the classifier. In contrast, PLSDA and SIMCA-based classifiers are more accurate in fault classification when no clear differences among warping profiles are found.

To cope with the complex number of asynchronisms batches may be affected with, a novel synchronization approach called Multisynchro was proposed, which is based on the DTW and RGTW algorithms. The new proposal is composed of two routines. The first one (high-level routine) is devoted to detect the different patterns of asynchronism of each particular batch based on the warping information derived from the RGTW or DTW algorithm. The second one (low-level routine) performs the batch synchronization using specific procedures based on the nature of the asynchronism. The new approach also includes a procedure that

performs abnormality detection and batch synchronization in an iterative way. This avoids batch abnormalities to affect synchronization quality. The simulated example has shown that the Multisynchro approach might outperform the standard approach of applying the same synchronization procedure, in particular, in the cases where incomplete batches are mixed with batches affected by other types of asynchronisms. These results support the claim that Multisynchro is a promising approach to perform reliable batch synchronizations with multiple and different classes of asynchronisms, which can be used for both post-batch and real-time applications. The main disadvantage of the Multisynchro algorithm is its complexity and non-linear interaction among the different techniques used. In particular, the complex interaction between the iterative and non-linear synchronization procedure with the iterative and linear modeling approach disables its automatic application. To ensure the best synchronization, the visualization and track of the algorithm actions is required.

III. Apply and compare the methods developed by using different industrial scenarios.

Along this thesis, the drawbacks of the state-of-the-art approaches were analyzed and new methods overcoming these problems were proposed. With the objective of transferring the knowledge generated through this research work to academia, and especially to industry, a user-interface named MVBatch Toolbox that integrates the new developments was implemented in Matlab. The design of this tool was based on the main steps of the bilinear modeling of batch processes: data alignment, calibration and monitoring. In each of these methods, the most used techniques in process chemometrics as well as those proposed in this thesis are available in a user-friendly way. The features that make this tool different from other commercial software packages are: 1) its flexibility to synchronize batch data with very diverse types of asynchronism for both post-batch and real-time applications through Multisynchro in its post-batch -DTW- and real-time -RGTW- versions, and 2) its versatility to model complex data by using modeling approaches covering the whole spectrum of possibilities: from variable-wise,

passing through batch dynamic and batch-wise models, to the multi-phase models. Along the manuscript, a comparison of the most used synchronization approaches, such as the IV (implemented in ProMV), and TLEC (implemented in ProMV and SIMCA), has been performed in different scenarios of asynchronisms. Through simulated examples, it has been shown that the proposed synchronization techniques (RGTW and Multisynchro) have better performance than those implemented in commercial software packages in terms of quality of synchronization, perseverance of the time-varying covariance structure, and reduction of the false alarm rate.

Future lines

This dissertation opens some research lines to address in the future:

a) **Technology transfer and development of fast synchronization methods**

New methodologies for batch synchronization and monitoring have been proposed and illustrated with two realistic simulated processes. In order to verify the findings and the performance of these methods, an effort of transferring the generated knowledge into industry is required. This technology transfer has been already initiated in Shell Global Solutions International B.V. The new synchronization approaches are currently being used in the exploratory data analysis of several batch processes in refineries and chemical plants. In addition, the synchronization method RGTW has been successfully implemented and currently it is used for the real-time monitoring of a chemical batch process, with plans of extending its deployment to different plants worldwide within Shell. Based on the application of the methods proposed in this thesis, new needs have raised, in particular, the development of new synchronization approaches that can be applied to scenarios of big data, in which a few Gigabytes of sensor data are available per minute. Due to confidentiality reasons, examples of the application of the new methods to real Shell processes have not been shown in this thesis, and for the same reason, further information

about the nature of these processes and the objectives of the bilinear modeling cannot be disclosed.

b) **Grey modeling of multi-scale biological information of bioprocesses.**

Currently, biotechnical industries are devoted to the production of economically important enzymes and active recombinant proteins from the cultivation of microorganisms genetically modified. The main goal of these industries is to maximize protein yields, decreasing the fermentation duration as much as possible. The production of this high-added value products is governed by a highly correlated factors that requires a multidisciplinary approach (biochemistry, molecular biology, process engineering, biotechnology, etc.). The development of accurate monitoring schemes to control the manufacturing process becomes a challenging task due to the lack of sensors not considering the strong dynamics of the biochemical synthesis process. These dynamics are mainly associated with the intracellular reaction fluxes, much faster than extracellular dynamics and, hence, they are difficult to be measured. Only a few process variables can be measured in biofermentations, such as, pH, temperature, stirring speed, substrate consumption, among others. In this kind of processes, the measurements belonging to biological process variables (inner fluxes of the metabolic network) are key for achieving an accurate process control. The inability to directly measure these key process variables does not mean one cannot extract valuable information related to biological features from the process. In order to develop novel monitoring schemes, the study of the different cellular behaviors to the synthesis process of high value-added products is crucial. This would allow us to know what key variables are strongly involved in the regulation pathway and the regulatory networks.

In this context, first-principles-based models of microbial systems can be developed to discern the principles that govern cellular behavior and to achieve a predictive understanding of cellular functions. For this purpose, a metabolic reaction network of an organism can be modeled assuming that certain constrains operate at steady-state, such as mass balances or reactions irreversibilities; this is the so-called *constraint-based modeling*. The imposed constraints define a solutions space that encloses all the possible states of the

network (i.e., flux distribution through the reactions). The use of statistical techniques to sample the constrained feasible solution space may be useful to obtain a large database where the essential cellular behavior are expressed. In this context, multivariate methods based on projection to latent structures (empirical models) may play an important role in biological systems understanding and dimensionality reduction of metabolic networks. The applications of the projection methods to latent structures may be helpful to discover patterns of heterogeneous protein production, predicting output variables that are extremely difficult to measure, and correlating intra and extracellular reactions in order to understand how the internal state of the cells determines their observed behavior.

Hence, the design of grey models that combine data-driven and knowledge-based information at different scales, showing the process state from different points of view, are of primary interest for the development of the biochemical industries.

c) **Application of pattern recognition methods in multivariate statistical batch process control.**

Hidden Markov Models (HMM) is a pattern recognition method that may be useful as a tool for batch process modeling. Hidden Markov modeling is a stochastic tool where a series of observations are approximated as a sequence of chain events linked by transition probabilities. An HMM can be considered as the simplest dynamic Bayesian network. These models are widely used in the field of continuous speech recognition due to their simpleness to be implemented and their computational feasibility. These models may be used as a tool to synchronize similar key events within batch trajectories. Add to it, the use of the HMM in batch process modeling opens new future perspectives to obtain more parsimonious models which capture more complex dynamical structures in bioprocesses. Also, a new multi-phase framework combining HMM with projection methods to latent structures models may be developed both for offline and online applications. As a future research, the generation of a modeling framework based on HMM that tackles complex, dynamic and, linear and/or non-linear relationships among process variables is proposed. For completion of the study, the comparison with other pattern

recognition techniques is also suggested.

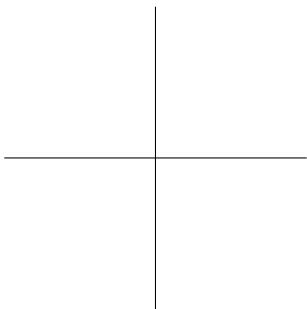
d) **Development of adaptive models for non-stationary dynamics processes**

The monitoring schemes based on projection techniques to latent structures have a major limitation. In industrial processes, changes occur in the process due to unexpected disruptions, e.g. fluctuations in raw material quality, deterioration of the electronic equipment, changes in cellular activity of microorganisms, etc. In this context, the behavior of the process is time-varying and non-stationary, making difficult the implementation of MSPC methods. The time-varying characteristics of an industrial process include: (i) changes in the mean, (ii) changes in the variance, and (iii) changes in the correlation structure between variables, including changes in the number of relevant principal components. In the case that projection methods to latent structures are used for non-stationary process monitoring by assuming the correlation structure keeps constant, false alarms are expected. Hence, the development of adaptive models for non-stationary process monitoring is relevant in the current scientific and industrial context.

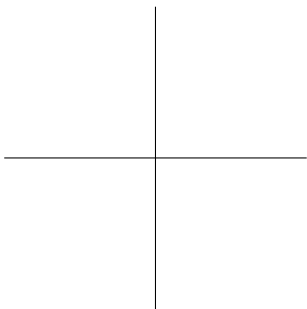
e) **Development of adaptive sensor system models (staircase models) for batch process monitoring**

In the current process industries, thanks to the development and innovation of new measuring instruments and electronic systems, hundreds of measurements of different types at a high sampling frequency can be recorded. The collected variables may be associated with temperatures, pressures, flows, amount of dissolved oxygen in a mixture, etc. (related to process features), or viscosity, redox potential, pH, turbidity, etc. (related to biochemical features). Measurements associated with physical and chemical features of the process can be further obtained from infrared spectroscopy (NIR), gas chromatography (GC), etc. In addition, product quality characteristics, such as color or texture, can be measured by imaging. Using infrared or RGB cameras can be very useful for obtaining information on the turbidity or the amount of alive microorganisms in the mixture of a biofermentation. Due to the cost and time required to obtain certain measurements, some of them derived from complex laboratory analysis, a rational approach is required. The development of adaptive sensor system models (stair-

case models), parsimonious and sensitive, capable of using different types of available measurements depending on the current state of the process (in-control or out-of-control) is of great interest for pharmaceutical and biochemical industries. When the process derives an undesirable state or the process starts to deviate from the target, non-invasive measurements from spectrometers or chromatograms may be used to improve the model sensitivity in terms of fault detection and diagnosis. The use of multiblocks models may be useful to implement the so-called staircase models: a model using variables corresponding to process features when the process is in control (low level); a model taking into account not only the aforementioned variables, but also variables associated with biochemical features in order to carry out a more intensive monitoring when the process is deviating from the target (intermediate level); and a model where all the possible variables are considered, included those obtained from sophisticated non-invasive measurements, to monitor an out-of-control process, making the fault detection and diagnosis easier (high level).



Part VI
Appendices

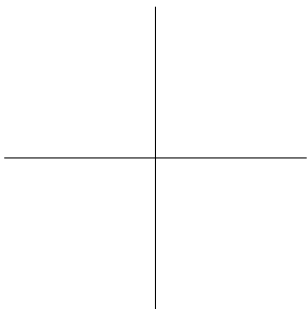


Notation

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

$\tilde{\mathbf{X}}_2$	$(N_2 \times J \times K_{ref})$ three-way array containing the measurements from \mathbf{X}_2 of J process variables synchronized at K_{ref} sampling points in N_2 batches.
$\tilde{\mathbf{X}}_3$	$(N_3 \times J \times K_{ref})$ three-way array containing the measurements from \mathbf{X}_3 of J process variables synchronized at K_{ref} sampling points in N_3 batches.
$\tilde{\mathbf{X}}_4$	$(N_4 \times J \times K_{ref})$ three-way array containing the measurements from \mathbf{X}_4 of J process variables synchronized at K_{ref} sampling points in N_4 batches.
\mathbf{X}_B	$(B_L \times J \times K_b)$ three-way array containing the original measurements of J process variables measured at K_b sampling points in B_L faulty batches, which were isolated in the L iterations of the iterative batch synchronization/abnormalities detection procedure.
$\tilde{\mathbf{X}}_B$	$(B_l \times J \times K_{ref})$ three-way array containing the measurements from \mathbf{X}_B of J process variables synchronized at K_{ref} sampling points in B_l faulty batches, which were isolated at the l -th iteration of the iterative batch synchronization/abnormalities detection procedure.
$\mathbf{X}_{c\#1}$	$(N_1 \times J \times K_{n_1})$ three-way array containing the simulated measurements of J process variables measured at K_1 different sampling points in N_1 batches with class III asynchronism.
$\mathbf{X}_{c\#2}$	$(N_2 \times J \times K_{n_2})$ three-way array containing the simulated measurements of J process variables measured at K_2 different sampling points in N_2 batches with class IV asynchronism.
$\mathbf{X}_{c\#3}$	$(N_3 \times J \times K_{n_3})$ three-way array containing the simulated measurements of J process variables measured at K_3 different sampling points in N_3 batches with class II and III asynchronism.
$\mathbf{X}_{c\#4}$	$(N_4 \times J \times K_{ref})$ three-way array containing the simulated measurements of J process variables measured at K_{ref} sampling points in N_4 batches with class I asynchronism.
\mathbf{X}_n	$(K_n \times J)$ matrix containing the J batch trajectories measured at K_n sampling points of the n -th batch.
$\tilde{\mathbf{X}}_n$	$(K_{ref} \times J)$ matrix containing the synchronized batch trajectories of the n -th batch \mathbf{X}_n .
\mathbf{X}_G	$(G \times J \times K_g)$ three-way array containing the measurements of J process variables measured at K_G sampling points in G normal batches.
$\tilde{\mathbf{X}}_G$	$(G \times J \times K_{ref})$ three-way array containing the measurements from \mathbf{X}_G of J process variables synchronized at K_{ref} sampling points in G normal batches.
\mathbf{X}_{ref}	$(K_{ref} \times J)$ matrix containing the J batch trajectories measured at K_{ref} sampling points in the batch selected as reference for synchronization.



List of Acronyms

AHKM	Adaptive Hierarchical K -Models
ALS	Alternating Least Squares
ANOVA	ANalysis Of VAriance
ARL	Average Run Length
BD	Batch Dynamic
BD1	Batch dynamic unfolding adding 1 lagged measurement vector
BMSPC	Batch Multivariate Statistical Process Control
BNR	Biological Nutrient Removal
BW	Batch-wise
CA	Cluster Analysis
CD	Current Deviations
CGF	Criterion of Goodness of Fit
CMD	Correlation Matrix Distance
COW	Correlation Optimized Warping
CUSUM	CUmulative SUM
CVA	Canonical Variate Analysis
DA	Discriminant Analysis
DDTW	Derivative Dynamic Time Warping
DMoDX	Distance to model
DOF	Degree of freedom
DTW	Dynamic Time Warping
EM	Expectation-Maximization
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
EWMA	Exponentially Weighted Moving Average
FDA	United States Food and Drugs Administration
GCA	Grey Component Analysis
HDDTW	Hybrid Derivative Dynamic Time Warping
IA	Iterative Algorithm
ISL	Imposed Significance Level
IV	Indicator Variable
IV & SCT	Synchronization performed using a SCT-based method after synchronizing batch data with IV
IV1	IV-based synchronization using the fermentation rate as IV
IV2	IV-based synchronization using the biomass concentration as IV
IV1-DTW	DTW-based synchronization after performing a IV-based synchronization with the fermentation rate as IV
IV2-DTW	DTW-based synchronization after performing a IV-based synchronization with the biomass concentration as IV
IV1-RGTW	RGTW-based synchronization after performing a IV-based synchronization with the fermentation rate as IV
IV2-RGTW	RGTW-based synchronization after performing a IV-based synchronization with the biomass concentration as IV
KNN	K-Nearest Neighbors
LCL	Lower Control Limit
LDA	Linear Discriminant Analysis
LM	Local K -model
LMVs	Lagged Measurement Vectors
LSD	Least Significant Difference
LVs	Latent Variables
MCC	Matthews Correlation Coefficient

MCR	Multivariate Curve Resolution
MD	Missing data imputation
MLS	Multivariate Linear Regression
MP	Membership Probability
MSPC	Multivariate Statistical Process Control
NIPALS	Non-linear Iterative PARTial Least Squares
NN	Neural Networks
NOC	Normal Operating Conditions
NOC-WICC	NOC Warping Information-based Control Chart
NSD	Normalized Squared Difference
OLS	Ordinary Least Squares
OWU	Observation Wise Unfolding
OWU-TBWU	Observation Wise Unfolding-T Scores Batch Wise Unfolding
OTI	Overall Type I
OTII	Overall Type II
PARAFAC	PARAllel FACTor
PAT	Process Analytical Technology
PCs	Principal Components
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
PLSDA	Partial Least Squares Discriminant Analysis
PRESS	PREdicted Residual Sum of Squares
QDA	Quadratic Discriminant Analysis
ROC	Receiver Operator Characteristic
RDDTW	Robust Derivative Dynamic Time Warping
RGTW	Relaxed Greedy Time Warping
RR	Ridge Regression
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
SCT	Stretching, Compressing and Translating

SIMCA	Soft Independent Modeling of Class Analogy
SPC	Statistical Process Control
SPE	Squared Prediction Error
SS	Explained Sum of Squares
SSR	Sum of Squared Residuals
STATIS	<i>Structuration des Tableaux A Trois Indices de la Statistique</i>
SVD	Singular Value Decomposition
SVM	Support Vector Machines
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
Trajectory C&S	Trajectory centering and scaling
TSR	Trimmed Score Regression
UCL	Upper Control Limit
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
Variable C&S	Variable centering and scaling
VW	Variable-Wise unfolding
VW-TCS	Variable-wise unfolding after trajectory centering and scaling
VW-VCS	Variable-wise unfolding after variable centering and scaling
WWTP	WasteWater Treatment Plant
ZD	Zero deviations

Projection methods to latent structures for BMSPC

In this appendix, a theoretical description of the projection methods to latent structures most used in process chemometrics, both two-way and N -way models, is presented. Some guidelines to perform batch process monitoring using these multivariate projection methods are also provided. Furthermore, a brief description of the techniques to select the optimum number of latent variables and an ample discussion on the best approach based on the purpose of the use of the model is posed.

A.1 Bilinear models

A.1.1 Principal Component Analysis

PCA is a method to apply to $(N \times J)$ two-way arrays, where N is the number of observations and J are the variables measured on each object. The main ideas of principal component analysis have its origin in the work developed by Pearson [259]. Namely, PCA decomposes the data matrix \mathbf{X} in orthogonal directions of maximum variance, so-called PCs, which are linear combinations of the original variables. So, such method can be seen as a tool to reduce the original J -dimensional variable space to a new A -dimensional subspace ($A \ll J$). PCA methods can be also used to identify patterns on data, trends, clusters, and outliers. The decomposition carried out by PCA can be expressed as

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{P}^T + \mathbf{E} \quad (\text{A.1})$$

where \mathbf{X} is the $N \times J$ data matrix, \mathbf{P} is the $J \times A$ loadings matrix, being A the number of principal components extracted, \mathbf{T} is the $N \times A$ scores matrix and \mathbf{E} is the $N \times J$ residuals matrix. The orthonormal loadings vectors \mathbf{p}_a , $a = 1, \dots, A$, obtained from the eigenvectors of the covariance matrix $\frac{\mathbf{X}^T \cdot \mathbf{X}}{N-1}$, provide the directions with maximum variability, whereas the scores vectors represent the coordinates of the objects into the reduced subspace. As such scores are orthogonal, each vector \mathbf{t}_a (columns of \mathbf{T}) represents different and uncorrelated latent structures in data. In order to calculate the principal components in a sequential manner, the *Non-linear Iterative Partial Least Squares* (NIPALS) algorithm can be used [260]. When a new normalized object of the population is available, it can be projected onto the reduced subspace to obtain its corresponding scores vector as

$$\boldsymbol{\tau}_{new}^t = \mathbf{x}_{new}^t \cdot \mathbf{P} \quad (\text{A.2})$$

where \mathbf{x}_{new}^t is a $1 \times J$ row vector of measurements of J variables, and \mathbf{P} is the $J \times A$ loadings matrix. The appropriate number of principal components can be determined by applying the cross-validation [261, 262]. Nonetheless, depending on the application at hand, the determination of the number of principal components should be addressed in a different way. When the latent model is subjected to be used for prediction or, process understanding or monitoring, the optimum A should be determined from different criteria [263]. For a survey on the matter, readers are referred to Section A.3.

In batch processes, PCA is usually used to analyze the L conditions (e.g. raw material properties, operation shifts, equipment) measure for the N batches contained in the two-way array \mathbf{Z} ($N \times L$), the H landmark features of the process variable trajectories (e.g. minimum, maximum, average temperatures, levels, pressures, values of the process variables at specific time points, slopes, duration of stages) contained in the two-way array \mathbf{X}_L ($N \times H$) and even two-way matrices unfolded by using any of the approaches explained in Chapter 1.

Regression

Let us assume two matrices are available, a $(N \times J)$ process data matrix \mathbf{X} containing measurements of engineering process variables, and also a $(N \times M)$ output matrix \mathbf{Y} belonging to quality measurements of the process. The goal is to estimate the matrix \mathbf{Y} using the data available in \mathbf{X} . Such linear regression problem can be formulated as

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{B} + \mathbf{E} \quad (\text{A.3})$$

where \mathbf{B} is the $(J \times M)$ matrix of regressors and \mathbf{E} constitutes the $(N \times M)$ matrix of residuals.

A solution to the above regression problem can be reached by using *Ordinary Least Squares* (OLS), which minimizes the sum of squares of the residual (\mathbf{E}):

$$\hat{\mathbf{B}}_{OLS} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{Y} \quad (\text{A.4})$$

Nonetheless, the least-squares solution cannot be calculated when the independent variables are highly correlated since $\mathbf{X}^T \cdot \mathbf{X}$ is ill-conditioned and the variance of the OLS estimators becomes large (unstable parameters). Hence, the good performance of this solution is highly dependent on the degree of independence of the variables from the matrix \mathbf{X} . In the literature several regression methods that overcome the inversion problem of ill-conditioned matrices, such as PLS [255], *Principal Component Regression* (PCR) [264, 265] and *Ridge Regression* (RR) [266] have been proposed. In the following, the two first methods will be described briefly since both are the most used in the literature, as they are able to deal with singular covariance matrices.

A.1.2 Partial Least Squares

PLS is an alternative to solve the linear regression problem presented in Equation A.3 in scenarios where OLS cannot be used. The intention of PLS is to predict the matrix \mathbf{Y} by using the latent variables or components of the matrix \mathbf{X} that maximizes its covariance with the latent structure of \mathbf{Y} . In this case, the partial linear regression problem between the normalized matrices \mathbf{X} and \mathbf{Y} can be defined as

$$\begin{aligned}\mathbf{X} &= \mathbf{T} \cdot \mathbf{P}^T + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T} \cdot \mathbf{Q}^T + \mathbf{F}\end{aligned}\tag{A.5}$$

where \mathbf{T} is a $N \times A$ matrix of latent variables or scores, \mathbf{P} is a $J \times A$ matrix of \mathbf{X} loadings, \mathbf{Q} is a $M \times A$ matrix of \mathbf{Y} loadings (\mathbf{P} and \mathbf{Q} show how the latent variables are related to the original \mathbf{X} and \mathbf{Y} variables, respectively), and \mathbf{E} and \mathbf{F} are the $N \times J$ and $N \times M$ matrices of residuals of \mathbf{X} and \mathbf{Y} , respectively. As the latent variables in \mathbf{T} are orthogonal and the number of components is quite lower than the number of observations, the inversion of the matrix $\mathbf{T}^T \cdot \mathbf{T}$ is feasible. Hence, let us express the PLS model based on Equation A.5 as

$$\hat{\mathbf{Y}} = \mathbf{X} \cdot \mathbf{B}_{PLS} = \mathbf{T} \cdot \mathbf{Q}^T\tag{A.6}$$

where $\mathbf{T} = \mathbf{X} \cdot \mathbf{W}^* = \mathbf{X} \cdot \mathbf{W}(\mathbf{P}^T \cdot \mathbf{W})^{-1}$, and \mathbf{W} is a $J \times A$ matrix of weights. So

$$\mathbf{B}_{PLS} = \mathbf{W}(\mathbf{P}^T \cdot \mathbf{W})^{-1} \cdot \mathbf{Q}^T\tag{A.7}$$

Hence, if a new observation vector \mathbf{x}_{new} is available to be projected onto the PLS model, its prediction vector $\hat{\mathbf{y}}_{new}$ can be estimated as

$$\hat{\mathbf{y}}_{new}^T = \mathbf{x}_{new}^T \cdot \mathbf{B}_{PLS}\tag{A.8}$$

In addition to overcoming the multicollinearity presented in data and the inversion problem in ill-conditioned matrices, PLS handles missing data. In order to estimate the PLS regressors, several algorithms, such as NIPALS [267] and SIMPLS [268], can be used. The latter algorithm has the advantage of being computational faster than the NIPALS algorithm.

A.1.3 Principal Component Regression

PCR performs a solution to the linear regression problem (see Equation A.3) in a similar way than PLS when the variables are closed to be collinear. In this case, PCR extracts the A directions of maximum variance (PCs) of the matrix \mathbf{X} and later on, \mathbf{Y} is regressed onto these A principal components as

$$\mathbf{Y} = \mathbf{T} \cdot \mathbf{B}_{PCR} + \mathbf{E}\tag{A.9}$$

$$\mathbf{B}_{PCR} = (\mathbf{T}^T \cdot \mathbf{T})^{-1} \cdot \mathbf{T}^T \cdot \mathbf{Y} \quad (\text{A.10})$$

where \mathbf{T} is a $N \times A$ scores matrix. Due to the orthogonality of the scores, the inversion of the matrix $\mathbf{T}^T \mathbf{T}$ can be carried out, yielding accurate estimations of the least squares regression coefficients ($\hat{\mathbf{B}}_{PCR}$). Such regressors can be used together with the scores matrix \mathbf{T} to estimate \mathbf{Y} as

$$\hat{\mathbf{Y}} = \mathbf{T} \cdot \mathbf{B}_{PCR} \quad (\text{A.11})$$

As $\mathbf{T} = \mathbf{X} \cdot \mathbf{P}$, where \mathbf{P} is the $J \times A$ loadings matrix of the PCA model, the matrix of regressors coefficients used to estimate \mathbf{Y} from \mathbf{X} can be defined as

$$\mathbf{B}_{PCR} = \mathbf{P} \cdot (\mathbf{T}^T \cdot \mathbf{T})^{-1} \cdot \mathbf{T}^T \cdot \mathbf{Y} \quad (\text{A.12})$$

A.2 N -way modeling

Let us assume batch process data are arranged in a three-way array $\underline{\mathbf{X}}$ ($N \times J \times K$), where K is the number of sampling points at which J variables were recorded in all the N batches. This three-way batch data array can be analyzed with several approaches of different nature. One of the most used in the literature is the unfold principal component analysis (uPCA) or also called multiway-PCA [82]. Other methods used to analyze this kind of data are the so-called trilinear methods, such as PARAFAC, Tucker3 and N-PLS.

A.2.1 Parallel Factor Analysis

The PARAFAC model is an extension of bilinear factor models to multilinear data [269]. Such type of modeling is based on Catell's principle of Parallel Proportional Profiles [270]. In the context of batch process monitoring, the PARAFAC model of the three-way array $\underline{\mathbf{X}}$ with A components can be defined by the following expression:

$$x_{njk} = \sum_{a=1}^A t_{na} p_{ja}^v p_{ka}^t + e_{njk} \quad (\text{A.13})$$

where x_{njk} is the value of the three-way array $\underline{\mathbf{X}}$ ($N \times J \times K$) for the j -th variable at the k -th sampling time of the n -th batch,

A is the number of extracted factors, t_{na} is an element of the scores matrix \mathbf{T} ($N \times A$) belonging to the n -th batch in the a -th factor (batch mode), p_{ja}^v is an element of the loadings matrix \mathbf{P}^v ($J \times A$) belonging to the j -th variable in the a -th component (variable mode), p_{ka}^t is an element of the loadings matrix \mathbf{P}^t ($K \times A$) belonging to the k -th sampling time in the a -th factor (time mode) and e_{njk} is an element of the residuals matrix \mathbf{E} ($N \times J \times K$) corresponding to the x_{njk} value. The PARAFAC model in Eq. A.13 can be also expressed in a two-way matrix representation as

$$\mathbf{X} = \mathbf{T}(\mathbf{P}^t \circ \mathbf{P}^v)^T + \mathbf{E} \quad (\text{A.14})$$

where \circ denotes the Khatri-Rao product.

The main feature of PARAFAC model is its no rotational freedom under certain conditions, i.e., the scores matrix \mathbf{T} and the loadings matrices \mathbf{P}^v and \mathbf{P}^t cannot be changed without changing the residuals. This provides the uniqueness property that makes such modeling technique really interesting in different research fields. Another important constraint of PARAFAC models is that the factors in the different modes (batch, variable or time mode) can only interact factor-wise. In other words, the a -th factor or component in the first mode can only interact with the a -th factor in the second and third mode. Consequently, it is expected the same number of factors is extracted in each mode.

The algorithm most used to fit a PARAFAC model is the *Alternating Least Squares* (ALS) due to its simplicity to be implemented and to incorporate additional constraints. For further details, readers are referred to [271, 272].

When batch process data registered under normal operating conditions (NOC) have been modeled by PARAFAC models, data corresponding to a new batch \mathbf{x}_{new} ($KJ \times 1$) can be compared with NOC data. For that, the vector \mathbf{x}_{new} is projected onto the PARAFAC model to estimate both the new score vector $\boldsymbol{\tau}_{new}$ ($A \times 1$) and the new residual vector \mathbf{e}_{new} ($KJ \times 1$) as

$$\begin{aligned} \boldsymbol{\tau}_{new} &= [(\mathbf{P}^t \circ \mathbf{P}^v)^T (\mathbf{P}^t \circ \mathbf{P}^v)]^{-1} (\mathbf{P}^t \circ \mathbf{P}^v)^T \mathbf{x}_{new} \\ \mathbf{e}_{new} &= \mathbf{x}_{new} - (\mathbf{P}^t \circ \mathbf{P}^v) \boldsymbol{\tau}_{new} \end{aligned} \quad (\text{A.15})$$

Some applications of PARAFAC models for batch process monitoring can be found in the literature [141, 273, 274, 275, 276].

A.2.2 Tucker-3

Similar to PARAFAC model, the Tucker-3 model [277, 278, 279] is an extension of bilinear factor analysis with the difference that the latter is more flexible allowing interactions between factors of different modes. Following with the batch process modeling, it is possible to fit a Tucker-3 model to the three-way array \mathbf{X} with A_b , A_v and A_t components or factors for the first (batch), second (variable) and third (time) mode by following the expression

$$x_{njk} = \sum_{a_b=1}^{A_b} \sum_{a_v=1}^{A_v} \sum_{a_t=1}^{A_t} t_{na_b} p_{ja_v}^v p_{ka_t}^t g_{a_b a_v a_t} + e_{njk} \quad (\text{A.16})$$

where x_{njk} is the value of the three-way array \mathbf{X} ($N \times J \times K$) for the j -th variable at the k -th sampling time of the n -th batch; A_b, A_v, A_t are the number of extracted factors for each of the modes (batch, variable and time); t_{na_b} is an element of the scores matrix \mathbf{T} ($N \times A_b$) belonging to the n -th batch in the a_b -th factor (batch mode); $p_{ja_v}^v$ is an element of the loadings matrix \mathbf{P}^v ($J \times A_v$) belonging to the j -th variable in the a_v -th component (variable mode); $p_{ka_t}^t$ is an element of the loadings matrix \mathbf{P}^t ($K \times A_t$) belonging to the k -th sampling time in the a_t -th factor (time mode); e_{njk} is an element of the residuals matrix \mathbf{E} ($N \times J \times K$) corresponding to the x_{njk} value; and $g_{a_b a_v a_t}$ is the element of the three-way core array \mathbf{G} associated to the interaction among the a_b -th, a_v -th and a_t -th factors of the three modes. A two-way representation of the Tucker-3 model is given by

$$\mathbf{X} = \mathbf{T}\mathbf{G}(\mathbf{P}^t \otimes \mathbf{P}^v)^T + \mathbf{E} \quad (\text{A.17})$$

where \otimes indicates the Kronecker product and \mathbf{G} ($A_b \times A_v \times A_t$) is the two-way representation of the three-way core array \mathbf{G} .

It is worth commenting that Tucker-1 and Tucker-2 models (belonging to the Tucker family models) perform a compression of one and two modes, respectively, unlike the Tucker-3 where the compression is carried out in the three modes.

Once the Tucker-3 model is built, a new batch \mathbf{x}_{new} ($KJ \times 1$) can be projected on the latent model to identify whether such batch is statistically 'good' or 'bad' in reference to NOC batches.

This projection is achieved by assessing the score vector $\boldsymbol{\tau}_{new}$ ($A_b \times 1$) and the new residual vector \mathbf{e}_{new} ($KJ \times 1$) as

$$\begin{aligned}\boldsymbol{\tau}_{new} &= (\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{G}[(\mathbf{P}^t \otimes \mathbf{P}^v)^T(\mathbf{P}^t \otimes \mathbf{P}^v)]^{-1}(\mathbf{P}^t \otimes \mathbf{P}^v)^T\mathbf{x}_{new} \\ \mathbf{e}_{new} &= \mathbf{x}_{new} - (\mathbf{P}^t \otimes \mathbf{P}^v)\mathbf{H}^T\boldsymbol{\tau}_{new}\end{aligned}\tag{A.18}$$

As in PARAFAC models, the ALS algorithm is widely used. In [280, 141, 281, 273, 282, 283] applications of such models in batch process monitoring can be found.

A.2.3 Other models

Another model focused on three-way data analysis is the so-called Structuration des Tableaux A Trois Indices de la Statistique (STATIS) method [284]. Unlike N-way analysis methods, this approach explores each mode separately. Each one of the samples is denoted as a slice of a three-way array and the covariance matrix of that slice is estimated. The basic idea of STATIS is to apply a PCA model to a global covariance matrix formed by a linear combination of the covariance matrices corresponding to each one of the slices. For further details, readers are referred to [284, 285].

On the other hand, when a quality variables matrix \mathbf{Y} is available, as in the case of PCR or PLS, a PARAFAC or a Tucker-family model can be fitted from the three-way array \mathbf{X} , and subsequently, the matrix \mathbf{Y} can be regressed on the score matrix obtained previously. Furthermore, an extension of the PLS algorithm (N-PLS) [286] was proposed to N -way data. A discussion on some theoretical aspects of using different three-way models for batch process data and some practical consequences can be found in [280, 287, 288].

A.3 Selection of the number of principal components

The selection of the appropriate number of PCs relies on the application of the multivariate model at hand [289, 290]. In [181], the most common objectives pursued in the use of PCA models in chemometrics were distinguished based on the type of application: when the interest is in the observations (a), in the

latent variables (b), and in the distributions in latent variables and residuals (c). In the first category, the goal is to fit a PCA model with a certain number of PCs such that the estimation of the original variables is the most accurate. Examples of application in this category is data compression and imputation, in which a precise representation of raw data is needed. When the focus is on the latent structure, for instance in exploratory analysis, the main goal is to interpret the multivariate model to gain data and process understanding. For this purpose, the first few components are usually used since they are the factors capturing the main and largest variation. Nevertheless, special caution should be taken since systematic information neither may have larger variance than noise [181] nor be irrelevant [291]. Hence, it is advised in this case not to fix the number of principal components, but exploring the different components and verifying at once that the relevant information is captured in the model. Finally, one of the broadest use of PCA in industry is the design of monitoring schemes. When using PCA for other purpose, the loading matrix \mathbf{P}_A is considered as the model itself since it is the unique model parameter employed to check the membership of incoming samples to the in-control latent structure. In statistical process monitoring, the scores \mathbf{T}_A and residuals \mathbf{E}_A are used to estimate the limits of their corresponding control charts though. Consequently, these parameters are considered part of the model as well. In this case, the objective is to select the optimum number of PCs so that the distributions of the model parameters defined from in-control historical data are representative of the distributions of incoming data, as long as they remain under control. At this point it is worth stressing that it is recommendable to validate the number of components in terms of the actual goal with the aim of ensuring that the useful information is retained in the multivariate model.

Much work has been devoted, mainly in PCA, to discuss and propose techniques to assess the suitable number of *Latent Variables* (LVs) in such a way that the model describes the systematic variation in process data and not in noise. Bartlett's test [292] is a statistical test for equality of the eigenvalues. This procedure tests the null hypothesis that the correlation matrix is an identity matrix in which all of the diagonal elements (eigenvalues) are 1 and all off diagonal elements are 0. The alternative hypothesis is that the largest one is different from the others. When the null hypothesis is rejected this means the

largest eigenvalue is different from the remaining ones, therefore, the corresponding principal component or factor is retained. This test is repeated until the null hypothesis is accepted, i.e. the belonging eigenvalue is not different from the rest, so the remaining PCs are not retained. The main drawback is its sensitivity to non-normal errors. A simpler method used in the literature is to retain those principal components whose eigenvalues exceed a specified value, usually a value greater than for autoscaled data (so-called Kaiser's rule or Kaisers-Guttman's rule) [185]. However, it is not recommended to be used without having prior knowledge about data. Another intuitive tool to select the number of PCs is the use of the SCREE test [293]. It is based on the assumption that relevant information is larger than random noise and that the magnitude of the variation of random noise seems to gradually decrease as a function of the number of components. Typically, a figure where the eigenvalues are plotted in decreasing order of value is used. The decision about the factors to be retained is based on the PC whose eigenvalue represents an 'elbow' in the shape of the curve. This method is not suitable from a pragmatic point of view because the decision is very subjective and cannot be implemented in an automatic framework for the selection of the number of components. A more realistic cut off for the eigenvalues is obtained through the broken stick rule [294]. This procedure assumes that if the total variance is divided randomly among the various components, then the expected distribution of the eigenvalues will follow a broken-stick distribution. Actual eigenvalues are considered interpretable if they exceed eigenvalues generated by the broken-stick model. A survey on stopping rules for determining the number of factors to retain in PCA can be found in [295].

Cross-validation has probably become the standard method to determine the optimum number of principal components since Wold proposed it [261]. In cross-validation, data are divided in k groups. Each time, a model is calibrated from the whole data-set but a group. Afterward, the data from that group are predicted using the model, and a *Criterion of Goodness of Fit* (CGF) is computed. This is repeated until all groups are left out once and only once, and a total CGF for a model is obtained. In PCA the CGF is computed for the models with A PCs. From the shape of the CGF, the optimum number of PCs is estimated. Wold suggested to retain PCs while the following index does not exceed the value of 1, which indicates that the predictions are

improved with the inclusion of the last component:

$$R_a = \frac{PRESS_a}{SSE_{a-1}} \quad (\text{A.19})$$

where SSE_{a-1} is the sum of squared residuals after retaining $a-1$ principal components. Later on, Eastment and Krzanowski proposed an alternative method to Wold's approach [262] based on the *Singular Value Decomposition* (SVD) algorithm. In the selection of PCs to be extracted, they proposed to add the a -th principal components as long as the following index W exceeds the unity:

$$W_a = \frac{(PRESS_{a-1} - PRESS_a)/DOF_a}{PRESS_a/DOF_a^{rem}} \quad (\text{A.20})$$

where DOF_a is the number of degrees of freedom used in fitting the a -th component and DOF_a^{rem} is the number of degrees of freedom remaining after fitting the a -th principal component.

The simplest cross-validation method is the so-called row-wise k -fold cross validation algorithm or *rkf* (see Algorithm A.1). In this method, the observations -rows- of the data matrix \mathbf{X} are arranged in k groups, typically in a random way. In each cross-validation iteration, one group is left out ($X_{\#}$) and a PCA model is calibrated from the rest (X_*). Then, the residual error $\mathbf{E}_{\#}^A$ matrix of the left out group of observations is computed with the model. This procedure is repeated until every sample is left out once and only once. Once all the iterations have been reached, the matrix of predictions errors \mathbf{E}^A and the PRESS, which is used as CGF, can be computed. The *rkf* method yields strictly decreasing curves of PRESS in terms of A . To determine the number of PCs, a threshold can be applied. Thus, if the decrease of PRESS when adding the a -th PC is lower than the threshold, the PC is discarded and the model selected contains $a-1$ PCs. Also, the curve can be corrected with the degrees of freedom consumed [289]. This method has been criticized by several authors due to the lack of independence in the estimation of prediction errors since the PCA estimates are computed using the actual values as input [263, 289].

In order to solve the problem of independence in the reconstruction error, the element-wise k -fold (*ekf*) algorithm (originally proposed by Wold as an alternative algorithm to the proposed one in [261]) is suggested to be used. This method is an extension of the *rkf* algorithm. This performance is based on two

Algorithm A.1: Row-wise k -fold (rkf) cross-validation algorithm

For each PC ($a = 1, \dots, A$)
 For each group of samples ($g = 1, \dots, G$)
 Form \mathbf{X}_* with data from all groups but g
 Form $\mathbf{X}_\#$ with data from g
 Fit a PCA model from \mathbf{X}_* , obtaining \mathbf{P}_*^a and \mathbf{T}_*^a
 $\mathbf{T}_\#^a = \mathbf{X}_\# \cdot \mathbf{P}_*^a$
 $\hat{\mathbf{X}}_\# = \mathbf{T}_\#^a \cdot (\mathbf{P}_*^a)^T$
 $\mathbf{E}_g^a = \mathbf{X}_\# - \hat{\mathbf{X}}_\#$
 end
 Combine matrices \mathbf{E}_g^a in \mathbf{E}^a
 Calculate $PRESS_a = \sum_{n=1}^N \sum_{j=1}^J e_{i,j}^a$
 end

Algorithm A.2: Element-wise k -fold (ekf) cross-validation algorithm

For each PC ($a = 1, \dots, A$)
 For each group of samples ($g = 1, \dots, G$)
 Form \mathbf{X}_* with data from all groups but g
 Form $\mathbf{X}_\#$ with data from g
 Fit a PCA model from \mathbf{X}_* , obtaining \mathbf{T}_*^a and \mathbf{P}_*^a
 For each group of variables ($j = 1, \dots, J$)
 Set $\mathbf{X}_{\#,j} = 0$
 Repeat until $\mathbf{X}_{\#,j}$ converges
 $\mathbf{T}_\#^a = \mathbf{X}_\# \cdot \mathbf{P}_*^a$
 $\hat{\mathbf{X}}_\# = \mathbf{T}_\#^a \cdot (\mathbf{P}_*^a)^T$
 end
 Restore its actual value to $\mathbf{X}_{\#,j}$
 $\mathbf{E}_{g,j}^a = \mathbf{X}_{\#,j} - \hat{\mathbf{X}}_{\#,j}$
 end
 end
 Combine matrices $\mathbf{E}_{g,j}^a$ in \mathbf{E}^a
 Calculate $PRESS_a = \sum_{n=1}^N \sum_{j=1}^J (e_{n,j}^a)^2$
 end

main ideas: a) in order to guarantee the independence, the data used to fit a PCA model in each iteration are not the same as those used to predict; and b) since the PCA model establishes relation structures among variables, the goodness of fit should be measured by predicting a variable from the remaining ones. Both ideas are implemented in the loops of the *ekf* algorithm (see Algorithm A.2). As can be seen, the first main idea is implemented in the outer loop where data are split into the test (the data group g) and training sets (the remaining data groups). The second idea is performed through the inner loop, iterating through all the variables in order to set the variable v as missing by imposing the value 0 ($\mathbf{X}_{\#,j} = 0$). Later on, the value of such variable is reconstructed through the latent variables of the PCA model fitted from \mathbf{X}_* . Afterward, the reconstruction error of the values corresponding to the j -th variable is estimated, guaranteeing the independence between estimated and actual values and the actual ones. Once all the A PCs set by the user have been iterated, the algorithm provides the matrix of prediction errors \mathbf{E}^a (with elements $e_{n,j}^a$ in the n -th row and j -th column).

The *ekf* algorithm provides a PRESS curve that usually presents a valley shape, with a minimum value. In principle, the lowest value of the PRESS curve denotes the appropriate number of PCs. For the sake of clarity, the preprocessing issue in cross-validation have not been directly denoted in the *rkf* and *ekf* algorithms shown in Algorithm A.1 and A.2. Nevertheless, a preprocessing step based on data centering and scaling is required. As the *ekf* algorithm is used in this dissertation to determine the optimum number of PCs to build predictive models, the preprocessing information is estimated from the training set \mathbf{X}_* instead of \mathbf{X} . Although the use of cross-validation is widely spread, its indiscriminate use has been criticized recently [181, 263].

A.4 Online application of projection methods to latent structures

For the online application of PCA/PLS batch-wise models (multivariate projection method applied to the two-way array after unfolding the three-way array in the batch direction and properly preprocessing), the imputation of future values is required. Only the values of the J process variables measured from the beginning up to k -th sampling point are known whereas the

remaining trajectory, values from the $(k + 1)$ -th sampling point until the end of the batch is unknown.

In the case of the batch-dynamic models, only the measurements during the initial period need to be imputed. For instance, assume that n_L LMVs have added to the variable-wise unfolded array \mathbf{X} ($NK \times J$), yielding the two-way array $\underline{\mathbf{X}}$ ($(K - n_L)N \times (n_L + 1)J$). Hence, provided that $k < n_L + 1$, the remaining future $n_L + 1 - k$ measurement vectors must be imputed. Once $(n_L + 1)$ sampling times have elapsed, the projection of the measurement vectors is straightforwardly done as explained in previous sections. Note that the length of the measurement to be imputed varies based on the number of LMVs included in the model. For multi-phase models, the same procedure as in the batch-dynamic models is applied since there are as many dynamic models as phases.

Several approaches have been proposed in the literature to overcome the problem of missing trajectories in on-line applications. Nomikos and MacGregor [82] suggested using: i) *Zero deviations* (ZD), ii) *Current Deviations* (CD) or iii) *Missing data imputation* (MD) approach. The first approach assumes that the unknown part (or future observations) will behave at the desired level stated by NOC data, i.e. the remaining part of the trajectories is expected to follow the mean trajectory. In this case, the unknown part of the measurement vectors is filled with values equal to zero once the average trajectory has been subtracted from the known part. The CD approach assumes that the current deviation of the mean-centered and autoscaled trajectories from the mean trajectory at the k -th sampling time point will remain up to the batch completion. Hence, the CD approach is performed by adding the current deviation to the values of the unknown part. The MD approach is based on replacing the unknown values $\tilde{\mathbf{x}}_{new,k+1:K}$ by missing data imputation computed from the PCA/PLS model. A good survey on missing imputation techniques for process chemometrics can be found in [136]. Garcia *et al.* [168] presented a comparative study of several missing data imputation techniques, leading to the conclusion that TSR is one of the recommended methods when the orthogonality, coherence and accuracy of the score estimates (and hence accuracy of missing data predictions) are considered for on-line trajectory prediction using NOC batches. However, García *et al.* found from their simulated case study that fault monitoring performance showed no statistical evidence

of a difference among the methods because the control charts are tailored to the missing data imputation method used. They also noted that a faulty batch breaks the "in-control" assumption and so breaks the assumptions of the theoretical basis for the various methods.

In this manuscript, the TSR method for PCA [169] and PLS [89] is used to compute the imputation of the future measurement vectors. TSR is performed by estimating the value of the scores from the trimmed scores, i.e. the scores obtained by filling the missing values with zeros (ZD approach). Assuming that the new batch is Trajectory C&S preprocessed and thereafter batch-wise unfolded at the k -th sampling time point, the measurements vector of such batch can be expressed as follows:

$$\mathbf{z}_k = [z_{11}, \dots, z_{J1}, \dots, z_{1k}, \dots, z_{Jk}, 0, \dots, 0]^T \quad (\text{A.21})$$

where z_{jk} is the value of the j -th variable at the k -th sampling point of batch \mathbf{z} .

The array \mathbf{z} can be split up in two parts, the known $\mathbf{z}_{1:k}^*$ and unknown $\mathbf{z}_{k+1:K}^\#$ part:

$$\mathbf{z}_k = [\mathbf{z}_{1:k}^*, \mathbf{z}_{k+1:K}^\#] \quad (\text{A.22})$$

Likewise, the loading \mathbf{P}^T and weighting \mathbf{W}^T matrices obtained from the PCA and PLS models, respectively, can be also divided into two parts:

$$\mathbf{P}^T = [\mathbf{P}_{A,1:kJ}^{*T}, \mathbf{P}_{A,kJ+1:KJ}^{\#T}] \quad (\text{A.23})$$

$$\mathbf{W}^T = [\mathbf{W}_{A,1:kJ}^{*T}, \mathbf{W}_{A,kJ+1:KJ}^{\#T}] \quad (\text{A.24})$$

The trimmed scores at the k -th sampling time point are calculated by using the PCA or PLS model as follows, respectively:

$$\boldsymbol{\tau}_A^{PCA} = \mathbf{P}_A^T \cdot \mathbf{z}_k \quad (\text{A.25})$$

$$\begin{aligned} \boldsymbol{\tau}_A^{PLS} &= \mathbf{R}_A^T \cdot \mathbf{z}_k \\ \mathbf{R}_A &= \mathbf{W}_A \cdot (\mathbf{P}_A^T \cdot \mathbf{W}_A)^{-1} \end{aligned} \quad (\text{A.26})$$

Alternatively, the latter equations can be expressed as follows:

$$\boldsymbol{\tau}_A^{*PCA} = \mathbf{P}_A^* \cdot \mathbf{z}_k^* \quad (\text{A.27})$$

$$\begin{aligned}\tau_A^{*PLS} &= \mathbf{R}_A^{*T} \cdot \mathbf{z}_k^* \\ \mathbf{R}_A^* &= \mathbf{W}_A^* \cdot (\mathbf{P}_A^T \cdot \mathbf{W}_A)^{-1}\end{aligned}\quad (\text{A.28})$$

Note that in the case of PLS, the complete matrices \mathbf{P}_A and \mathbf{W}_A are used in the inversion of Equation A.28, thereby the prediction capability is enhanced by using the information of the complete model.

The calibration data in $\tilde{\mathbf{X}}$ can be used to improve the estimation of the missing values belonging to the unknown part of $\mathbf{z}_{k+1:K}^\#$ at the k -th sampling point. Let us call $\tilde{\mathbf{X}}^*$ the sub-matrix of $\tilde{\mathbf{X}}$ with the variables available in $\mathbf{z}_{1:k}^*$ and \mathbf{P}_A^* the corresponding submatrix of \mathbf{P} . Thus, the matrix of trimmed scores corresponding to the calibration data can be defined for PCA and PLS as follows, respectively:

$$\mathbf{T}_A^* = \mathbf{X}^* \cdot \mathbf{P}_A^* \quad (\text{A.29})$$

$$\mathbf{T}_A = \mathbf{X}^* \cdot \mathbf{W}_A \cdot (\mathbf{P}_A^T \cdot \mathbf{W}_A)^{-1} \quad (\text{A.30})$$

The score matrix of the data used to create the model \mathbf{T}_A can be regressed on the trimmed scores \mathbf{T}^* :

$$\mathbf{T}_A = \mathbf{T}_A^* \mathbf{B} + \mathbf{E} \quad (\text{A.31})$$

where the matrix of regressors \mathbf{B} can be computed from ordinary least squares since the inversion of $\mathbf{T}_A^{*T} \cdot \mathbf{T}_A^*$ is well-conditioned:

$$\hat{\mathbf{B}} = (\mathbf{T}_A^{*T} \cdot \mathbf{T}_A^*)^{-1} \cdot \mathbf{T}_A^{*T} \cdot \mathbf{T}_A \quad (\text{A.32})$$

Afterward, \mathbf{B} is used to improve the estimation of the scores in Equations A.27 and A.28 for PCA and PLS, respectively:

$$\hat{\tau}_A^{PCA} = \hat{\mathbf{B}}^T \cdot \mathbf{P}_A^{*T} \cdot \mathbf{z}_k^* \quad (\text{A.33})$$

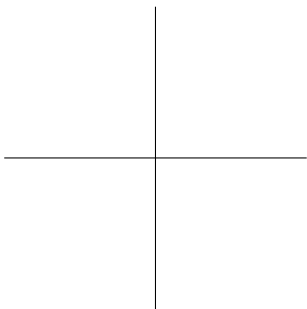
$$\hat{\tau}_A^{PLS} = \hat{\mathbf{B}}^T \cdot \mathbf{R}_A^{*T} \cdot \mathbf{z}_k^* \quad (\text{A.34})$$

The expression for \mathbf{B} derived for PCA [169] and PLS [89] can respectively be expressed as:

$$\hat{\mathbf{B}}^{PCA} = (\mathbf{R}_A^{*T} \cdot \mathbf{S}^{**} \cdot \mathbf{R}_A^*)^{-1} \cdot \mathbf{R}_A^{*T} \cdot \mathbf{P}_A^* \cdot \Theta_A \quad (\text{A.35})$$

$$\hat{\mathbf{B}}^{PLS} = (\mathbf{P}_A^{*T} \cdot \mathbf{S}^{**} \cdot \mathbf{P}_A^*)^{-1} \cdot \mathbf{P}_A^{*T} \cdot \mathbf{P}_A^* \cdot \Theta_A \quad (\text{A.36})$$

Finally, the score τ_A can be used to estimate the unknown part of the new observation.



Variance-covariance maps for process dynamics visualization

Let us assume that a three-way array $\underline{\mathbf{X}}$ ($I \times J \times K$) containing J trajectories measured at K different sampling points are collected in I batches. After applying Trajectory C&S preprocessing, the cross- and autocorrelations of order l can be estimated via a multivariate auto-regressive model which attempts to predict the $J \times 1$ measurement vector \mathbf{x}_k at the k -th sampling point as a linear combination of LMVs as follows:

$$\mathbf{x}_k = \sum_{l=1}^L \Phi_l \mathbf{x}_{k-l} + \mathbf{e}_k \quad (\text{B.1})$$

where L represents the maximum number of possible LMVs at the k -th sampling point ($L = K - 1$), Φ is a $J \times 1$ two-way array containing the AR coefficients corresponding to the l -th lag, which captures the auto-covariances and lagged cross-covariances of the variables, and \mathbf{e}_k the residuals of the AR(l) model.

The total variance-covariance matrix of $\underline{\mathbf{X}}$ is estimated as:

$$\Gamma = \frac{\left(\underline{\mathbf{X}}^{(K-1)}\right)^T \cdot \underline{\mathbf{X}}^{(K-1)}}{I - 1} \quad (\text{B.2})$$

where $\underline{\mathbf{X}}^{(K-1)}$ represents the $JK \times JK$ batch-wise unfolded two way array of $\underline{\mathbf{X}}$.

The dynamic and instantaneous-dynamic partial variance-covariance matrices are computed by calculating the autocorrelation between the k -th sample and all possible the lagged samples with the effects of other lagged samples removed. The difference among both maps lies on the instantaneous correlations are captured in the latter but not in the former. The way of computing this partial correlation are similar to the procedure to estimating partial correlations in AR models. In the following, the algorithms for their computation are described.

Algorithm B.1: Estimation of the $KJ \times KJ$ dynamic partial variance-covariance matrix Γ

Data: $\underline{\mathbf{X}} = \{\mathbf{X}_1, \dots, \mathbf{X}_K\}$
 Initialize $\underline{\mathbf{X}}^b = \underline{\mathbf{X}}$
 Initialize $\underline{\mathbf{X}}^f = \underline{\mathbf{X}}$
 Initialize $\Gamma = \frac{1}{I-1} \left(\underline{\mathbf{X}}^{(K-l)} \right)^T \underline{\mathbf{X}}^{(K-l)}$
for $l = 1$ **to** L (for each LMV) **do**
 for $k = K$ **to** $l + 1$ (for each sampling point) **do**
 $\Phi_{k-l,k} = \left[\left(\mathbf{X}_{k-l}^f \right)^T \mathbf{X}_{k-l}^f \right]^{-1} \left(\mathbf{X}_{k-l}^f \right)^T \mathbf{X}_k^b$
 $\hat{\mathbf{X}}_k^b = \mathbf{X}_{k-l}^f \cdot \Phi_{k-l,k}$
 $\Gamma_{k-l,k} = \frac{1}{I-1} \left(\mathbf{X}_{k-l}^f \right)^T \mathbf{X}_k^b$
 end
 for $k = 1$ **to** $K - l$ (for each sampling point) **do**
 $\mathbf{X}_k^f = \mathbf{X}_k^f - \mathbf{X}_{k+l}^b \left[\left(\mathbf{X}_{k+l}^b \right)^T \mathbf{X}_{k+l}^b \right]^{-1} \left(\mathbf{X}_{k+l}^b \right)^T \mathbf{X}_k^f$
 end
 $\underline{\mathbf{X}}_{l+1:K}^b = \underline{\mathbf{X}}_{l+1:K}^b - \hat{\underline{\mathbf{X}}}_{l+1:K}^b$
end
return ($JK \times KJ$) dynamic partial variance-covariance matrix Γ

Algorithm B.2: Estimation of the $KJ \times KJ$ dynamic partial variance-covariance matrix $\mathbf{\Gamma}$ taking into account the instantaneous relationships across dynamics.

Data: $\underline{\mathbf{X}} = \{\mathbf{X}_1, \dots, \mathbf{X}_K\}$

Initialize $\underline{\mathbf{X}}^b = \underline{\mathbf{X}}$

Initialize $\underline{\mathbf{X}}^f = \underline{\mathbf{X}}$

Initialize $\mathbf{\Gamma} = \frac{1}{T-1} \left(\underline{\mathbf{X}}^{(K-1)} \right)^T \underline{\mathbf{X}}^{(K-1)}$

for $k = 1$ **to** K (for each time point) **do**

for $j = 1$ **to** J (for each variable) **do**

$$\mathbf{\Phi}_{k,k}^j = \left[(\mathbf{X}_{k,\neq j})^T \mathbf{X}_{k,\neq j} \right]^{-1} (\mathbf{X}_{k,\neq j})^T \mathbf{x}_{k,j}$$

$$\mathbf{x}_{k,j}^0 = \mathbf{x}_{k,j} - \mathbf{X}_{k,\neq j} \mathbf{\Phi}_{k,k}^j$$

end

end

for $l = 1$ **to** L (for each LMV) **do**

for $k = 1$ **to** $K - l$ (for each time point) **do**

for $j = 1$ **to** J (for each variable) **do**

$$\mathbf{\Phi}_{k+l,k}^j = \left[(\mathbf{X}_{k+l,\neq j}^b)^T \mathbf{X}_{k+l,\neq j}^b \right]^{-1} (\mathbf{X}_{k+l,\neq j}^b)^T \mathbf{x}_{k,j}^f$$

$$\mathbf{x}_{k,j}^{0f} = \mathbf{x}_{k,j}^f - \mathbf{X}_{k+l,\neq j}^b \mathbf{\Phi}_{k+l,k}^j$$

end

$$\hat{\mathbf{X}}_k^f = \mathbf{X}_{k+l}^b \left[(\mathbf{X}_{k+l}^b)^T \mathbf{X}_{k+l}^b \right]^{-1} (\mathbf{X}_{k+l}^b)^T \mathbf{X}_k^f$$

end

for $k = K$ **to** $l + 1$ (for each time point) **do**

$$\mathbf{\Gamma}_{k-l,k} = \frac{1}{T-1} \left(\mathbf{X}_{k-l}^{0f} \right)^T \mathbf{X}_k^0$$

$$\mathbf{\Phi}_{k-l,k} = \left[(\mathbf{X}_{k-l}^f)^T \mathbf{X}_{k-l}^f \right]^{-1} (\mathbf{X}_{k-l}^f)^T \mathbf{X}_k^b$$

$$\mathbf{X}_k^b = \mathbf{X}_k^b - \mathbf{X}_{k-l}^f \cdot \mathbf{\Phi}_{k-l,k}$$

for $j = 1$ **to** J (for each variable) **do**

$$\hat{\mathbf{x}}_{k,j}^0 = \mathbf{x}_{k,j}^{0f} \left[(\mathbf{x}_{k,j}^{0f})^T \mathbf{x}_{k,j}^{0f} \right]^{-1} (\mathbf{x}_{k,j}^{0f})^T \mathbf{x}_{k,j}^0$$

$$\mathbf{x}_{k,j}^0 = \mathbf{x}_{k,j}^0 - \hat{\mathbf{x}}_{k,j}^0$$

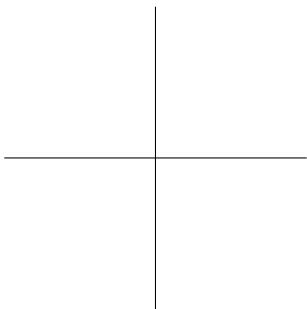
end

end

$$\mathbf{X}_{1:K-l}^f = \mathbf{X}_{1:K-l}^f - \hat{\mathbf{X}}_{1:K-l}^f$$

end

return ($JK \times KJ$) instantaneous-dynamic partial variance-covariance matrix $\mathbf{\Gamma}$.



Monitoring of multiple asynchronous batches

In this appendix, the OWU scores and DModX control charts for the NOC and faulty (type I, II, and III) test batches affected by five different types of asynchronism (cases 1, 2, 3, 4, and 5) are provided¹. For this purpose, the OWU phase of the observationwise unfolding-T scores batchwise unfolding (OWU-TBWU) approach is used. Note that these results come from the comparative study performed between Multisynchro approach and the TLEC method in Chapter 8. The main goal of this study is to find out to what extent the accuracy of detection of faults from the monitoring scheme based on the aforementioned modeling approach is affected by scenarios of multiple asynchronisms and the synchronization technique applied.

The structure of this appendix is divided into four sections. Section C.1 shows the results of the monitoring of NOC batches affected by the five cases of asynchronism. Section C.2, Section C.3, and Section C.4 show the results of the monitoring of batches affected by the five cases of asynchronism and abnormalities of type I, II and III, respectively.

¹All the figures that appear in Appendix C are reprinted with permission from "Effect of Synchronization on Bilinear Batch Process Modeling. J. M. González-Martínez, R. Vitale, O. E. de Noord, and A. Ferrer. *Industrial & Engineering Chemistry Research* 2014, 53 (11), 4339-4351". Copyright 2014 American Chemical Society

C.1 Synchronization and monitoring of NOC batches

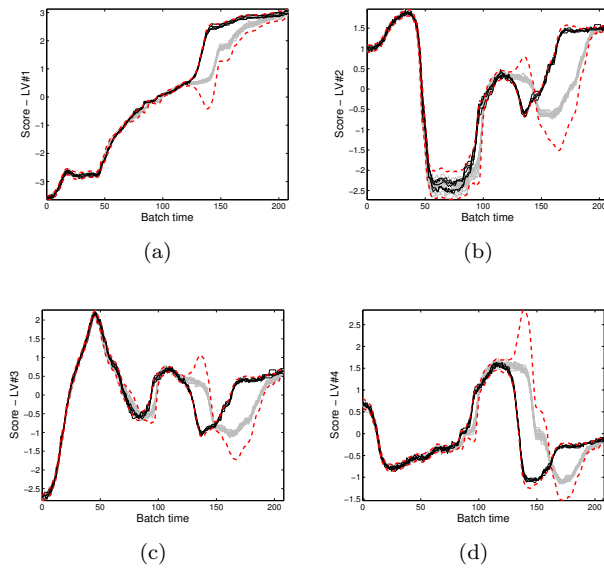


Figure C.1. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class I asynchronism in black lines.

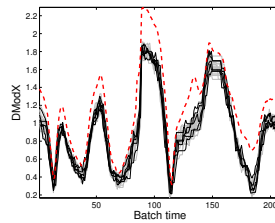


Figure C.2. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class I asynchronism in black lines.

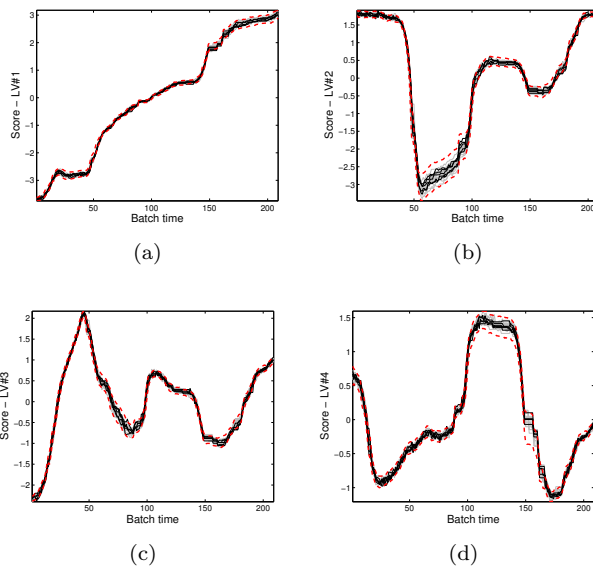


Figure C.3. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class I asynchronism in black lines.

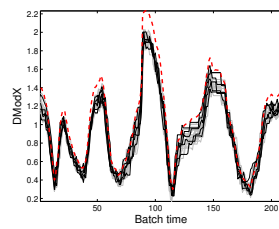


Figure C.4. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class I asynchronism in black lines.

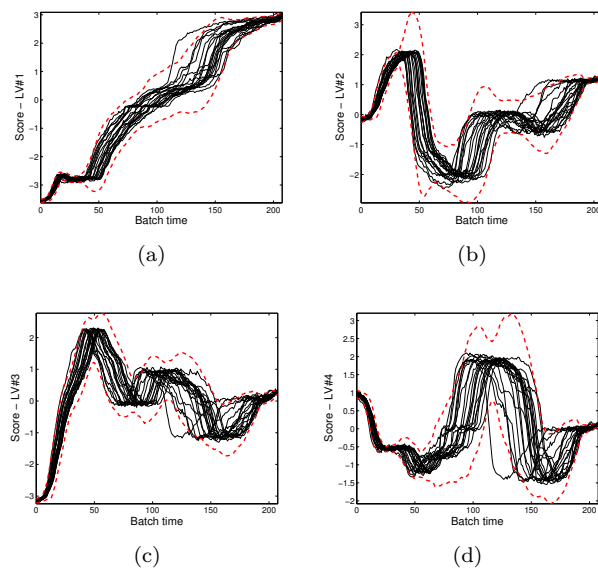


Figure C.5. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

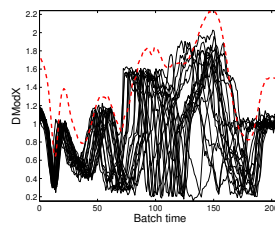


Figure C.6. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

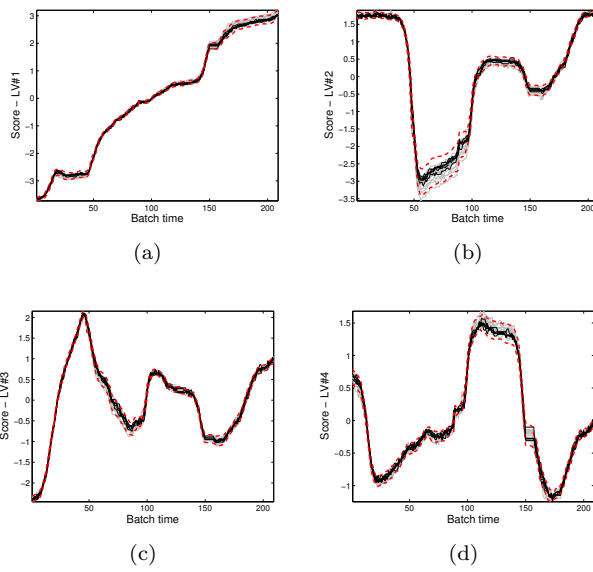


Figure C.7. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

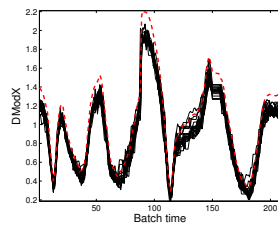


Figure C.8. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

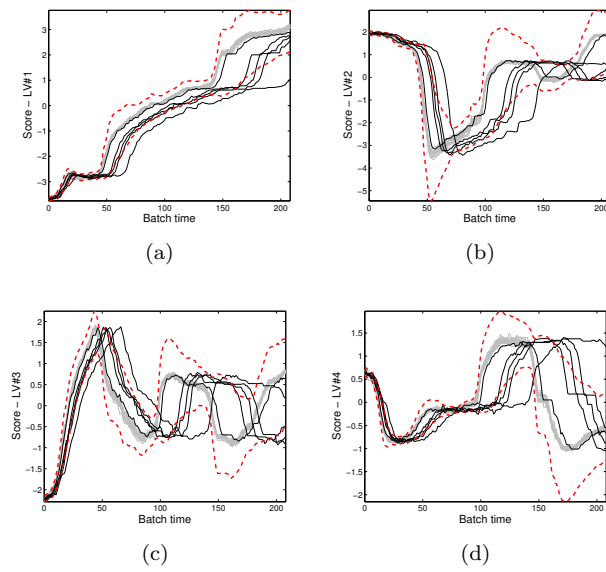


Figure C.9. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class III asynchronism in black lines.

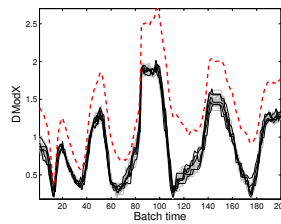


Figure C.10. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class III asynchronism in black lines.

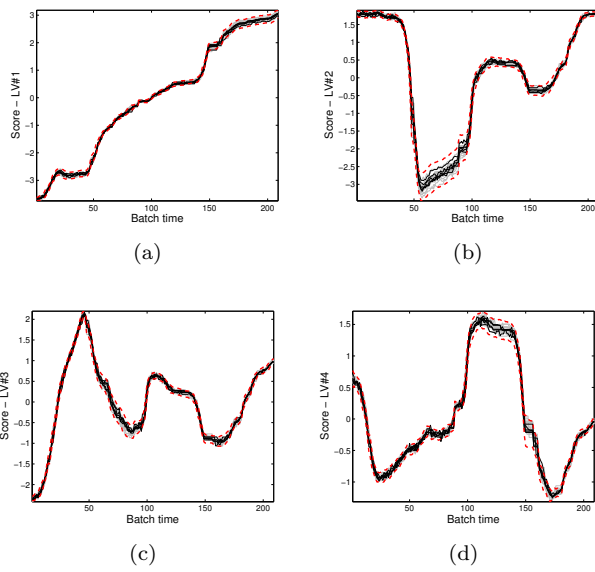


Figure C.11. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class III asynchronism in black lines.

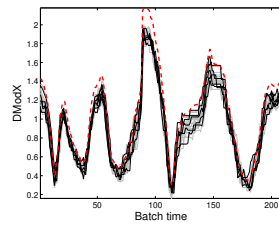


Figure C.12. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class III asynchronism in black lines.

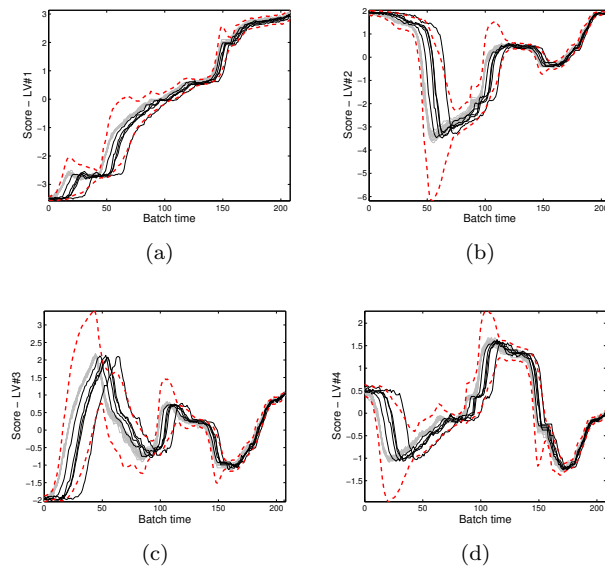


Figure C.13. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class IV asynchronism in black lines.

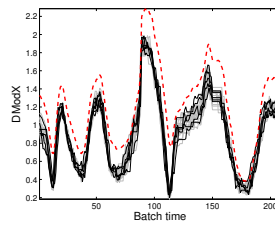


Figure C.14. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class IV asynchronism in black lines.

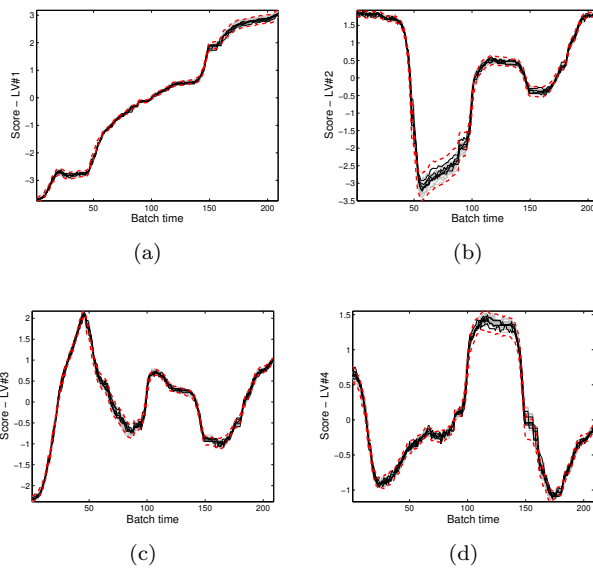


Figure C.15. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class IV asynchronism in black lines.

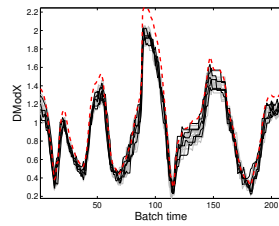


Figure C.16. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. The batch trajectories with different asynchronism patterns are distinguished by black and grey lines. Batches with class IV asynchronism in black lines.

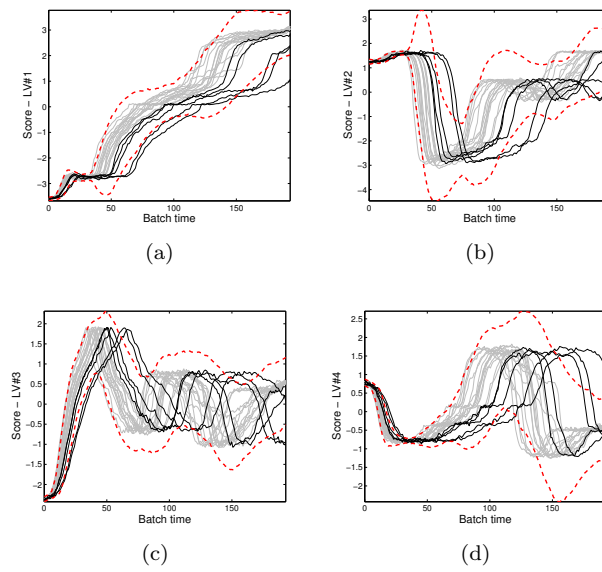


Figure C.17. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

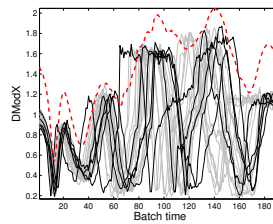


Figure C.18. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

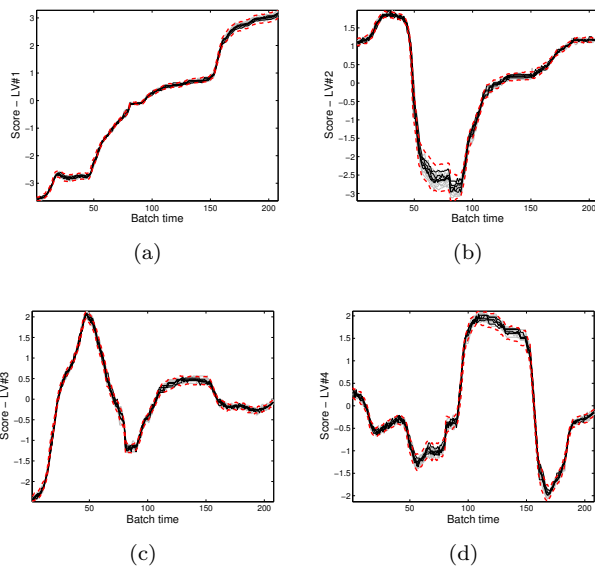


Figure C.19. OWU scores obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

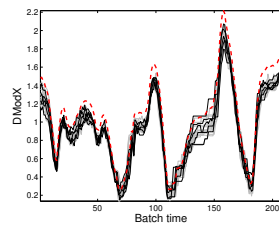


Figure C.20. DModX obtained from the projection onto the PLS latent structure of the NOC test batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

C.2 Synchronization and monitoring of type I faulty batches

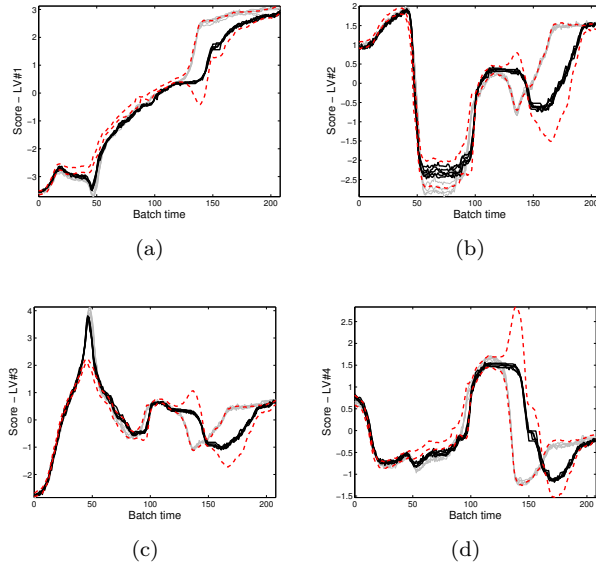


Figure C.21. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

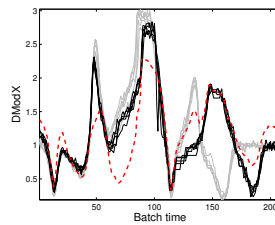


Figure C.22. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

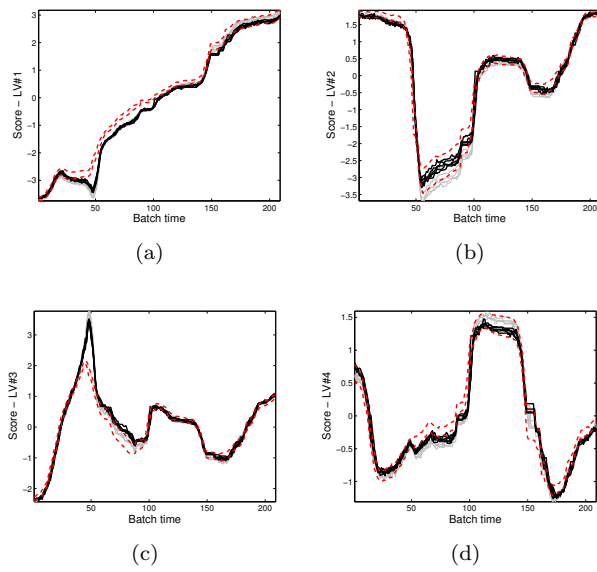


Figure C.23. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

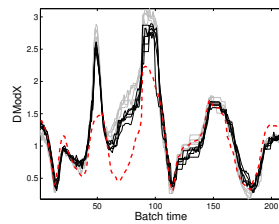


Figure C.24. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

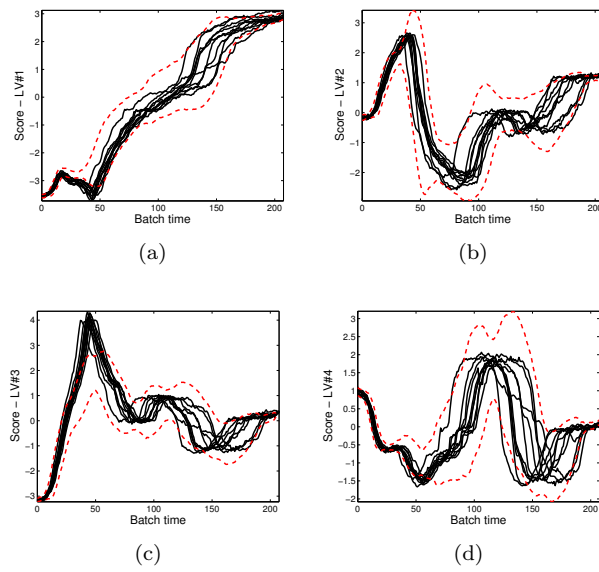


Figure C.25. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

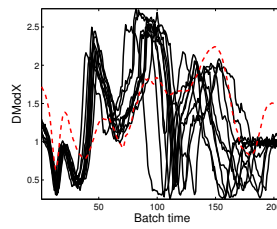


Figure C.26. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

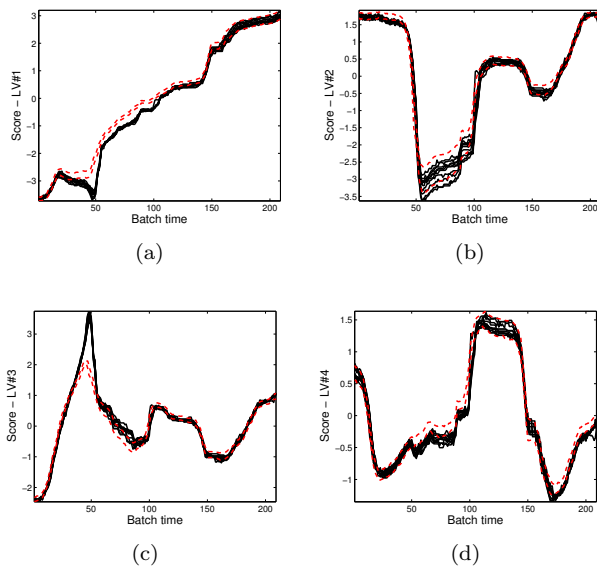


Figure C.27. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

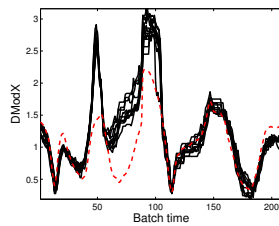


Figure C.28. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

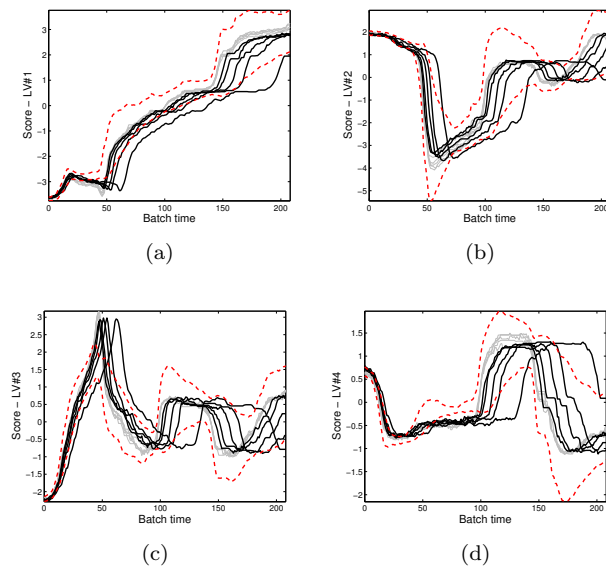


Figure C.29. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

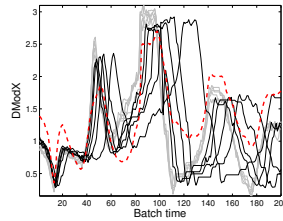


Figure C.30. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

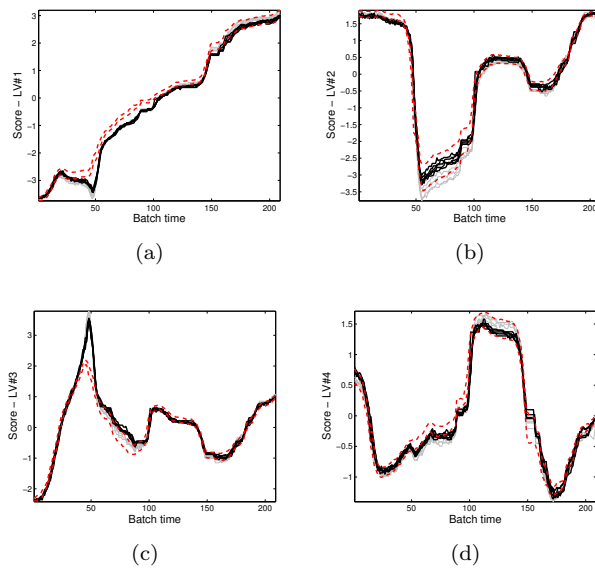


Figure C.31. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

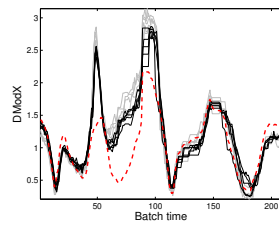


Figure C.32. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

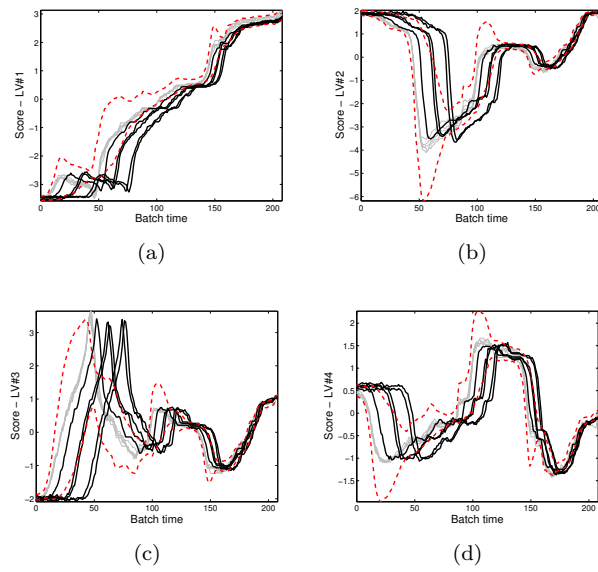


Figure C.33. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

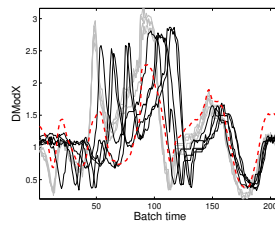


Figure C.34. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

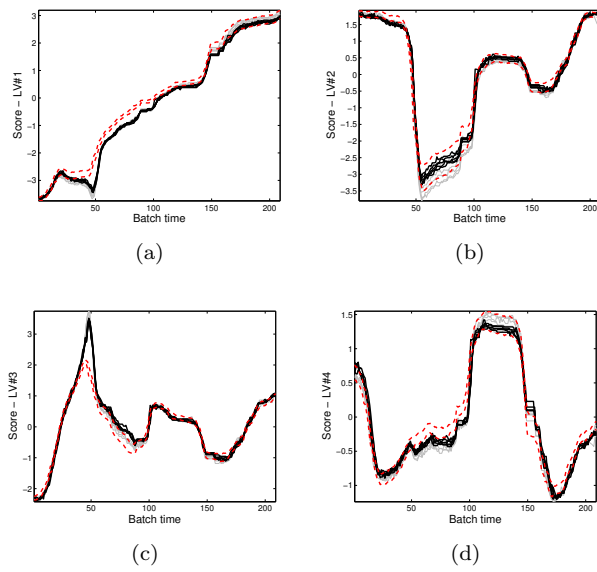


Figure C.35. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

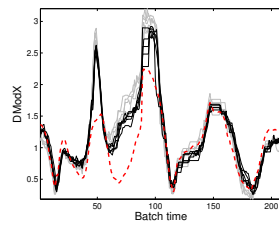


Figure C.36. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

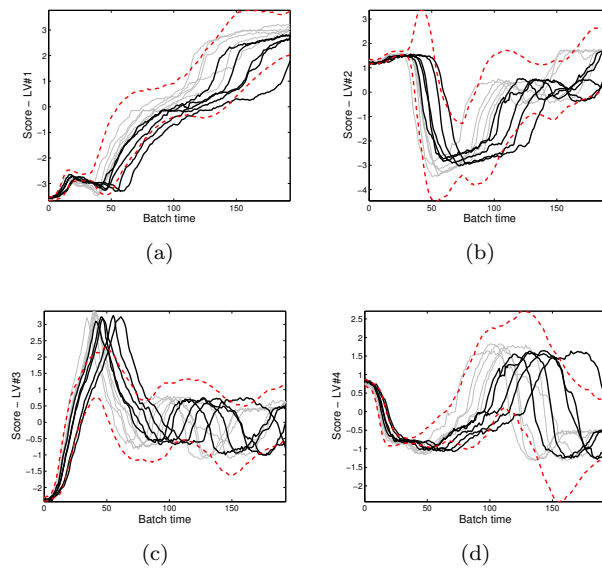


Figure C.37. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and class III asynchronism in black lines.

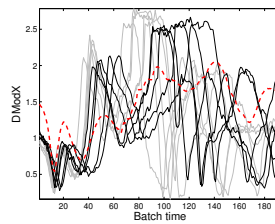


Figure C.38. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

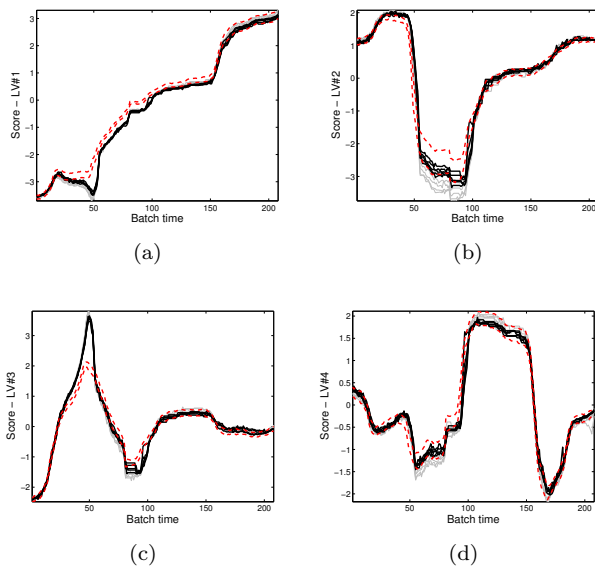


Figure C.39. OWU scores obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

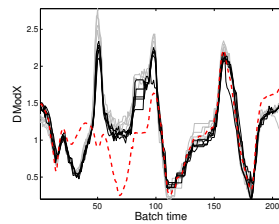


Figure C.40. DModX obtained from the projection onto the PLS latent structure of the test type I-faulty batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

C.3 Synchronization and monitoring of type II faulty batches

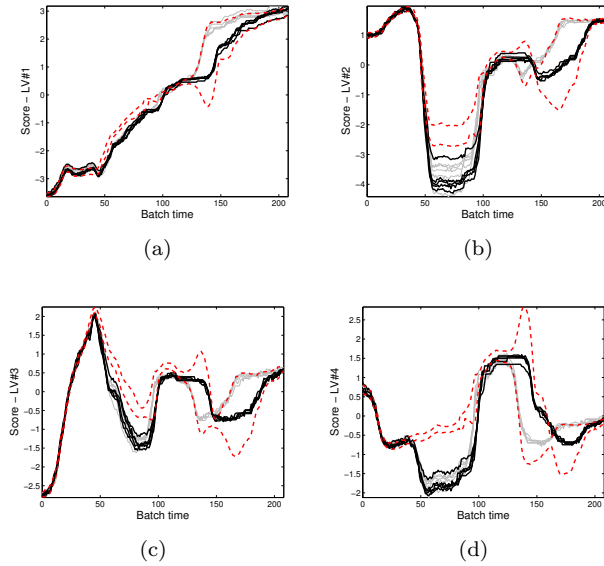


Figure C.41. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

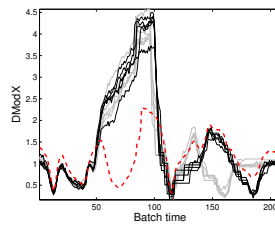


Figure C.42. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

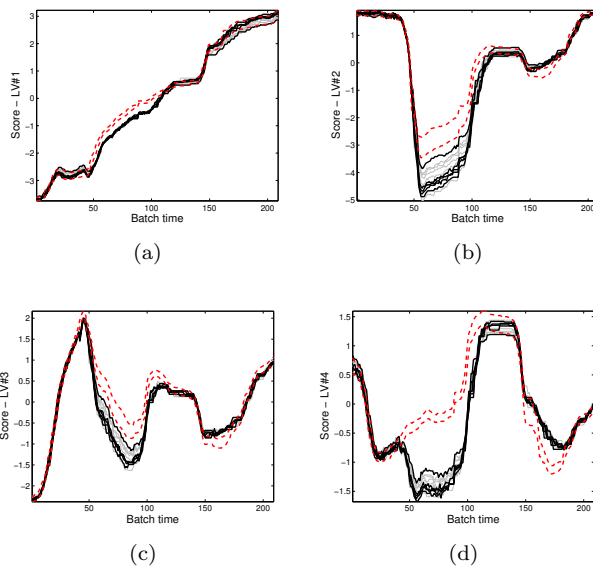


Figure C.43. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

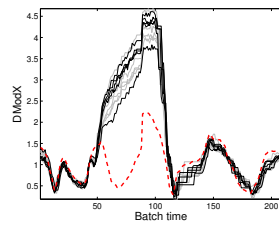


Figure C.44. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

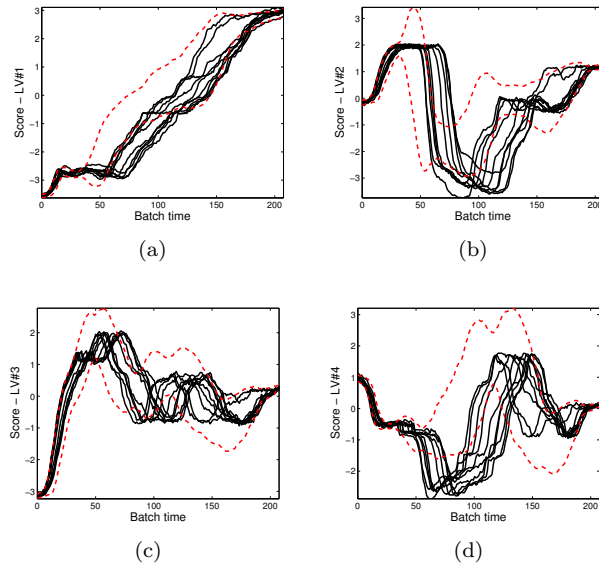


Figure C.45. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

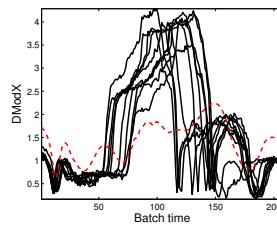


Figure C.46. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

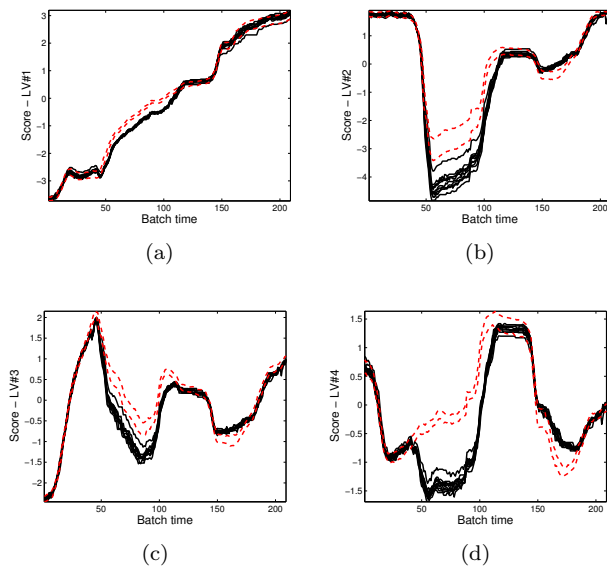


Figure C.47. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

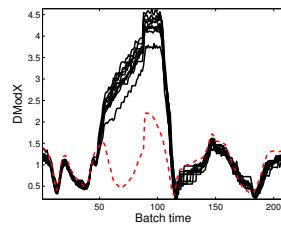


Figure C.48. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

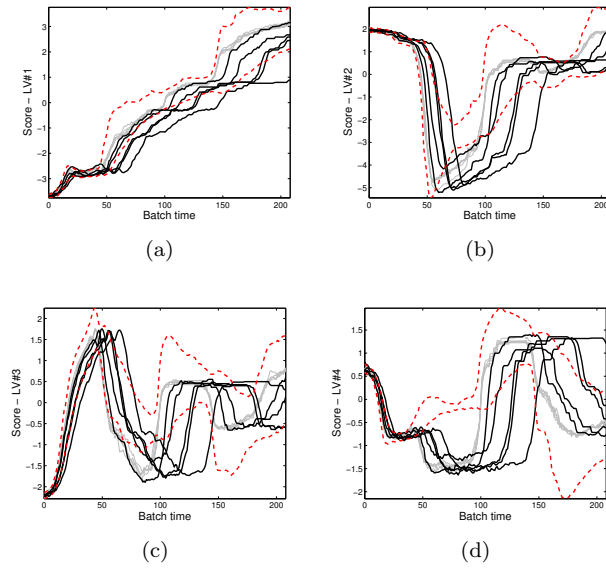


Figure C.49. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

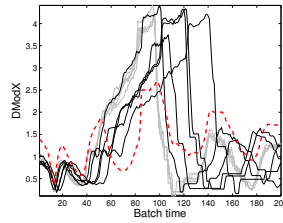


Figure C.50. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

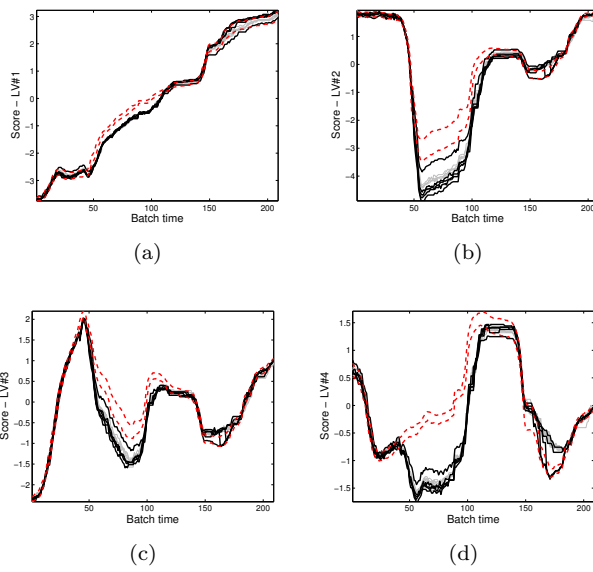


Figure C.51. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

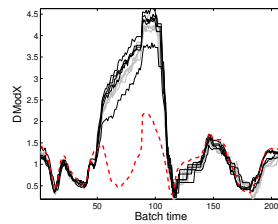


Figure C.52. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

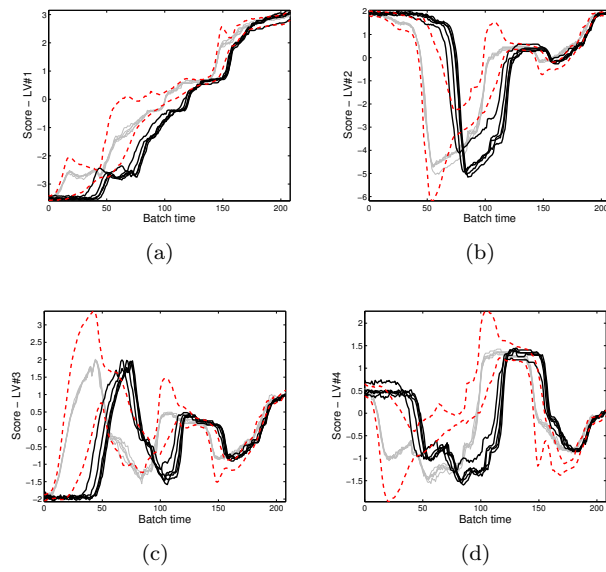


Figure C.53. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

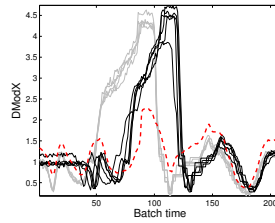


Figure C.54. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

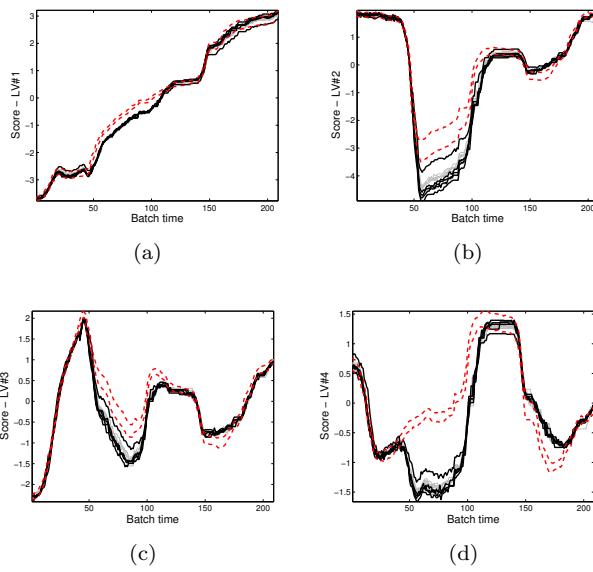


Figure C.55. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

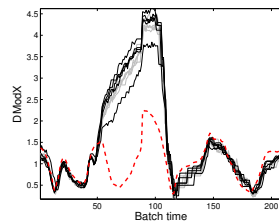


Figure C.56. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

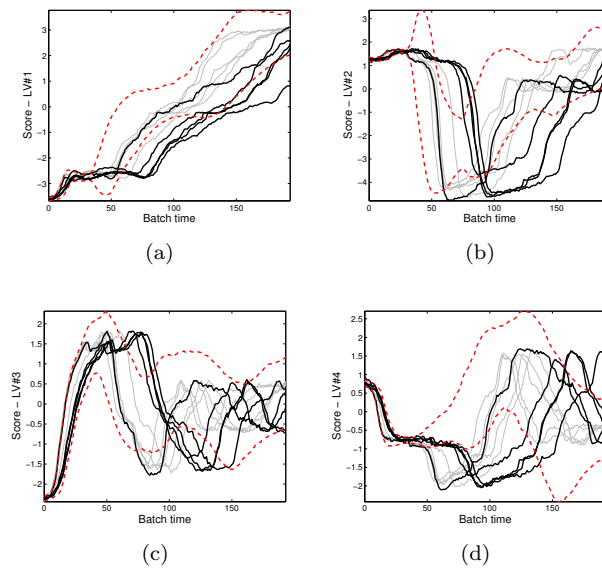


Figure C.57. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

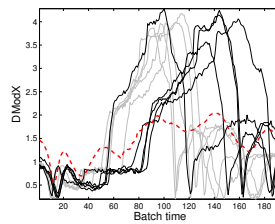


Figure C.58. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

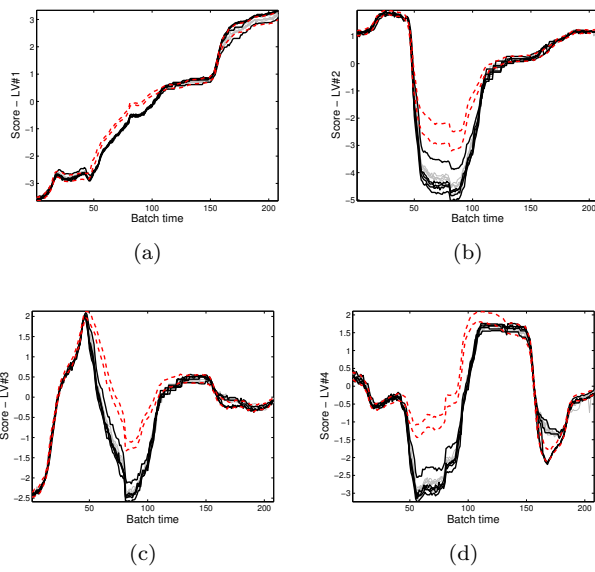


Figure C.59. OWU scores obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

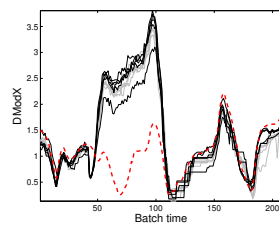


Figure C.60. DModX obtained from the projection onto the PLS latent structure of the test type II-faulty batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

C.4 Synchronization and monitoring of type III faulty batches

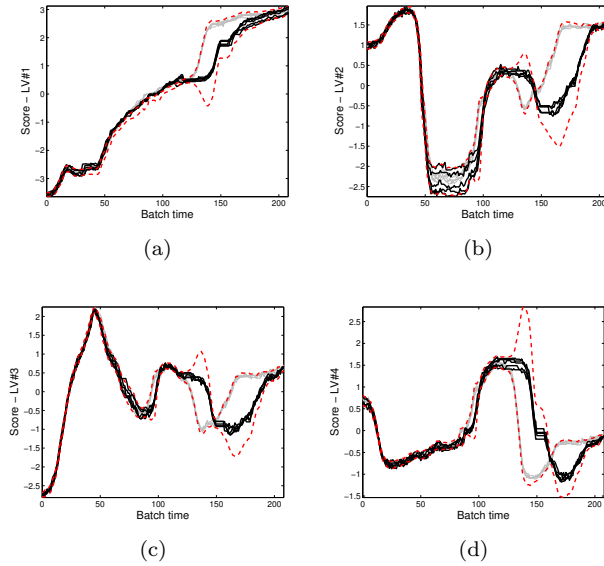


Figure C.61. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

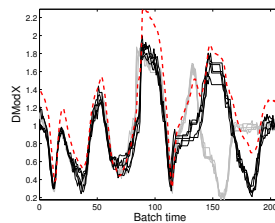


Figure C.62. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class I asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

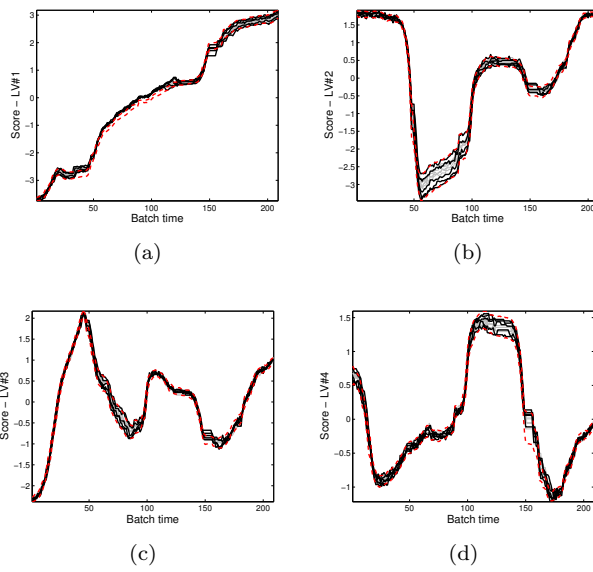


Figure C.63. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

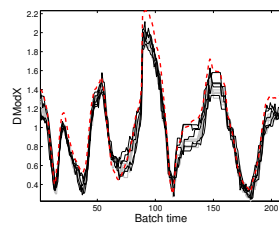


Figure C.64. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class I asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class I asynchronism in black lines.

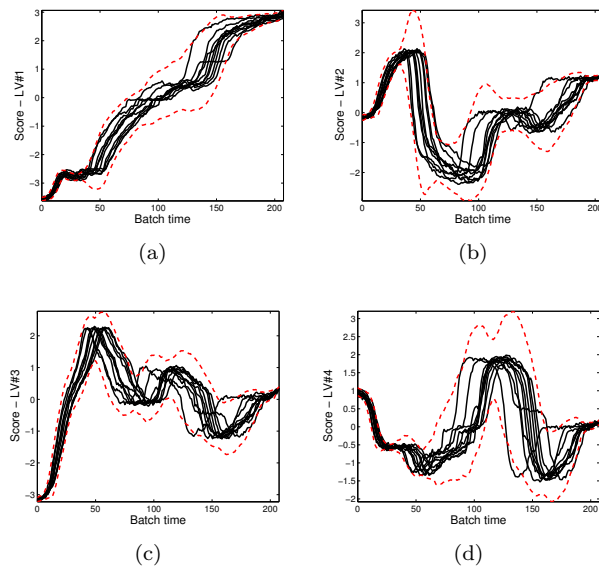


Figure C.65. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

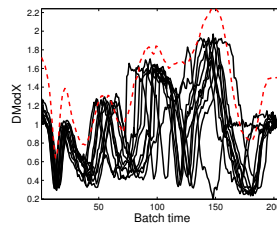


Figure C.66. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

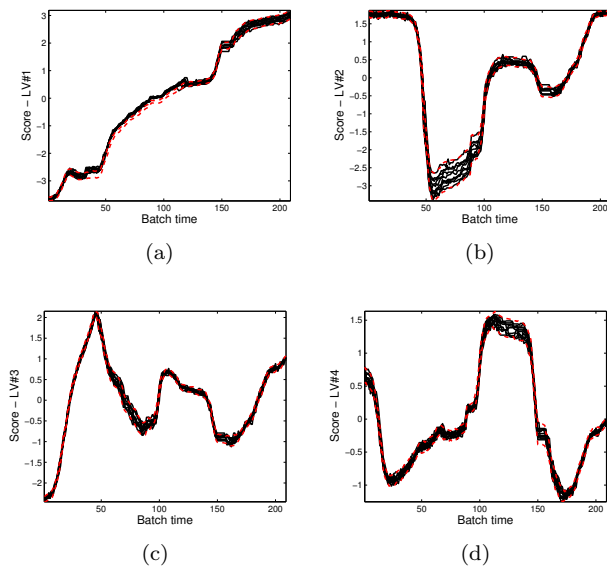


Figure C.67. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

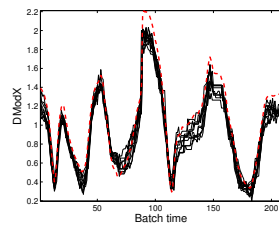


Figure C.68. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Note that only black lines are depicted because all batches are affected by class II asynchronism.

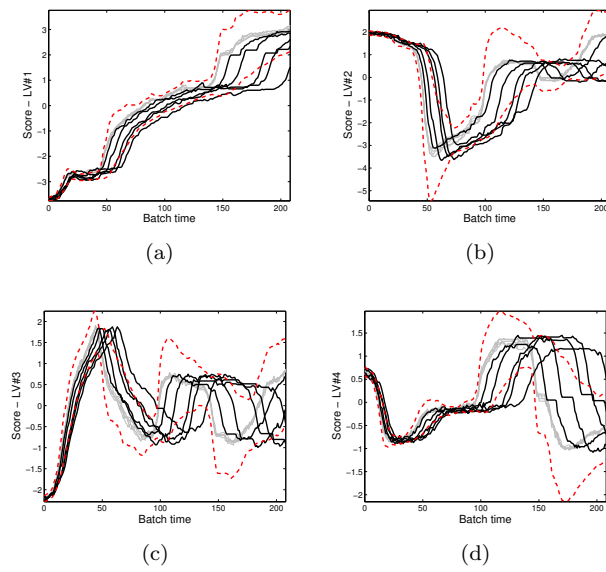


Figure C.69. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

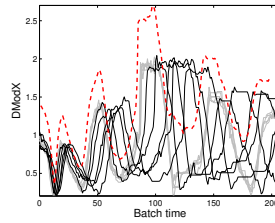


Figure C.70. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class III asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

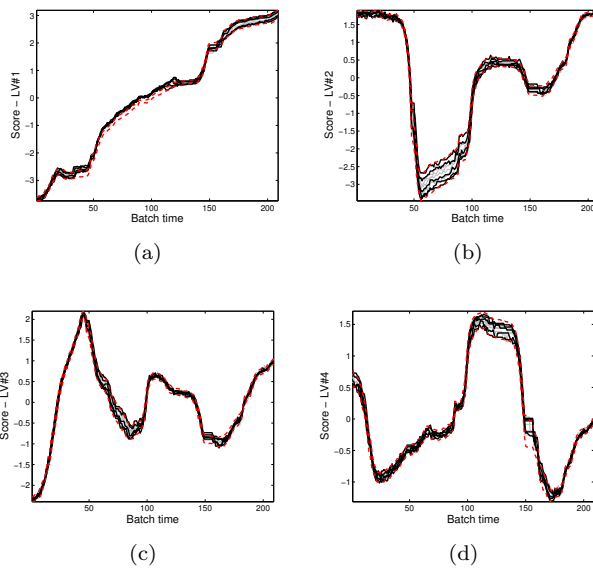


Figure C.71. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

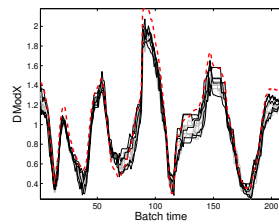


Figure C.72. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class III asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class III asynchronism in black lines.

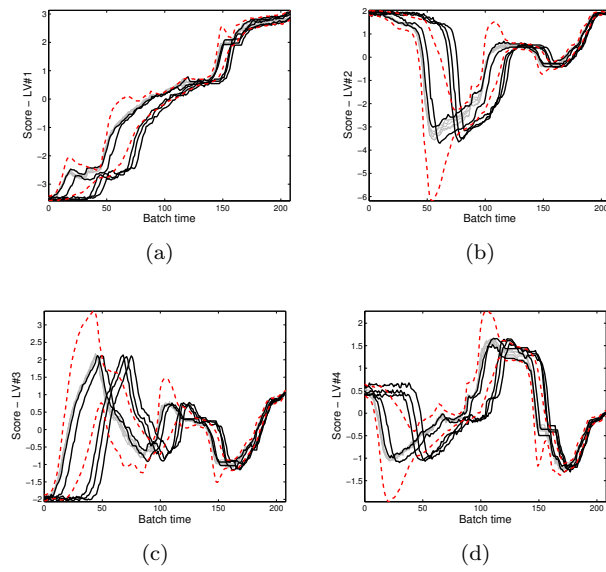


Figure C.73. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

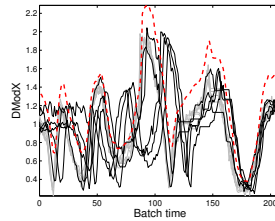


Figure C.74. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class IV asynchronism and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

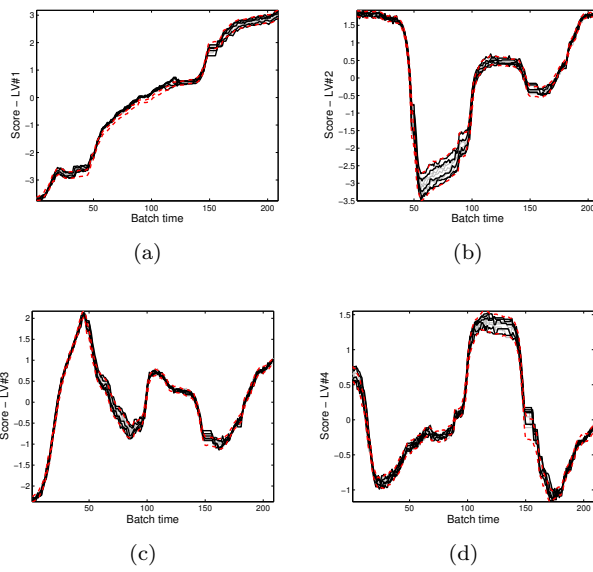


Figure C.75. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

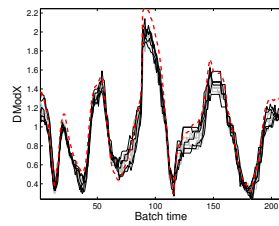


Figure C.76. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class IV asynchronism and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class IV asynchronism in black lines.

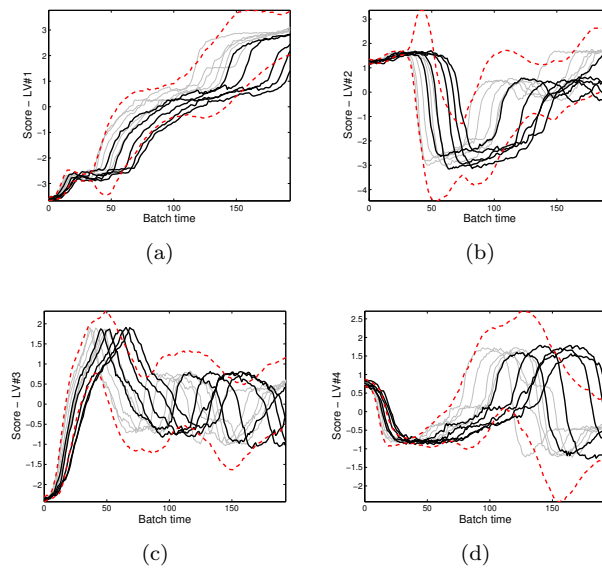


Figure C.77. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

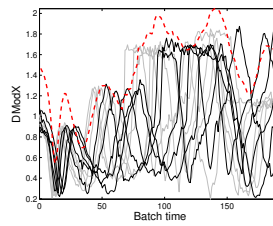


Figure C.78. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II and class III asynchronisms and synchronized by TLEC. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

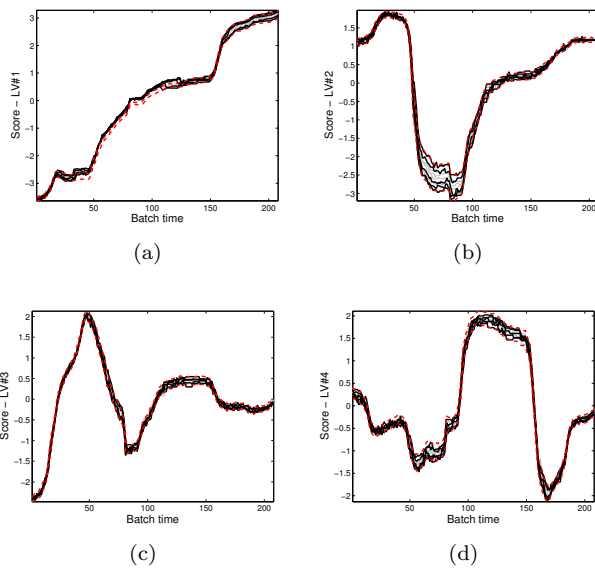


Figure C.79. OWU scores obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.

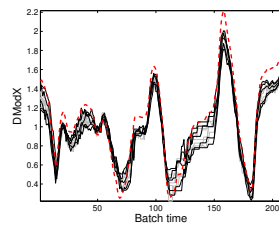
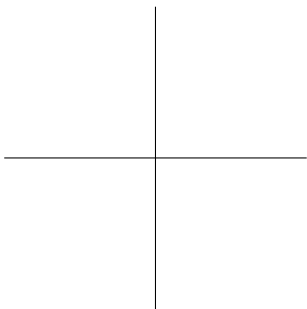


Figure C.80. DModX obtained from the projection onto the PLS latent structure of the test type III-faulty batches affected by class II and class III asynchronisms and synchronized by applying the Multisynchro approach. Control limits at 95% confidence level in dashed red lines. Batches with class II and III asynchronism in black lines.



References

- [1] J.M. González-Martínez, A. Ferrer, and 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*, 105:195–206, 2011.
- [2] J.M. González-Martínez, J.A. Westerhuis, and A. Ferrer. Using warping information for batch process monitoring and fault classification. *Chemometrics and Intelligent Laboratory Systems*, 127:210–217, 2013.
- [3] J.M. González-Martínez, J. Camacho, and A. Ferrer. Bilinear modeling of batch processes. Part III: parameter stability. *Journal of Chemometrics*, 28:10–27, 2014.
- [4] J.M. González-Martínez, O.E. de Noord, and A. Ferrer. Multisynchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms. *Journal of Chemometrics*, 28:462–475, 2014.
- [5] J.M. González-Martínez, R. Vitale, O.E. de Noord, and A. Ferrer. Effect of synchronization on bilinear batch process modeling. *Industrial & Engineering Chemistry Research*, 53:4339–4351, 2014.
- [6] J.M. González-Martínez, J. Camacho, and A. Ferrer. MV-Batch Toolbox: a MATLAB graphical interface for bilinear batch process modeling. *In elaboration*.
- [7] J.M. González-Martínez, J. Camacho, and A. Ferrer. A comparison of synchronization methods based on the correlation structure of batch data. *In elaboration*.
- [8] J. Camacho, J.M. González-Martínez, and A. Ferrer. Equalization of batch data: challenges and solutions. *In elaboration*.

- [9] J.M. González-Martínez, J. Camacho, and A. Ferrer. Modeling time-varying process dynamics through latent models and control charts. *In elaboration*.
- [10] J. Camacho, J.M. González-Martínez, and A. Ferrer. Chapter 1: Introduction to batch processing. batch processes: Monitoring and process understanding. *Wiley-VCH Verlag GmbH*, due in 2016.
- [11] J. Camacho, J.M. González-Martínez, and A. Ferrer. Chapter 2: Latent structures based models. batch processes: Monitoring and process understanding. *Wiley-VCH Verlag GmbH*, due in 2016.
- [12] J. Camacho, J.M. González-Martínez, and A. Ferrer. Chapter 3: Batch process data. batch processes: Monitoring and process understanding. *Wiley-VCH Verlag GmbH*, due in 2016.
- [13] J. Camacho, J.M. González-Martínez, and A. Ferrer. Chapter 4: Bilinear modeling of batch process data. batch processes: Monitoring and process understanding. *Wiley-VCH Verlag GmbH*, due in 2016.
- [14] J. Camacho, J.M. González-Martínez, and A. Ferrer. Chapter 5: Batch process analysis and understanding. batch processes: Monitoring and process understanding. *Wiley-VCH Verlag GmbH*, due in 2016.
- [15] J. Camacho, J.M. González-Martínez, and A. Ferrer. Chapter 6: On-line batch process monitoring. batch processes: Monitoring and process understanding. *Wiley-VCH Verlag GmbH*, due in 2016.
- [16] J.M. González-Martínez and A. Ferrer. A comparison of different methods for synchronization of batch trajectories. In *Proceedings of 11th Scandinavian Symposium on Chemometrics*, page 83, Løen (Norway), 2009.
- [17] A. Ferrer, J. Camacho, and J.M. González-Martínez. Issues on batch multivariate statistical process control. In *Proceedings of Eastern Analytical Symposium & Exposition*, page 56, New Jersey (USA), 2010.
- [18] J.M. González-Martínez, A. Ferrer, and J.A. Westerhuis. Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping. In *Proceedings of the 12th Conference on*

- Chemometrics in Analytical Chemistry*, pages 229–230, Antwerpen (Belgium), 2010.
- [19] J.M. González-Martínez, A. Ferrer, F. Arteaga, D. Aguado, and J. Ribes. Error-proof latent-based multivariate process control of a continuous biological nutrient removal process. In *Proceedings of the 12th Conference on Chemometrics in Analytical Chemistry*, pages 149–150, Antwerpen (Belgium), 2010.
- [20] J.M. González-Martínez, A. Ferrer, F. Llaneras, M. Tortajada, and J. Picó. Metabolic flux understanding of *Pichia Pastoris* grown on heterogenous culture media. In *Proceedings of the 12th Conference on Chemometrics in Analytical Chemistry*, page 86, Antwerpen (Belgium), 2010.
- [21] J.M. González-Martínez, A. Ferrer, F. Llaneras, M. Tortajada, and J. Picó. Metabolic flux understanding: a grey modeling approach. In *Proceedings of the Workshop on Chemometrics for Young Researchers*, pages 47–48, A Coruña (Spain), 2011.
- [22] J.M. González-Martínez and A. Ferrer. Batch synchronization: a paramount step before bilinear modeling in batch multivariate statistical process control. In *Proceedings of the Workshop on Chemometrics for Young Researchers*, pages 29–30, A Coruña (Spain), 2011.
- [23] A. Ferrer, J. Camacho, and J.M. González-Martínez. Chemometrics tools for process understanding and monitoring with bilinear models: a review of novel proposals. In *Proceedings of the 5th International Chemometrics Research Meeting*, page 11, Nijmegen (The Netherlands), 2011.
- [24] J.M. González-Martínez, A. Ferrer, and J.A. Westerhuis. Relaxed-greedy time warping: a new tool for real-time synchronization of batch trajectories for batch MSPC. In *Proceedings of the 5th International Chemometrics Research Meeting*, page 32, Nijmegen (The Netherlands), 2011.
- [25] J.M. González-Martínez, A. Ferrer, F. Arteaga, D. Aguado, and J. Ribes. Multivariate statistical process control of a continuous biological removal process: Designing efficient monitoring schemes robust to sensor malfunctioning. In *Proceedings of the 2nd European Conference on Process*

- Analytics and Control technology (EUROPACT)*, page 149, Glasgow (UK), 2011.
- [26] O.E. de Noord and J.M. González-Martínez. Recent developments in multivariate data analysis and monitoring of chemical manufacturing processes. In *Proceedings of the 2nd African-European Conference on Chemometrics (AFRODATA 2012)*, Stellenbosch (South Africa), 2012.
- [27] J.M. González-Martínez, J. Camacho, and A. Ferrer. Enhancement of batch process understanding and monitoring: a matter of parameters stability. In *Proceedings of the 13th Conference on Chemometrics in Analytical Chemistry*, page 39, Budapest (Hungary), 2012.
- [28] J.M. González-Martínez, O.E. de Noord, and A. Ferrer. A novel approach for batch synchronization in scenarios of multiple asynchronies. In *Proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 28, Djurönäset (Sweden), 2013.
- [29] A. Ferrer, J.M. González-Martínez, and J. Camacho. Practical implications of synchronization, preprocessing and bilinear modelling of batch processes for MSPC. In *Proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 26, Djurönäset (Sweden), 2013.
- [30] J.M. González-Martínez, R. Vitale, O.E. de Noord, and A. Ferrer. Does synchronization matter in batch multivariate statistical process control? In *Proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 58, Djurönäset (Sweden), 2013.
- [31] J.M. González-Martínez, J. Camacho, O.E. de Noord, and A. Ferrer. Equalization and data-driven compression as a prior step to batch modelling. In *Proceedings of the 13th Scandinavian Symposium on Chemometrics (SSC13)*, page 59, Djurönäset (Sweden), 2013.
- [32] J. Camacho, J.M. González-Martínez, and A. Ferrer. MVStat toolbox for matlab. <https://mseg.webs.upv.es/Software.html>, 2014.
- [33] J.M. González-Martínez and O.E. de Noord. Enhanced process understanding and monitoring of manufacturing batch processes of specialty chemicals, 2013.
- [34] J.M. González-Martínez, A. Folch-Fortuny, F. Llaneras, M. Tortajada, J. Picó, and A. Ferrer. Metabolic flux

- understanding of pichia pastoris grown on heterogeneous culture media. *Chemometrics and Intelligent Laboratory Systems*, 134:89–99, 2014.
- [35] D. Bonne and S.B. Jorgensen. Data-driven modeling of batch processes. In *Proc. of 7th International Symposium on Advanced Control of Chemical Processes, ADCHEM*. In: *Proc. of 7th International Symposium on Advanced Control of Chemical Processes, ADCHEM*, 2004.
- [36] Z.K. Nagy and R.D. Braatz. Nonlinear model predictive control for batch processes. In W.S. Levine, editor, *The Control Handbook*, volume Control System Advance Methods, chapter 15, pages 1–23. CRC Press, 2011.
- [37] Food and Drug Administration. Guidance for industry: PAT - a framework for innovative pharmaceutical development, manufacturing, and quality assurance, 2004. Accessed 31 May 2011.
- [38] K.J. Åström. *Introduction to Stochastic Control Theory*. Dover, New York, 2006. Reprint. Originally published by Academic Press 1970.
- [39] S.P. Gurden, J.A. Westerhuis, S. Bijlsma, and A.K. Smilde. Modelling of spectroscopy batch process data using grey models to incorporate external information. *Journal of chemometrics*, 15:101–121, 2001.
- [40] D. Bonvin. Optimal operation of batch reactors-a personal view. *Journal of Process Control*, 8:355–368, 1998.
- [41] P.M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge based redundancy. a survey and some new results. *Automatica*, 26:459–474, 1990.
- [42] K.B. Konstantinov and T. Yoshida. A method for on-line reasoning about the time-profiles of process variables. In *Procs. IFAC Symp. On-Line Fault Detection and Supervision in the Chemical Process Industries*, April 1992.
- [43] L.E. Holloway and B.H. Krogh. On-line trajectory encoding for discrete -observation process monitoring. In *Procs. IFAC Symp. On-Line Fault Detection and Supervision in the Chemical Process Industries*, April 1992.
- [44] R. Steuer and B.H. Junker. *Computational Models of Metabolism: Stability and Regulation in Metabolic Networks*. Wiley: New York, 2008.

- [45] T. Kourti. Process analysis and abnormal situation detection: From theory to practice. *IEEE Control Systems Magazine*, 22(5):10–25, 2003.
- [46] T. Kourti and J.F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.
- [47] 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*, 19:311–325, 2007.
- [48] M.D. Mesarovic, S.N. Sreenath, and J.D. Keene. Search for organising principles: understanding in systems biology. *Systems Biology, IEE*, 1(1):19–27, 2004.
- [49] K.J. Kauffman, P. Prakash, and J.S. Edwards. Advances in flux balance analysis. *Current Opinion in Biotechnology*, 14(5):491 – 496, 2003.
- [50] S. Feyodeazevedo, B. Dahm, and F. Oliveira. Hybrid modelling of biochemical processes: A comparison with the conventional approach. *Computers & Chemical Engineering*, 21:S751–S756, 1997.
- [51] Y. Takane and T. Shibayama. Principal component analysis with external information on both subjects and variables. *Psychometrika*, 56:97–120, 1991.
- [52] Y. Takane, H.A.L. Kiers, and J. de Leeuw. Component analysis with different sets of constraints on different dimensions. *Psychometrika*, 60:259–280, 1995.
- [53] H. Ramaker, E.N.M. van Sprang, S.P. Gurden, J.A. Westerhuis, and A.K. Smilde. Improved monitoring of batch processes by incorporating external information. *Journal of Process Control*, 12:569–576, 2002.
- [54] E.N.M. van Sprang, H.J. Ramaker, J.A. Westerhuis, A.K. Smilde, and D. Wienke. Statistical batch process monitoring using gray models. *AIChE Journal*, 51(3):931–945, 2005.
- [55] J.A. Westerhuis, E.P.P.A Derks, H.C.J Hoefsloot, and A.K. Smilde. Grey component analysis. *Journal of Chemometrics*, 21:474 – 485, 2007.

- [56] C.W. Ng and M.A. Hussain. Hybrid neural network - prior knowledge model in temperature control of a semi-batch polymerization process. *Chemical Engineering and Processing: Process Intensification*, 43(4):559–570, 2004.
- [57] B. Sohlberg. Grey box modelling for model predictive control of a heating process. *Journal of Process Control*, 13:225–238, 2003.
- [58] H. Bechmann, H. Madsen, N.K. Poulsen, and M.K. Nielsen. Grey box modeling of first flush and incoming wastewater at a wastewater treatment plant. *Environmetrics*, 11:1–12, 2000.
- [59] R. Bro and S. de Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11:393–401, 1997.
- [60] R. Tauler. Multivariate curve resolution applied to second order data. *Chemometrics and Intelligent Laboratory System*, 30:133–146, 1995.
- [61] J.M.F. Ten Berge and A.K. Smilde. Non-triviality and identification of a constrained tucker3 analysis. *Journal of Chemometrics*, 16:609–612, 2002.
- [62] S. Wold, N. Kettaneh-Wold, J.F. MacGregor, and K.G. Dunn. Batch Process Modeling and MSPC. *Comprehensive Chemometrics - Chemical and Biochemical Data Analysis, Elsevier:Oxford*, 1:163–195, 2009.
- [63] C. Duchesne and J.F. MacGregor. Multivariate analysis and optimization of process variable trajectories for batch processes. *Chemometrics and Intelligent Laboratory Systems*, 51(51):125–137, 2000.
- [64] L. Eriksson, E. Johansson, N. Kettaneh-Wold, J. Trygg, C. Wikström, and S. Wold. *Multi- and Megavariate Data Analysis*. MKS Umetrics AB, 2006.
- [65] UMETRICS. SIMCA 13.0.3. Available at www.umetrics.com, accessed in 2014.
- [66] S. García-Muñoz, T. Kourti, and J.F. MacGregor. Troubleshooting of an industrial batch process using multivariate methods. *Industrial and Engineering Chemistry Research*, 42:3592–3601, 2003.
- [67] M. Zarzo and A. Ferrer. Batch process diagnosis: PLS with variable selection versus block-wise PCR. *Chemometrics and Intelligent Laboratory Systems*, 73:15–27, 2004.

- [68] ProSensus Inc. ProMV Batch Edition Release 13.02. Available at <http://www.prosensus.ca>, accessed in 2014.
- [69] D.J. Louwerse, A.A. Tate, A.K. Smilde, G.L.M. Koot, and 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(2):197 – 206, 1999.
- [70] P. Nomikos and J.F. MacGregor. Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40:1361–1375, 1994.
- [71] C. Undey and A. Çinar. Statistical monitoring of multi-stage, multiphase batch processes. *IEEE Control System Magazine*, 22(5):40–52, 2002.
- [72] N. Kaitsha and C.F. Moore. Extraction of event times in batch profiles for time synchronization and quality predictions. *Industrial & Engineering Chemistry Research*, 40(1):252–260, 2001.
- [73] J.O. Ramsay and B.W. Silverman. *Functional data analysis*. New York:Springer-Verlag, 1997.
- [74] S.W. Andersen and 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(2):327–344, 2012.
- [75] K. Wang and T. Gasser. Synchronizing sample curves nonparametrically. *The Annals of Statistics*, 27(2):439–460, 04 1999.
- [76] A. Kassidas, J.F. MacGregor, and P.A. Taylor. Synchronization of batch trajectories using dynamic time warping. *AIChE Journal*, 44:864–875, 1998.
- [77] Y. Zhang and T.F. Edgar. A robust dynamic time warping algorithm for batch trajectory synchronization. In *Proceedings of American Control Conference*, pages 2864–2869, 2008.
- [78] G. Gins, P. Van den Kerkhof, and J.F.M. Van Impe. Hybrid derivative dynamic time warping for online industrial batch-end quality estimation. *Industrial & Engineering Chemistry Research*, 51(17):6071–6084, 2012.
- [79] T. Kourti. Abnormal situation detection, three-way data and projection methods; robust data archiving and model-

- ing for industrial applications. *Annual Reviews in control*, 27:131–139, 2003.
- [80] J.A. Westerhuis, T. Kourti, and J.F. MacGregor. Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, 13(3-4):397 – 413, 1999.
- [81] S.P. Gurden, J.A. Westerhuis, R. Bro, and A.K. Smilde. A comparison of multiway regression and scaling methods. *Chemometrics and Intelligent Laboratory Systems*, 59(1?2):121 – 136, 2001.
- [82] P. Nomikos and J.F. MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995.
- [83] S. Wold, N. Kettaneh, H. Friden, and A. Holmberg. Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemometrics and Intelligent Laboratory Systems*, 44:331–340, 1998.
- [84] W. Ku, R.H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179 – 196, 1995.
- [85] A. Wachs and D.R. Lewin. Improved pca methods for process disturbance and failure identification. *AIChE Journal*, 45(8):1688–1700, 1999.
- [86] J. Chen and K. Liu. On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chemical Engineering Science*, 57:63–75, 2002.
- [87] J. Camacho, J. Picó, and A. Ferrer. Bilinear modelling of batch processes. Part I: Theoretical discussion. *Journal of Chemometrics*, 22(5):299–308, 2008.
- [88] A. Simoglou, P. Georgieva, E.B. Martin, A.J. Morris, and S. Foyo de Azevedo. On-line monitoring of a sugar crystallization process. *Computers & Chemical Engineering*, 29:1411–1422, 2005.
- [89] J. Camacho, J. Picó, and A. Ferrer. Bilinear modelling of batch processes. Part II: A comparison of PLS soft-sensors. *Journal of Chemometrics*, 22(10):533–547, 2008.
- [90] J. Camacho, J. Picó, and A. Ferrer. On-line monitoring of batch processes based on PCA: Does the modelling

- structure matter? *Analytica chimica acta*, 642:59–69, 2009.
- [91] H. Ramaker, E.N.M. van Sprang, J.A. Westerhuis, and A.K. Smilde. Fault detection properties of global, local and time evolving models for batch process monitoring. *Journal of Process Control*, 15(7):799–805, 2005.
- [92] B. Lennox, G.A. Montague, H.G. Hiden, G. Kornfeld, and P.R. Goulding. Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, 74:125, 2001.
- [93] D. Neogi and C. Schlags. Multivariate statistical analysis of an emulsion batch process. *Industrial and Engineering Chemistry Research*, 37:3971–3979, 1998.
- [94] J. Camacho, J. Picó, and A. Ferrer. Multi-phase analysis framework for handling batch process data. *Journal of Chemometrics*, 22:632–643, 2008.
- [95] Y. Yao and F. Gao. A survey on multistage/multiphase statistical modeling methods for batch processes. *Annual Reviews in Control*, 33(2):172 – 183, 2009.
- [96] C. Undey, S. Ertunç, and A. Çinar. Online batch/fed-batch process performance monitoring, quality, prediction, and variable-contribution analysis for diagnosis. *Industrial & Engineering Chemistry Research*, 42:4645–4658, 2003.
- [97] S. Reinikainen and A. Höskuldsson. Multivariate statistical analysis of a multi-step industrial processes. *Analytica Chimica Acta*, 595(1–2):248 – 256, 2007.
- [98] D. Dong and T.J. McAvoy. Batch tracking via nonlinear principal component analysis. *AIChE Journal*, 42(8):2199–2208, 1996.
- [99] J.A. Westerhuis, T. Kourti, and J.F. MacGregor. Analysis of multiblock and hierarchical pca and pls models. *Journal of Chemometrics*, 12(5):301–321, 1998.
- [100] P. Facco, M. Olivi, C. Rebuscini, F. Bezzo, and M. Barolo. Multivariate statistical estimation of product quality in the industrial batch production of a resin. cancun (mexico), june 6-8, vol. 2, p.93-98. In *Proc. DYCOPS 2007 - 8th IFAC Symposium on Dynamics and Control of Process Systems (B. Foss and J. Alvarez, Eds.)*, 2007.

- [101] X. Doan and R. Srinivasan. Online monitoring of multi-phase batch processes using phase-based multivariate statistical process control. *Computers & Chemical Engineering*, 32(1-2):230 – 243, 2008.
- [102] N. Lu, F. Gao, and F. Wang. Sub-PCA modeling and on-line monitoring strategy for batch processes. *AIChE Journal*, 50(1):255–259, 2004.
- [103] S. Rännar, J.F. MacGregor, and S. Wold. Adaptive batch monitoring using hierarchical PCA. *Chemometrics and Intelligent Laboratory Systems*, 41:73–81, 1998.
- [104] A. Ferrer. Statistical control of measures and processes. *Comprehensive Chemometrics - Chemical and Biochemical Data Analysis*, Elsevier:Oxford, 1:97–126, 2009.
- [105] D.C. Montgomery. *Intorduction to Statistical Quality Control*, 5th ed. Wiley: New York, 2005.
- [106] E.S. Page. Cumulative sum control charts. *Technometrics*, 3(1):1 – 9, 1961.
- [107] D.M. Hawkins and D.H. Olwell. *Cumulative Sum Charts and Charting for Quality Improvement*. springer: New York, 1988.
- [108] S. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–102, 1959.
- [109] J.S. Hunter. Exponentially weighted moving average. *Journal of Quality Technology*, 18:97–102, 1986.
- [110] N.D. Tracy and R.L. Mason J.C. Young. Multivariate control charts for individual observations. *Journal of Quality Technology*, 24:88, 1992.
- [111] W.H. Woodall and N.M. Ncube. Multivariate cusum quality control procedures. *Technometrics*, 27:285 – 292, 1985.
- [112] C.A. Lowry, W.H. Woodall, C.W. Champ, and S.E. Rigdon. Multivariate exponentially weighted moving average control chart. *Technometrics*, 34:46 – 53, 1992.
- [113] H. Hotelling. *Multivariate Quality Control. Techniques of Statistical Analysis*. C. Eisenhart, M. Hastay, W.A. Wallis (eds.) MacGraw-Hill: New York: 111-184, 1947.
- [114] T. Kourti and J.F. MacGregor. Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 28(4):409–428, 1996.

- [115] J.E. Jackson and G.S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21:341, 1979.
- [116] G.E.P. Box. Some theorems on quadratic forms applied in the study of analysis of variance problems: Effect of inequality of variance in one-way classification. *The Annals of Mathematical Statistics*, 25:290–302, 1954.
- [117] J. Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502, 2003.
- [118] F. Lei, M. Rotbøll, and S.B Jørgensen. A biochemically structured model for *Saccharomyces cerevisiae*. *Journal of Biotechnology*, 88:205–221, 2001.
- [119] J. Ferrer, A. Seco, J. Serralta, J. Ribes, J. Manga, E. Asensi, J.J. Morenilla, and F. Llavador. DESASS: a software tool for designing, simulating and optimising WWTPs. *Environmental Modelling and Software*, 23:19–26, 2008.
- [120] A. Seco, J. Ribes, J. Serralta, and J. Ferrer. Biological nutrient removal model no. 1 (bnrm1). *Water Science and Technology*, 50(6):69–78, 2004.
- [121] J.B. Copp. Development of standardised influent files for the evaluation of activated sludge control strategies. *IAWQ Scientific and Technical Report Task Group: Respirometry in Control of the Activated Sludge Process - Internal report*, 1999.
- [122] N.F. Thornhill, M.A.A.S Choudhury, and S.L. Shah. The impact of compression on data-driven process analyses. *Journal of Process Control*, 14(4):389–398, 2004.
- [123] J.F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403 – 414, 1995.
- [124] B.R. Bakshi and G. Stephanopoulos. Compression of chemical process data by functional approximation and feature extraction. *AIChE Journal*, 42(2):477–492, 1996.
- [125] M.J. Watson, A. Liakopoulos, D. Brzakovi, and C. Georgakis. A practical assessment of process data compression techniques. *Industrial & engineering chemistry research*, 37(1):267–274, 1998.

- [126] M. Misra, S. Joe Qin, S. Kumar, and D. Seemann. On-line data compression and error analysis using wavelet technology. *AIChE Journal*, 46(1):119–132, 2000.
- [127] AspenTech. Analysis of data storage technologies for the management of real-time process manufacturing data, June 2015.
- [128] E.H. Bristol. Swinging door trending: Adaptive trend recording? In *ISA National Conf. Proc.*, pages 749–753, 1990.
- [129] OSIsoft. The compression algorithm - kb00699, June 2015.
- [130] J. Pettersson and P. Gutman. Automatic tuning of the window size in the box car back slope data compression algorithm. *Journal of Process Control*, 14(4):431 – 439, 2004.
- [131] S. García-Muñoz, M. Polizzi, A. Prpich, C. Strain, A. Lalonde, and V. Negron. Experiences in batch trajectory alignment for pharmaceutical process improvement through multivariate latent variable modelling. *Journal of Process Control*, 21(10):1370 – 1377, 2011.
- [132] Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.
- [133] Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.
- [134] T. Kourti. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17:93–109, 2003.
- [135] F. Arteaga and A. Ferrer. Framework for regression-based missing data imputation methods in on-line MSPC. *Journal of Chemometrics*, 19:439–447, 2005.
- [136] F. Arteaga and A. Ferrer. Missing data. *Comprehensive Chemometrics - Chemical and Biochemical Data Analysis*, Elsevier:Oxford, 3:285–314, 2009.
- [137] A. Folch-Fortuny, F. Arteaga, and A. Ferrer. PCA model building with missing data: New proposals and a comparative study. *Chemometrics and Intelligent Laboratory Systems*, 146:77 – 88, 2015.

- [138] Carl Runge. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. *Zeitschrift für Mathematik und Physik*, 46(224-243):20, 1901.
- [139] N. Lu, F. Gao, Y. Yang, and F. Wang. PCA-based modeling and on-line monitoring strategy for uneven-length batch processes. *Industrial and Engineering Chemical Research*, 43:3343–3352, 2004.
- [140] J. Camacho and J. Picó. Online monitoring of batch processes using multi-phase principal component analysis. *Journal of Process Control*, 10(16):1021–1035, 2006.
- [141] D.J. Louwerse and A.K. Smilde. Multivariate statistical process control of batch processes based on three-way models. *Chemical Engineering Science*, 55:1225 – 1235, 1999.
- [142] S. Lakshminarayanan, R. Gudi, and S. Shah. Monitoring batch processes using multivariate statistical tools: extensions and practical issues. In *Proceedings of IFAC Worm Congress*, pages 241–246, 1996.
- [143] O. Marjanovic, B. Lennox, D. Sandoz, K. Smith, and M. Crofts. Real-time monitoring of an industrial batch process. *Computers & Chemical Engineering*, 30(10–12):1476 – 1481, 2006.
- [144] J. Wan, O. Marjanovic, and B. Lennox. Uneven batch data alignment with application to the control of batch end-product quality. *{ISA} Transactions*, 53(2):584 – 590, 2014.
- [145] C. Duchesne, T. Kourti, and J.F. MacGregor. Multivariate SPC for startups and grade transitions. *AIChE Journal*, 48(12):2890–2901, 2002.
- [146] Y. Zhang, M. Dudzic, and V. Vaculik. Integrated monitoring solution to start-up and run-time operations for continuous casting. *Annual Reviews in Control*, 27(2):141 – 149, 2003.
- [147] S.G. Rothwell, E.B. Martin, and A.J. Morris. Comparison of methods for handling unequal length batches. In *Proceedings of IFAC DYCOPS5, Corfu, Greece*, pages 66–71, 1998.
- [148] R. Srinivasan and M.S. Qian. Online fault diagnosis and state identification during process transitions using

- dynamic locus analysis. *Chemical Engineering Science*, 61(18):6109 – 6132, 2006.
- [149] R. Srinivasan and M.S. Qian. Off-line temporal signal comparison using singular points augmented time warping. *Industrial & Engineering Chemistry Research*, 44(13):4697–4716, 2005.
- [150] R. Srinivasan and M.S. Qian. Online temporal signal comparison using singular points augmented time warping. *Industrial & Engineering Chemistry Research*, 46(13):4531–4548, 2007.
- [151] J. Chen and J. Liu. Post analysis on different operating time processes using orthonormal function approximation and multiway principal component analysis. *Journal of Process Control*, 10(5):411 – 418, 2000.
- [152] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.
- [153] N. Nielsen, J. Carstensen, and J. Smedsgaard. Aligning of single and multiple wavelength chromatographic profiles for chemometrics data analysis using correlation optimised warping. *Journal of Chromatography*, 805:17–35, 1998.
- [154] V. Pravdova, B. Walczak, and D. Massart. A comparison of two algorithms for warping of analytical signals. *Analytica chimica acta*, 456:77–92, 2002.
- [155] G. Tomasi and F. van den Berg. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *Journal of Chemometrics*, 18:231–241, 2004.
- [156] D. Clifford, G. Stone, I. Montoliu, S. Rezzi, F. Martin, P. Guy, S. Bruce, and S. Kochhar. Alignment using variable penalty dynamic time warping. *Analytical Chemistry*, 81(3):1000–1007, 2009.
- [157] P.H.C. Eilers. Parametric time warping. *Analytical Chemistry*, 76(2):404–411, 2004.
- [158] T.G. Bloemberg, J. Gerretzen, A. Lunshof, R. Wehrens, and L.M.C. Buydens. Warping methods for spectroscopic and chromatographic signal alignment: A tutorial. *Analytica Chimica Acta*, 781(0):14 – 32, 2013.

- [159] K. Gollmer and C. Posten. Supervision of bioprocesses using a dynamic time warping algorithm. *Control Engineering Practice*, 4(9):1287–1295, 1996.
- [160] H. Ramaker, E.N.M. van Sprang, J.A. Westerhuis, and A.K. Smilde. Dynamic time warping of spectroscopic batch data. *Analytica Chimica Acta*, 498:133–153, 2003.
- [161] M. Fransson and S. Folestad. Real-time alignment of batch process data using COW for on-line process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 84:56–61, 2006.
- [162] G. Gins, J. Espinosa, I. Y. Smets, W. Van Brempt, and J.F.M. Van Impe. Data alignment via dynamic time warping as a prerequisite for batch-end quality prediction. In *Proceedings of the 6th Industrial Conference on Data Mining Conference on Advances in Data Mining: Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, ICDM'06, pages 506–510, Berlin, Heidelberg, 2006. Springer-Verlag.
- [163] E.J. Keogh and M.J. Pazzani. Derivative dynamic time warping. In *In First SIAM International Conference on Data Mining (SDMO2001)*, 2001.
- [164] Y. Zhang, B. Lu, and T.F. Edgar. Batch trajectory synchronization with robust derivative dynamic time warping. *Industrial & Engineering Chemistry Research*, 52(35):12319–12328, 2013.
- [165] A. Cinar, S.J. Parulekar, C. Undey, and G. Birol. *Batch Fermentation: Modeling, Monitoring, and Control*. Chemical Industries. Taylor & Francis, 2003.
- [166] Athanassios Kassidas. Fault detection and diagnosis in dynamic multivariable chemical processes using speech recognition methods. *Open Access Dissertations and Theses. Paper 3386.*, 1997.
- [167] M. Herdin, N. Czink, H. Ozelik, and E. Bonek. Correlation matrix distance, a meaningful measure for evaluation of non-stationary mimo channels. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 1, pages 136–140 Vol. 1, 2005.
- [168] S. García-Muñoz, T. Kourti, and J.F. MacGregor. Model predictive monitoring for batch processes. *Industrial and Engineering Chemistry Research*, 43(18):5929–5941, 2004.

- [169] F. Arteaga and A. Ferrer. Dealing with missing data in MSPC: several methods, different interpretations, some examples. *Journal of Chemometrics*, 16:408–418, 2002.
- [170] M. Barker and W. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.
- [171] S. Wold. Pattern recognition by means of disjoint principal components models. *Pattern Recognition*, 8(3):127 – 139, 1976.
- [172] Kowalski B.R., editor. *Chemometrics: Theory and Application*. American Chemical Society, Washington, D.C., 1977.
- [173] J.A. Westerhuis, H.C.J. Hoefsloot, S. Smit, D.J. Vis, A.K. Smilde, E.J.J. van Velzen, J.P.M. van Duijnhoven, and F.A. van Dorsten. Assessment of plsda cross validation. *Metabolomics*, 4:81–89, 2008.
- [174] B.W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozymes. *Biochimica et Biophysica Acta - Protein Structure*, 405(2):442–451, 1975.
- [175] E.N.M. van Sprang, H. Ramaker, J.A. Westerhuis, S.P. Gurden, and A.K. Smilde. Critical evaluation of approaches for on-line batch process monitoring. *Chemical Engineering Science*, 57:3979–3991, 2002.
- [176] Nicolaas (Klaas) M Faber and Rasmus Bro. Standard error of prediction for multiway pls: 1. background and a simulation study. *Chemometrics and Intelligent Laboratory Systems*, 61(1–2):133 – 149, 2002.
- [177] F. Lindgren, B. Hansen, W. Karcher, W. Sjöström, and L. Eriksson. Model validation by permutation tests: applications to variable selection. *Journal of Chemometrics*, 10(5-6):521–532, 1996.
- [178] Y. Xu, S. Zomer, and R. Brereton. Support vector machines: a recent method for classification in chemometrics. *Critical Reviews in Analytical Chemistry*, 36:177–188, 2005.
- [179] G. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.
- [180] D. Coomans and D.L. Massart. Alternative k-nearest neighbour rules in supervised pattern recognition. part

1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta.*, 136:15–27, 1982.
- [181] J. Camacho and A. Ferrer. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects. *Chemometrics and Intelligent Laboratory Systems*, 131(0):37 – 50, 2014.
- [182] Yingwei Zhang and S Joe Qin. Fault detection of nonlinear processes using multiway kernel independent component analysis. *Industrial & Engineering Chemistry Research*, 46(23):7780–7787, 2007.
- [183] T. Kourti. Application of latent variable methods to process control and multivariate statistical process control in industry. *Int. J. Adapt. Control Signal Process.*, 19:213–246, 2005.
- [184] J.E. Jackson. *A User's Guide to Principal Components*. Wiley-Interscience, England, 2003.
- [185] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.
- [186] J.M. Lee, C. Yoo, and I.B. Lee. On-line batch process monitoring using different unfolding method and independent component analysis. *Journal of Chemical Engineering of Japan*, 36(11):1384–1396, 2003.
- [187] C. Zhao, F. Wang, Z. Mao, N. Lu, and M. Jia. Improved batch process monitoring and quality prediction based on multiphase statistical analysis. *Industrial & Engineering Chemistry Research*, 47:835–849, 2008.
- [188] E. Martin, J. Morris, and S. Lane. Monitoring process manufacturing performance. *IEEE Contr. Syst. Mag.*, pages 26–39, 2002.
- [189] C. Ündey, E. Tatara, and A. Çinar. Real-time batch process supervision by integrated knowledge-based systems and multivariate statistical methods. *Engineering Applications of Artificial Intelligence*, 16:555–566, 2003.
- [190] C. Ündey, E. Tatara, and A. Çinar. Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations. *Journal of Biotechnology*, 108(1):61 – 77, 2004.
- [191] P. Facco, F. Doplicher, F. Bezzo, and M. Barolo. Moving average PLS soft sensor for online product quality

- estimation in an industrial batch polymerization process. *Journal of Process Control*, 19(3):520 – 529, 2009.
- [192] J. Mingxing, L. Fengxiang, and G. Shouping. Optimal PCA-based modeling and fault diagnosis for uneven-length batch processes. In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pages 1731–1736, 2010.
- [193] H. Huang and H. Qu. In-line monitoring of alcohol precipitation by near-infrared spectroscopy in conjunction with multivariate batch modeling. *Analytica Chimica Acta*, 707(1–2):47 – 56, 2011.
- [194] U. Jeppsson, J. Alex, M.N. Pons, H.i Spanjers, and P. Vanrolleghem. Status and future trends of ica in wastewater treatment - a european perspective. *WATER SCIENCE AND TECHNOLOGY*, 45(4-5):485–494, 2002.
- [195] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311, 2003.
- [196] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313–326, 2003.
- [197] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yinn. A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & chemical engineering*, 27(3):327–346, 2003.
- [198] Rolf Isermann. Process fault detection based on modeling and estimation methods—a survey. *Automatica*, 20(4):387 – 404, 1984.
- [199] M. Basseville. Detecting changes in signals and systems—a survey. *Automatica*, 24(3):309 – 326, 1988.
- [200] J.J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Syst. Mag.*, 8(6):3–11, 1988.
- [201] S. Yoon and J. F. MacGregor. Fault diagnosis with multivariate statistical models part i: using steady state fault signatures. *Journal of Process Control*, 11(4):387 – 400, 2001.

- [202] S. Yoon and J.F. MacGregor. Statistical and causal model-based approaches to fault detection and isolation. *AICHE Journal*, 46(9):1813–1824, 2000.
- [203] J.F. MacGregor and A. Cinar. Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods. *Computers & Chemical Engineering*, 47:111–120, 2012.
- [204] B.M. Wise, N.L. Ricker, D.F. Veltkamp, and B.R. Kowalski. A theoretical basis for the use of principal component models for monitoring multivariate processes. *Process control and quality*, 1(1):41–51, 1990.
- [205] T.E. Marlin J.V. Kresta, J.F. MacGregor. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, 69(1):35–47, 1991.
- [206] K.A. Kosanovich M.J. Piovoso and J.P. Yuk. Process data chemometrics. *Instrumentation and Measurement, IEEE Transactions on*, 41(2):262–268, 1992.
- [207] B. M. Wise and N. B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329 – 348, 1996.
- [208] T. Kourti, J. Lee, and J.F. MacGregor. Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers & Chemical Engineering*, 20:S745–S750, 1996.
- [209] R.L. Mason, C.W. Champ, N.D. Tracy, S.J. Wierda, and J.C. Young. Assesment of multivariate process control techniques. *Journal of Quality Technology*, 29(2):140–143, 1997.
- [210] P. R. Goulding, B. Lennox, D. J. Sandoz, K. J. Smith, and O. Marjanovic. Fault detection in continuous processes using multivariate statistical methods. *International Journal of Systems Science*, 31(11):1459–1471, 2000.
- [211] B.M. Wise, N.B. Gallagher, S.W. Butler, D. White, and G.G. Barna. Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process: impact of measurement selection and data treatment on sensitivity. In *IFAC SAFEPROCESS*, volume 97, pages 35–42. Citeseer, 1997.

- [212] W. Li, H.H. Yue, S. Valle-Cervantes, and S. Joe Qin. Recursive {PCA} for adaptive process monitoring. *Journal of Process Control*, 10(5):471 – 486, 2000.
- [213] H.H. Yue, J. Qin, R.J. Markle, C. Nauert, and M. Gatto. Fault detection of plasma etchers using optical emission spectra. *Semiconductor Manufacturing, IEEE Transactions on*, 13(3):374–385, 2000.
- [214] I. Miletic, S. Quinn, M. Dudzic, V. Vaculik, and M. Champagne. An industrial perspective on implementing on-line applications of multivariate statistics. *Journal of Process Control*, 14(8):821–836, 2004.
- [215] Y. Zhang and M.S. Dudzic. Online monitoring of steel casting processes using multivariate statistical technologies: From continuous to transitional operations. *Journal of Process Control*, 16(8):819 – 829, 2006.
- [216] M. Kano and Y. Nakagawa. Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers & Chemical Engineering*, 32(1):12–24, 2008.
- [217] K.A. Kosanovich, K.S. Dahl, and M.J. Piovoso. Improved process understanding using multiway principal component analysis. *Engineering Chemical Research*, 35:138–146, 1996.
- [218] M. Sánchez, U. Cortés, J. Béjar, J. De Gracia, J. Lafuente, and M. Poch. Concept formation in wwtp by means of classification techniques: a compared study. *Applied Intelligence*, 7(2):147–165, 1997.
- [219] P. Teppola, S.P. Mujunen, and P. Minkkinen. Partial least squares modeling of an activated sludge plant: A case study. *Chemometrics and Intelligent Laboratory Systems*, 38(2):197 – 208, 1997.
- [220] C. Rosen and G. Olsson. Disturbance detection in wastewater treatment plants. *Water Science and Technology*, 37(12):197 – 205, 1998.
- [221] R.K. Tomita, S.W. Park, and O.Z. Sotomayor. Analysis of activated sludge process using multivariate statistical tools—A PCA approach. *Chemical Engineering Journal*, 90(3):283–290, 2002.

- [222] Union européenne. Direction générale de la recherche. *The COST simulation benchmark: description and simulator manual*. Directorate-General for Research, 2002.
- [223] D. Garcia-Alvarez. Fault detection using principal component analysis (PCA) in a wastewater treatment plant (WWTP). In *Proceedings of the International Student Scientific Conference*, 2009.
- [224] C. Rosen, J. Röttorp, and U. Jeppsson. Multivariate on-line monitoring: challenges and solutions for modern wastewater treatment operation. *Water Science & Technology*, 47(2):171–179, 2003.
- [225] M.V. Ruano, J. Ribes, A. Seco, and J. Ferrer. Low cost-sensors as a real alternative to on-line nitrogen analysers in continuous systems. *Water Science & Technology*, 60(12):3261–3268, 2009.
- [226] Gustaf Olsson. {ICA} and me – a subjective review. *Water Research*, 46(6):1585 – 1624, 2012.
- [227] Christian Rosen. *A chemometric approach to process monitoring and control-with applications to wastewater treatment operation*. Lund University, 2001.
- [228] F Tsung. Statistical monitoring and diagnosis of automatic controlled processes using dynamic pca. *International Journal of Production Research*, 38(3):625–637, 2000.
- [229] C. Wikström, C. Albano, L. Eriksson, H. Fridén, E. Johansson, A. Nordahl, S. Rännar, M. Sandberg, N. Kettaneh-Wold, and S. Wold. Multivariate process and quality monitoring applied to an electrolysis process: Part ii. multivariate time-series analysis of lagged latent variables. *Chemometrics and intelligent laboratory systems*, 42(1):233–240, 1998.
- [230] Lennart Eriksson, T Byrne, E Johansson, Johan Trygg, and C Vikström. *Multi-and megavariable data analysis basic principles and applications*. Umetrics Academy, 2013.
- [231] A. Negiz and A. Çinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal*, 43(8):2002–2020, 1997.
- [232] C.Y. Yoo, S.W. Choi, and I. Lee. Dynamic monitoring method for multiscale fault detection and diagnosis in mspc. *Industrial & engineering chemistry research*, 41(17):4303–4317, 2002.

- [233] A. Raich and A. Cinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE Journal*, 42(4):995–1009, 1996.
- [234] M. Kano, S. Hasebe, I. Hashimoto, and H. Ohno. Statistical process monitoring based on dissimilarity of process data. *AIChE journal*, 48(6):1231–1240, 2002.
- [235] R. Rengaswamy and V. Venkatasubramanian. A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engineering Applications of Artificial Intelligence*, 8(1):35–51, 1995.
- [236] J. Wang and Q. P. He. Multivariate statistical process monitoring based on statistics pattern analysis. *Industrial & Engineering Chemistry Research*, 49(17):7858–7869, 2010.
- [237] V. Venkatasubramanian and K. Chan. A neural network methodology for process fault diagnosis. *AIChE Journal*, 35(12):1993–2002, 1989.
- [238] J.Y. Fan, M. Nikolaou, and R.E. White. An approach to fault diagnosis of chemical processes via neural networks. *AIChE Journal*, 39(1):82–88, 1993.
- [239] J.C. Hoskins, K.M. Kaliyur, and D.M. Himmelblau. Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE Journal*, 37(1):137–141, 1991.
- [240] S. Zhou, J. Zhang, and S. Wang. Fault diagnosis in industrial processes using principal component analysis and hidden markov model. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5680–5685. IEEE, 2004.
- [241] R. Dunia, J. Qin, T.F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42(10):2797–2812, 1996.
- [242] H Henry Yue and S Joe Qin. Reconstruction-based fault identification using a combined index. *Industrial & engineering chemistry research*, 40(20):4403–4414, 2001.
- [243] R. Dunia and J. Qin. Subspace approach to multidimensional fault identification and reconstruction. *AIChE Journal*, 44(8):1813–1831, 1998.
- [244] A. Raich and A. Çinar. Diagnosis of process disturbances by statistical distance and angle measures. *Computers & Chemical Engineering*, 21(6):661 – 673, 1997.

- [245] P. He, J. Qin, and J. Wang. A new fault diagnosis method using fault directions in fisher discriminant analysis. *AIChE journal*, 51(2):555–571, 2005.
- [246] G. Garcia M. Fuente and G.I. Sainz. Fault diagnosis in a plant using fisher discriminant analysis. In *Control and Automation, 2008 16th Mediterranean Conference on*, pages 53–58. IEEE, 2008.
- [247] D. Garcia-Alvarez, M.J. Fuente, P. Vega, and G. Sainz. Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant. In *Proceedings of the 7th IFAC international symposium on advanced control of chemical processes, Turkey*, 2009.
- [248] L.H. Chiang, E.L. Russell, and R.D. Braatz. Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and intelligent laboratory systems*, 50(2):243–252, 2000.
- [249] Y. Ignacio, G. Escudero, M. Graells, and L. Puigjaner. Performance assessment of a novel fault diagnosis system based on support vector machines. *Computers & chemical engineering*, 33(1):244–255, 2009.
- [250] K.P. Detroja, R.D. Gudi, S.C. Patwardhan, and K. Roy. Fault detection and isolation using correspondence analysis. *Industrial & engineering chemistry research*, 45(1):223–235, 2006.
- [251] S. Yoon and J.F. MacGregor. Statistical and causal model-based approaches to fault detection and isolation. *AIChE Journal*, 46(9):1813–1824, 2000.
- [252] P. Miller, R.E. Swanson, and C.E. Heckler. Contribution plots: a missing link in multivariate quality control. *Applied mathematics and computer science*, 8:775–792, 1998.
- [253] J.A. Westerhuis, S.P. Gurden, and A.K. Smilde. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51(1):95–114, 2000.
- [254] S. Vidal-Puig and A. Ferrer. Fingerprints contribution plot a new approach for fault diagnosis in multivariate statistical process control. In *Proceedings of the 11th International Conference on Chemometrics in Analytical*

- Chemistry, Montpellier, France.*, CAC'08, pages 27–31, 2008.
- [255] S. Wold, P. Geladi, K. Esbensen, and J. Ohman. Multiway principal components-and-PLS-analysis. *Journal of Chemometrics*, 1:41 – 56, 1987.
- [256] P. Nomikos. *Statistical process control of batch processes*. PhD Thesis, McMaster University, Hamilton, ON, 1995.
- [257] J. Camacho. *New Methods Based on the Projection to Latent Structures for Monitoring, Prediction and Optimization of Batch Processes*. PhD Dissertation, Universidad Politécnic de Valencia, 2007.
- [258] J. Camacho, P. Padilla, J. Díaz-Verdejo, K. Smith, and D. Lovett. Least-squares approximation of a space distribution for a given covariance and latent sub-space. *Chemometrics and Intelligent Laboratory Systems*, 105(2):171 – 180, 2011.
- [259] K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2:559–572, 1901.
- [260] H. Wold. *Nonlinear estimation by iterative least squares procedures*. In *Research papers in Statistics, David, F. (ed.)*, Wiley: New York: 441-444, 1966.
- [261] S. Wold. Cross-validatory estimation of the number of components in factor and principal components. *Technometrics*, 20(4):397 – 405, 1978.
- [262] H.T. Eastment and W.J. Krzanowski. Cross-validatory choice of the number of components from a principal component analysis. *Technometrics*, 24(1):73–77, 1982.
- [263] J. Camacho and A. Ferrer. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical aspects. *Journal of Chemometrics*, 27(7):361–373, 2012.
- [264] H. Hotelling. the relations of the newer multivariate statistical methods to factor analysis. *British Journal of Statistical Psychology*, 10:69–79, 1957.
- [265] M.G. Kendall. *A Course in Multivariate Analysis*. Griffin: London, 1957.
- [266] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 185:1–17, 1986.

- [267] P. Geladi and B.R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.
- [268] S. de Jong. SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1992.
- [269] R.A. Harshman. Foundations of the parafac procedure: models and conditions for an 'explanatory' multimodal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970.
- [270] R.B. Cattell. Parallel proportional profiles and other principles for determining the choice of factor by rotation. *Psychometrika*, 9:267–283, 1944.
- [271] A.K. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis, Application in the Chemical Sciences*. John Wiley & Sons: England, 2003.
- [272] R. Bro. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149, 1997.
- [273] X. Meng, A. Morris, and E. Martin. On-line monitoring of batch processes using a PARAFAC representation. *Journal of Chemometrics*, 17(1):65–81, 2003.
- [274] B.M. Wise, N.B. Gallagher, and E.B. Martin. Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch. *Journal of Chemometrics*, 15:285–296, 2001.
- [275] P.P. Mortensen and R. Bro. Real-time monitoring and chemical profiling of a cultivation process. *Chemometrics and Intelligent Laboratory Systems*, 84(1–2):106 – 113, 2006.
- [276] S. Matero, S. Poutiainen, J. Leskinen, S. Reinikainen, J. Ketolainen, K. Järvinen, and A. Poso. Monitoring the wetting phase of fluidized bed granulation process using multi-way methods: The separation of successful from unsuccessful batches. *Chemometrics and Intelligent Laboratory Systems*, 96(1):88 – 93, 2009.
- [277] L.R. Tucker. *The extension of factor analysis to three-dimensional matrices. in: Frederiksen n, gulliksen h. contributions to mathematical psychology*. New York: Holt, rinehart and winston. pages 110-162, 1964.

- [278] P.M. Kroonenberg. *Three-Mode Principal Component Analysis: Theory and Applications*. DSWO Press: Leiden, 1983.
- [279] R. Henrion and C.A. Andersson. A new criterion for simple-structure transformations of core arrays in N-way principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 47:189–204, 1999.
- [280] A.K. Smilde. Comments of three-way analysis used for batch process data. *Journal of Chemometrics*, 15:19–27, 2001.
- [281] A.K. Smilde and S. van der Wiel. Application of tucker models for batch monitoring. *AIChE Journal*, 62:1320, 1999.
- [282] L.H. Chiang, R. Leardi, R.J. Pell, and M.B. Seasholtz. Industrial experiences with multivariate statistical analysis of bath process data. *Chemometrics and Intelligent Laboratory Systems*, 81:109–119, 2006.
- [283] L. Luo, S. Bao, Z. Gao, and J. Yuan. Batch process monitoring with gtucker2 model. *Industrial & Engineering Chemistry Research*, 53(39):15101–15110, 2014.
- [284] I. Stanimirova, B. Walczak, D.L. Massart, V. Simeonov, C.A. Saby, and E.D. Crescenzo. STATIS, a three-way method for data analysis. application to environmental data. *Chemometrics and Intelligent Laboratory Systems*, 72(2):219–233, 2004.
- [285] S. Gourvéneq, I. Stanimirova, C-A. Saby, C.Y. Airiau, and D.L. Massart. Monitoring batch processes with the STATIS approach. *Journal of chemometrics*, 19:288–300, 2005.
- [286] R. Bro. Multiway calibration. multi-linear PLS. *Journal of Chemometrics*, 10:47–61, 1996.
- [287] R. Boqué and A.K. Smilde. Monitoring and diagnosing batch processes with multiway covariates regression models. *AIChE Journal*, 45(7):1504–1520, 1999.
- [288] B.M. Wise, N.B. Gallagher, S.W. Butler, D.D. White, and G.G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13(3-4):379–396, 1999.

- [289] R. Bro, K. Kjeldahl, A.K. Smilde, and H.A.L. Kiers. Cross-validation of component models: A critical look at current methods. *Journal of Chemometrics*, 390:1241–1251, 2008.
- [290] R. Bro and A.K. Smilde. Principal component analysis. *Analytical Methods*, 6:2812–2831, 2014.
- [291] O.E. de Noord. The influence of data preprocessing on the robustness and parsimony of multivariate calibration models. *Chemometrics and Intelligent Laboratory Systems*, 23(1):65 – 70, 1994.
- [292] M.S. Barlett. Tests of significance in factor analysis. *The British Journal of Psychology*, 3:77–85, 1950.
- [293] R.B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1:245–276, 1966.
- [294] S. Frontier. Étude de la décroissance des valeurs propres dans une analyse en composantes principales: Comparaison avec le modèle du bâton brisé. *Journal of Experimental Marine Biology and Ecology*, 25(1):67 – 75, 1976.
- [295] P.R. Peres-Neto, D.A. Jackson, and K.M. Somers. How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974 – 997, 2005.