UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA CHIMICA E DEI PROCESSI INDUSTRIALI

Tesi di Laurea Magistrale in
Ingegneria Chimica e dei Processi Industriali

Batch process monitoring using an assumption-free modeling methodology

Relatore: prof. Massimiliano Barolo
Correlatore: prof. Pierantonio Facco

Laureanda: ALICE FRACASSETTO

ANNO ACCADEMICO 2021 – 2022

To my parents.

Always grateful.

Abstract

Batch-wise unfolded multiway principal component analysis (MPCA) is a powerful tool for online monitoring of batch processes. However, it is constrained by the fact that all calibration and validation batches must have the same duration, otherwise they need to be time-aligned (synchronized) before being unfolded into a two-dimensional matrix. To overcome this problem, Westad et al. (2015) proposed a methodology based on a variable-wise unfolded PCA, which aims at modelling a normal trajectory of the process in the score space though a grid-search algorithm. This modelling methodology is called assumption-free, because the assumption that all batches must have the same length is no longer required. In this thesis, assumption-free models and batch-wise unfolded models are developed and tested to monitor five benchmark processes with different characteristics. For each process, the number of missed faults, false faults and false alarms are computed. The assumption-free model is able to recognize most faulty batches, without raising a significant number of false alarms; however, its performances are strongly affected by the shape of the normal trajectory of the process and by the quality of calibration batches. Moreover, the assumption-free models tend to recognize batches according to the number of consecutive scores (or residuals) out of the confidence limits, but they do not consider the overall path of the scores: this may lead to missing fault detection. The batch-wise unfolded models are able to recognize abnormal batches, but they may raise several false alarms (especially for normal batches).

Riassunto

I processi batch sono ampiamente utilizzati nell'industria manufatturiera per la produzione di prodotti ad elevato valore aggiunto. Un efficiente monitoraggio di questi processi, unito a un intervento tempestivo da parte dell'operatore in caso di anomalia, è fondamentale per ridurre eventuali sprechi di materiale, tempo e denaro. L'elevato numero di variabili che influenzano il processo, e la numerosità delle reazioni chimiche che possono aver luogo contemporaneamente, rendono complesso il monitoraggio del processo. Uno strumento modellistico che negli ultimi anni si è rilevato molto utile non solo per la comprensione delle relazioni tra le variabili misurate, ma anche per il monitoraggio dell'intero processo, è la PCA (*principal component analysis*). Tale metodologia consiste nel rappresentare grandi quantità di informazioni, relative alle variabili di processo, in uno spazio di ridotta dimensione. Per poter calibrare un modello PCA, è necessario che la matrice di calibrazione, contenente misurazioni effettuate su processi batch in condizioni operative normali, sia in forma bidimensionale. Dati tridimensionali possono essere organizzati in matrici *variable-wise unfolded* o *batch-wise unfolded*. Nel caso di quest'ultima, è necessario che tutti i batch di calibrazione abbiano la stessa durata (stesso numero di istanti di tempo campionati), situazione che non sempre si verifica in un comune impianto industriale, rendendo necessario il ricorso a metodi di sincronizzazione delle traiettorie temporali, i quali possono avere conseguenze sulle prestazioni del modello. Per evitare di ricorrere ad un processo di sincronizzazione dei batch, Westad et al. (2015) hanno proposto un modello di monitoraggio che non richiede come condizione l'eguaglianza della durata dei batch. Questo approccio viene denominato *assumption-free*. Tale modello PCA è basato su una matrice di calibrazione *variable-wise unfolded*, e utilizza un algoritmo di ricerca a griglia per modellare una traiettoria nello *score plot*, rappresentativa di un processo in condizioni operative normali. In questa tesi, due tecniche modellistiche, una *assumption-free* e una *batch-wise*, sono sviluppate e testate con 5 diversi dataset, al fine di determinare quale dei due è il più appropriato per il monitoraggio di processi batch.

Per quanto riguarda il modello *assumption-free*, sono considerate diverse configurazioni di griglia sullo *score plot*, e per ogni cella di ogni griglia si ricercano gli *scores* in essa contenuti: se una cella contiene almeno uno *score* per ogni batch di calibrazione, allora è ritenuta "valida". L'algoritmo seleziona come griglia ottimale quella che, con il maggior numero di celle valide, è in grado di catturare almeno il 95% di tutti gli *scores* di calibrazione. Per ogni cella valida (della griglia ottimale), è calcolata la media di tutti gli scores in essa contenuti; quindi, per interpolazione di tutte le medie calcolate, si ottiene la traiettoria rappresentativa di un processo normale. Per ogni cella valida, si calcolano la distanza degli *scores* dalla traiettoria e i residui *Q* corrispondenti, quindi vengono calcolati i limiti relativi agli *scores* e ai

residui *Q*. Infine, gli allarmi sullo *score plot* e sui residui sono calibrati considerando i batch di calibrazione.

Il modello in *batch-wise unfolding* è sviluppato con lo scopo di effettuare un monitoraggio in tempo reale; pertanto, ad ogni istante di tempo è necessario stimare i valori mancanti delle variabili, relativi agli istanti di tempo futuri. In questo caso, si adotta la procedura suggerita da Nomikos & MacGregor (1994).

Al termine del test dei due modelli con tutti i batch disponibili per la convalida, il modello *assumption-free* si rivela in grado di riconoscere i batch anomali. Tuttavia, una forma complessa della traiettoria del processo, data ad esempio dalla presenza di rapidi cambi di direzione (curve strette o angoli), modellata con un esiguo numero di celle valide, può abbassare drasticamente la sensibilità degli allarmi, portando al mancato riconoscimento dei batch anomali e compromettendo quindi le prestazioni del modello. Un mancato riconoscimento dell'anomalia può verificarsi anche nel caso in cui il dataset di calibrazione contenga almeno un batch la cui traiettoria si discosta molto da quelle degli altri batch: dal momento che l'allarme è calibrato considerando tutti i batch allo stesso modo, un solo batch di calibrazione con un elevato numero di *scores* consecutivi fuori dai limiti di confidenza è sufficiente per ridurre la sensibilità dell'allarme. Un ulteriore limite di questo modello, osservato nel caso del dataset n.4, consiste nell'incapacità di riconoscere un batch anomalo nel caso in cui gli *scores* si trovino all'interno dell'area di confidenza, ma seguano una traiettoria diversa da quella rappresentativa del processo normale (modellata attraverso l'algoritmo di ricerca a griglia). Il modello *batch-wise*, invece, è in grado di riconoscere i batch anomali in ogni occasione; tuttavia, presenta numerosi falsi allarmi nel caso di batch normali; nel caso dei dataset n.3, n.4 e n.5, tutti i batch normali risultano essere anomali.

Table of contents

Introduction

Batch processes are very common in the industrial manufacturing of several high-value compounds like chemicals, drugs, fermented foods, polymers and semi-conductors, as mentioned by Kosanovich et al. (1996), Wang (2015) and Jeffy et al. (2018). Most of batch processes involve expensive raw materials, and an online process monitoring would allow materials, time and money savings. As mentioned by Chai et al. (2013), the complexity of process monitoring is due to the high number of variables affecting the process, the nigh number of samples collected, the complexity of the process itself, with several reactions occurring at the same time, and the limited time available for process monitoring (and control). As mentioned before, batch products are high-value compounds. Therefore, detecting promptly a fault occurring in the process and acting to bring the manufacturing process to normal operating conditions is paramount to saving money and time, to avoiding waste of raw materials, and to increasing efficiency and product quality. Statistical process control (SPC) methodologies, and in particular multiway principal component analysis (MPCA), have become the main tool for on-line process monitoring and fault detection in last years for two main reasons. The first reason is that they allow an easier and more effective process understanding by compressing data and projecting them onto a low-dimensional space, in which main correlations between variables can be identified clearly (Nomikos and MacGregor, 1994). The second reason is that they allow one to build a model of the process under normal operating conditions without any knowledge about the process and its kinetics, exploiting only data collected from an appropriate number of batches running in normal operating conditions (NOC) and whose products are within specifications (Camacho et al., 2009). New samples of the batch to monitor are projected onto the model representing a normal process, and faults can be identified; then, the engineer can exploit his or her knowledge to manipulate variables in order to obtain a final product within specifications. Industrial data are usually collected in three-dimensional matrix that need to be unfolded before performing a PCA analysis. Several unfolding methodologies have been reported by Camacho et al. (2008) and Camacho et al. (2009), and two main approaches can be identified: the batch wise unfolding and the variable-wise unfolding.

As mentioned by Camacho et al. (2008), the first one allows for a representation of the complete batch, while the second one treats data collected at each time instant: while in the second case new data can be projected onto the model as they are at each time instant, in the case of batch-wise unfolding the entire matrix of new samples must be completed; however only at the end of the process all real data are available. The consequence is that if a new batch need to be projected in real time onto a batch-wise model, missing data for future time instants need to be predicted. To this purpose, Nomikos and MacGregor (1994) suggested that keeping the deviation from the mean of the last time instant constant for the remaining time instants revealed to be a good prediction of missing future samples. The other issue regarding the batch-wise unfolding model is related to batch length: in many manufacturing processes the duration of a batch may vary across batches. In order to apply batch-wise unfolding PCA, data collected need to be aligned such that all batches have the same number of samples (Camacho et al., 2008).

A solution to avoid these issues was proposed by Westad et al. (2015), and consists in a variable-wise unfolding model which aims at modelling a normal trajectory of the process, without the need of missing data imputation for data alignment. The model is based on a grid-search algorithm that is able to model the process trajectory based on data collected from batches under normal operating conditions; confidence limits are calculated to define the region inside which new projected data are deemed normal. New data collected from the process at every time instant are projected onto the model and compared to the trajectory modelled and its limits: if a fault is detected (i.e., if the new batch trajectory deviates from the normal one, thus going out of confidence limits), the main causes are investigated and corrective actions can be taken.

The objective of this thesis is to investigate the monitoring approach proposed by Westad et al. (2015), and to compare it with a batch-wise monitoring approach in order to assess which one is more suitable for process monitoring. It is important to notice that the variable-wise model is not described in detail by Westad et al. (2015); in particular, neither the grid-search algorithm criteria and parameters, nor assumptions and methods for confidence limits calculation, are described in detail in the original manuscript.

The thesis is organized in 4 Chapters. Chapter 1 contains the principles of principal component analysis and of a typical batch-wise model. Chapter 2 includes the description of the 5 datasets available in the literature, with different characteristics and related to different processes. In Chapter 3 the procedure used to develop the assumption-free model is discussed. In Chapter 4 all the case studies are presented together with their results.

Chapter 1

Process monitoring models

Two modelling strategies are considered in this thesis for the purpose of process monitoring. Both of them use principal component analysis (PCA) (Nomikos and MacGregor, 1994; Jeffy et al., 2018) as a modelling platform. The first modelling approach is based on batch-wise unfolded dataset, and exploits the score plot, Hotelling T^2 statistics and Q residuals to detect process abnormalities. The second one is an assumption-free model calibrated with variable-wise unfolded dataset and consists in a trajectory in the score plot representing batch in normal operating conditions: a new batch is considered like faulty if it deviates from this trajectory.

1.1 Multi-way principal component analysis (MPCA)

As discussed by Nomikos & MacGregor (1994), multi-way principal component analysis is a powerful statistical technique that allows one to explain the variance and covariance within a multivariate dataset through a linear combination of few terms. The dataset is decomposed in order to capture directions of maximum variability: these directions define the new low-dimension coordinate system on which the original data are projected, allowing an easier overview of batch history and correlations between variables. Industrial datasets are usually available in three-dimensional arrays in the form $\boldsymbol{X_{3D}}(I \times J \times K)$, like reported in Figure 1.1, where I represents the number of batches sampled, J is the number of variables and K the number of time instants sampled for each variable for each batch.

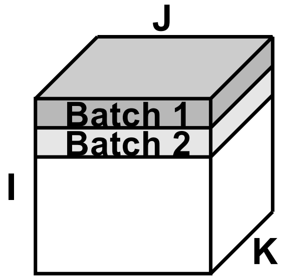


fig-1.1.jpg

Figure 1.1. *Structure of a three-dimensional matrix. I is the number of batches, J is the number of column and K is the number of samples (time instants). Each horizontal "layer" contains K samples for J variables of one batch*

Carrying out an MPCA is like performing a PCA on a large two-dimensional dataset obtained by unfolding the original three-dimensional dataset while preserving the dimension of I

batches (obtaining a batch-wise unfolded matrix of dimensions $I \times JK$), as described by Nomikos and MacGregor (1994) and Camacho et al. (2009) or the dimension of J variables (obtaining a variable-wise unfolded matrix of dimensions $KI \times J$). The matrix obtained after unfolding is autoscaled: the mean of each column is subtracted to the column itself, which is then scaled on its variance. The resulting pre-processed matrix has all columns with mean equal to zero and unit variance. In order to define the reduced latent space, directions of maximum variability of the data are calculated starting from the covariance matrix defined by Wise et al. (2006) like

$$\mathrm{cov}(\mathbf{X}) = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{m-1}, \tag{1.1}$$

where **X** is the unfolded matrix and m is its number of rows. Its eigenvalues and corresponding eigenvectors are then calculated by Wise et al. (2006) according to

$$\mathrm{cov}(\mathbf{X})\mathbf{p}_n = \lambda_n \mathbf{p}_n \,, \tag{1.2}$$

in which λ_n is the eigenvalue associated to the eigenvector $\mathbf{p}_n$. $\mathbf{p}_n$ are called "loadings" and are vectors that provide directions of maximum variability of the data. Multiplying the loading matrix $\mathbf{P}_{all}$ by the unfolded matrix **X**, the projections of original data onto the new low-dimensional space can be obtained (Jeffy et al., 2018; Wise et al., 2006):

$$\mathbf{T} = \mathbf{X}\mathbf{P}_{all} \,. \tag{1.3}$$

T is the score matrix and contains coordinates of original data into the reduced space. Since the objective of the PCA model is to simplify data inspection representing them in a low-dimensional space, a good approach is to build the PCA model using only few principal components PC (i.e., few dimensions in the new coordinate system) to represent data, without significant loss of information. For this purpose, eigenvalues and corresponding eigenvectors are ordered in descending order: the higher the eigenvalue, the higher the variance of data explained by its eigenvector (loading). As a consequence, the **X** matrix results to be

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E} \,, \tag{1.4}$$

in which **P** is the truncated matrix of loadings, and **E** is the error matrix containing the part of data unexplained by the model (data modelled by discarded principal components, and usually representing measurement noise). Each score $\mathbf{t}_i$ is representative of one sampled batch (row of **X**), so information about how batches are related to each other can be extracted from the score plot, in which clusters can be identified. The loadings $\mathbf{p}_i$ are representative of variables (columns of **X**), and correlations between different variables can be identified from the loading plot. Different criteria can be adopted to select the number of principal components (PCs) to retain into the PCA model. In this thesis the root-mean-square-error of cross validation (RMSECV) is used for every case study.

In Wise et al. (2006), the RMSECV is defined as

$$\mathrm{RMSECV}_n = \sqrt{\frac{1}{Z}\sum_{l}^{l=Z}(\hat{y}_l - y_l)^2} \quad , \tag{1.5}$$

in which the $\hat{y}_l$ are predictions for samples that are not included in model formulation, and y_l are Z real samples that are not included in model formulation. n refers to the number of principal components used to build the model on which the RMSECV is then calculated. The optimal number of principal components to use to build the model is the one at which the curve of the RMSECV reaches its minimum, or the one at which the curve RMSECV vs PCs shows an “elbow”. A measure of the variation of each sample within the PCA model is given by the Hotelling T^2 statistic, which is defined by Jeffy et al. (2018) and Wise et al. (2006) as the sum of the squares of scores:

$$T_i^2 = \mathbf{t}_i\mathbf{\Lambda}^{-1}\mathbf{t}_i^{\mathbf{T}} = \mathbf{x}_i\ \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^{\mathbf{T}}\mathbf{x}_i^{\mathbf{T}}\ . \tag{1.6}$$

The $\mathbf{t}_i$ is i^{th} row of the score matrix **T**, while $\mathbf{x}_i$ is the i^{th} row of the unfolded matrix **X**.

Λ is the diagonal matrix containing eigenvalues λ_n up to the last one retained in the PCA model:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix}. \tag{1.7}$$

From a mathematical point of view, T_i^2 represents the distance of the projection of the sample i onto the new space from the origin of the coordinate system (i.e. from the mean of multivariate samples): very high T_i^2 means that the sample is well fitted by the model, but it deviates a lot form the mean of other samples (values of variables are much larger or smaller than the ones of other samples). A sample that is not fitted appropriately by the model shows a very large Q statistic, instead, defined by Jeffy et al. (2018) and Wise et al. (2006) as

$$Q_i = \mathbf{e}_i\mathbf{e}_i^{\mathbf{T}} = \mathbf{x}_i(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathbf{T}})\mathbf{x}_i^{\mathbf{T}}\ , \tag{1.8}$$

where Q_i is the Q statistic for the sample i, $\mathbf{e}_i$ is the vector of errors for sample i (i^{th} row in the error matrix **E**), and **I** is the identity matrix. Q is a measure of the orthogonal distance of the sample from the plane of the new space, so it is an index of the amount of information of a sample that are not represented by the PCA model. A large Q is common when a fault occurs in the process causing a changing in the correlation structure of variables.

In order to classify new samples as normal or abnormal, it is necessary to establish control limits for scores, Hotelling T^2 and Q statistic. Limits for the scores are calculated according to the Student's t-distribution: considering the n^{th} PC, 1-α confidence limit for the scores on principal component n is calculated as

$$t_{n,\alpha} = \pm\sqrt{\lambda_n} t_{m-1,\alpha/2}\,, \tag{1.9}$$

where λ_n is the eigenvalue corresponding to the principal component n, m is the number of samples (rows in the unfolded matrix) and $t_{m-1,\alpha/2}$ is the probability point on the single-sided t-distribution (e.g., for 95% confidence limits $\alpha = 0.05$).

Most of the variance of the original dataset is typically captured by the first two principal components. For this study, only two-dimensional score plots will be considered and used for process monitoring.

Confidence limits for the Hotelling T^2 are calculated by Wise et al. (2006) considering the F-distribution, according to the formula:

$$T^2_{N,m,\alpha} = \frac{N(m-1)}{m-N} F_{N,m-N,\alpha}\,, \tag{1.10}$$

where N is the number of principal components retained by the model and $F_{N,m-N,\alpha}$ the (1-α) probability point of the F-distribution.

Limits for the Q statistic are calculated by Wise et al. (2006) with the formula:

$$Q_\alpha = \Theta\left[\frac{c_\alpha\sqrt{2\Theta_2 h_0^2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0(h_0-1)}{\Theta_2^2}\right]^{\frac{1}{h_0}}, \tag{1.11}$$

where

$$\Theta_i = \sum_{j=N+1}^{M} \lambda_j^i \qquad \text{for } i\text{=1,2,3} \tag{1.12}$$

and

$$h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2}\,. \tag{1.13}$$

c_α is the standard normal deviate corresponding to the 1-α upper percentile.

These approaches for confidence limits calculation are based on the assumption that samples are randomly distributed, thus scores are normally distributed: if this assumption is violated, confidence limits are not completely reliable.

1.2 Batch-wise unfolded MPCA model

A “batch-wise unfolded model” is MPCA model in which the original three-dimensional matrix is unfolded along the variable direction: if the original matrix is $\mathbf{X}_{\mathbf{3D}}(I\times J\times K)$ with I batches, J variables and K time instants, the batch-wise unfolded matrix $\mathbf{X}(I\times JK)$ is a matrix with I rows and JK columns (Nomikos and MacGregor, 1994; Camacho et al., 2009), as showed in Figure 1.2.

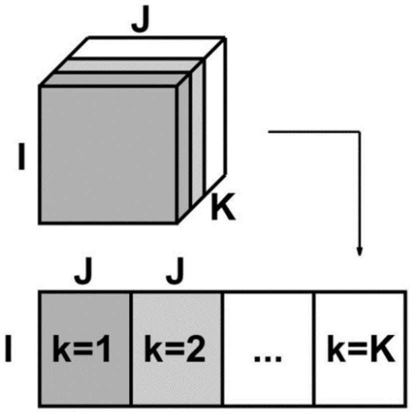


fig-1.2.jpg

Figure 1.2. *Scheme of a batch-wise unfolded matrix. Three-dimensional matrix is “sliced” vertically: each “slice” corresponds to a k time instant and contains J variables sampled for all I batches. A I×(JK) matrix results*

Each i^{th} row of the unfolded matrix contains all samples of all time instants of the i^{th} batch, while first J columns are samples of J variables at the first time instant for all I batches. In order to carry out a batch-wise unfolding, a fundamental prerequisite must be respected by the dataset: all batches must have the same length (i.e., same number of time instants sampled).

Performing a principal component analysis on the batch-wise unfolded matrix, using the RMSECV criteria described in §1.1 for the selection of the N number of principal components to retain into the model, a $I\times(JK)$ score matrix and a $(JK)\times N$ loading matrix are obtained.

As mentioned in §1.1, scores relate to rows of the unfolded matrix (in this case, each row corresponds to one batch) and it is common to represent the scores by considering only first two principal components: Figure 1.3 is an example of a score plot in which scores are multinormally distributed, so the fundamental assumption on which confidence limits for scores, Hotelling T^2 and Q statistic are calculated is respected. Batches inside the confidence ellipse are considered normal batches, while batch no.12 and batch no.16, that are out of the confidence area, are probably abnormal batches.

Despite it is very useful for a preliminary identification of a faulty batch, the score plot has a limit: since each score resumes the entire history of a batch, it may happen that if some variables are abnormal in excess and others are abnormal in defect, a compensation effect occurs and the batch score results to be inside the confidence region.

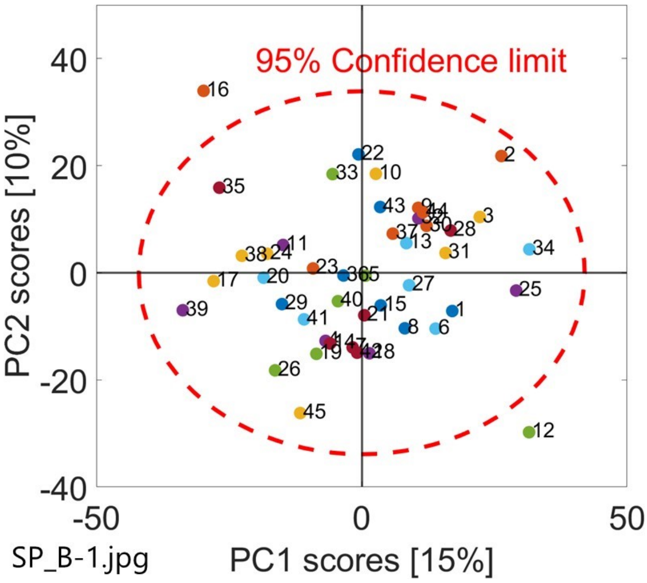


Figure 1.3. *Example of a score plot for a batch-wise unfolding model. Each dot represents a batch (e.g. dot no.26 represents batch no.26), resuming its entire process considering all variables over all time. Dashed line corresponds the 95% confidence limit for scores. Percentage in squared brackets is the variance captured by the corresponding principal component. This figure is related to the dataset described in §2.1 and used by* Nomikos and MacGregor (1994)

Monitoring the process in real time is useful to detect promptly when a fault occurs, and possibly act on the manipulated variables: at every time instant, the new batch dataset is projected into the model so that at the end of the process a trajectory of the batch is available in the score, Hotelling T^2 and Q residual plots. As mentioned at the beginning of this section, a new batch can be projected onto the model only if the number of samples (time instants) is the same of the one of calibration batches. In the case of online monitoring, only k samples for each variable are available at time k: in order to be able to project the new batch at time instant k (so during the entire process and not only at the end), the remaining K-k future samples need to be estimated. As discussed in Nomikos and MacGregor (1994), a good prediction of the score $\mathbf{t}_{new}$ of the new batch that would result if the new matrix was complete is obtained by assuming that future deviations from the mean of calibration batches remain constant for the rest of the process and equal to the ones of the last observation k. New data projection is obtained in the following way:

1. The new matrix available up to time instant k is scaled on the calibration matrix (truncated at time instant k): mean of calibration matrix is subtracted to the new matrix, which is then divided by the standard deviation of the calibration matrix;

2. The last sample of the scaled matrix is repeated *K*-*k* times, until filling the new matrix $\mathbf{X}_{new}$;
3. Score of the new batch, Hotelling T^2 and Q residual are calculated with equations (1.3), (1.6), and (1.8), putting $\mathbf{X} = \mathbf{X}_{new}$:

$$\mathbf{t}_{new} = \mathbf{X}_{new}\mathbf{P}\,, \tag{1.14}$$

$$T^2_{new} = \mathbf{t}_{new}\mathbf{\Lambda}^{-1}\mathbf{t}^{\mathbf{T}}_{new} = \mathbf{X}_{new}\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^{\mathbf{T}}\mathbf{X}^{\mathbf{T}}_{new}\,, \tag{1.15}$$

$$Q_{new} = \mathbf{e}_{new}\mathbf{e}^{\mathbf{T}}_{new} = \mathbf{X}_{new}(\mathbf{I}-\mathbf{P}\mathbf{P}^{\mathbf{T}})\mathbf{X}^{\mathbf{T}}_{new}\,, \tag{1.16}$$

where $\mathbf{t}_{new}$, T^2_{new} and Q_{new} are the score, the Hotelling T^2 and Q residual of the new batch.

In this way the evolution of the new batch is represented in the model.

As mentioned in §1.1, the Hotelling T^2 calculated with the (1.6) is useful to recognise outlier batches: a batch with a high T^2 is a batch which operating conditions are far from the mean of other batches (see batch no.12 and batch no.16 in Figure 1.4).

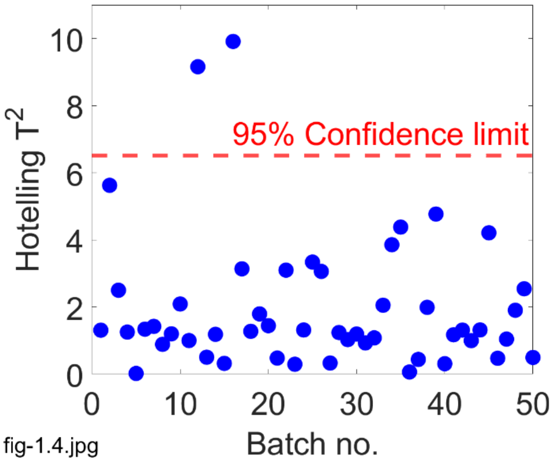


Figure 1.4. *Example of a Hotelling T^2 plot. Each dot represents a batch. The 95% confidence limit is reported. This figure is related to the dataset described in §2.1 and used by* Nomikos and MacGregor (1994)

An example of a Hotelling T^2 plot is reported in Figure 1.4: similarly to the score plot, batches represented by points that are below the confidence limit are considered like normal batches, while batches with a T^2 over the limit may be in faulty conditions. In this case the assumption of random distribution on which the calculation of limits is based is verified, and the confidence limit reported can be considered reliable for a primarily batch classification (i.e., to say if a batch is normal or not). As mentioned in §1.1, also the Q residuals calculated with (1.8) are useful to detect outlier batches: an example of Q residual plot is reported in Figure 1.5. The Q residual corresponds to the orthogonal distance between data and the reduced space, and can be interpreted like the part of data not represented by the model: a batch with a large Q is not fitted well by the model and shows a different correlation structure between variables, which can be due to a fault in the process. Since residuals are calculated over rows, the number of points appearing in the plot is equal to the number of batches in the case of a batch-wise unfolded matrix. Like in the case of the Hotelling T^2 plot, also in this one the assumption of randomly distributed points (Q residuals) is respected, and the limit can be considered reliable for an appropriate calibration of the PCA model.

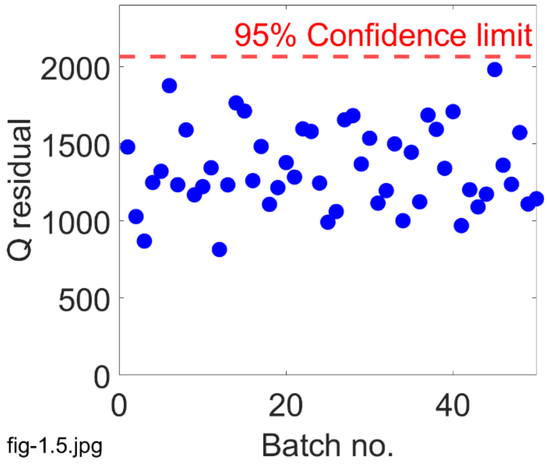


Figure 1.5. *Example of a Q residual plot. All batches result to be inside the 95% confidence region, whose limit is identified by the dashed line. This figure is related to the dataset described in §2.1 and used by* Nomikos and MacGregor (1994)

The fault diagnostics in the case of a batch-wise unfolding model can be done by analysing the contribution plots for both the Hotelling T^2 and the Q residual. The t contribution quantifies the contribution of each variable at each time instant to a batch score $\mathbf{t}_i$ and is defined by Wise et al. (2006) like

$$\mathbf{t}_{con,i} = \mathbf{t}_i \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{P}^{\mathrm{T}} = \mathbf{x}_i \mathbf{P} \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{P}^{\mathrm{T}} \,, \tag{1.17}$$

where $\mathbf{t}_{con,i}$ is the vector containing contributions of all variables at all time instants to the score of batch i. From the score contribution, the T^2 contribution is then calculated (Wise et al., 2006) like

$$\mathbf{T}^{2}_{con,i} = \mathbf{t}_{con,i} {\mathbf{t}_{con,i}}^{\mathrm{T}} \,, \tag{1.18}$$

where $\mathbf{T}^{2}_{con,i}$ is the vector containing contributions of all variables at all time instants to the Hotelling T^2 of batch i.

In order to determine which are variables responsible of the fault, some limits inside which variables contributions should lay must be defined. Limits are not the same for all variables and vary along the time, so they must be calculated for each time instant. For the (1-α) % confidence limit calculation, the basic assumption is that T^2 contributions of each variable at each time instant are normally distributed with mean and standard deviation equal to the mean and standard deviation of contributions of that variable at that time instant.

Similarly, the Q contribution quantifies the contribution of each variable, at every time instant, to the total Q residual of a sample (batch). The Q contribution for a batch i corresponds to the i^{th} row of the error array **E,** according to Wise et al. (2006):

$$\mathbf{Q}_{con,i} = \mathbf{e}_i \, . \tag{1.19}$$

Differently from the T^2 contributions, the Q contributions retain the sign of the deviation. Also in this case, limit calculation is done for each variable at each time instant and based on the assumption of normal distribution of errors, with mean and standard deviation equal to the mean and standard deviations of the error related to a variable at a specific time instant.

As mentioned in §1.1, scores in the score plot represent batches considering all their variables along the entire process, so some compensations phenomena may occur and a score could be inside confidence ellipse in the score plot also if the batch is abnormal, especially if the model is calibrated with non-random batches (as discussed in §2.2). For this reason, the Hotelling T^2 and the Q residual plots should always be checked to avoid missing fault detection, then contribution plots should be analysed to identify the cause of the abnormality and its magnitude.

Considering the on-line monitoring in the Hotelling T^2 plot and Q residual plots, in both cases the alarm is set to start after 3 consecutive points out of confidence limits for all cases reported in §4.

1.3 Assumption-free model

The assumption-free model is a variable-wise unfolding-based model proposed by Westad et al. (2015). The models developed in this thesis are an attempt to reproduce it; however, since all modelling steps are not described in detail in the paper, some assumptions and modelling decisions have been necessary. Differently from the batch-wise model, the variable-wise one does not require a dataset containing batches with the same number of samples: this is a great advantage considering that it is very common to have different batch durations to obtain the same product. The objective of this approach is to model a trajectory of a normal process in the score plot using a dataset of batches in normal operating conditions: monitoring of a new batch is made comparing its trajectory with the normal one and its confidence limits. The procedure followed to develop the assumption-free model is described in the flowchart of Figure 1.6.

First of all, the three-dimensional matrix is unfolded while preserving the dimension of variables (usually data are already available in a variable-wise form and unfolding is not necessary; moreover, in real industry processes usually have different durations and a three-dimensional matrix is not available), then it is autoscaled and a principal component analysis (PCA) is performed as described in §1.1, using the RMSECV criteria for the selection of the number of principal components. A grid search algorithm is used to model the trajectory of the process: it considers different grid resolutions and selects the one that gives the highest number of grid elements (i.e., a trajectory with the highest number of points). For each grid element, the overall mean of all samples and the mean for each batch are calculated: the first one is used for trajectory modelling, while the second one is used for the calculation of the confidence limits around trajectory. According to Westad et al. (2015), all scores must be included into grid elements, so all of the scores must be used for trajectory modelling. Overall means are interpolated to draw the trajectory, while batch means are projected into trajectory and their distance in the model space is estimated. The standard deviation of distances is calculated and limits are plotted following the direction of the trajectory, avoiding crossing. For each grid element *Q* residuals are calculated and a limit for each grid element is calculated. Methods and assumptions for limits calculation are not provided by the author.

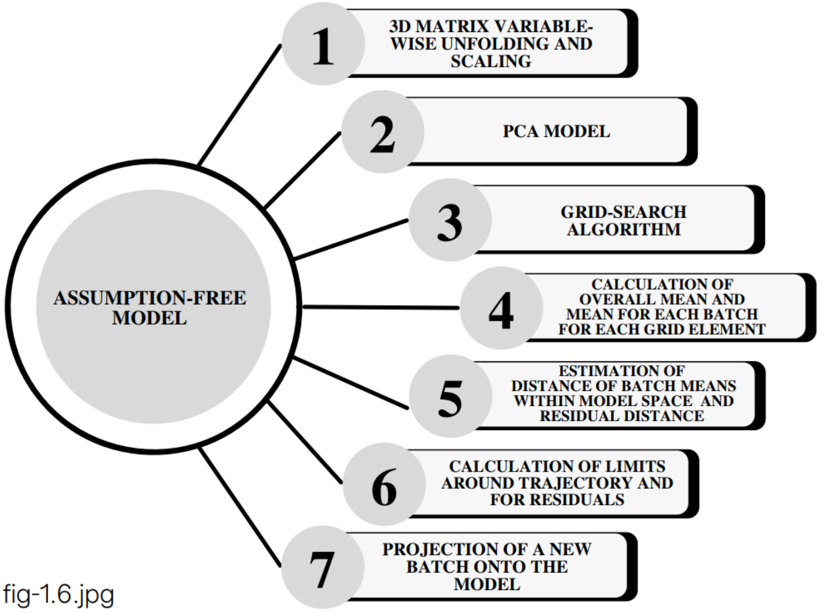

ASSUMPTION-FREE MODEL
1
3D MATRIX VARIABLE-WISE UNFOLDING AND SCALING
2
PCA MODEL
3
GRID-SEARCH ALGORITHM
4
CALCULATION OF OVERALL MEAN AND MEAN FOR EACH BATCH FOR EACH GRID ELEMENT
5
ESTIMATION OF DISTANCE OF BATCH MEANS WITHIN MODEL SPACE AND RESIDUAL DISTANCE
6
CALCULATION OF LIMITS AROUND TRAJECTORY AND FOR RESIDUALS
7
PROJECTION OF A NEW BATCH ONTO THE MODEL
fig-1.6.jpg

Figure 1.6. *Procedure for assumption-free model development, according to Westad et al. (2015)*

In the end, a new batch can be projected onto the model for fault detection and diagnosis: the distance between scores of the new batch data and trajectory and *Q* residuals are calculated. The state of the new process (relative time) is estimated according to the relative position of new scores with respect to trajectory modelled. The procedure followed to develop the assumption-free model and to implement the grid-search algorithm is reported in §3.

Chapter 2

Available datasets

To test the models developed, datasets of different batch processes have been considered. Experimental and simulated data found in the literature (sources will be reported for each dataset) have been reorganized in order to have all datasets with a similar structure. Unfortunately, not all information are available for each dataset: in some cases, the units of measure are unknown. A calibration dataset and a validation one with normal and faulty batches are provided for each process. Table 2.1 summarizes available datasets: 2 of them (dataset no.1 and dataset no.3) are simulated datasets obtained though mathematical models, while the other 3 contain real industrial data. Not all datasets contain batches with the same number of samples: dataset no.3 and dataset no.5 contain batches with different lengths, which means that an alignment procedure is needed before carrying out a batch-wise unfolding MPCA analysis.

Table 2.1. *Available datasets summary*

Dataset no.	Description	Experimental/ simulated	Equal no. of samples for all batches
1	SBR polymerization	Simulated	Yes
2	Industrial batch polymerization	Experimental	Yes
3	*Saccharomyces Cerevisiae* production	Simulated	No
4	Baker's yeast production	Experimental	Yes
5	Herbicide production	Experimental	No

More information about all datasets, such as the number of calibration batches available, the number and the description of variables sampled, and the number of samples, are reported in the paragraph corresponding to each single dataset.

2.1 Dataset 1 – Polymerization of a styrene-butadiene rubber

The dataset is related to a simulation of the semibatch polymerization of styrene-butadiene rubber (SBR) and has been tested firstly by Nomikos and MacGregor (1994) using a batch-wise unfolding model to discriminate normal and faulty batches after process completion.

2.1.1 Process description

According to the model developed by Broadhead et al. (1985), before starting the process the reactor is charged with all raw materials necessary to obtain the desired product: SBR particles, an initiator (S_2O_8), a chain transfer agent (aliphatic mercaptan), an emulsifier (fatty acid soap), water and a small quantity of styrene and butadiene monomers, while more of these monomers will be fed to the reactor at constant rate until the end of the process. The jacked-reactor is assumed to be perfectly mixed with a cylindrical geometry. The temperature inside is kept under control manipulating the cooling water flowrate to the jacket. Steady state concentration of the initiator, located only in the water phase, is assumed. Other reactions, not involved in radical initiation, can be considered negligible. The reaction starts with the decomposition of S_2O_8 into radicals, according to reaction reported in Broadhead (1984):

$$\mathrm{S_2O_8} \rightarrow \ 2\,\mathrm{SO_4^-}\cdot, \tag{2.1}$$

$$\mathrm{SO_4^-}\cdot\ + \mathrm{M}\ \rightarrow\ \mathrm{SO_4^-} + \mathrm{M}\cdot, \tag{2.2}$$

where M can be either styrene (S) or butadiene (B).

With radical monomers the propagation phase begins: double bonds in position cis,1-4 and trans,1-4 are assumed to have equal reactivity, while bonds 1,2 are the most reactive ones. Diffusion-controlled propagation need to be taken into account only in the case of very high proportions of styrene: this is not the case because styrene and butadiene flowrates are equal. Propagation can occur with different combinations:

$$\sim\mathrm{S}\cdot\ + \mathrm{S}\ \rightarrow\ \sim\mathrm{SS}\cdot, \tag{2.3}$$

$$\sim\mathrm{S}\cdot\ + \mathrm{B}\ \rightarrow\ \sim\mathrm{SB}\cdot, \tag{2.4}$$

$$\sim\mathrm{B}\cdot\ + \mathrm{S}\ \rightarrow\ \sim\mathrm{BS}\cdot, \tag{2.5}$$

$$\sim\mathrm{B}\cdot\ + \mathrm{B}\ \rightarrow \sim\mathrm{BB}\cdot, \tag{2.6}$$

Radical termination, instantaneous for small particles, is assumed to occur only in the polymer phase because of chain transfer to monomer, polymer or modifier (chain transfer agent).

A noise has been added to the initial charge purity and butadiene flowrate. Additional measurement noise has been introduced in the feed's temperature measurements. How the noise has been introduced is not explained in Nomikos and MacGregor (1994), however it could be reasonable to think that it consists in a random numerical noise (random number) added during the numerical implementation of the model.

2.1.2 Calibration dataset

As summarized in Table 2.2, the original calibration dataset contains normal batches with equal number of samples, so no alignment of batches is necessary. All 45 simulated batches have a duration of 1000 min, corresponding to 200 samples. Every 5 min 9 variables are measured, as reported in Table 2.3: the styrene and butadiene flowrates, the rubber density and the temperature of the feed, the reactor, the cooling water and the jacket of the reactor.

The total conversion and the instantaneous net rate of energy released are estimated though energy balance around the reactor.

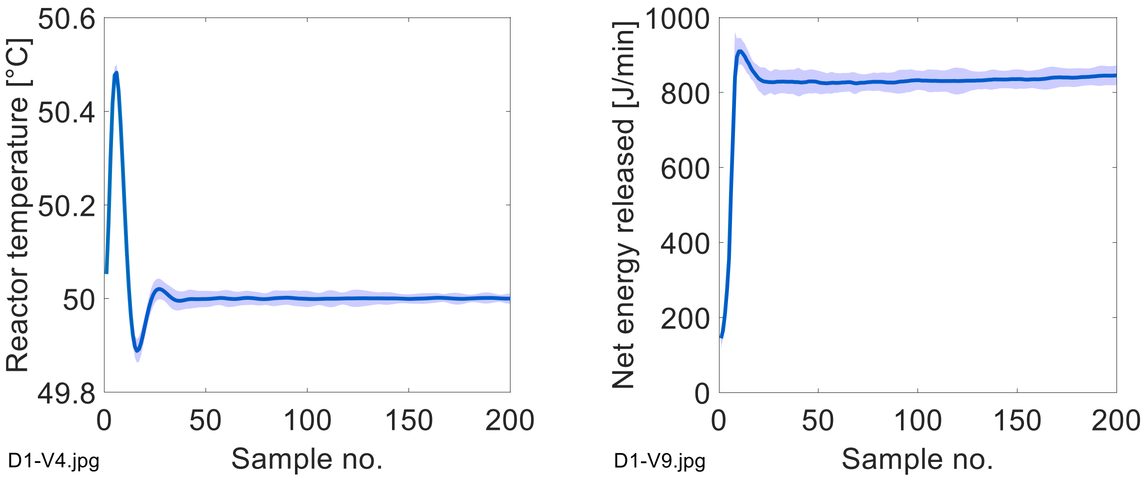
Table 2.2. *Dataset 1: Calibration data summary*

Experimental/ simulated	No. of batches	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables	3D matrix dimensions
Simulated	45	1000 min	Yes	200	9	50×9×200

Table 2.3. *Dataset 1: Measured and estimated variables*

Variable no.	Type	Description	Units
1	Measured	Styrene flowrate	
2	Measured	Butadiene flowrate	
3	Measured	Feed temperature	°C
4	Measured	Reactor temperature	°C
5	Measured	Cooling water temperature	°C
6	Measured	Reactor jacket temperature	°C
7	Measured	Latex density in the reactor	g/L
8	Estimated	Total conversion	-
9	Estimated	Instantaneous rate of energy	J/min

The total conversion is dimensionless and varies between 0 and 1. In this case the maximum conversion reached is lower than 0.7, as showed in the figure D1-V8.jpg reported in §A.1.1.



(a) (b)

Figure 2.1. *Mean profiles of (a) temperature and (b) net energy released along the process duration. The shaded area corresponds to variability across batches*

As shown in Figure 2.1, the reactor temperature profile has a peak at the beginning of the process, indicating a very fast dynamics of the reaction; however it tends to stabilize very quickly remaining constants for the rest of the process duration, except for some fluctuations. The net energy released increases rapidly at the beginning of the process, when the polymerization reaction rate is very fast due to high concentrations monomers; then, it stabilizes at about 800 J/min and remains constant for the rest of the process duration, except for some fluctuations. Profiles of other variables are reported in Appendix 1 at §A.1.1.

2.1.3 Validation dataset

The validation dataset available, described in Table 2.4, includes 8 batches: 6 in normal operating conditions and 2 in abnormal conditions. All batches have the same duration of 1000 min, which corresponds to the one of calibration batches. Also for the validation datasets, 9 variables are measured at 209 time instants (samples).

Table 2.4. *Dataset 1: Validation dataset summary*

Experimental/ simulated	No. of batches	Batches type	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables
Simulated	8	6 normal 2 faulty	1000 min	Yes	209	9

Table 2.5. *Dataset 1: Validation batches characteristics*

Batch no.	Type	Fault time	Fault cause
1-5, 53	Normal		
99	Faulty	Half-way of the process	Contamination in butadiene feed
106	Faulty	Beginning of the process	Contamination in butadiene feed

As reported in Table 2.5, the fault batch no.106 consists in a contamination in the butadiene feed at the beginning of the process, while batch no.99 presents the same type of fault halfway through the process.

2.2 Dataset 2 – Industrial batch polymerization

This dataset is a collection of real industrial data of a polymerization process carried out in a DuPont batch reactor, as reported by Nomikos and MacGregor (1995). In this case, units of measure of the variables and a detailed description of the process (including also reactions and raw materials) are missing, to protect data confidentiality.

2.2.1 Process description

The process is carried out in two stages, each one lasting approximatively 1 h, with reactants loaded into the reactor at the beginning of the first stage. The first part of the process consists in the removal of the solvent in which raw materials are initially dissolved to be charged into the reactor, through a vigorous vaporization without the need of stirring. Reaction is then completed in the second stage, at the end of which the final polymer product is obtained and can be discharged from the vessel. In order to keep the pressure and temperature profiles under control for all the reaction duration, the flows of the heating/cooling medium are adjusted during the entire process.

2.2.2 Calibration dataset

As summarized in Table 2.6, the experimental (industrial) dataset available is already aligned: all calibration and validation batches have the same duration (2 h) and number of samples (100 samples) for each of the 10 variables measured, reported in Table 2.7: 3 temperatures, 3 pressures, 2 temperatures of the heating/cooling medium and 2 flowrates.

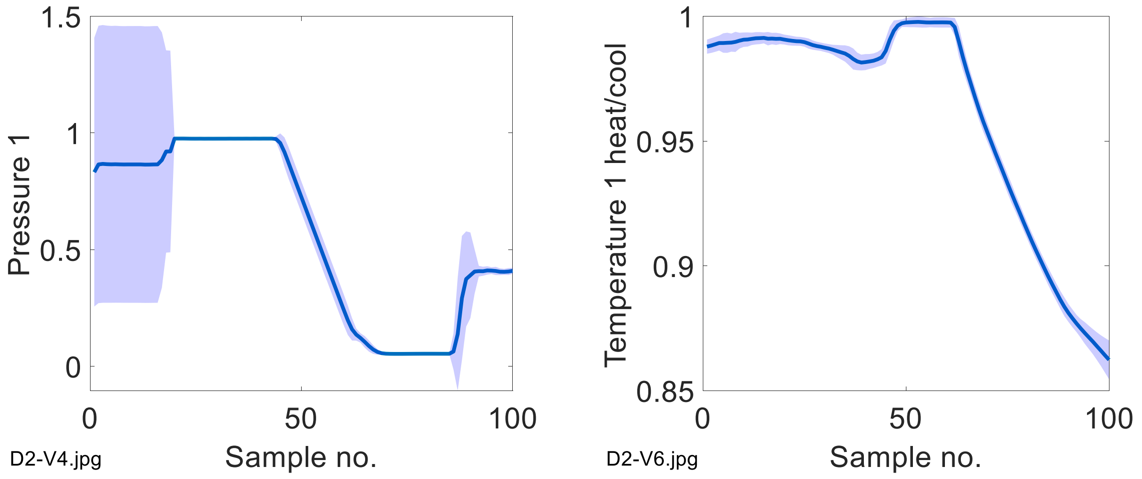
Table 2.6. *Dataset 2: Calibration dataset summary*

Experimental/ simulated	No. of batches	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables	3D matrix dimensions
Experimental	50	2 h	Yes	100	10	50×10×100

Table 2.7. *Dataset 2: Measured variables*

Variable no.	Description	Units
1	Temperature 1	
2	Temperature 2	
3	Temperature 3	
4	Pressure 1	
5	Flowrate 1	
6	Temperature 1 (heat/cool medium)	
7	Temperature 2 (heat/cool medium)	
8	Pressure 2	
9	Pressure 3	
10	Flowrate 2	

Figure 2.2 shows two examples of variables profiles meaned over all batches and the interval of variation (coloured area) between batches.



(a) (b)

Figure 2.2. *Mean profiles of (a) pressure 1 and (b) temperature 1 in the heating/cooling system along the process duration. The shaded area corresponds to variability across batches*

Pressure 1 remains constant for almost half of the process, then it decreases rapidly and increases again towards the end. Temperature 1 of the heating/cooling medium remains almost constant until half of the reaction, then it decreases rapidly. Profiles of other variables are reported in Appendix 1 at §A.1.2.

2.2.3 Validation dataset

The validation dataset available, described in Table 2.8, includes 4 normal batches and only 1 faulty batch. The 5-batch validation dataset is created with the batch of the original dataset indicated as faulty and 4 normal batches randomly selected from the ones in normal operating conditions. All batches have the same duration (2 h) of the calibration dataset. The same 10 variables of Table 2.7 are measured 100 times.

Table 2.8. *Dataset 2: Validation dataset summary*

Experimental/ simulated	No. of batches	Batches type	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables
Experimental	5	4 normal 1 faulty	2 h	Yes	100	10

Table 2.9. *Dataset 2: Validation batches characteristics*

Batch no.	Type	Fault time	Fault cause
2	Normal		
10	Normal		
15	Normal		
39	Normal		
49	Faulty	Beginning of the process	

As reported in Table 2.9, in batch no.49 the fault occurs at the beginning of the process (i.e. at the first time instant), but its cause is unknown. Having only one faulty batch is quite limiting: more faulty batches will be useful to calibrate and test models in a more appropriate way.

2.3 Dataset 3 – *Saccharomyces Cerevisiae* production

This dataset is included in the MVBatch Toolbox, freely available for Matlab at https://github.com/jogonmar/MVBatch/releases, as reported in González-Martínez et al. (2018). The simulated process is the fermentation of the Saccharomyces Cerevisiae cultivation, under normal and abnormal operating conditions, whose model has been developed by Lei et al. (2001). Both calibration and validation dataset contain batches with different number of samples. Variables units of measure are unknown.

2.3.1 Process description

The fermentation process consists in 4 phases: a lag phase, two phases of exponential growth, and a stationary final phase. The first phase, in which the yeast acclimates to the heterogeneous media for a couple of hours, is followed by two exponential growth phases whose reactions are schematized in Figure 2.3. In the first growth phase, glucose fed to the reactor is metabolized into pyruvate through *r_1* (catabolic reaction) and biomass through *r_7* (anabolic reaction).

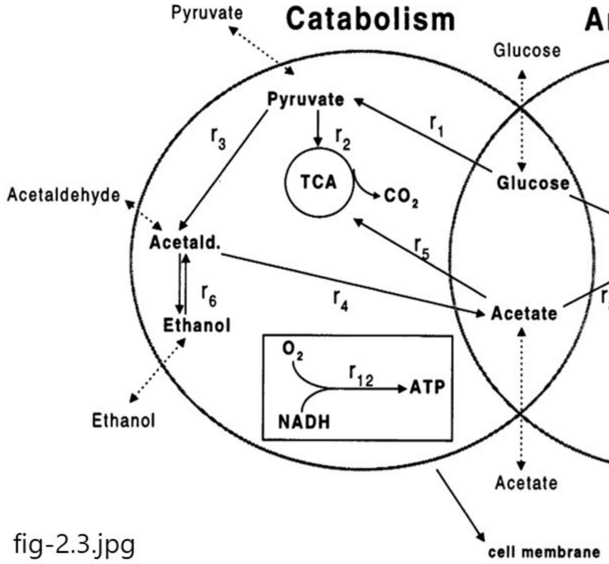


fig-2.3.jpg

Figure 2.3. *Scheme of the catabolic and anabolic reactions occurring inside the reactor. Glucose and acetate lead to biomass growth (anabolic path), however they participate also to catabolic reactions that produce ethanol and carbon dioxide. This figure is from* Lei et al. (2001)

At low glucose flowrate, pyruvate is completely converted into TCA (tricarboxylic acid) through *r_2* and consequently into CO_2, but when the flowrate increases, pyruvate dehydrogenase saturates and pyruvate is consumed in *r_3* leading to acetaldehyde formation. Acetaldehyde is then consumed by the main reaction *r_4*, increasing the acetate concentration in the reactor; however at higher concentration of acetaldehyde, the acetaldehyde dehydrogenase saturates and the *r_6* equilibria side reaction occurs leading to ethanol formation. When all glucose is consumed and it can’t be used as nutrient by the growing cell, ethanol is used as substrate in the second exponential growth: it is converted into acetate, which can be used in the catabolic reaction *r_5*, leading to CO_2 formation, or in the anabolic reaction *r_8*, leading to biomass formation. A perfect abiotic system is assumed.

2.3.2 Calibration dataset

The available calibration dataset contains batches with a different number of samples: in order to use the dataset with both MPCA models (assumption-free model and the one in batch-wise unfolding), a multy-sinchro alignment has been performed though the MVBatch Toolbox, as suggested by González-Martínez et al. (2018). The dataset is summarized in Table 2.10 and includes 40 simulated batches with a duration of about 35 h, varying from batch to batch. The number of samples is the same (209 samples) for all batches only after alignment.

10 variables are measured and described in Table 2.11: concentrations of glucose, pyruvate, acetaldehyde, acetate, ethanol, biomass, active cell material and acetaldehyde dehydrogenase, the specific oxygen uptake rate and the specific CO_2 evolution rate.

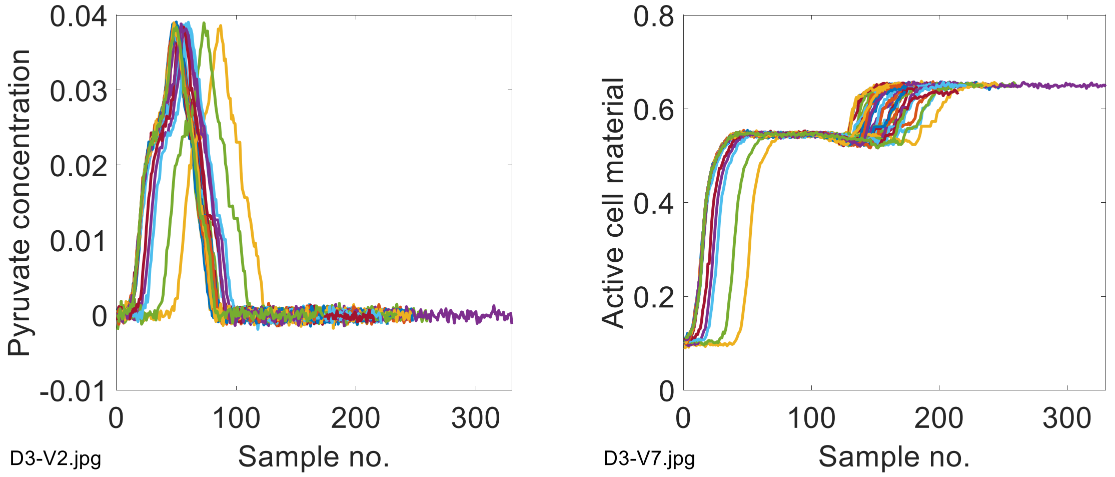
Table 2.10. *Dataset 3: Calibration dataset summary*

Experimental/ simulated	No. of batches	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables	3D matrix dimensions
Simulated	40	≃35 h	No	209	10	40×10×209

Table 2.11. *Dataset 3: Measured variables*

Variable no.	Description	Units
1	Glucose concentration	
2	Pyruvate concentration	
3	Acetaldehyde concentration	
4	Acetate concentration	
5	Ethanol concentration	
6	Biomass concentration	
7	Active cell material	
8	Acetaldehyde dehydrogenase	
9	Specific oxygen uptake rate	
10	Specific CO_2 evolution rate	

Only substances concentrations are measured, while any information related to temperature and pressure is not provided: they do not appear between measured variables.



(a)

(b)

Figure 2.4. *Mean profiles of (a) pyruvate concentration and (b) active cell material along the process duration. The shaded area corresponds to variability across batches*

Two examples of variables profiles along batch time are shown in Figure 2.4: during the first exponential growth of cell material, pyruvate is produced rapidly, then after about one-fourth of the process cells stop growing and pyruvate is rapidly consumed. In the last part of the reaction cells are subjected to another exponential growth, while pyruvate concentration fluctuates around zero. In Appendix 1 at §A.1.3 are reported figures with profiles of all variables.

2.3.3 Validation dataset

A 55-batch validation dataset is already available in the toolbox. As reported in Table 2.12, the first 25 sampled batches are in normal operating conditions, while the remaining 30 batches are faulty. The batch duration and number of samples (209 samples) are the same for all batches (and equal to the ones of calibration batches) only after alignment. The same 10 variables of Table 2.11 are measured for each batch.

Table 2.12. *Dataset 3: Validation dataset summary*

Experimental/ simulated	No. of batches	Batches type	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables
Simulated	55	25 normal 30 faulty	≃35 h	No	209	10

Table 2.13. *Dataset 3: Validation batches characteristics*

Batch no.	Type	Fault time	Fault cause
1-25	Normal	N/A	
26-55	Faulty		Glucose uptake system, ethanol formation, biomass concentration sensor

In Table 2.13, three types of faults are reported for faulty batches: the first one is related to the glucose uptake system and the glycolytic pathway, the second one is due to ethanol formation from acetaldehyde, while the third one is a fault of the biomass concentration sensor. Times at which faults occur are not provided, neither the exact identity of fault (among the three possible causes) for every faulty batch.

2.4 Dataset 4 – Baker's yeast production

This dataset has been provided by Jästbolaget AB (Sweden) and consists in an industrial dataset regarding baker's yeast batch production. Two examples of MPCA monitoring approach using this dataset are discussed in Eriksson et al. (2013). Variables units of measure for this dataset are unknown.

2.4.1 Process description

The dataset is related to the last of the five phases that constitute yeast's production process. The process, briefly described in George et al. (1998), starts when ammonia and a mixture mainly constituted by sucrose are fed to the reactor as yeast's carbon source. Molasses flowrate is increased during the first part of the process, causing an exponential biomass growth, and it is set constant in a second moment in order to avoid cooling limitations and an overflow metabolism that would results in an excessive ethanol production. Yeast invertase hydrolyses molasses into a mixture of ethanol, glucose and fructose (greatest part): glucose and fructose are consumed first, then when their concentrations decrease ethanol is consumed and a higher yield is reached in the end. During the final stage of the process, ammonia and molasses flowrates are reduced to zero. At the end of the process, cells are harvested and dewatered, then yeast is packed. The process is carried out in sugar limitation: when the sugar exceeds a critical value (critical concentration), cells are not able to fully consume the entire amount of sugar provided, which starts to be converted into ethanol. Also if ethanol is then used for biomass growth, the total biomass yield from glucose is lower if combustion of glucose passes through ethanol. More details about process reactions are not provided by authors, however the process is similar to the one of dataset §2.3: what differs is that in this case the process includes only the last part of yeast production and dataset contains real industrial data (not simulated).

2.4.2 Calibration dataset

As reported in Table 2.14, 16 calibration batches available have all the same duration of 14 h, corresponding to 83 time instants sampled for each batch: no dataset alignment is needed in this case. 7 variables are measured, as described in Table 2.15: the ethanol content, the temperature, the molasses, ammonia and air flowrates entering the reactor, the tank level and the pH.

Table 2.14. *Dataset 4: Calibration dataset summary*

Experimental/ simulated	No. of batches	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables	3D matrix dimensions
Experimental	16	14 h	Yes	83	7	16×7×83

Table 2.15. *Dataset 4: Measured variables*

Variable no.	Description	Units
1	Ethanol content	
2	Temperature	
3	Molasses flowrate	
4	NH_3 flowrate	
5	Air flowrate	
6	Tank level	
7	pH	

Variability between batches is very high due to variable fluctuations and changes in their relationships during the batch process: this phenomenon is particularly present in variables like the ethanol content, the reactor temperature, the ammonia flowrate and the pH.

All mean profiles of variables are reported in Appendix 1 at §A.1.4, where a wide coloured area represents large variability between batches profiles.

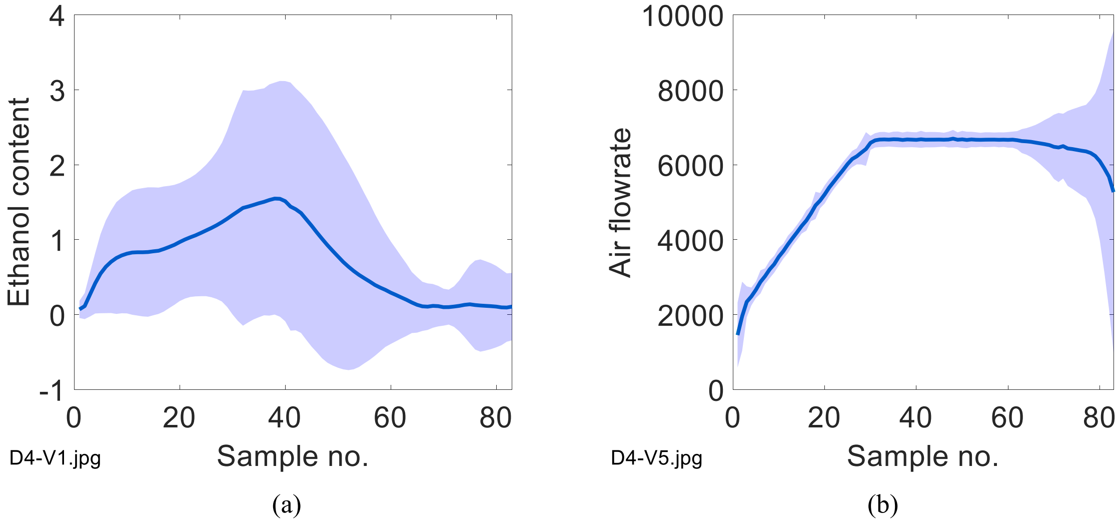


Figure 2.5. *Mean profiles of (a) ethanol content and (b) air flowrate along the process duration. The shaded area corresponds to variability across batches*

As shown in Figure 2.5, ethanol is produced in the first 30 samples and then consumed in the second part of the process. The air flowrate is constantly increased at the beginning of the process, then it is kept constant for about 30 samples before being decreased towards the end of the procedure.

2.4.3 Validation dataset

For validation, 17 batches are available each one containing 83 samples, the same number of the one of calibration batches, as reported in Table 2.16. Similarly to the calibration dataset of Table 2.14, the number of samples is the same for each batch, so no alignment is needed to perform a batch-wise PCA analysis.

Table 2.16. *Dataset 4: Validation dataset summary*

Experimental/ simulated	No. of batches	Batches type	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables
Experimental	17	Unknown	14 h	Yes	83	7

Differently from previous datasets, in this case information about the type of validation batches (i.e., normal or faulty) are not available, and so it is also for the time at which the fault eventually occurs. This represents a limitation for the purpose of the study because it is not possible to evaluate in an appropriate way the performance of two models without knowing what the model should detect and at which time.

2.5 Dataset 5 – Herbicide production

A dataset provided by FMC corporation, as reported in the Aspen ProMV Getting Started Guide (2017), and containing data of an industrial batch dryer reactor for an herbicide production is available in Aspen ProMV, an AspenTech software, at the path C:\ProgramData\AspenTech\Aspen ProMV Desktop\Examples (both the dataset and the Getting Started Guide can be downloaded). It contains normal batches (i.e., batches whose final quality variables are on specifications) and off-specification batches, and it is available in both aligned and unaligned forms. For the purpose of the study, dataset is reorganized into two datasets: one for calibration containing normal batches and one for validation containing normal and out of specifications batches. Units of measure for all variables are not available.

2.5.1 Process description

As described by García-Muñoz et al. (2003), the purpose of the process is to dry an herbicide product evaporating the solvent present in the wet cake and collecting it in a tank. A scheme of the batch process is reported in Figure 2.6, imported from Aspen ProMV Getting Started Guide (2017) . Reactions are not available. The total real duration of the process is unknown.

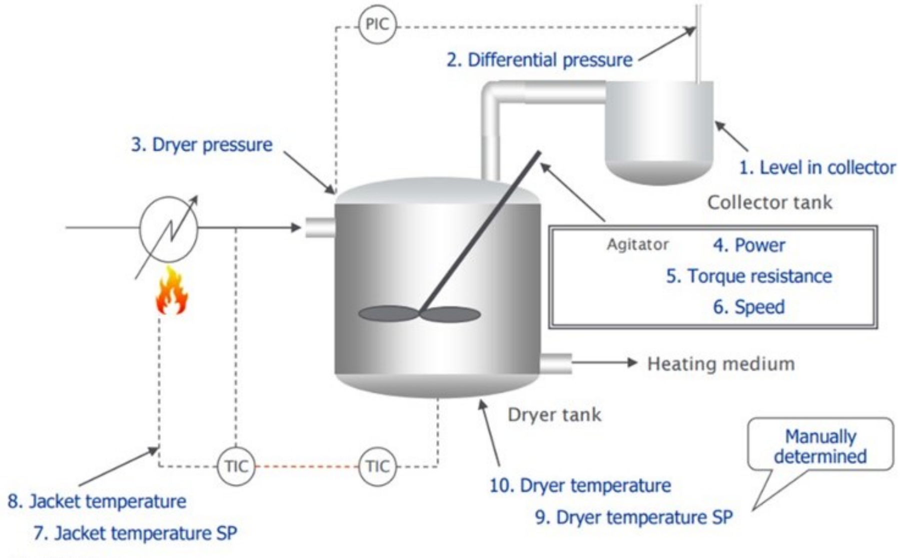


Figure 2.6. *Scheme of the drying process for an herbicide production in an FMC Corporation plant. The stirred reactor is heated by hot water flowing into the jacket, while evaporating solvent is collected in a separate tank. The agitator speed and the temperatures set-points are adjusted according to properties of the cake. (*Aspen ProMV Getting Started Guide, 2017*)*

The operation starts with the charging of the wet cake, whose volume can vary from batch to batch, into the reactor: while the tank level is measured, the amount of solvent present in the cake is unknown. At the beginning of the process the agitator runs at low speed while the hot water is already flowing into the jacket making the temperature inside the batch increasing. In a second moment, determined by the control system according to properties of the material inside the reactor, the agitator speed is increased rapidly and then decreased just before the temperature inside the reactor reaches its maximum. After the temperature peak, the product is cooled down; then, toward the end of the process, the agitator speed is increased for some time. All the solvent vaporized is recovered in a separated tank.

2.5.2 Calibration dataset

The calibration dataset, summarized in Table 2.17, contains 30 batches in normal operating conditions with different number of samples: an alignment process is needed to perform a batch-wise unfolding; however the aligned dataset, with batches lasting all 325 time instants, is already available in the software together with the unaligned one. For each batch, 10 variables reported in Table 2.18 are measured: the solvent collector tank level, the differential pressure in the dryer, the dryer pressure, the power provided to the agitator and its speed, the torque resistance, the set points of the jacket and dryer temperatures and the actual jacket and dryer temperatures.

Table 2.17. *Dataset 5: Calibration dataset summary*

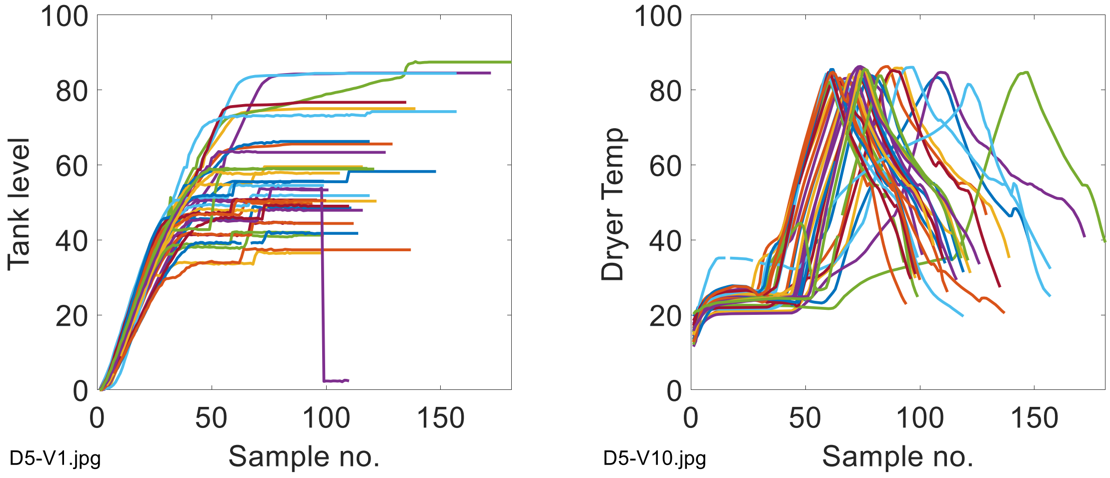
Experimental/ simulated	No. of batches	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables	3D matrix dimensions
Experimental	30	N/A	No	325	10	30×10×325

Table 2.18. *Dataset 5: Measured variables*

Variable no.	Description	Units
1	Solvent collector tank level	
2	Differential pressure	
3	Dryer pressure	
4	Power	
5	Agitator speed	
6	Torque	
7	Jacket temperature set point	
8	Jacked temperature measured	
9	Dryer temperature set point	
10	Dryer temperature measured	

The amount of solvent contained in every wet cake is not constant (as evidenced by the large variability of the level of the solvent collector tank reported in Figure 2.7), so times at which set points of temperatures and agitator speed change are not fixed from batch to batch.

All of the profiles of variables and their variability are reported in Appendix 1 at §A.1.5.



(a) (b)

Figure 2.7. *Mean profiles of (a) solvent collector tank and (b) dryer temperature along the process duration. The shaded area corresponds to variability across batches*

In Figure 2.7 it can be observed how the level of the solvent collector tank and the temperature inside the dryer change during operation. The tank level increases almost constantly for the greatest part of the process and tends to stabilize at the end: the cake inside the reactor is completely dry and no more solvent is vaporized. The temperature inside the drier increases rapidly in the last part of the process: energy is continuing to be provided to the reactor through hot water flowing into the jacket, however since there is no more solvent to be vaporized inside the reactor, the energy causes a rapid increasing in temperature.

2.5.3 Validation dataset

The validation dataset contains 41 batches, as reported in Table 2.19: 3 batches are normal, while the others are faulty, which means that their final product quality is out of specification. The number of samples for each batch is different in the case of the unaligned dataset and it is the same of the ones of calibration batches (325 samples) in the case of the aligned dataset.

Table 2.19. *Dataset 5: Validation dataset summary*

Experimental/ simulated	No. of batches	Batches type	Batch duration	Equal no. of samples for all batches	No. of samples (aligned dataset)	No. of variables
Experimental	41	3 normal 38 faulty	N/A	No	325	10

For this dataset times at which faults occur and fault causes are not known: also if the batch can be classified by the model like normal of faulty, it is not possible to state if alarms are false or they are reporting a real fault and if the real cause of fault is detected.

Chapter 3

The assumption-free model

This section contains the description of how the assumption-free model has been developed in this thesis and in particular how the grid-search algorithm has been implemented. To explain step by step the procedure, the dataset of an industrial polymerization reaction is considered (Nomikos and MacGregor, 1995) and described in §2.2.

3.1 The PCA model

Industrial data are usually collected in a variable-wise form, which consists in a matrix with a number of columns equal to the number of variables sampled, and a number of rows equal to the sum of all time instants sampled of all batches (Camacho et al., 2009). Since the number of samples can vary from batch to batch, an additional column representing time is necessary (usually, it corresponds to the first column of the matrix) to identify the beginning and the end of each batch. Specifications on how data should be arranged to be loaded into the algorithm are reported in a file attached to the algorithm Matlab files. If all batches have the same duration, data collected from batches under normal operating conditions can be arranged into a three-dimensional matrix and then unfolded as reported in Figure 3.1.

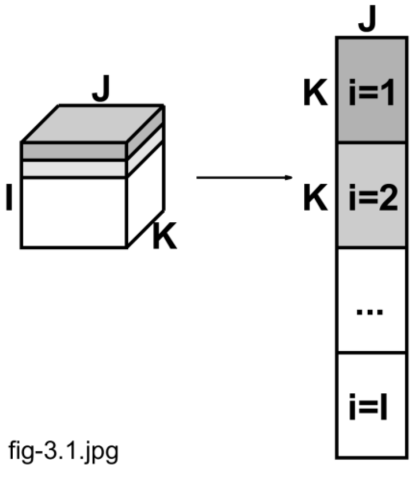


Figure 3.1. *Scheme of a variable-wise unfolded matrix. Three-dimension matrix is "sliced" horizontally: each "slice" corresponds to a i batch and contains J variables sampled for K time instants. The resulting matrix is a (KI)×J matrix*

After performing a PCA analysis on the unfolding matrix, using RMSECV criteria described in §1.1 for the selection of the number of principal components to retain into the model, a

(*KI*)×*J* score matrix and a *J*×*NPCs* loading matrix are obtained. More details on the PCA model are provided in §4.2.1, where the entire case study with validation is reported.

The loading plot is useful to capture correlations between variables considering the entire process duration. In this case, the loading plot resulting from the principal component analysis is reported in Figure 3.2.

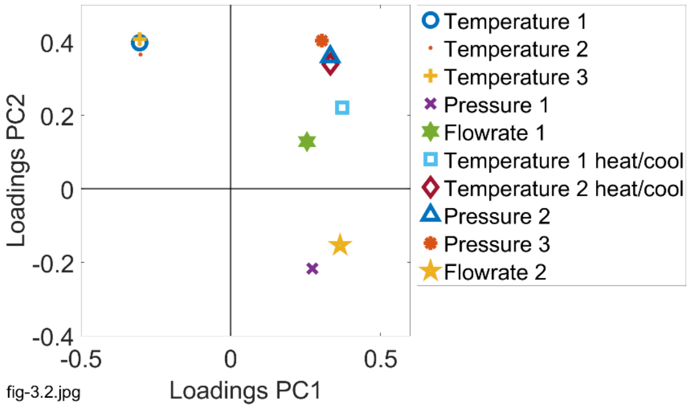


Figure 3.2. *Example of a loading plot for a variable-wise unfolding PCA. Each circle represents a variable considering the entire process duration. This figure is related to dataset no.2*

Considering the first principal component PC1, pressures, flowrates and temperatures in the heating/cooling medium are all correlated (on the right side of the plot), while they are anti-correlated to temperatures 1, 2 and 3 (on the left side of the plot). Correlation between two variables means that if a variable increases, the other increases too; on the other hand, if a variable increases and the other decreases, the situation is of anti-correlation.

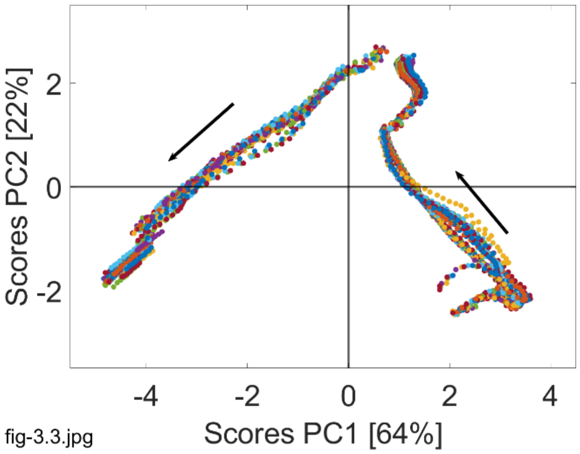


Figure 3.3. *Example of a score plot. Each score (point) represents a batch at a specific time instant. Arrows indicate directions along which the process evolves: from the south-east to the south-west area. Percentage in squared brackets is the variance captured by the corresponding principal component. The model has been calibrated using all 50 calibration batches of the dataset no.2*

In the score plot of Figure 3.3 the entire process history for all batches is reported: the assumption-free model is built on these 50 batch normal trajectories.

The score plot is at the base of this modelling approach: a grid-search algorithm is applied to it in order to capture the normal evolution of a batch process, modelling a trajectory based on some batches running under normal operating conditions.

3.2 Grid search algorithm

The grid-search algorithm allows one to model a normal trajectory of the process in the score plot. By setting 2 parameters (reported in Table 3.1), it evaluates several grid configurations over the score plot and looks for scores that are contained in each grid cell. Eventually, the optimal grid is chosen and the normal batch trajectory is modelled, assuming that the trajectory can be represented using only two principal components.

Table 3.1. *Grid-search algorithm parameters*

Parameter no.	Symbol	Description	Value	Units
1	β	Fraction of batches per cell	100	% of batches
2	γ	Total fraction of scores captured	95	% of total scores

The two parameters that need to be set before running the algorithm are: 1) the fraction of batches included in a cell for the cell to be considered valid, and 2) the total fraction of scores captured by valid cells of the grid and used for trajectory modelling. The concept of "valid cell" will be explained later.

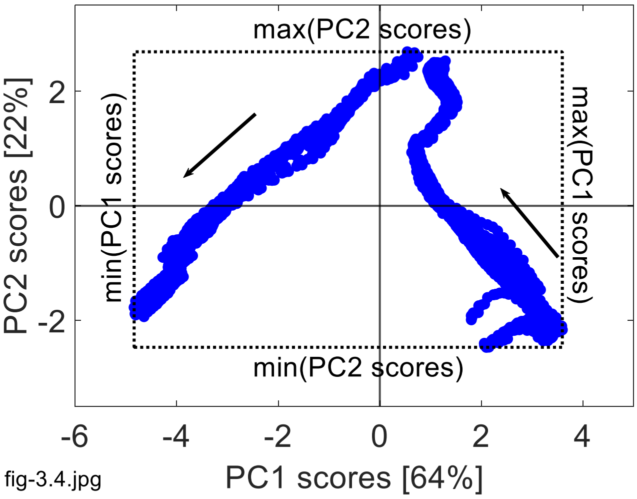


Figure 3.4. *Boundaries of grid in the score plot. The left-bound is the minimum of PC1 scores, the right-bound is the maximum of PC1 scores, the lower-bound is the minimum of PC2 scores, the upper-bound is the maximum of PC2 scores*

The left-right and lower-upper boundaries of the grid are set equal to the minimum and maximum of PC1 scores, and to the minimum and maximum of PC2 scores, respectively, as shown in Figure 3.4. Boundary values are kept constant for all grid configurations, independently of the number of cells considered: this way, all scores are always included in the overall grid space.

Considering a given dataset, all cells have the same dimension. The cell dimension is a function only of the grid resolution (overall number of cells). Once boundaries are fixed, several grid configurations are considered. A good choice is to start from a high-resolution grid (i.e., a grid with a large number of cells), and then to consider grids with a decreasing number of cells (lower resolution). The initial number of cells can be chosen a priori: the only requirement is that it is “large enough”, so that the optimal grid found by the algorithm has a resolution that is lower than the initial one. If the number of cells of the optimal grid is the same of the initial one, then the starting resolution has to be set higher. In all cases reported in this thesis (and also in this one used as example), a 15×15 initial grid is considered, meaning a grid with 15 rows and 15 columns. It should be noticed that “grid resolution” and “trajectory resolution” have two different meanings: the first one refers to the total number of cells (both valid and invalid) of the grid considered, while the second one is related to the number of points on which the trajectory is built by interpolation, and corresponds to the number of valid cells.

For each grid, not all cells are taken into account for trajectory modeling, but only those which are identified as valid. A cell is considered valid if all batches are present inside the cell with at least one score for each batch (β=100%). For a clear identification, in this thesis valid cells are denoted with a white background, and invalid cells are denoted with a grey background.

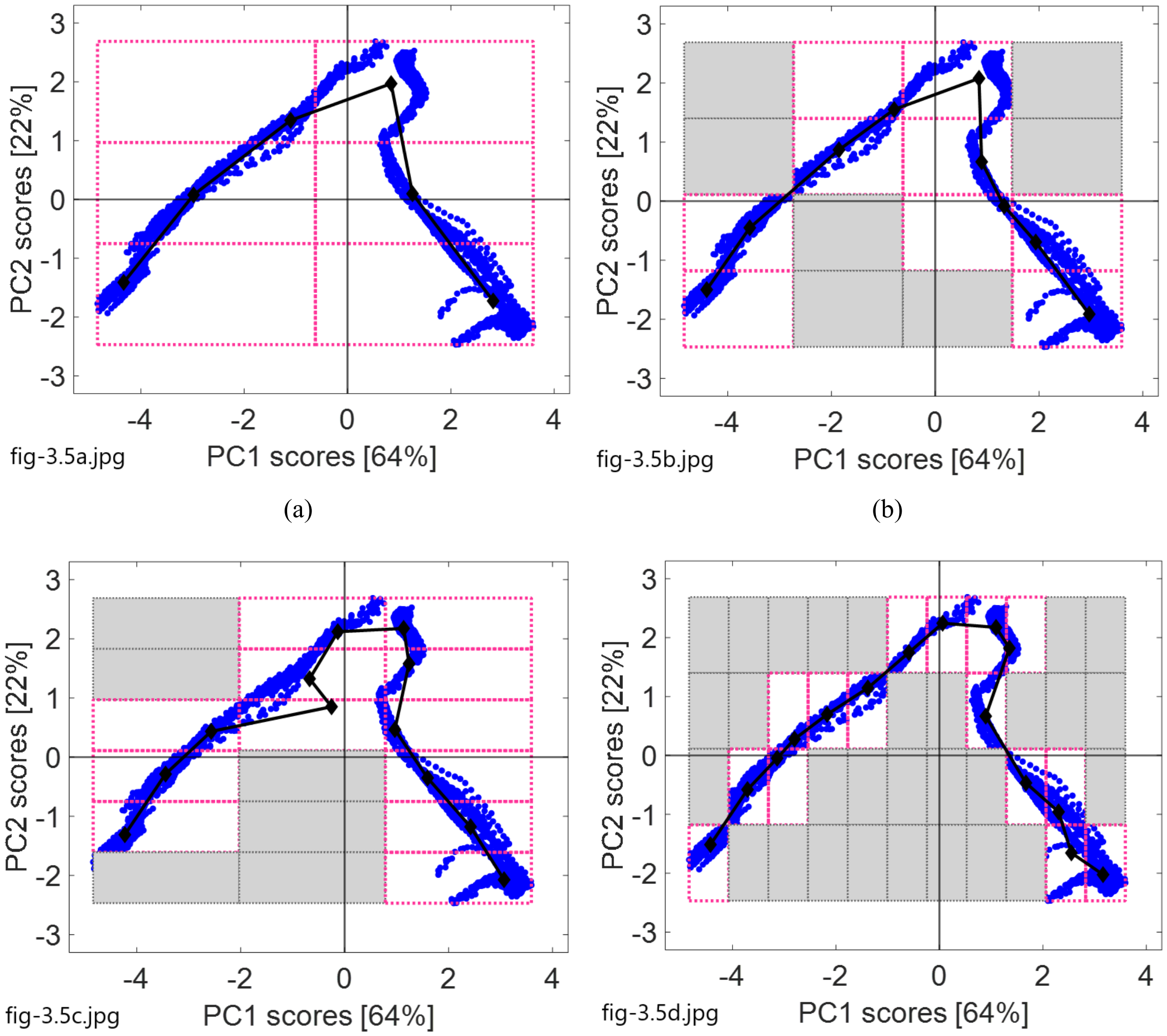
For each valid cell, the overall mean of scores $\bar{\bar{\mathbf{t}}}_{cell}$, with component $\bar{\bar{t}}_{1,cell}$ along PC1 and $\bar{\bar{t}}_{2,cell}$ along PC2, are calculated:

$$\bar{\bar{t}}_{a,cell} = \frac{1}{M}\sum_{m=1}^{M} t_{a,cell,m} \qquad (a = 1,2) \tag{3.1}$$

$$\bar{\bar{\mathbf{t}}}_{cell} = (\bar{\bar{t}}_{1,cell}, \bar{\bar{t}}_{2,cell}) \tag{3.2}$$

where *a* identifies the principal component the value of score refers to (*a*=1 for component on PC1, and *a*=2 for component on PC2), *cell* identifies the cell considered, and *M* is the total number of scores included in *cell*. The trajectory is then obtained by connecting all the means of scores calculated with the (3.2).

Some iterations of the grid-search algorithm are reported in Figure 3.5.



(a) (b)

(c) (d)

Figure 3.5. *Some iterations of the grid-search algorithm. Invalid cells are marked in grey, and the calibration scores (dots) included in them are not shown. For each valid cell, marked in white, the overall mean of scores is calculated (diamonds). All means are interpolated to draw the trajectory (black line). All grid configurations capture 95% of the calibration scores. Configuration in (a) gives a 6-points trajectory, configuration in (b) gives a 9-point trajectory, and configuration in (c) gives a 12-point trajectory. Configuration in (d) gives the trajectory with the highest resolution (15 points): this is the optimal configuration. In this example, dataset no.2 is considered*

At the end, the grid that allows to model a trajectory that best represents the normal batches is chosen among different configurations, and corresponds to grid of Figure 3.5 (d), which has a resolution of 4×11 cells (i.e., grid with 4 rows and 11 columns). The criteria for grid selection is: the grid that allows to obtain the trajectory with the largest number of means of scores $\bar{\bar{\mathbf{t}}}_{cell}$ (i.e., highest trajectory resolution) *and* which is able to include in all its valid cells at least an assigned fraction γ of all calibration scores. The performance of the algorithm in trajectory modelling decreases if the percentage of scores captured by all valid cells is much smaller than 95%: too many calibration data are lost, and the trajectory is poorly representative of the normal conditions of the process.

Despite the procedure of Westad et al. (2015) imposes that *all* scores should be included into grid elements (valid cells), setting a percentage of captured scores equal to 100% is not recommended: if the trajectory of one of the calibration batches deviates a lot from trajectories of other calibration batches (as in the case discussed at §4.5.1), to include also those scores in valid cells, the required cell dimension must be very large, and the resulting normal process trajectory would be too coarse.

3.3 Confidence limits calculation

Calculation of confidence limits is based on the optimal grid found on which the trajectory is modelled, and on the distance of the calibration scores from the trajectory. To calculate the distance between scores and the trajectory, a sufficiently large number *P* of points (e.g., 15000 points) is identified on the trajectory, as showed in Figure 3.6.

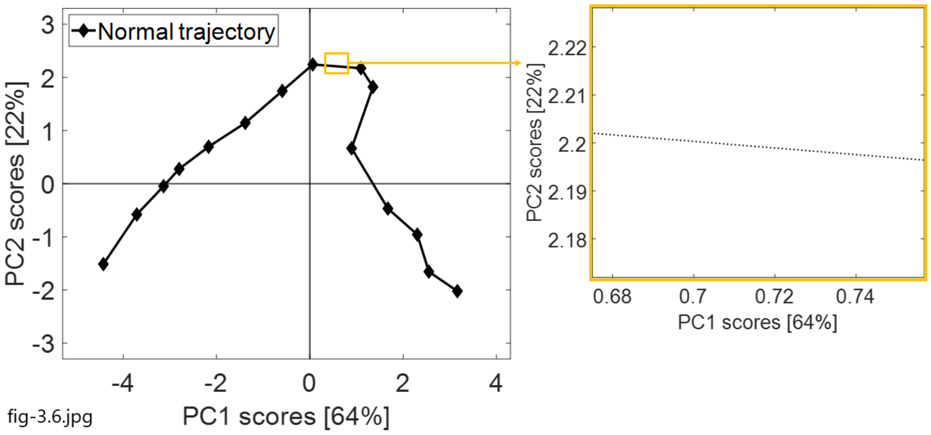


Figure 3.6. *P points identified onto the normal trajectory. In this case, the number of points identified is equal to 15000. Diamonds are overall means of scores calculated for each valid cell, while the solid line represents the normal trajectory obtained by interpolating overall means of scores. Points onto trajectory are identified by interpolation*

For each grid cell on which an overall mean of scores has been calculated, the closest point *pclosest* of the trajectory to each score belonging to the cell is found using a *k*-nearest neighbours algorithm. Then, the Euclidean distance between each calibration score *m* inside the cell and its closest point in the trajectory is calculated:

$$d_{traj,cell,m} = \sqrt{\sum_{a=1}^{2}(t_{a,cell,m} - t_{a,traj,pclosest})^2}\ , \tag{3.3}$$

where $d_{traj,cell,m}$ is the distance between the score *m* and its closest *p* point *pclosest*, $t_{a,cell,m}$ is the component *a* of score *m* of cell *cell*, and $t_{a,traj,pclosest}$ is the component *a* of the point *pclosest*. Since *pclosest* is the point in the trajectory that is the closest to the score *m,* distance

$d_{traj,cell,m}$ can be approximated as the minimum distance between score m of cell *cell* and the trajectory. An example is reported in Figure 3.7.

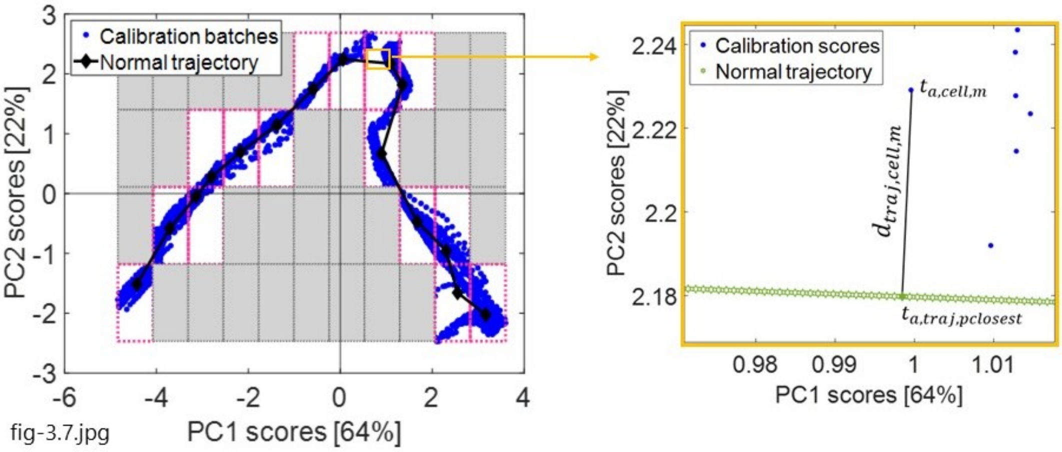


Figure 3.7. *Example of calculation of distance between a calibration score ($\mathbf{t}_{cell,m}$) and its closest point onto trajectory ($\mathbf{t}_{traj,pclosest}$) for cell no.7. This distance can be approximated like the minimum distance between $\boldsymbol{t}_{cell,m}$ and trajectory*

The distance is computed for all scores in each valid cell, while scores belonging to invalid cells are neglected.

For each cell, the distance for the (1-α)% confidence limit is determined, such that 95% of the distances $d_{traj,cell,m}$ is smaller than that distance:

$$d_{(1-\alpha)\%\,c.l.,cell} = d_{traj,cell} | (1-\alpha)\% \; d_{traj,cell,m} < d_{(1-\alpha)\%\,c.l.,cell} \,, \tag{3.4}$$

where $d_{(1-\alpha)\%\,c.l.,cell}$ is the distance of the (1-α)% confidence limit for cell *cell*, and $d_{traj,cell}$ is a distance from trajectory for cell *cell*. The histogram of Figure 3.8 shows an example of how the confidence limit is calculated.

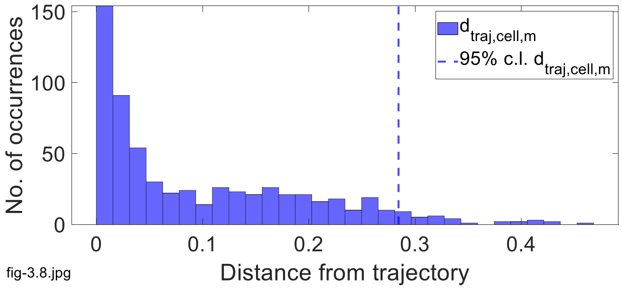


Figure 3.8. *Histogram of distances between calibration scores and trajectory for cell no.7. The distance for the 95% confidence limit is marked with dashed line: the 95% of distances is smaller than the distance for the 95% confidence limit*

For cell no.7, the number of large distances is very small (few occurrences): scores corresponding to these distances (which correspond to the 5% of the overall number of scores in the cell) are the furthest from trajectory and are left out from confidence limit (dashed line).

The approach for limits calculation suggested by Westad et al. (2015) is slightly different: it consists in the calculation of the mean of scores of each batch in each (valid) cell, and computing the distance between mean of each batch and its orthogonal projection onto trajectory. The component *a* of the mean of scores of batch *i* for the cell *cell* is calculated as

$$\bar{t}_{a,cell,i} = \frac{1}{L}\sum_{l=1}^{L} t_{a,cell,i,l} \tag{3.5}$$

$$\bar{\mathbf{t}}_{cell,i} = (\bar{t}_{1,cell,i}, \bar{t}_{2,cell,i}) \tag{3.6}$$

where $t_{a,cell,i,l}$ is the component *a* of the score *l* of batch *i* in the cell *cell*, *L* is the total number of scores of batch *i* inside cell *cell* and $\bar{\mathbf{t}}_{cell,i}$ the mean of scores of batch *i* into cell *cell*.

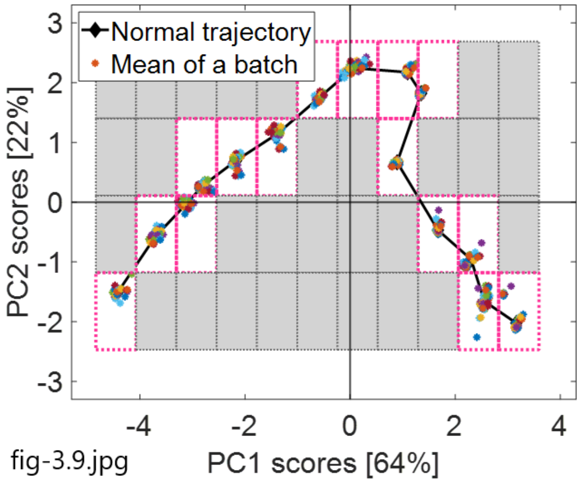


Figure 3.9. *Resulting score plot after calculation of mean of scores of each batch for each cell. Each valid cell contains 50 means (asterisks): one mean for each batch*

While in Figure 3.5 (d) all calibration scores included in valid cells are reported (4831 points), in Figure 3.9 only one point per batch (the mean of its scores) is reported in each valid cell; since the number of batches is equal to 50 and valid cells are 15, 750 means are reported in Figure 3.9. The distance between the mean of scores of batch *i* in the cell *cell* and the trajectory is calculated as

$$d_{traj,cell,i} = \sqrt{\sum_{a=1}^{2}(\bar{t}_{a,cell,i} - \bar{t}_{a,cell,i\perp traj})^2}\,, \tag{3.7}$$

where $\bar{t}_{a,cell,i\perp traj}$ is the component *a* of the projection of $\bar{\mathbf{t}}_{cell,i}$ into trajectory. Figure 3.11. Figure 3.10 shows an example of calculation of Euclidean distance between the mean of scores of one batch included in one cell (cell no.7) and its projection onto the trajectory. From the comparison of Figure 3.7 with Figure 3.10, it is clear that the means of batches (in Figure

3.10) are much closer to the trajectory than single calibration scores (in Figure 3.7). This result is confirmed by comparing also two histograms of distances of Figure 3.8 and Figure 3.11.

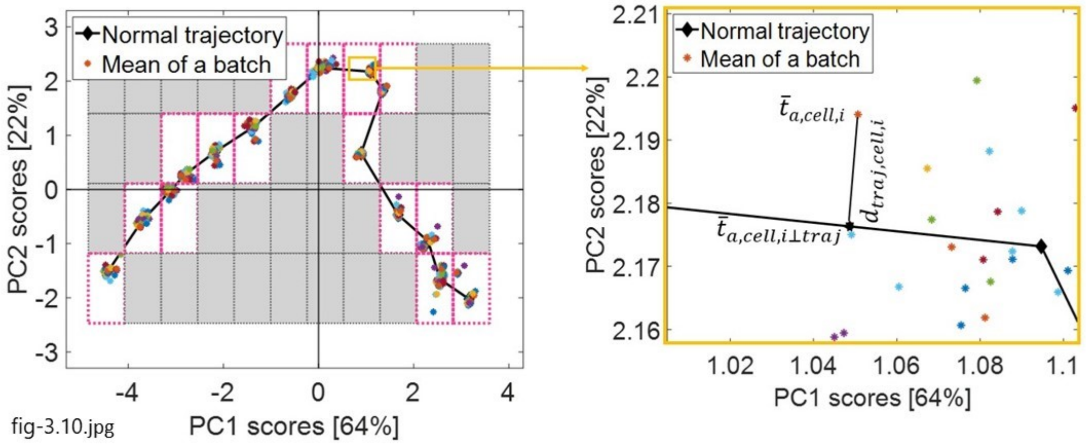


Figure 3.10. *Example of calculation of distance between a batch mean of scores* $(\bar{\mathbf{t}}_{\boldsymbol{cell,i}})$ *and its projection onto trajectory* $(\bar{\mathbf{t}}_{\boldsymbol{cell,i\perp traj}})$ *for cell no.7. Cell no. 7 is zoomed on the right side: each asterisk represents the mean of scores of one batch in one cell, while the solid line with diamond markers represents the normal trajectory. One batch mean is considered* $(\bar{\mathbf{t}}_{\boldsymbol{cell,i}})$ *and the distance* $\boldsymbol{d_{traj,cell,i}}$ *from its projection onto trajectory* $\bar{\mathbf{t}}_{\boldsymbol{cell,i\perp traj}}$ *is indicated*

Following this approach, the limit for each cell is calculated as

$$d_{(1-\alpha)\%\,c.l.\mathrm{W},cell} = d_{traj,cell}\big|(1-\alpha)\%\ d_{traj,cell,i} < d_{(1-\alpha)\%\,c.l.\mathrm{W},cell}, \qquad (3.8)$$

where $d_{95\%\,c.l.\mathrm{W},cell}$ is the distance from the trajectory for the limit calculated with the Westad et al. (2015) approach. Figure 3.11 is an example of histogram of distances $d_{traj,cell,i}$, calculated for cell no.7.

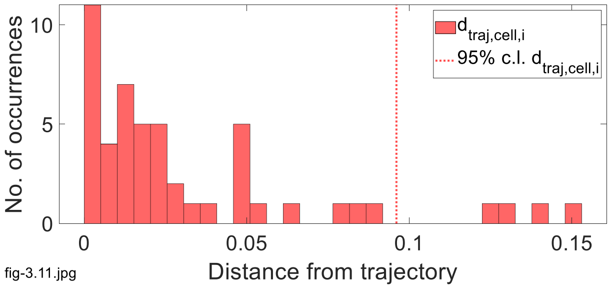
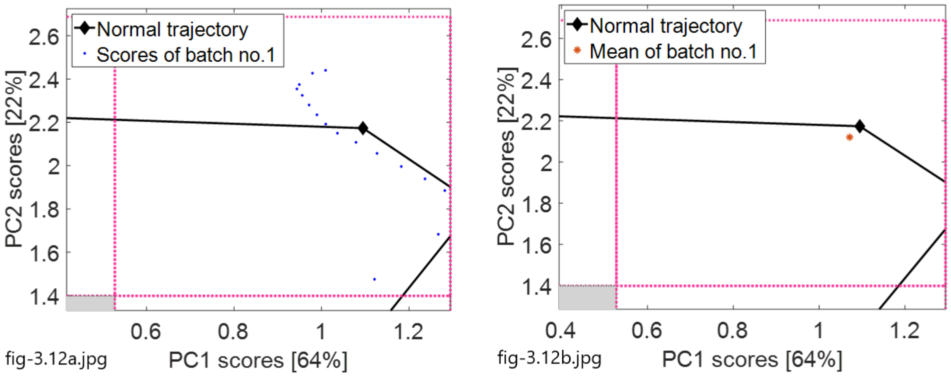


Figure 3.11. *Histogram of distances between batch means and their projection onto trajectory for cell no.7. The distance for the 95% confidence limit is marked with dotted line: the 95% of distances is smaller than the distance for the 95% confidence limit*

In this case the number of distances (occurrences) calculated is smaller than in Figure 3.8 because in this case only 50 points (one mean of scores for each batch) are considered in one cell. Since the means of batches are closer to trajectory than single scores, distances in Figure

3.11 are much smaller than distances in Figure 3.8. The distance for the (1-α)% confidence limit calculated with the approach of Westad et al. (2015) in Figure 3.11 is about one half smaller than the distance for limit calculated with the new approach in Figure 3.8: Westad et al. (2015) limits are closer to the trajectory.

The first problem of this method is that only the variance *across* batches is considered, while the variance *within* batches (i.e., variance inside the process) is not considered. Figure 3.12 shows as example batch no.1 in cell no.7: in Figure 3.12 (a) all scores are considered and the variance *within* the batch is represented, while in Figure 3.12 (b) only the mean of scores of batch no.1 is considered and the batch evolution inside cell no.7 is neglected.



(a) (b)

Figure 3.12. *Batch no.1 in cell no.7 (a) with all of its scores represented and (b) with only its mean of scores represented. In (a) the entire evolution of batch no.1 is considered, while in (b) only a mean point of batch no.1 is considered*

This approach is not appropriate because the objective of the study is to build a trajectory which is able to represent the process along its entire duration. Considering the means of batches $\bar{\mathbf{t}}_{cell,i}$, part of the process evolution is neglected, but when a new batch is projected onto the model for process monitoring, all of its samples are projected, so the entire process evolution should be represented by the model.

The second problem of this method for limits calculation is that the distance between means of batch scores and trajectory is calculated considering the orthogonal projection of the mean of scores of each batch onto the trajectory. It must be remembered that the trajectory is obtained by interpolation of overall means of scores found inside all valid cells, therefore it is a line made by a series of segments. When a point is projected orthogonally onto the trajectory, it is projected on a segment of the trajectory; however it may happen that in some cases the orthogonal projection of a point onto a trajectory segment doesn’t belong to the trajectory. An example is reported in Figure 3.13: the mean of one batch in cell no.6 can’t be projected directly onto the trajectory and its two closest trajectory segments need to be prolonged to perform the orthogonal projections. However, in both cases the projection doesn’t belong to the trajectory, and it is not possible to determine which projection (distance) should be considered for limits calculation. Since limits are referred to the trajectory, it is the distance between the mean of the batch and the trajectory that should be considered, and not the distance between the mean of the batch and a trajectory prolongation. In this case, this distance is larger than the distance computed with the orthogonal projection. The same reasoning is valid if the orthogonal projection is applied for the calculation of the distance of all single calibration scores from the trajectory.

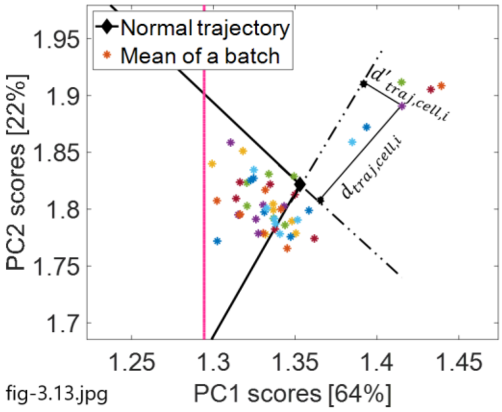
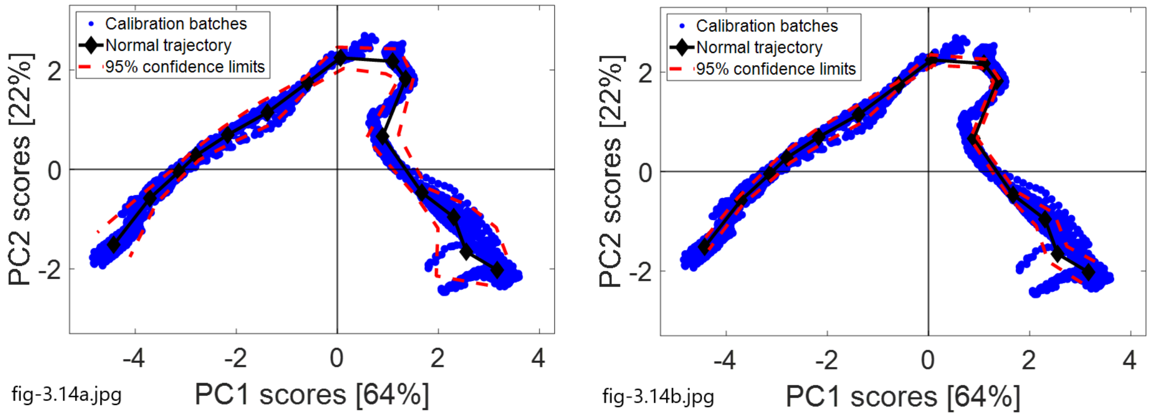


Figure 3.13. *Example of orthogonal projection of the mean of scores of one batch in cell no.6. The mean of the batch is far from the trajectory and two orthogonal projections are possible, however it is not possible to determine which distance ($\boldsymbol{d}_{traj,cell,i}$ or $\boldsymbol{d'}_{traj,cell,i}$) should be considered for limits calculation. In both cases the projection doesn't belong to trajectory but to the prolongation of its segments*

The new approach for limits calculation, which suggests to compute distances between single calibration scores and their closest points onto trajectory, is a good approximation of the distance calculated performing an orthogonal projection, as proposed by Westad et al. (2015), and overcomes the problem of projections external to the trajectory (see Figure 3.13).



(a) (b)

Figure 3.14. *95% confidence limits calculated with (a) the new approach and (b) the approach of* Westad et al. (2015). *In the second case limits are much tighter and the 34% of calibration scores is out of confidence limits (instead of 5%). In the first case only the 7% of scores is out of confidence limits*

Figure 3.14 shows that limits calculated with (3.8) are much tighter than the ones calculated with (3.4). Moreover, while with the new approach for limits calculation the amount of calibration scores out of confidence limits is equal to 7%, using the approach suggested by Westad et al. (2015) 34% of the calibration scores is out of limits. In all cases reported in §4, the limits are calculated on the distance between all scores and trajectory, according to (3.4).

A plot of the confidence limits for each cell is done by considering the bisector of the angle given by the segments of the trajectory that connects the mean of scores of the current cell to the mean of scores of the previous cell, and the mean of scores of the following cell, as shown in Figure 3.15. The bisector is extended for a length equal to the distance for the confidence limit $d_{(1-\alpha)\%\ c.l.,cell}$ of the cell whose mean of scores $\bar{\bar{\mathbf{t}}}_{cell}$ is the vertex of the angle.

For the first and last cells, since an angle is not present, the line orthogonal to trajectory is considered.

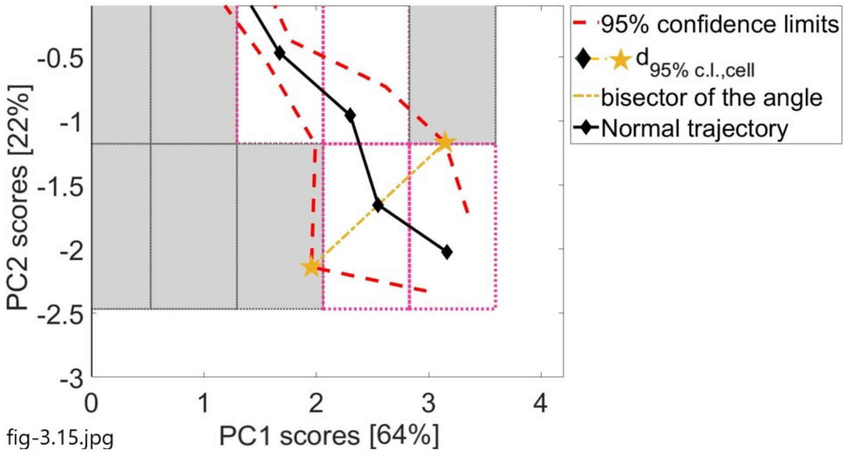


Figure 3.15. *Plot of confidence limits for a cell. Diamonds are mean points found with the grid search algorithm and the solid line is the trajectory modelled. Stars are points along the bisector distant* $\mathbf{d}_{95\%\ c.l.,cell}$ *from the mean point of the corresponding cell. Dashed lines are confidence limits obtained interpolating limits (stars) of all cells*

All limit points are then linearly interpolated to draw confidence limits around the trajectory, as show the dashed lines in Figure 3.16. Limits are plotted in such a way that they do not intersect, either with trajectory or between them: some manual adjustments may be necessary for each dataset (process). In the end, the resulting model is similar to the one reported in Figure 3.16.

For each valid cell, residuals and their limits are calculated. The Q residuals are the same obtained with the PCA model calibration, given by the error in the data fitting. For each cell, Q associated to the scores belonging to the cell are collected and the (1-α)% limit is calculated, similarly to the distance calculated for limits around the trajectory in the score plot. The assumption of normal distribution is not verified for both distances of scores from trajectory and the Q residuals, so limits are calculated by sorting values in ascending order and selecting the value such that (1-α)% of other values are smaller than the limit value.

Note that confidence limits are calculated on cells and then interpolated. Therefore if the trajectory is modelled with only few points, the final limits obtained are a rough approximation of the (1-α)% confidence limits.

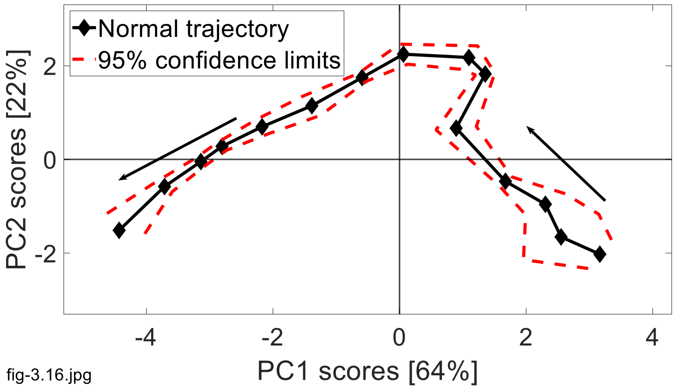


Figure 3.16. *Example of a process normal trajectory (continuous line) with means of scores (diamonds) and confidence limits (dashed lines) obtained with assumption-free modeling approach. Arrows indicate the direction in which the process evolves*

This approximation is worsened by the fact that some calibration scores are initially neglected by the grid-search algorithm (according to parameter γ), thus the number of scores used for limits calculation is only the one captured by the grid, which is smaller than the total number of scores of the PCA model.

3.4 Fault detection and diagnosis

When a new batch is available, it can be projected onto the model, and its evolution can be compared to the one of a normal process in the score plot, as showed in Figure 3.17.

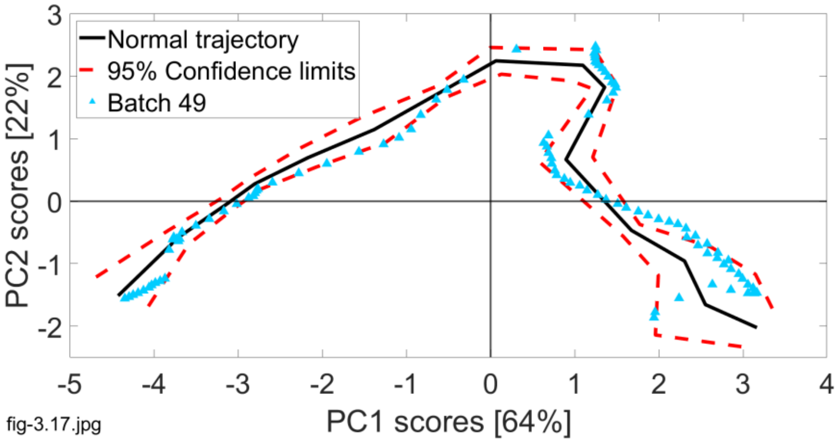


Figure 3.17. *Example of projection of a new batch (batch no.49) onto the model. The solid line represents the trajectory of a normal process, modelled with the grid-search algorithm. Dashed lines represent confidence limits. In the south-east region of the score plot, some scores are out of confidence limits*

The advantage of this model is that, differently from the batch-wise unfolding model discussed in §1.2, available samples are sufficient to project the batch onto the model at any time point, and no future data need to be estimated. The trajectory modelled in §3.2 and confidence limits calculated in §3.3 are representative of normal behaviour of the process: when a new batch is projected into the model, to be classified as normal its scores should follow the normal trajectory remaining inside confidence limits, otherwise it is considered as abnormal batch. Confidence limits calculated are of the (1-α)%, so some scores are expected to be out of limits also if the batch is in normal operating conditions. The alarm for scores is calibrated by setting the maximum number of consecutive scores allowed to be out of confidence limits equal to the maximum number of consecutive scores out of confidence limits that were found for calibration normal batches. The alarm for *Q* residuals is calibrated in the same way, by considering the maximum number of consecutive *Q* residuals out of confidence limits for calibration batches.

The distance of scores of the new batch from trajectory is calculated like

$$d_{traj,new} = \sqrt{\sum_{a=1}^{2}(t_{a,new} - t_{a,traj,pclosest})^2}, \tag{3.9}$$

where $t_{a,new}$ is the component *a* of the score of the new sample, and $t_{a,traj,pclosest}$ is the component *a* of the point *pclosest* in the trajectory which is the closest to the new score $\mathbf{t_{new}}$. Plotting the difference between the distance of the new sample from the trajectory and the distance of the limit from trajectory, a clear overview of time instants at which new scores are out of confidence limits can be obtained.

Also the *Q* residuals of the new batch are compared to the limits of *Q* residual calculated in §3.3, so that abnormalities in the correlation structure of the process can be detected.

From the projection of the new scores into trajectory, an approximation of the process state (or relative time) of the new batch can be estimated: the new score is projected onto the score space and its closest trajectory point among the ones calculated with (3.2) is found. The position in the trajectory of the closest point found is then divided by the total number of points into the trajectory, corresponding to the number of overall means of scores calculated (and to the number of valid cells of the grid). To make the relative time independent of the number of trajectory points, it is scaled to 0-100, where 0 is the beginning of the batch and 100 its end.

Once the fault is identified, its diagnosis is done through the loading plot: a relationship between the direction on which scores go out of confidence limits and the location of variables in the loading plot exists.

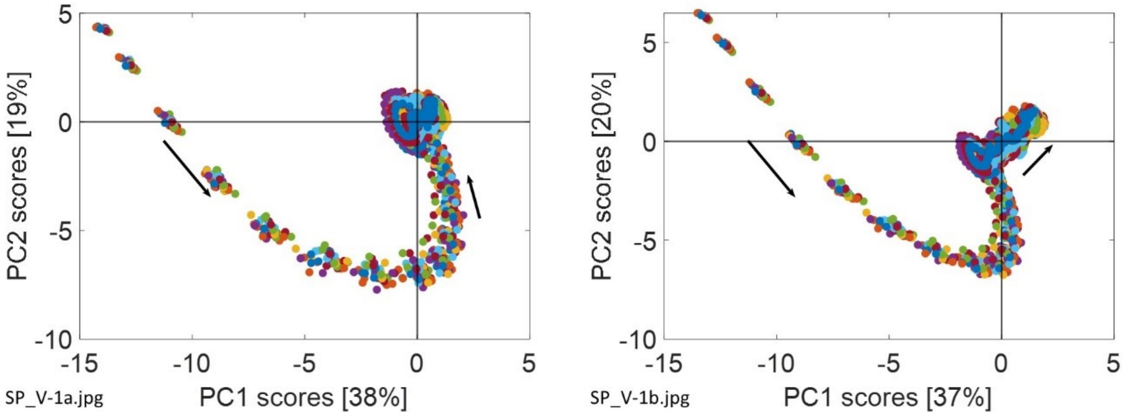
Chapter 4

Model testing

The assumption-free model and the variable-wise model have been calibrated and validated with 5 different datasets. The performance of each model is evaluated considering the capability of fault detection, the detection time and the number of false alarms occurring in the case of both normal and faulty batches. The number of batches available for validation is not the same for all datasets.

4.1 Dataset 1

This dataset contains samples of a simulated SBR polymerization reaction. 9 variables are sampled for 45 normal batches at 200 time instants. A description of the process and of datasets available is provided by Nomikos and MacGregor (1994) and reported at §2.1. From a first screening of the process variables profiles, reported in §A.1.1, the process results to be almost stationary during its entire duration; this situation is confirmed by the score plot of the assumption-free model of Figure 4.1 (a), where after the 15th time instant scores accumulate around the origin of the axes for the rest of the batch duration (up to 200th time instant).



(a) (b)

Figure 4.1. *Dataset 1. Score plot resulting from the variable-wise unfolded PCA (a) without time variable and (b) with time variable. All time instants for all 45 batches are reported in both cases. Arrows indicate the direction on which the process evolves. Percentage in square brackets represents the variance captured by the corresponding principal component*

The accumulation of scores at the origin of the score space corresponds to the part of the process on which values of variables remain almost constant, as can be seen from the profiles of variables reported at §A.1.1. In order to try to capture the dynamics of the process, the time variable is included as additional variable, then a PCA is performed and the corresponding score plot is reported in Figure 4.1 (b). Comparing the two models, the total amount of variance captured by the first two principal components is the same (57%); however, in the model including the time variable also the last part of the process can be somewhat captured by the trajectory modelling. After the alarm calibration, the number of consecutive samples out of confidence limits for fault detection and alarm activation in the second case (35 samples in the score plot and 7 samples in the *Q* residual plot) is lower than in the first case (36 samples in the score plot and 10 samples in the *Q* residual plot). For these reasons, in the study reported in §4.4.1 the model including also the time variable is used, to obtain a better representation of the process dynamics (more detailed trajectory modelling) and a faster fault detection.

In order to compare consistently the batch-wise and the variable-wise models, they have to be calibrated on the same number of variables: the time variable is considered also for the batch-wise model calibration. Note that the "time variable" is considered only in this dataset.

4.1.1 Monitoring with an assumption-free model

The variable-wise unfolded matrix with the additional time variable is used to calibrate a PCA model, summarized in Table 4.1. For the selection of the number of principal components to retain into the model, the RMSECV criteria is adopted. With 3 principal components the RMSECV is equal to 0.68, while the variance captured is about 68% of the total one.

Table 4.1. *Dataset 1: variable-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	3.68	36.85	36.85	0.89
2	2.00	20.06	56.91	0.75
3	1.23	12.25	69.16	0.68
4	1.00	10.00	79.16	1.50
5	0.99	9.89	89.05	12.98

From the $J \times NPC$s loading matrix, and in particular from the loading plot reported in Figure 4.2, the correlation structure between variables can be analysed.

Considering the first principal component (which is the one capturing the largest amount of variance), time, total conversion and the energy released are all positively correlated, while they are anti-correlated to the latex density and to the reactor, jacket and cooling water temperatures. These relationships are not surprising: since the SBR polymerization is an exothermic reaction, the conversion of reactants into product is favored by low temperature in the reactor, which is obtained through cooling water flowing into the jacket. Styrene and butadiene flowrates and the feed temperature are kept constant for the entire process duration and have no influence on the process dynamics.

The score matrix can be plotted into a two-coordinate system considering first two principal components: as can be noticed from Figure 4.1 (b), the process can be divided into an initial phase with a marked dynamics (lasting about 15 time instants), and a second phase with very slow dynamics in which the scores are located close to the origin of axes for the rest of the batch duration.

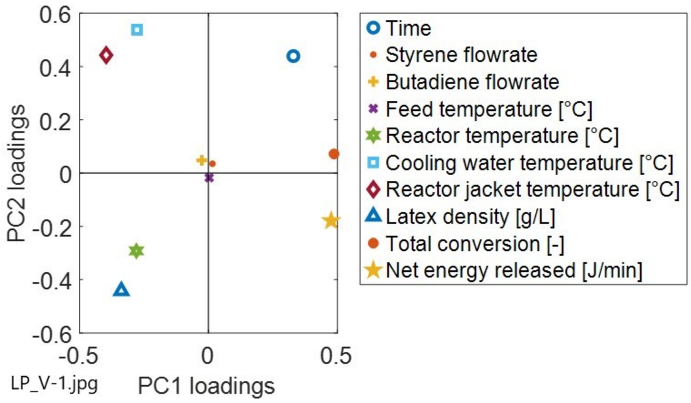


Figure 4.2. *Dataset 1. Loading plot resulting from the variable-wise unfolded PCA*

The grid-search algorithm is applied to the score plot to model a normal trajectory of the process as described in §3.2. As reported in Table 4.2, overall mean points are calculated on cells containing at least one score of each batch (β=100%) and at least the 95% of scores is required to be captured by overall valid cells.

Table 4.2. *Dataset 1: parameters set for grid-search algorithm*

Parameter no.	Symbol	Value	Unit
1	β	100	% of batches
2	γ	95	% of scores

The initial grid has a 15×15 resolution, while the optimal grid found by the algorithm is a 9×4 grid, showed in Figure 4.3. In the end the total number of scores included in trajectory modelling is equal to 98% (> γ=95%). The overall mean of scores for each valid cell is calculated and all points found are interpolated to model the final normal trajectory with 9-point resolution. The number of points on which the trajectory is calculated is very small: because of the modest dynamics of the process, the largest part of the process is not modelled because scores start to accumulate after few time instants, therefore this part of the process, despite it includes more than the three-fourth of the total duration, is modelled by the grid-search algorithm with only 2 points.

For each cell, the confidence limit of 95% is calculated with the method of Westad et al. (2015) and with the new approach proposed, both described in §3.3.

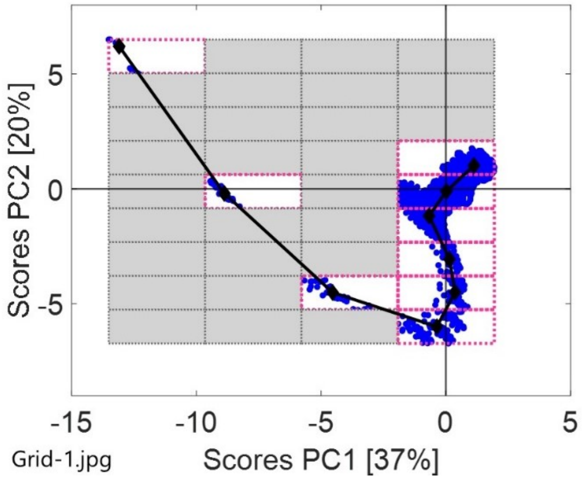
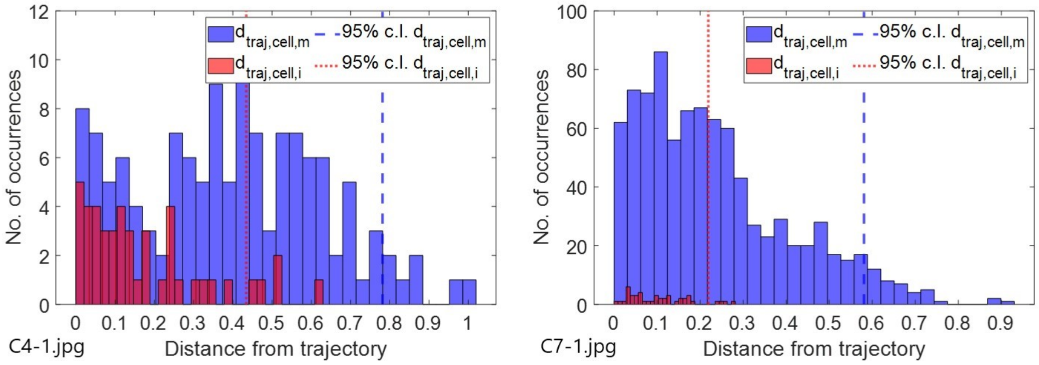


Figure 4.3. *Dataset 1. Optimal grid found by the grid-search algorithm. Valid cells are denoted with white background, while invalid cells have grey background. Normal trajectory is marked with solid line, while diamonds represent the means of scores calculated in each valid cell*

In Figure 4.4, the distances of each batch mean from trajectory and distances of scores from trajectory are reported for each cell: it is evident that limit distances calculated with the approach proposed by Westad et al. (2015) are smaller than the ones calculated with the new approach. The first method, in fact, suppresses part of the process dynamics by summarizing the scores of each batch into a single point for each cell, and the limit found is underestimated.



(a) (b)

Figure 4.4. *Dataset 1. Plot of distances and of limits calculated for (a) cell no.4 and (b) cell no.7. Distances of single scores from trajectory are in blue, while distances of mean of scores of each batch from trajectory are in red. The dotted line marks the limit calculated with approach of* Westad et al. (2015)*, while the dashed line marks the limit calculated with the new approach*

This approach does not correctly represent the dynamics of the process and is not appropriate for monitoring when all samples of a new batch are projected onto the model. The new approach proposed is chosen and all limit points are interpolated to draw continuous limits around the trajectory. Depending on the shape of the trajectory, some adjustments in the plot of limits (plot_lim.m Matlab function) may be necessary, as mentioned also by Westad et al. (2015), in order to avoid intersections with the normal trajectory.

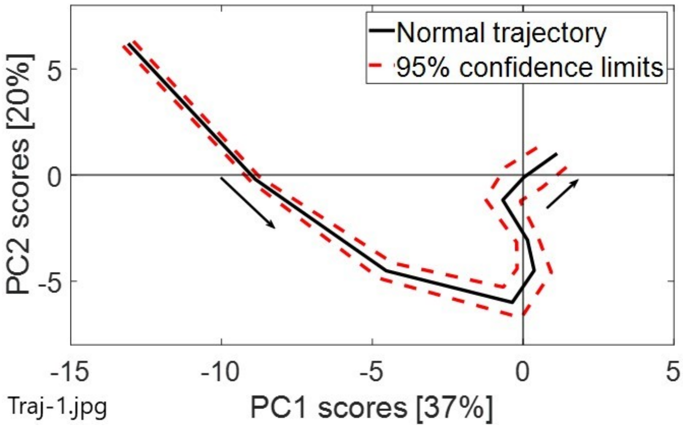


Figure 4.5. *Dataset 1. Final normal trajectory of the process (solid line) obtained through the grid-search algorithm with 95% confidence limits (dashed lines)*

At the end, the final trajectory with limits obtained is reported in Figure 4.5. Calibration batches are then projected onto the model to evaluate the real number of scores out of the confidence limits: despite limits are calculated at 9 points considering a 95% confidence area and then approximated by interpolation, only the 7% of scores results to be out of the confidence limits, indicating a good approximation.

The alarm is calibrated on normal calibration batches, both for the scores and residuals, as described in §3.4: the maximum number of consecutive scores allowed to be out of the confidence limits for normal batches is equal to 34, while the maximum number of consecutive residuals out of the confidence limits for normal batches is equal to 6. The number of scores out of the confidence limits before alarm activation in this case is very high, so the alarm has a low sensitivity and the fault is detected with a substantial delay: this is due to the fact that trajectory in the last part of the process is able to model only part of the dynamics, and lots of calibration scores are left out of the confidence limits.

Eight new batches are available for validation as described at §2.1.3: six are normal (batch no.1-5 and batch no.53), while the other two are faulty (batch no.99 and batch no.106). An example of validation is reported here considering batch no.99 as a new batch. The new data are projected onto the model: from the score plot of Figure 4.6, it can be noticed that, as expected, lots of scores are out of the confidence limits when trajectory approaches the origin of the coordinate system. Despite it is known that the fault occurs half-way of the process (i.e., after about 100 time instants), the abnormality is detected by the model only at the 140th time instant, so with a delay of 40 time instants; however looking at the distances from trajectory reported in Figure 4.7 (a), it is evident that after 102 time instants the distance between scores and confidence limits (and so the distance between scores and the normal trajectory) increases rapidly and remains high until the 153th time instant.

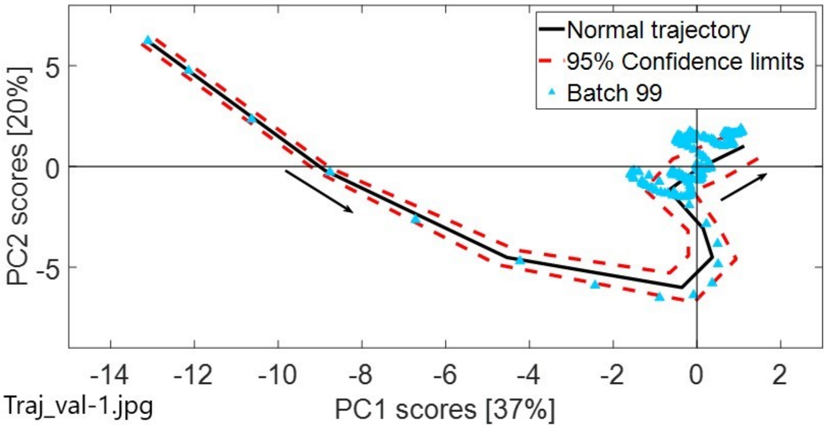


Figure 4.6. *Dataset 1. Score plot of the projection of batch no.99 onto the model. Normal trajectory and confidence limits are represented respectively by the solid line and by dashed lines. Triangles corresponds to new samples projected onto the model*

It can be concluded that the delay in fault detection is due to the number of consecutive scores out of confidence limits on which the alarm is calibrated, but the plot of distances is able to highlight earlier that an abnormal event is taking place.

Residuals of each sample are plotted together with confidence limits calculated, like reported in Figure 4.7 (b): in this case the number of consecutive residuals out of limits is lower than 7 and the fault is not detected.

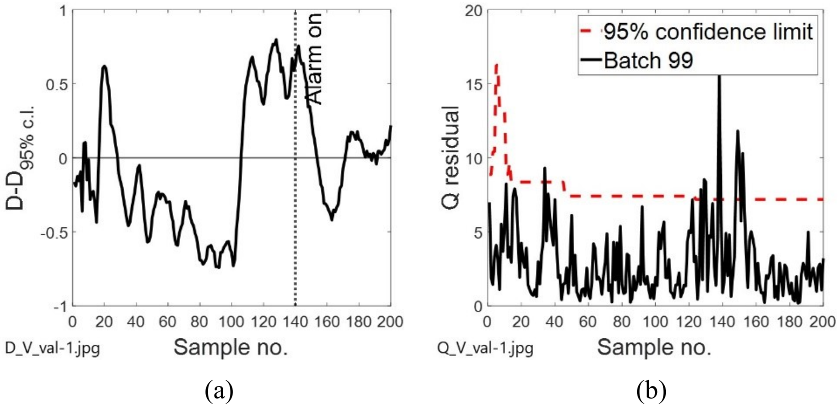


Figure 4.7. *Dataset 1. Plot of (a) the difference between the distance from trajectory of a new sample (D) and the limit distance ($D_{95\% c.l.}$), and of (b) Q residuals along the sample number. The 95% confidence limits for residuals is reported (dashed line). Time at which the alarm turns on is marked with dotted line*

The loading plot of Figure 4.2 is considered for the fault diagnosis: reactor jacket temperature (and so the cooling water temperature), the reactor temperature and the latex density seem to be the variables responsible of the abnormal conditions of the new batch. Considering that the latex density is a consequence of the reaction process taking place inside the reactor, it can't be considered like the source of the fault. Reactor jacket temperature and cooling water temperature are probably controlled by a temperature controller, according to the temperature inside the reactor, which can be considered like the measured variable responsible of the abnormality detected. It is known by literature that the fault is given by a contamination in the butadiene feed: the reaction rate, and so the velocity of latex production and the energy released by the reaction are proportional to reactants concentration in the model of Broadhead et al. (1985), thus introducing impurities in the feed, the concentrations of reactants decrease and the rate of energy decreases too. The net energy released by the exothermic reaction is lower, the temperature inside the reactor is lower than normal operating conditions and the cooling power demand is lower than expected too (higher reactor jacket temperature and higher cooling water temperature). Unfortunately, only the feed flowrate is measured and not its purity, thus an engineering interpretation is fundamental to reconstruct the real cause of the fault. It must be remembered that these considerations are possible only assuming that all sensors and controllers in the reactor work appropriately, otherwise also malfunctions in the instrumentation should be investigated. As described in §3.4, from the projection of scores of new samples into trajectory and the scaling on the number of points on which the trajectory is modelled, the relative time is evaluated and plotted at each sample in Figure 4.8. From the profile of the process state (or relative time), the process seems to be completed after 122 time instants, also if the batch duration is equal to 200 time instant. The slope of the "curve" at the initial time instants confirms the fast dynamics of the process, as discussed at the beginning of this section, while the almost flat profile after 50 time instants is associated to the static part of the process, in which scores accumulate close to the last two points of the normal trajectory modelled with the gird-search algorithm.

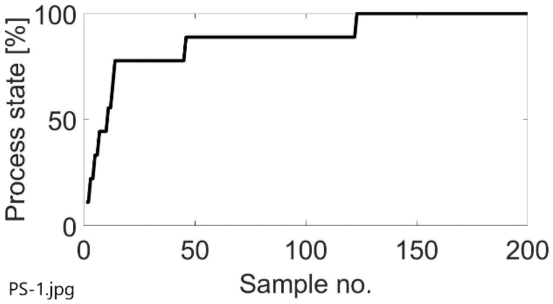


Figure 4.8. *Dataset 1. Plot of the process state (or relative time) against sample number of batch no.99. The process results to be complete more than 50 time instants before its end*

Validation is completed testing also other two batches and reporting results in Table 4.3. Six normal batches are available for validation and none of them is classified as faulty batch by the model, so no false faulty batches occur.

Table 4.3. *Dataset 1: assumption-free model results*

Description	Value
Confidence limits	95%
Calibration scores out of c.l.	7%
Consecutive scores out of c.l. for alarm on scores	35
Consecutive scores out of c.l. for alarm on residuals	7
False faulty batches	0/6
Missed fault detection	1/2
False alarms	0

Batch no.106 presents contamination in the butadiene flowrate since the beginning of the process, however this is not detected by the assumption-free model and the batch results to be normal: one of the two faulty batches is not recognised by the model. Considering batch no.99, the fault occurs half-way of the process and the fault is detected after 140 sample, so no false alarms occur.

4.1.2 Monitoring with a batch-wise unfolded MPCA model

In this dataset, all 45 calibration batches have the same duration (i.e., same number of time instants sampled) and no alignment procedure is needed to build a batch-wise unfolded matrix. A PCA model is calibrated and summarized in Table 4.4, adopting the RMSECV criterion for the selection of the number of principal components to retain into the model.

Table 4.4. *Dataset 1: batch-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	268.15	14.91	14.91	0.94
2	174.57	9.71	24.62	0.92
3	107.38	5.97	30.60	0.92
4	98.10	5.46	36.05	0.93
5	92.36	5.14	41.19	0.93

Two principal components are selected for model calibration; however, the cumulative variance captured by the model is only the 25%. The RMSECV is equal to 0.92 with 2 principal components, and it neither increases nor decreases too much if the number of principal components is increased.

From the score plot reported in Figure 4.9, calibration batches result to be normally distributed, therefore they respect the assumption on which the calculation of limits for scores, T^2 statistic and Q statistic is based.

Batches no.12 and no.16 are out of the confidence ellipse.

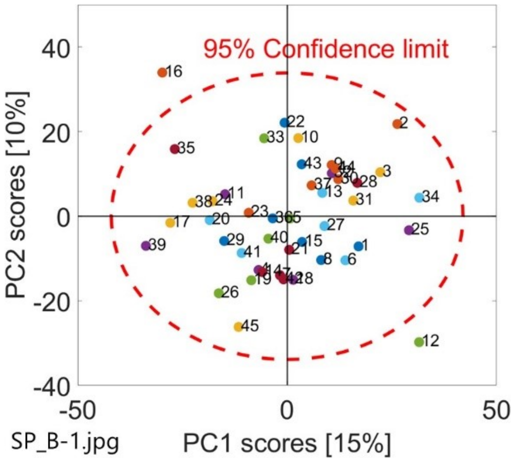
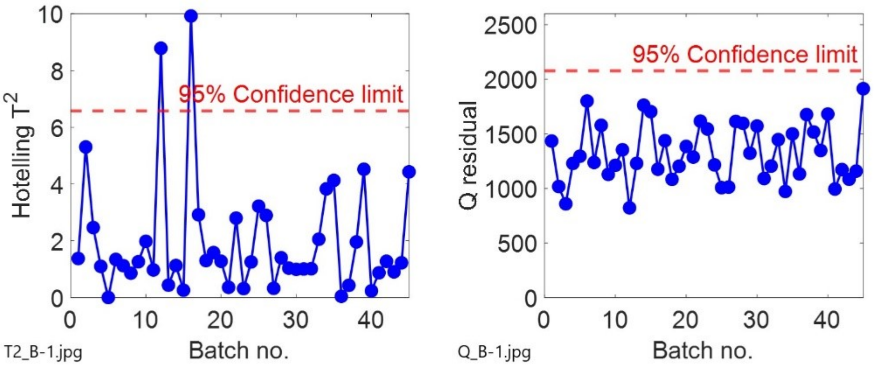


Figure 4.9. *Dataset 1. Score plot resulting from the batch-wise unfolded PCA. All 45 batches are reported with the 95% confidence limit (dashed line). Percentages in squared brackets represent the variance captured by the corresponding principal component*

Normal distribution of calibration batches is confirmed by the Hotelling T^2 statistic and Q residual plots of Figure 4.10.



(a) (b)

Figure 4.10. *Dataset 1. (a) Hotelling T^2 and (b) Q residual. Each dot represents one batch, while 95% confidence limits are marked with dashed line*

An example of validation is provided here projecting batch no.99 as a new batch. The batch evolution is represented in the score plot of Figure 4.11, computing missing future time instants at each time *k* according to the procedure explained in §1.2.

Batch no.99 starts inside the limits but evolves out of the confidence area, where it ends after 200 time instants. From the evolution of the batch trajectory, a fault is suspected occurring at about half-way of the process.

The fault detection is confirmed also by the plots of the Hotelling T^2 and the Q statistic, whose alarms are set to start after 3 consecutive points out of confidence limits.

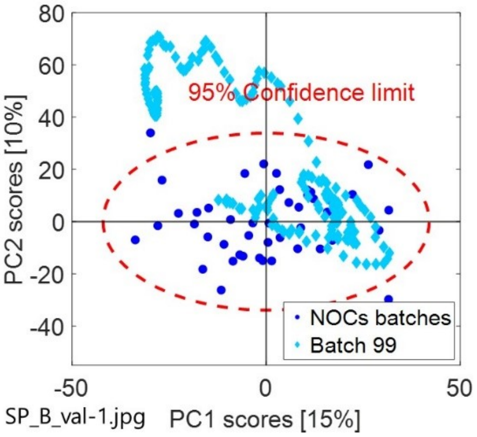
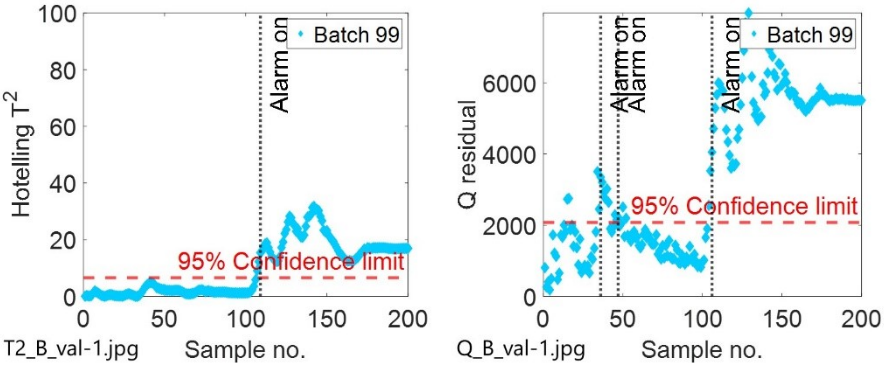


Figure 4.11. *Dataset 1. Score plot of the projection of batch no.99 onto the model. Dots represent normal batches, while diamonds represent evolution of the new batch at every time instant. Dashed ellipse limits the 95% confidence area*

In Figure 4.12 times at which the alarm occurs are marked with dotted line: two false alarms take place in the Q residual plot, then after the third alarm the batch remains out of the 95% confidence limit for the rest of the process duration. In the literature it is reported that in batch no.99 a contaminant enters in the butadiene feed flowrate half-way of the process, however the butadiene purity, which is the variable in which the anomaly is present, is not a measured variable.



(a) (b)

Figure 4.12. *Dataset 1. Plots of (a) Hotelling T^2 and (b) Q residuals for batch no.99. Diamonds represent samples of batch no.99, while the dashed line marks the 95% confidence limit. An alarm occurs in the T^2 at sample no.108. 3 alarms occur in the Q residual plot: after the last one at time 106, the process remains out of confidence limit until the end*

The fault diagnosis is carried out considering the contribution plots of the T^2 and Q: at samples no.106 and no.108, which are samples at which the fault appears in the T^2 statistic and Q residual control charts, respectively, variables that most contribute to the fault (i.e., their contribution is out of the confidence limits) are the reactor temperature, the latex density, the conversion and the net energy released: these are all signs that the reaction rate is lower than expected. The energy released is the first variable to go out of the confidence limits and it depends on butadiene and styrene concentrations, as reported in the model of Broadhead et al. (1985), so a decrease in the concentrations of monomers can be suspected if the reaction slows down. The jacket and the cooling water temperature go out of limits after the reactor temperature, which means that they are probably controlled by a temperature controller in the reactor and their abnormal profile is only the consequence of the abnormal reactor temperature. All contribution plots of variables to the Hotelling T^2 and Q residual can be obtained with the validation_BWU.m file reported in Appendix 2.

Table 4.5. *Dataset 1: batch-wise model results*

Description	Value
Confidence limits	95%
False faulty batches	4/6
Missed fault detection	0/2
False alarms	9

Validation is carried out also for other 7 batches available (six normal batch and one abnormal) and final results are summarized in Table 4.5. Batch no.2 and batch no.53 result to be normal, and they actually are, while the other four normal batches are recognised as faulty on the control chart of Q. In both faulty batches the fault is detected quite promptly, and 2 false alarms are present in the case of batch no.99.

As mentioned before, the feed purity is not a measured variable, so in order to detect the fault it is necessary to wait until its effect appears in the measured variables. Since the fault can be detected only indirectly (purity is not a measure variable), it is unlikely that the fault appears in the model control charts as soon as it occurs: also if the first real alarm turns on at the 106th time instant (in the Q residual plot) for batch no.99 and at 6th time instant (in the Q residual plot) or batch no.106, it can be concluded that the model has high performance and it is able to detect the fault promptly.

While the assumption-free model is able to recognise only one out of two faulty batches, the batch-wise model detects all abnormal batches. However, several false alarms are raised in the second model, and four out of six normal batches are wrongly classified as faulty by the Q control chart. It can be concluded that, while the assumption-free model lacks sensitivity, the batch-wise model is too sensible for this dataset.

4.2 Dataset 2

This dataset contains samples of an industrial polymerization reaction carried out in a DuPont plant. 10 variables are sampled for 50 normal batches, all with the same duration. A description of the process and of datasets available is provided by Nomikos and MacGregor (1995) and reported at §2.2, while all profiles of variables are reported in Appendix 1 at §A.1.2.

4.2.1 Monitoring with an assumption-free model

The variable-wise matrix is used to perform a PCA. The column of "time", which is present in the initial variable-wise matrix, is not considered as a variable in the PCA model: it has been added to allow the algorithm to distinguish single batches, but it is not part of the original dataset provided by DuPont. The RMSECV criteria is used for the selection of the number of principal components to retain into the model.

Table 4.6. *Dataset 2: variable-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	6.39	63.92	63.92	0.66
2	2.16	21.56	85.48	0.45
3	0.95	9.50	94.98	0.36
4	0.25	2.52	97.50	0.35
5	0.11	1.07	98.57	0.65

As reported in Table 4.6, the total amount of variance captured is of the 95%, while the RMSECV is equal to 0.36 using 3 principal components: the minimum of the RMSECV is reached with 4 principal components, however the difference between two values is not significant, so the model with a smaller number of principal components is preferred since the objective is to represents data in a low-dimensional space (Nomikos and MacGregor, 1994).

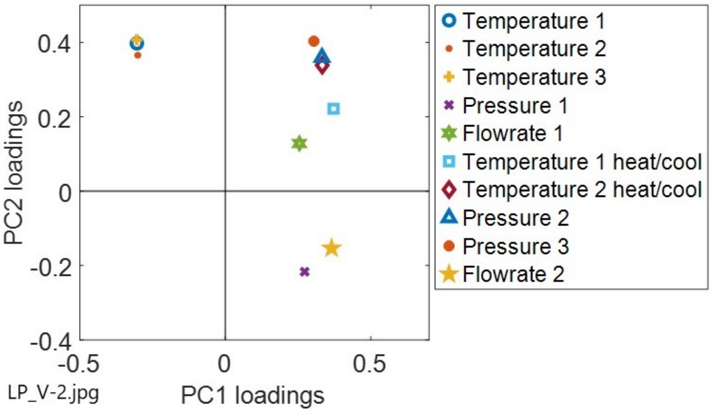


Figure 4.13. *Dataset 2. Loading plot resulting from the variable-wise unfolded PCA*

The loading plot resulting from the model calibration is presented in Figure 4.13.

Considering the first principal component, which is the one capturing the largest amount of variance, Temperature 1, 2 and 3 are anti-correlated to all of the other variables.

The score plot, reported in Figure 4.14, represents trajectories of all 50 batches along the entire process duration. The process evolves from right to left, as indicated by the black arrows reported in the plot.

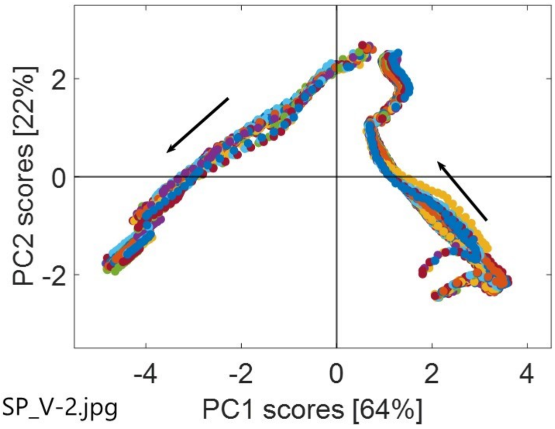


Figure 4.14. *Dataset 2. Score plot resulting from the variable-wise unfolded PCA. Arrows indicate the direction on which the process evolves. Percentages in squared brackets indicate the variance captured by the corresponding principal component*

The grid-search algorithm described in §3.2 is applied to the score plot to model a normal trajectory of the process. For this dataset, an initial grid resolution of 15×15 is used, and only cells containing at least one score for each batch (β=100%) are considered for trajectory modelling, as reported in Table 4.7. The minimum number of scores required to be captured by the grid is γ=95%. At the end the total number of scores included in trajectory modelling is equal to 97% (> γ=95%).

Table 4.7. *Dataset 2: parameters set for grid-search algorithm*

Parameter no.	Symbol	Value	Unit
1	β	100	% of batches
2	γ	95	% of scores

The optimal grid selected by the algorithm according to criteria reported in §3.2 has a resolution of 4×11 cells. For each cell the overall mean of scores is calculated and all points found are interpolated to obtain the final trajectory with 15-points resolution reported in Figure 4.15.

The 95% confidence limit is calculated for each cell, considering both approaches: the one adopted by Westad et al. (2015) and the new one proposed in this thesis.

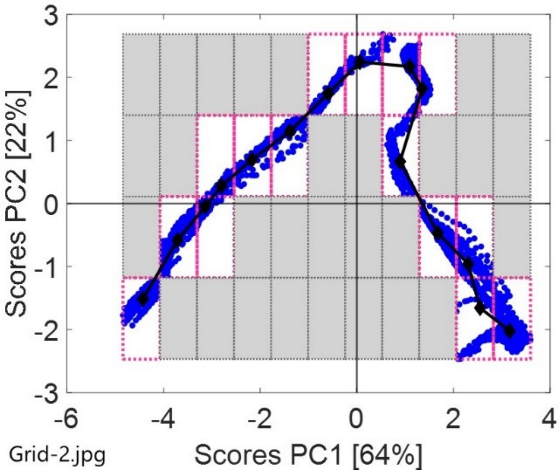


Figure 4.15. *Dataset 2. Optimal grid found by the grid-search algorithm. Valid cells are denoted with white background, while invalid cells have grey background. Normal trajectory is marked with solid line, while diamonds represent the means of scores calculated in each valid cell*

In Figure 4.16 the distances of two cells calculated with two approaches are reported, together with corresponding limits. Also in this case, as demonstrated in §4.1.1, the approach adopted by Westad et al. (2015) is not appropriate.

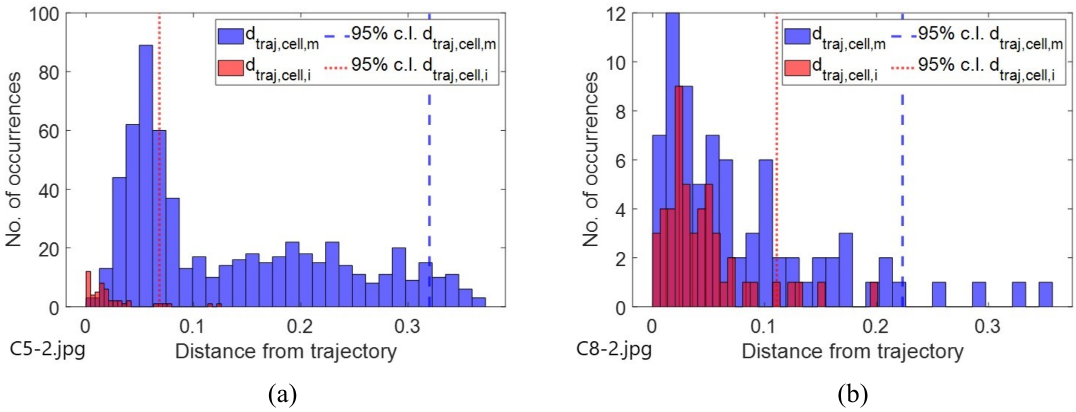


Figure 4.16. *Dataset 2. Plot of distances and of limits calculated for (a) cell no.5 and (b) cell no.8. Distances of single scores from trajectory are in blue, while distances of mean of scores of each batch from trajectory are in red. Dotted line marks the limit calculated with approach of* Westad et al. (2015)*, while the dashed line marks the limit calculated with the new approach*

In particular, from Figure 4.16 (a) it is evident that at the 5th trajectory point (i.e., 5th valid cell) distances of means of batches from trajectory are much smaller than distances of single scores from trajectory, so the limit calculated with the approach of Westad is about three times tighter than the limit calculated with the new approach: lots of calibration scores, despite their normal conditions, would result to be out of confidence limits if the Westad limit was adopted. At the 8th trajectory point of Figure 4.16 (b) the difference between two limits calculated is smaller than in the 5th cell, but the limit calculated with the Westad approach would be underestimated, because only the variability between batches would be represented instead of the variability within the process: also in this case, lots of calibration (and validation) scores would fall out of confidence limits. For this reason, the second approach is chosen for limits definition and all points are interpolated to plot continuous limits around the trajectory, as shown in Figure 4.17.

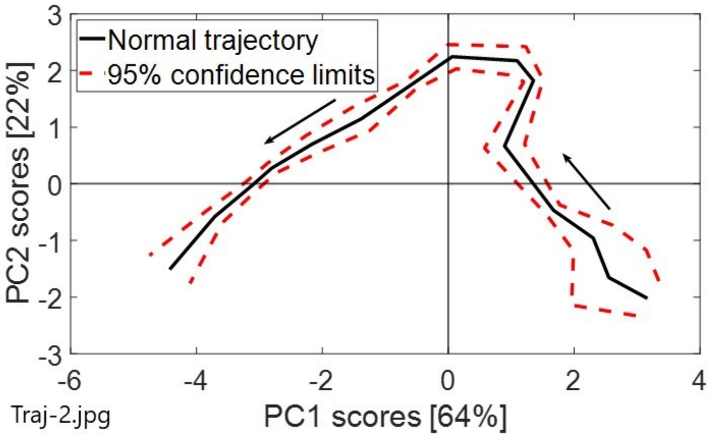


Figure 4.17. *Dataset 2. Final normal trajectory of the process obtained through the grid-search algorithm (solid line) with 95% confidence limits (dashed line)*

Calibration normal batches are then projected onto the model to evaluate the real number of scores out of confidence limits. Limits are calculated for each cell and despite continuous limits around trajectory are evaluated only at 15 points, and then approximated by interpolation, only the 7% of scores is actually out of confidence limits, indicating a good approximation.

In order to detect faults and classify new batches like normal or abnormal, two alarms are calibrated for the score plot and the *Q* residual plot.

The number of consecutive scores allowed out of the confidence limits for validation batches is set equal to the maximum number of consecutive scores out of the confidence limits for calibration batches: in this case it is equal to 9, so the alarm turns on at 10 consecutive scores out of confidence limits. Considering residuals, the alarm turns on after 13 consecutive values higher than the limit value. Five batches (4 normal and 1 faulty) listed in §2.2.3 are used for validation, and an example is reported here considering batch no.49 as a new batch.

The new data are projected onto the model and the resulting score plot with the new batch trajectory is reported in Figure 4.18: in the south-east region of the score space some scores are out of confidence limits, however their number is smaller than 10, so the fault is not detected in the score plot.

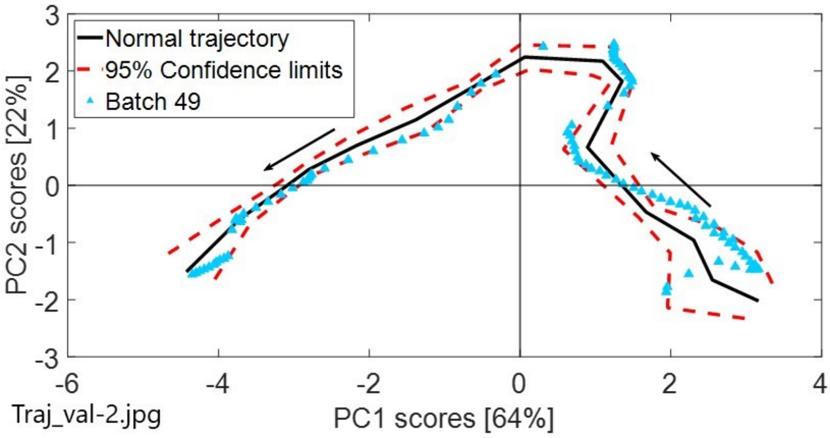


Figure 4.18. *Dataset 2. Projection of a new batch (batch no.49) onto the model. Normal trajectory and confidence limits are represented, respectively, by the solid line and the dashed line. Triangles represent new batch scores*

Instants at which scores are out of confidence limits can be identified more precisely by the plot of the difference between the distance of the new scores from trajectory and the limit distance (i.e., the distance of the limit from trajectory), shown in Figure 4.19 (a): in this case no alarm is reported because no fault is detected.

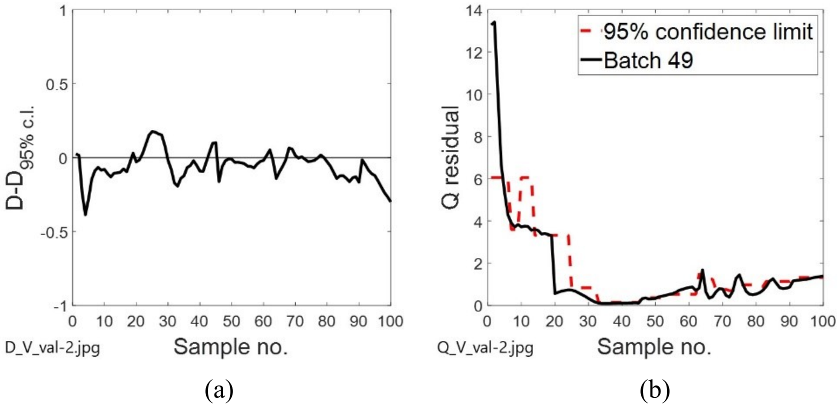


Figure 4.19. *Dataset 2. Plot of (a) the difference between the distance from trajectory of a new sample (D) and the limit distance ($D_{95\%\ c.l.}$), and of (b) Q residuals along the sample number. The 95% confidence limit for residuals is reported (dashed line)*

The *Q* residuals for the new sample can be plotted with the corresponding limit, calculated for each grid-cell. As showed in Figure 4.19 (b), at the beginning of the new batch residuals are much higher than limits allowed, evidencing a large amount of data not described by the model and a different correlation structure between variables. The fault is not detected by the alarm on scores, neither by the alarm on Q residual because they are calibrated on the number of consecutive points out of confidence limits, and scores and residuals are out of confidence limits only for few time instants; however, considering the magnitude of residuals at the beginning of the process, an anomaly can be suspected. The diagnosis of the cause can be done looking at the loading plot of Figure 4.13 and at the score plot of Figure 4.18. Scores go out of confidence limits in the right direction in the fourth quadrant of the plot: the variable that is located in the corresponding region of the loading plot is Pressure 1, so it can be considered like the one responsible of the fault. If the trajectory of this variables is plotted against the sample number together with calibration batches, it can be noticed that the starting point of the Pressure 1 is abnormal with respect to the normal batches, however the new batch is not the only one presenting this deviation: also some calibration batches have the same initial value for this variable and in the score plot they have a different starting point with respect to other calibration batches. The reason why the new batch is not recognized as faulty is that some calibration batches (from no.45 to no.50), considered like normal, behave similarly to a faulty batch (e.g., batch no.49).

The state of the process (or relative time) can be estimated considering the point into trajectory which is the closest to the new sample: the time is scaled on 0-100 to make it independent from the trajectory resolution (i.e., number of trajectory points).

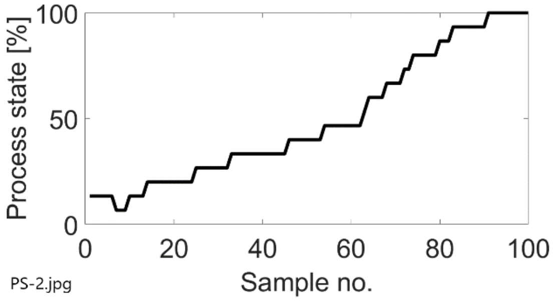


Figure 4.20. *Dataset 2. Plot of the process state (or relative time) against sample number of batch no.49. The process dynamics becomes faster after about 65 samples*

Figure 4.20 shows that the process dynamics is slower between samples 46 and 62: it can be associated to an accumulation of points corresponding to these samples in the first region of the score plot. A similar situation occurs towards the end of the process where the process dynamics is slower and scores accumulate in proximity of the end of the trajectory. All batches available for validation are projected onto the model and final results are summarized in Table 4.8. In this case, only considering the (null) number of alarms both for the score plot and the residual plot, the model is not able to recognise the faulty batch, due to the presence of some batches in the calibration dataset which have similar characteristics to the ones of the faulty batch.

Table 4.8. *Dataset 2: assumption-free model results*

Description	Value
Confidence limits	95%
Calibration scores out of c.l.	7%
Consecutive scores out of c.l. for alarm on scores	10
Consecutive scores out of c.l. for alarm on residuals	13
False faulty batches	0/4
Missed fault detection	1/1
False alarms	0

Since all validation normal batches are classified as normal, there are no false faulty batches. Since no alarm is present, no false alarm occurs.

4.2.2 Monitoring with a batch-wise unfolded MPCA model

The three-dimensional matrix is batch-wise unfolded and a PCA model is developed. For the selection of the number of principal components to retain into the model, the RMSECV criteria is used. Model characteristics are summarized in Table 4.9.

Table 4.9. *Dataset 2: batch-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	376.10	39.46	39.46	0.87
2	169.92	17.83	57.29	0.77
3	68.47	7.19	64.48	0.73
4	43.58	4.57	69.05	0.70
5	28.55	3.00	72.05	0.70

Four principal components are retained into the model: the RMSECV is equal to 0.70, while the total amount of variance captured by the model is equal to 69%.

From the score plot represented in Figure 4.21, it can be noticed that scores are not multinormally distributed: a diagonal cluster is recognized in the right side of the plot, while 5 batches are located in the second quadrant. Not only can 2 clusters be identified, but they also are made by consecutive batches: the central cluster includes batches from no.1 to no.44, while the second cluster includes batches from no.45 to no.50.

Although it is not verified the assumption of normal distribution on which confidence limits calculation is based, the formula for limits calculation remains the same described in §1.1, reminding that model performances may be affected by this situation.

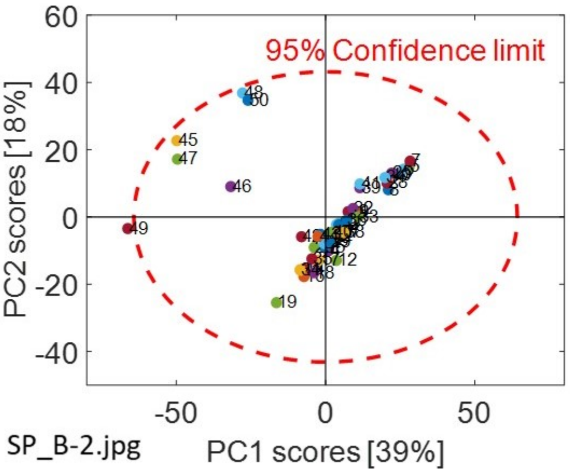
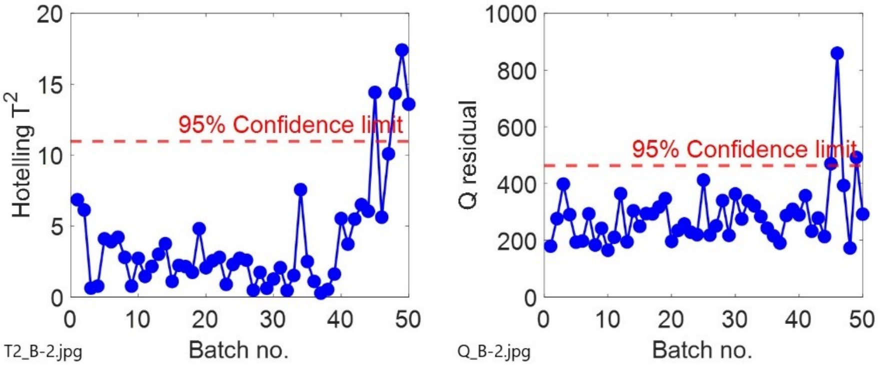


Figure 4.21. *Dataset 2. Score plot resulting from the batch-wise unfolded PCA. All 50 batches are reported with the 95% confidence limit (dashed line). Percentages in squared brackets represent the variance captured by the corresponding principal component*

The fact that samples are not randomly distributed is confirmed by the Hotelling T^2 and Q residual plot, reported in Figure 4.22.



(a) (b)

Figure 4.22. *Dataset 2. (a) Hotelling T^2 and (b) Q residual. In the T^2 plot, a path can be identified after batch no.40: assumption of random distribution of samples is not verified. 95% confidence limits are marked with dashed line*

In the Hotelling T^2, a path due to an increasing T^2 can be identified form batch no.40 to batch no.50: samples seem to be correlated and assumption of normal distribution is not verified. The same phenomenon is not present in the Q residual plot, where all values of residuals are randomly distributed.

As for §4.2.1, batch no.49 is reported as an example for validation.

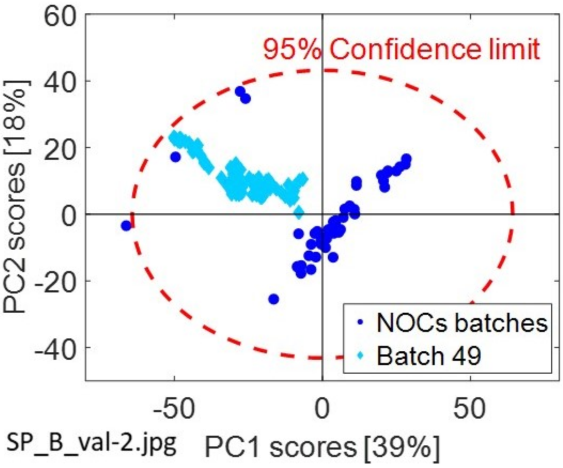
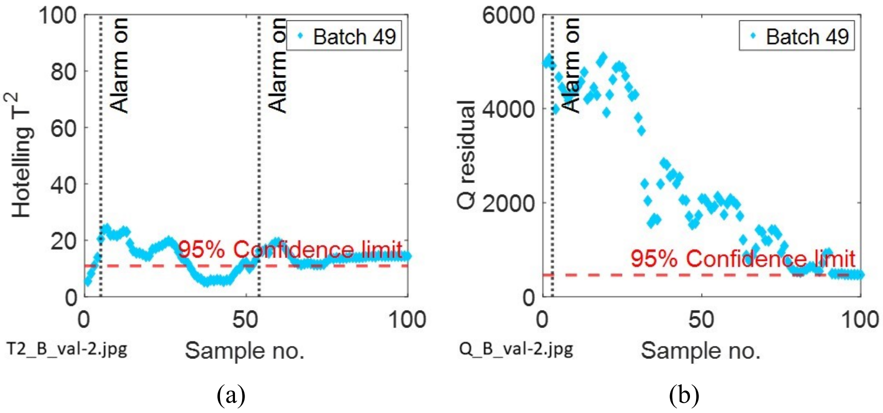


Figure 4.23. *Dataset 2. Score plot of the projection of batch no.49 onto the model. Dots represent normal batches, while diamonds represent evolution of the new batch at every time instant, from the center of the confidence region towards the 95% confidence limit (dashed line)*

The new batch is projected onto the model and the on-line monitoring is carried out as described in §1.2. The resulting score plot is reported in Figure 4.23: the batch starts around the center of the confidence region and evolves towards the limit dashed line.

The Hotelling T^2 and the Q residual are then considered and reported in Figure 4.24.



(a) (b)

Figure 4.24. *Dataset 2. Plots of (a) Hotelling T^2 and (b) Q residuals for batch no.49. Diamonds represent samples of batch no.49, while the dashed line marks the 95% confidence limit. Two alarms occur in the T^2 at sample no.5 and no.54. Values of Q residual are out of confidence limit since the beginning of the process and the alarm turns on at sample no.3*

The fault is detected for the first time by the model though the Q residual plot, indicating in the new batch a different correlation structure between variables with respect to the one of calibration batches. Residuals get closer to confidence limits during batch evolution, approaching the limit at the end of the process. Two alarms occur in the T^2 plot: one at sample no. 5 and one at sample no.54; after sample no.54, the batch remains out of confidence limits until the end of the process. The cause of the fault is investigated considering the contribution plot of T^2 and Q: in the Q contribution plots, Pressure 1 is the variable that exceeds the confidence limits since the beginning of the process. Considering the T^2 contribution, Pressure 2, Pressure 3, the Temperature 2 in the heating/cooling medium and the Flowrate 2 are all out of the upper confidence limit at the beginning of the process, while Pressure 1 starts out of the lower limit. Pressure 1 can be considered like the variable responsible of the fault: it is the only variable showing deviations in both contribution plots, moreover, exploiting the loading plot of Figure 4.13, it can be noticed that despite it is correlated to variables that in the T^2 contribution plot are in excess, in the same contribution plot it is in defect, showing a different correlation structure. Contribution plots for batch no.49 can be obtained through the validation_BWU.m file listed in Appendix 2.

Table 4.10. *Dataset 2: batch-wise model results*

Description	Value
Confidence limits	95%
False faulty batches	3/4
Missed fault detection	0/1
False alarms	12

All validations are performed, and results are reported in Table 4.10.

Although 4 validation batches are normal by literature, batches no.10, no.15 and no.39 present multiple alarms in the Q residual plot: 3 out of 4 normal batches are detected like faulty with a total of 12 false alarms. The only faulty batch available (batch no.49) is recognised like abnormal batch.

Similarly to dataset no.1, the batch-wise model is too sensible: 12 false alarms occur, and three out of four normal batches are recognized as faulty. On the other side, the assumption-free model is not able to recognize as faulty the only one abnormal batch available for validation. The cause of the over-sensitiveness of the batch-wise unfolded model is given by the control chart of the residuals. Therefore an appropriate adjustment of the confidence limit on the residuals may improve the performance of the model, thus reducing the occurrences of false alarms.

4.3 Dataset 3

This dataset contains data of a simulated fermentation of the Saccharomyces Cerevisiae cultivation. 40 batches are available for model calibration and 10 variables are measured for each of them for the entire process duration, which varies across batches. A description of the process and of datasets available is provided at §2.3, while all variables profiles are reported in Appendix 1 at §A.1.3.

4.3.1 Monitoring with an assumption-free model

In this case, the batches have different numbers of sampled time instants: since it is not possible to build a three-dimensional matrix using batches with this characteristic, data are already available in a variable-wise form. The variable-wise matrix to load into the Matlab script must include a first column containing the sampling time or the number of the sample; however, except for §4.1.1, this variable is always excluded from model calibration. A principal component analysis is performed on the dataset, adopting the RMSECV criteria for the selection of the number of principal components to retain into the model.

Table 4.11. *Dataset 3: variable-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	3.50	34.95	34.95	0.87
2	3.19	31.94	66.90	0.68
3	1.99	19.93	86.83	0.52
4	0.77	7.70	94.53	0.41
5	0.37	3.67	98.20	0.30
6	0.12	1.16	99.37	1.32

As reported in Table 4.11, the RMSECV reaches its minimum with 5 principal components and the total amount of variance captured by the model is equal to 87%.

The loading plot in Figure 4.25 shows correlations between variables: acetaldehyde, pyruvate and glucose concentration are all correlated and anti-correlated on PC1 (which is the component capturing the largest amount of variance) to the active cell material, the biomass concentration and the acetaldehyde dehydrogenase.

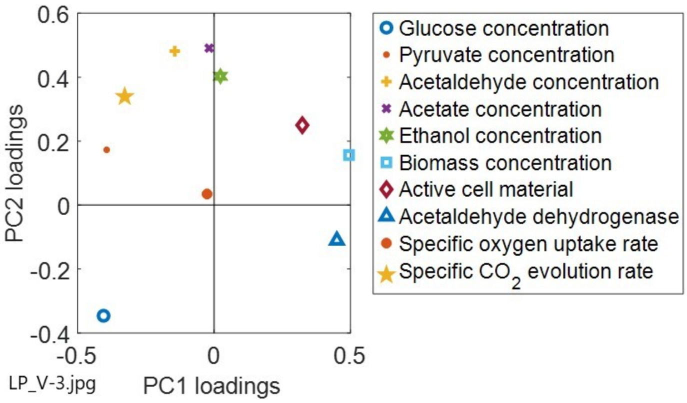


Figure 4.25. *Dataset 3. Loading plot resulting from the variable-wise unfolded PCA*

These relationships confirm the reasoning of Figure 2.3: the consumption of acetaldehyde, pyruvate and glucose leads to active cell material and biomass production.

In the score plot of Figure 4.26, all 40 calibration batches are reported. As indicated by arrows, the process starts in the third quadrant of the plot, corresponding to a condition of high glucose concentration, and evolves towards the right part of the plot, corresponding to high biomass concentration (end of the process).

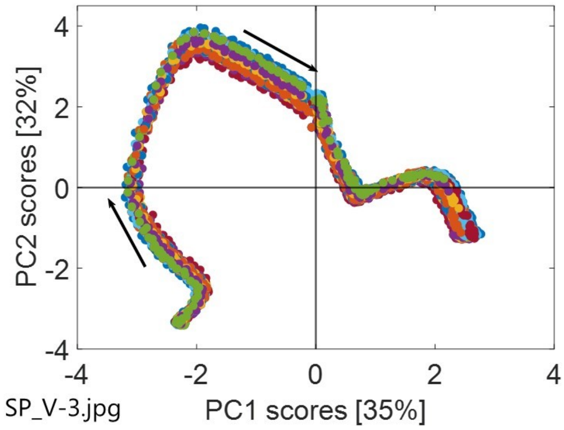


Figure 4.26. *Dataset 3. Score plot resulting from the variable-wise unfolded PCA. Arrows indicate the direction in which the process evolves. Percentages in squared brackets indicate the variance captured by the corresponding principal component*

The grid-search algorithm is applied with an initial grid of 15×15. At least the 95% if calibration scores (γ=95%) is required to be captured by cells that contain at least one sample of each batch (β=100%), as reported in Table 4.12.

Table 4.12. *Dataset 3: parameters set for grid-search algorithm*

Parameter no.	Symbol	Value	Unit
1	β	100	% of batches
2	γ	95	% of scores

The optimal grid found by the algorithm, shown in Figure 4.29, has a resolution of 4×7 cells and it is able to capture the 97% of scores (>γ=95%).

A 13-point resolution normal trajectory is obtained by interpolation of overall means calculated for each grid valid cell.

The 95% confidence limit is calculated for each cell with the method adopted by Westad et al. (2015) and with the new one proposed in this thesis, both described at §3.3.

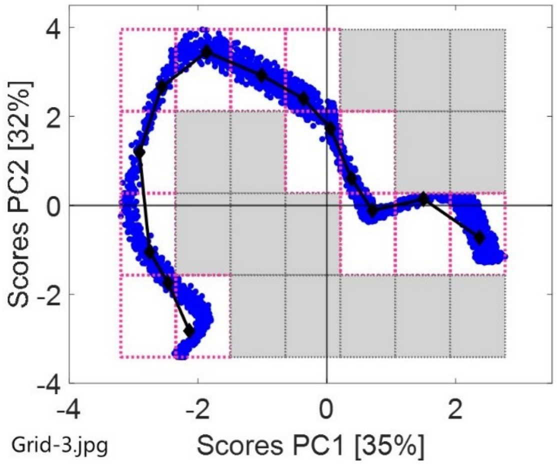
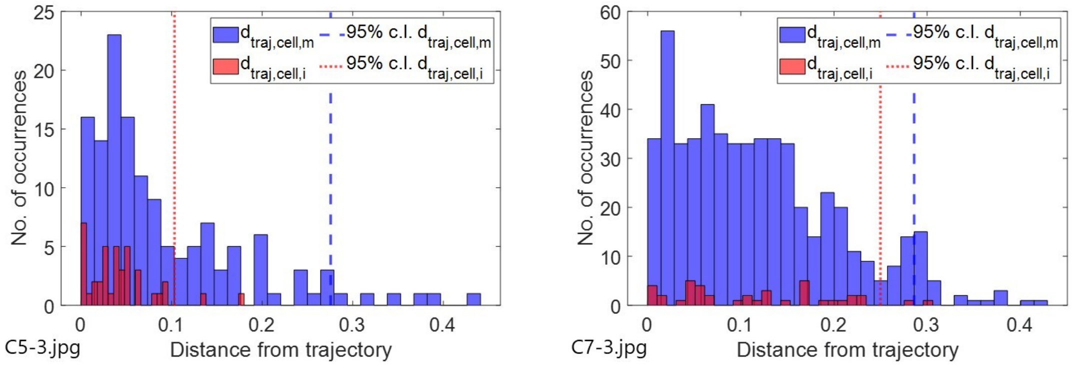


Figure 4.27. *Dataset 3. Optimal grid found by the grid-search algorithm. Valid cells are denoted with white background, while invalid cells have grey background. Normal trajectory is marked with solid line, while diamonds represent the means of scores calculated in each valid cell*

In Figure 4.28, cell no.5 and cell no.7 are reported as examples to show the difference between two limits calculated and their consequences.



(a) (b)

Figure 4.28. *Dataset 3. Plot of distances and of limits calculated for (a) cell no.5 and (b) cell no.7. Distances of single scores from trajectory are in blue, while distances of mean of scores of each batch from trajectory are in red. Dotted line marks the limit calculated with approach of* Westad et al. (2015)*, while the dashed line marks the limit calculated with the new approach*

Distances of the means of batches from the trajectory ($d_{traj,cell,i}$) are smaller than distances of single scores from the trajectory ($d_{traj,cell,m}$), in particular at the 5th trajectory point, reported in Figure 4.28 (a), the limit calculated with the Westad approach is more than one half smaller than the limit calculated with the new approach (i.e., limit calculated on distances of single scores from trajectory). The objective of the model is to allow the entire process monitoring, thus all the dynamics of the process should be represented by the model. However, if limits were calculated with the Westad approach, they would capture only the variability *across* different batches instead of the variability *within* the process: limits would be underestimated and lots of calibration scores would be out of the confidence limits, since their distance from trajectory is larger than the limit distance. The new approach proposed seems to be appropriate for monitoring purposes and it is chosen for limits calculation; then all points are interpolated and plotted following the direction of the trajectory, avoiding any intersection. At the end, the final model obtained is reported in Figure 4.29. Despite the limits are calculated on 95% confidence distance from trajectory, the total number of calibration scores out of confidence limits is about 11%: differently from cases in §4.1.1 and §4.2.1, here the model is affected by the fact that trajectory is more complex with respect to previous cases, and tight limits around corners left out lots of calibration scores. Moreover, it can be noticed that short and rapid changings in the direction of trajectory are not modelled by the grid-search algorithm: in particular at the beginning of the process, a rough trajectory modelling leads to very large confidence limits.

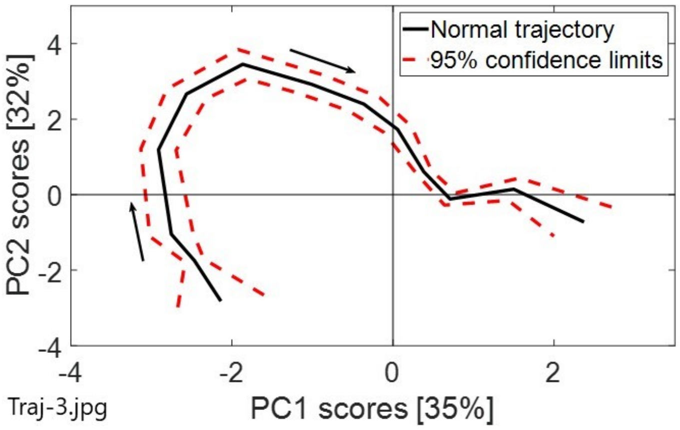


Figure 4.29. *Dataset 3. Final normal trajectory of the process obtained through the grid-search algorithm (solid line) with 95% confidence limits (dashed line)*

Alarms are calibrated on normal calibration batches: for fault detection, the minimum number of consecutive scores out of confidence limits is equal to 30, while the minimum number of consecutive residuals out of limits is 21.

The consequence of the low sensitivity of both alarms is that also if the fault occurs at the first time instant, it can’t be detected before the 21st time instant, with a substantial detection delay. According to the validation dataset description of Table 2.13, the first 25 batches are normal: an example is here reported considering the faulty batch no.36.

The new batch is projected onto the model and its 209-samples trajectory in the score plot is reported in Figure 4.30.

The batch starts inside confidence limits, deviates in the left direction in the second quadrant of the plot and then goes back inside the limits approaching the end of the process.

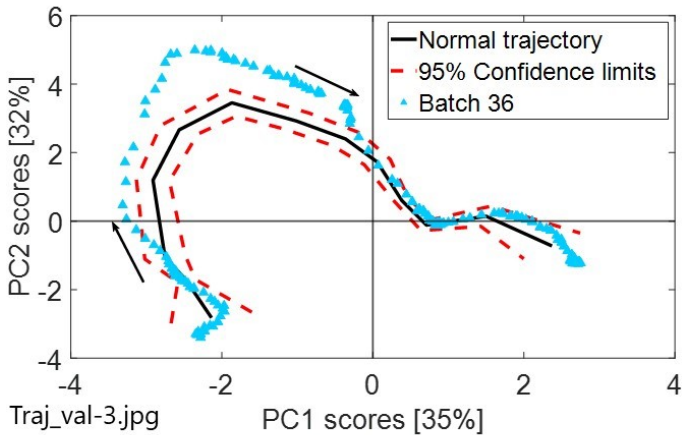
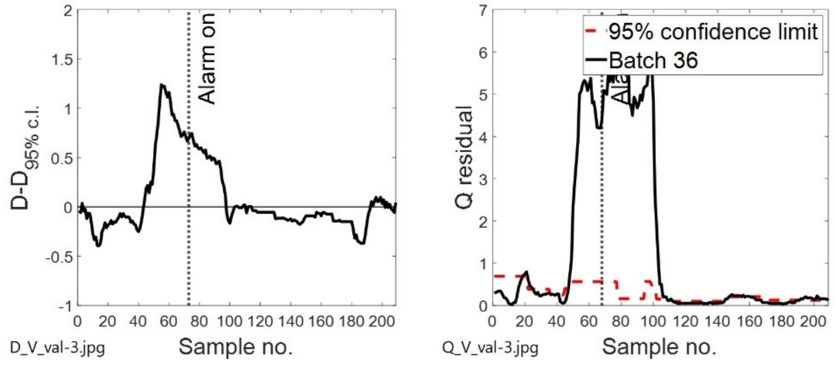


Figure 4.30. *Dataset 3. Projection of a new batch (batch no.36) onto the model. Normal trajectory and confidence limits are represented, respectively, by the solid line and the dashed lines. Triangles represent new batch scores*

The fault is detected firstly by the *Q* residual control chart of Figure 4.31 (b) at the 68th time instant, then an alarm occurs also in the distance-from-limit control chart of Figure 4.31 (a) at the 73rd sample. Since the fault firstly occurs in the profile of residuals along the process duration, it can be supposed that the responsible of the fault are one or more variables whose values are abnormal, giving a different correlation structure between all variables with respect to the one identified in the calibration dataset. Then, the fault appears also in the distance plot, which is the plot related to scores, because other variables are probably affected by the faulty one.



(a) (b)

Figure 4.31. *Dataset 3. Plot of (a) the difference between the distance from trajectory of a new sample (D) and the limit distance ($D_{95\%\ c.l.}$), and of (b) Q residuals along the sample number. The 95% confidence limit for residuals is reported (dashed line)*

The fault diagnosis is carried out considering the loading plot of Figure 4.25.

Since the deviation is in the north direction of the score plot, variables responsible of the fault should be looked for in the north-region of the loading plot, where acetaldehyde concentration, the specific CO_2 uptake rate and the acetate concentration are located. Also if the durations of batches are not the same, the projection of profiles of variables of the validation batch along the process duration, together with profiles of variables of calibration normal batches, is useful to identify which one, between variables mentioned before, is the real responsible of the fault. Although it is only a qualitative investigation because samples of batches are not synchronized, it is sufficient to conclude that the fault consists in an excessive acetaldehyde concentration. The exact time at which the fault occurs and the exact type of fault are not known, however considering that the alarm in the residual plot activates only after at least 21 scores out of confidence limit, and assuming that the trajectory of the new batch starts to deviate from the normal one as soon as the fault occurs, it can be concluded that the anomaly takes place probably after about 47 time instants after the process beginning. Among the possible types of fault described in literature and reported at §2.3.3, this is probably the case of the fault in the ethanol formation from acetaldehyde: the reaction rate is slower than expected, so acetaldehyde accumulates into the reactor while the ethanol formation is delayed.

The scores of the new batch are projected into trajectory and the relative time is estimated as described in §3.4. In the score plot accumulations of scores occur in the initial and final part of the process: considering Figure 4.32, this aspect reflects in a slower process dynamics during the first 25 time instants and the last 59 time instants.

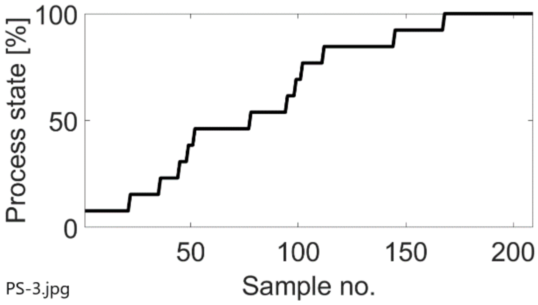


Figure 4.32. *Dataset 3. Plot of the process state (or relative time) against sample number of batch no.36. The process dynamics is slower in the initial and last phases of the process*

The lag in the initial time instants corresponds to the phase in which yeast acclimates to the heterogeneous media at the beginning of the process, while at the end of the process, concentration of ethanol (which is the main reactant for biomass growth when glucose is completely consumed) is very small and reaction slows down. Note that the process state does not indicate the rate of reactions, but it is an index of the relative position of the scores of the new batch with respect to normal trajectory: if the process state doesn't advance, it means that scores accumulate in a point in the score plot and no variation in the process variables occurs.

Table 4.13. *Dataset 3: assumption-free model results*

Description	Value
Confidence limits	95%
Calibration scores out of c.l.	11%
Consecutive scores out of c.l. for alarm on scores	30
Consecutive scores out of c.l. for alarm on residuals	21
False faulty batches	0/25
Missed fault detection	6/30
False alarms	N/A

All batches available for validation are projected onto the model and final results are summarized in Table 4.13.

25 normal batches result to be normal, so the validation procedure ends up without false faulty batches and no false alarm is present for normal batches. 6 out of 30 faulty batches are classified like normal by the model, since no fault is recognized neither in the score plot nor in the plot of residuals. In the other faulty batches, the fault is detected by the model; however, since the time at which the fault occurs is not notified in literature, it is not possible to determine if some alarms appearing in faulty batches are false.

4.3.2 Monitoring with a batch-wise unfolded MPCA model

Calibration batches with different lengths are synchronized using the "multi-synchro" method of the MVBatch Toolbox available for Matlab, as explained in §2.3.2, and organized into a *I*×*JK* batch-wise unfolded matrix. The synchronization procedure is a disadvantage in this case, because differently with respect to the assumption-free model, the data used to calibrate the batch-wise model are not the original ones directly measured on the reactor, but they depend on the method used for synchronization. A PCA model is calibrated, adopting the RMSECV criteria for the selection of the number of principal components to retain into the model, as described in §1.1.

Table 4.14. *Dataset 3: batch-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	595.59	28.50	28.50	0.92
2	117.32	5.61	34.11	0.92
3	111.72	5.35	39.46	0.90
4	89.24	4.27	43.73	0.90
5	89.08	4.26	47.99	0.89
6	82.57	3.95	51.94	0.87
7	69.87	3.34	55.28	0.85
8	57.35	2.74	58.03	0.84
9	53.20	2.55	60.57	0.84

As reported in Table 4.14, 8 principal components capture the 58% of total variance of the dataset and make a PCA model with a RMSECV of 0.84. Principal component no.8

corresponds to the second "elbow" of the RMSECV curve as a function of the number of principal components: the first "elbow" is at three principal components, however the cumulative variance explained in this case is only about 40%, so 8 PCs are preferred.

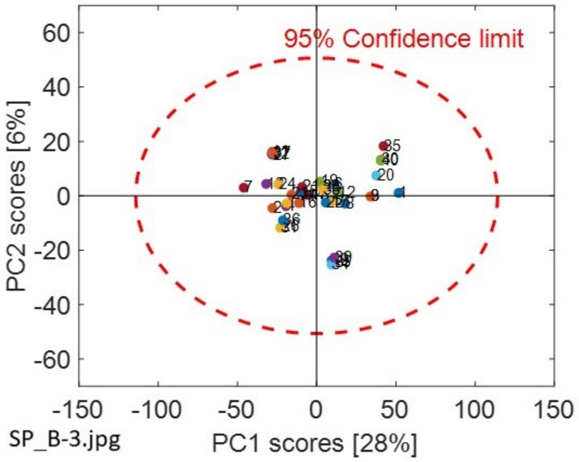
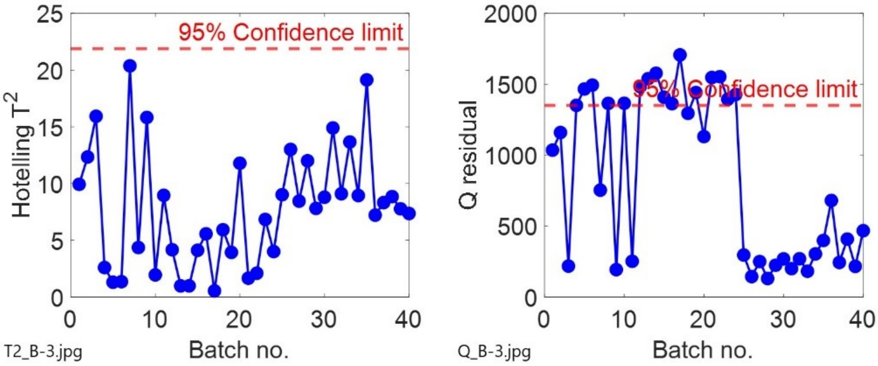


Figure 4.33. *Dataset 3. Score plot resulting from the batch-wise unfolded PCA. All 40 batches are reported with the 95% confidence limit (dashed line). Percentages in squared brackets represent the variance captured by the corresponding principal component*

The score plot of the model is reported in Figure 4.33: 40 calibration batches are projected and the confidence limit is calculated according to the assumption of normal distribution of scores, as explained in §1.1; however in this case it is evident that samples are not normally distributed in the score space and all batches result to be inside confidence limits.



(a) (b)

Figure 4.34. Dataset 3. *(a) Hotelling T^2 and (b) Q residual. 95% confidence limits are marked with dashed line. The assumption of random distribution in both cases is not verified*

In the example reported by González-Martínez et al. (2018), this dataset is used to calibrate a multi-phase model (and not a batch-wise model), and confidence limits are adjusted by the operator; however, the multi-phase model and the adjustment of limits are not part of this thesis. The batch-wise unfolded model is used, remembering that the lack of normal distribution makes the confidence limits calculated for the T^2 and Q residuals not completely reliable. The total amount of variance explained by the first two principal components is only about 34%, which is a quite small value.

Hotelling T^2 plot and the Q residual plot are reported in Figure 4.34 (a) and Figure 4.34 (b), respectively. T^2 of batches are not normally distributed and a path given by an increasing T^2 along the batch number after batch no.20 can be identified. Considering the residuals, also in this case the assumption of random distribution is not verified: for the first 20 batches, a large number of residuals is over the 95% confidence limit, then the residuals of remaining batches all lie below the limit.

An example of validation is reported here considering the same validation batch used for the assumption-free model in the previous section (batch no.36).

An online monitoring is carried out according to the procedure described in §1.2 and the resulting batch trajectory in the score plot is reported in Figure 4.35.

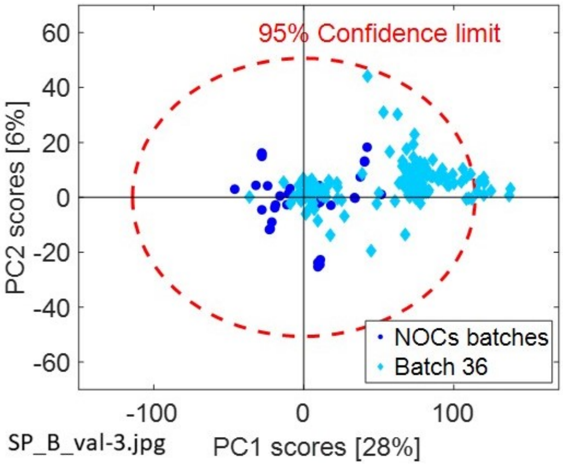
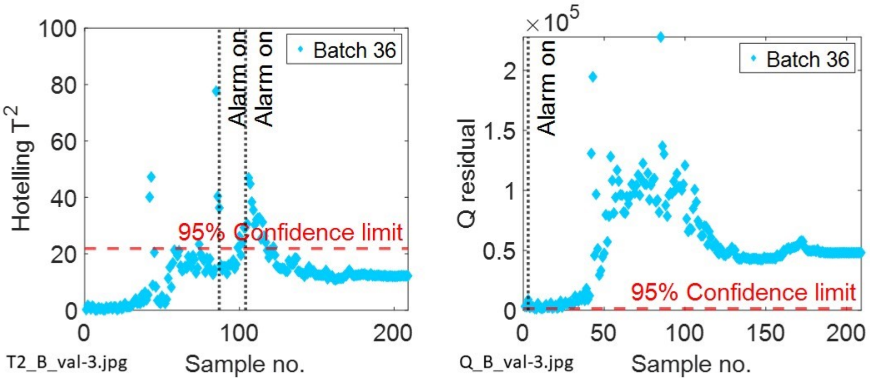


Figure 4.35. *Dataset 3. Score plot of the projection of the new batch (batch no.36) onto the model. Dots represent normal batches, while diamonds represent evolution of the new batch at every time instant, from the center of the score space to the confidence limit. The 95% confidence limit is marked with dashed line*

The new batch projected starts its process inside the confidence area and then evolves in the right direction, ending out of the right boundary of the region. Clearly, this can be classified like a faulty batch and the plots of the Hotelling T^2 and the Q residual are inspected to identify the time at which the fault occurs. Considering the T^2 plot, two alarms occur at samples no.87 and no.104, but after the second alarm the process goes back below the confidence limit, where it remains until its end. On residuals, the alarm turns on immediately at sample no.3: from this time residuals are always higher than the 95% confidence value and the alarm remains active until the end of the process.

The contribution plots of the residuals, which can be obtained through the validation_BWU.m script, show that at the beginning of the process a fault occurs because the concentration of glucose is lower than expected.



(a) (b)

Figure 4.36. *Dataset 3. Plots of (a) Hotelling T^2 and (b) Q residuals for batch no.36. Diamonds represent samples of batch no.36, while the dashed line marks the 95% confidence limit. Two alarms occur in the T^2 at sample no.87 and no.104. In the Q residual plot, alarm starts at sample no.3*

However the residuals in Figure 4.36 increase dramatically after about 50 samples: at this time the *Q* contribution of the acetaldehyde concentration increases rapidly, while the ethanol and the biomass concentrations result to be lower than expected (i.e., lower than normal operating conditions). The fault diagnosis is the same of the assumption-free model: the cause of the fault is a slower reaction rate regarding the ethanol formation from acetaldehyde, resulting in an accumulation of acetaldehyde in the reactor and a lower ethanol formation. Also if the residuals are slightly out of the confidence limit since the beginning of the process, the real fault can be assumed taking place after about 50 time instants, when both the Hotelling T^2 and the residuals start to increase.

Validations with all batches available are carried out and results are summarized in Table 4.15.

Table 4.15. *Dataset 3: batch-wise model results*

Description	Value
Confidence limits	95%
False faulty batches	25/25
Missed fault detection	0/30
False alarms	N/A

The 95% confidence limit for residuals is calculated on the assumption of random distribution of residuals of calibration normal batches; however, as mentioned before, in this case this assumption is not verified. The consequence of this situation is that, according to the *Q* residual plot, all normal batches available for validation are detected like faulty and lots of false alarms occur. The difference observed between residuals in normal batches and residuals in faulty batches consists in the magnitude of the Q statistic: in faulty batches residuals are much larger than residuals in normal batches. Since it is not possible to avoid alarms in normal batches only considering the number of residuals out of confidence limits, in order to avoid discarding normal batches, also the magnitude of the fault should be inspected. In all faulty batches the fault is detected by the alarm on the residual plot and, in most of the cases, also by the alarm in the T^2 plot. Since it is not known the time at which faults occur, it is not possible to determine if false alarms occur in abnormal batches.

This model classifies as faulty all normal batches; therefore several false alarms occur, while the assumption-free model is able to recognize all normal batches without any false alarms, and only 6 out of 30 faulty batches are not detected. For this dataset, the assumption-free model is probably more suitable for process monitoring: although in some cases batches that are not in normal operating conditions are not recognized, it avoids the rejection of all good batches.

4.4 Dataset 4

The dataset available contains 16 normal batches of a baker's yeast fermentation for which 7 variables are measured at the same number of time instants (83 time instants). A description of the process is provided by George et al. (1998) and reported also at §2.4.1, together with the summary of datasets available for calibration and validation. All profiles of variables are reported in §A.1.4.

4.4.1 Monitoring with an assumption-free model

Calibration batches available in a three-dimensional matrix are organized into a variable-wise matrix. An additional column representing time is added to the unfolded matrix, similarly to previous datasets, to allow the algorithm to recognize batches (this additional time column is not considered for model calibration).

Table 4.16. *Dataset 4: variable-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	2.94	42.01	42.01	0.86
2	2.11	30.14	72.15	0.71
3	0.86	12.32	84.47	0.71
4	0.49	7.07	91.54	0.82
5	0.35	5.00	96.54	1.47

A principal component analysis is then carried out, selecting the number of principal components to retain into the model according to the RMSECV criteria.

As reported in Table 4.16, with a model built on 2 principal components, the cumulative variance captured is about 72%, which means that the model is able to represent the largest amount of data. The *J*×*NPCs* loading matrix is plotted in the two-dimensional space of Figure 4.37, to identify the general correlation structure between variables.

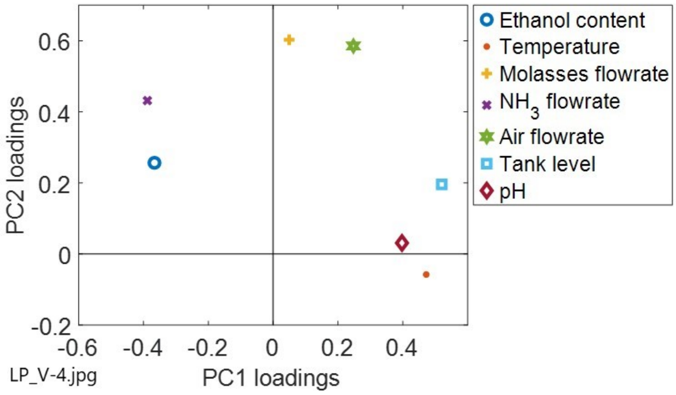


Figure 4.37. *Dataset 4. Loading plot resulting from the variable-wise unfolded PCA*

Considering the first principal component, a decrease in the ethanol content and in the ammonia flowrate is associated to an increase of the tank level, the pH, the air flowrate and the temperature: ethanol and ammonia are consumed to produce yeast, so the tank level increases when raw materials decrease.

From the score plot of Figure 4.38, it is evident that trajectories of scores in the score space are less close to each other than in the previous cases; in particular, at the beginning and at the end of the process, the variability is very strong.

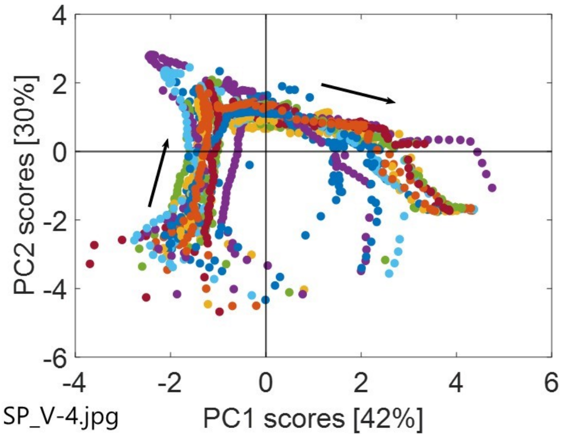


Figure 4.38. *Dataset 4. Score plot resulting from the variable-wise unfolded PCA. Arrows indicate the direction on which the process evolves. Percentages in squared brackets indicate the variance captured by the corresponding principal component*

The grid-search algorithm is run on the score plot of Figure 4.38 starting form an initial resolution of 15×15 cells.

At least the 95% of the calibration scores is required to be captured by the grid cells, each one containing at least one score for each batch, as indicated by the β and γ values of Table 4.17.

Table 4.17. *Dataset 4: parameters set for grid-search algorithm*

Parameter no.	Symbol	Value	Unit
1	β	100	% of batches
2	γ	95	% of scores

At the end, the optimal grid found by the algorithm (Figure 4.39) has a grid resolution of 4×2, i.e., a very coarse one: it is able to capture the 99% of calibration scores and allows to obtain a 5-point trajectory.

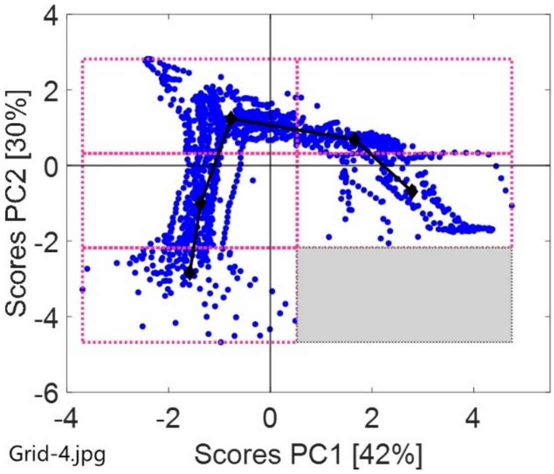
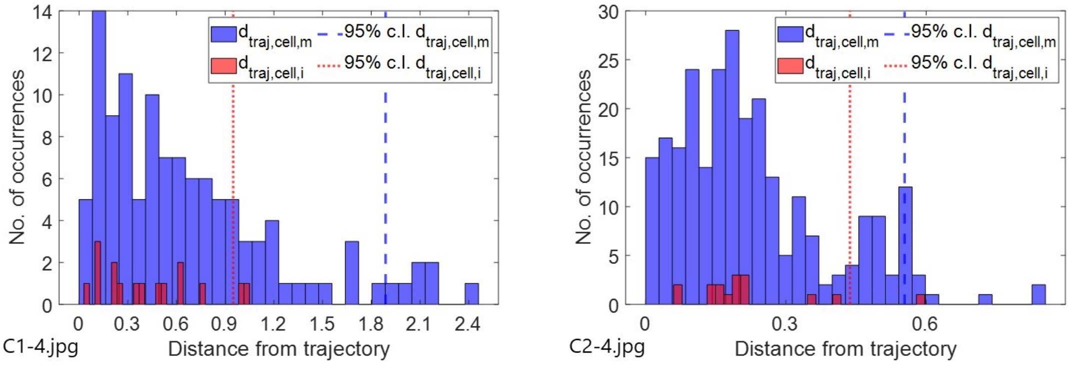


Figure 4.39. *Dataset 4. Optimal grid found by the grid-search algorithm. Valid cells are denoted with white background, while invalid cells have grey background. Normal trajectory is marked with solid line, while diamonds represent the means of scores calculated in each valid cell*

For each valid cell, the distance for the 95% confidence limit is calculated with the Westad et al. (2015) approach and with the new one proposed: two examples of limits calculated for cell no.1 and cell no.2 are reported in Figure 4.40: in cell no.1, the limit calculated with the approach proposed by Westad et al. (2015), and marked with dotted line, results to be about one half smaller than the distance for the limit calculated with the new approach, marked with dashed line, while in cell no.2 the difference between two limits calculated is smaller; however, for all cells, the limit calculated with Westad et al. (2015) procedure is always smaller than the limit calculated on distances of all single calibration scores from trajectory (new approach). Looking at the two histograms of Figure 4.40, it is evident that lots of scores have distance from the trajectory which is larger than the limit distance calculated with the Westad et al. (2015) approach: using the $d_{traj,cell,i}$ as limit distance, lots of scores (much more than 5%) would be left out of confidence limits.

The new approach is used to calculate the limit distance for each cell, then all points are interpolated avoiding any intersection with the normal trajectory, as showed in Figure 4.41.



(a) (b)

Figure 4.40. *Dataset 4. Plot of distances and of limits calculated for (a) cell no.1 and (b) cell no.2. Distances of single scores from trajectory are in blue, while distances of mean of scores of each batch from trajectory are in red. Dotted line marks the limit calculated with approach of* Westad et al. (2015)*, while the dashed line marks the limit calculated with the new approach*

Projecting the calibration scores onto the new normal trajectory model, the 6% of calibration scores result to be out of the 95% confidence limits: despite the limits are calculated only at 5 points and then joint by interpolation, the approximation is appropriate. It can be noticed that at the beginning and ta the end of the process, the large variance between scores leads to very wide confidence limits around the normal trajectory.

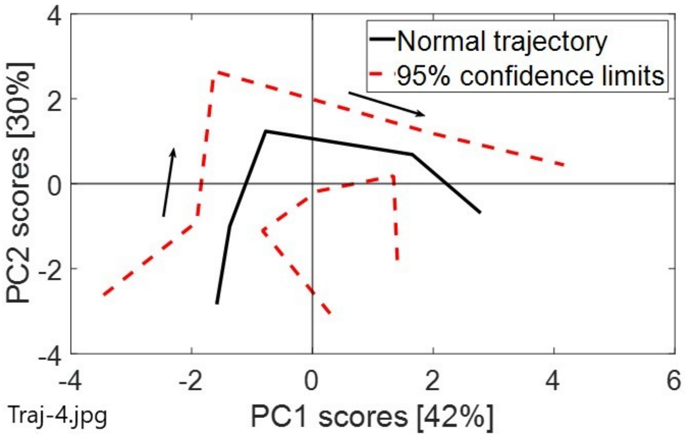


Figure 4.41. *Dataset 4. Final normal trajectory of the process obtained through the grid-search algorithm (solid line) with 95% confidence limits (dashed line)*

The two alarms of the score plot and of the *Q* residual plot are calibrated on scores of normal calibration batches: in the first one, the maximum number of consecutive scores allowed to be out of confidence limits is equal to 22, therefore at the 23rd score out of confidence limits an alarm turns on; in the second case, the alarm turns on at the 24th consecutive value of residual over the limit. The low sensitivity of the alarms (i.e., the high number of consecutive scores out of confidence limits before alarm activation) is due to some calibration batches (3 batches) that deviate from the others in the north-west region of the score plot. Among the 17 batches available, batch no.16 has been arbitrarily selected as a new batch to provide an example of validation.

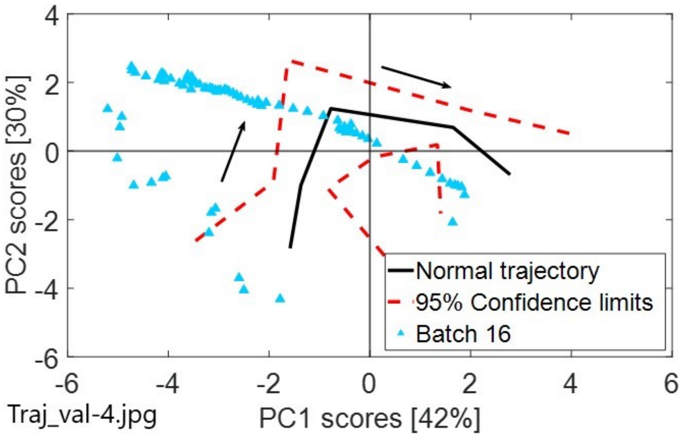
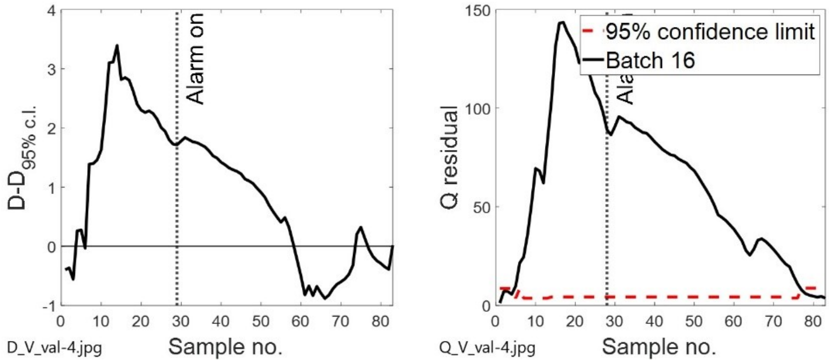


Figure 4.42. *Dataset 4. Projection of a new batch (batch no.16) onto the model. Normal trajectory and confidence limits are represented, respectively, by the solid line and the dashed lines. Triangles represent new batch scores*

The new batch is projected onto the model and its trajectory of scores is represented in the score plot of Figure 4.42. After 4 time instants, samples are already out of the confidence limits, following a trajectory (triangles) much wider than the normal one (solid line): the fault is probably present since the beginning of the process, however it is detected only after 29 time instants because of the low-sensitivity of the alarm in the score plot.



(a) (b)

Figure 4.43. *Dataset 4. Plot of (a) the difference between the distance from trajectory of a new sample (D) and the limit distance ($D_{95\%\ c.l.}$), and of (b) Q residuals along the sample number. The 95% confidence limit for residuals is reported (dashed line)*

In Figure 4.43 (a) the time at which the alarm turns on (29th sample) is marked with dotted line. Considering the plot of residuals in Figure 4.43 (b), the profile is similar to the one of *D-D*$_{95\%c.l.}$ of Figure 4.43 (a): after few time instants below the confidence limit, it increases rapidly and makes the alarm starting at the 28th sample. The loading plot of Figure 4.37 is useful for the fault diagnosis: the ethanol content, the molasses flowrate and the ammonia flowrate are probably variables responsible of the fault. Plotting the profiles of variables of the new batch together with profiles of variables of calibration normal batches (for this purpose, the validation_VWU.m script can be used), it is evident that the ethanol content is much higher than normal since the beginning of the process, while the ammonia and molasses flowrates don not increase as expected at about half of the process.

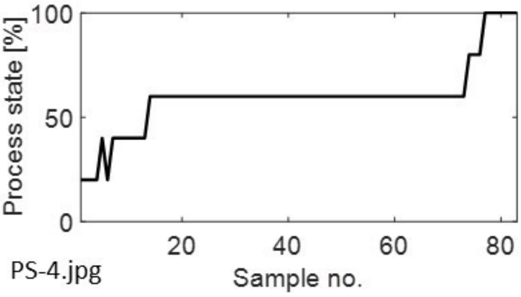


Figure 4.44. *Dataset 4. Plot of the process state (or relative time) against sample number of batch no.16. The process seems to be static for more than 30 samples*

Projecting the scores of the new batch onto the trajectory, it is possible to estimate the process state (or relative time) in order to identify whether the process is regular, delayed or in advance. In this case, the profile of the process state of Figure 4.44 does not represent correctly the dynamics of the process: the process seem to be static from about sample no.20 to sample no.75; however looking at its trajectory in Figure 4.42, this conclusion is wrong. In fact, scores do not accumulate around a single point, but they follow a diagonal trajectory from the second to the fourth quadrant. All batches available for validation are used to test the model and final results are reported in Table 4.18.

Table 4.18. *Dataset 4: assumption-free model results*

Description	Value
Confidence limits	95%
Calibration scores out of c.l.	6%
Consecutive scores out of c.l. for alarm on scores	23
Consecutive scores out of c.l. for alarm on residuals	24
Normal batches	14
Faulty batches	3
False alarms	N/A

As mentioned also in the validation dataset description at §2.4.3, it is not known which are the faulty batches and the normal ones, therefore the number of normal and abnormal batches is reported, instead of the number of false faulty batches and missed fault detections. At the end, batches no.1, no.8 and no.16 seem to be faulty, while all of the others are recognized as normal.

A particular case is represented by batch no.9: despite it is clearly faulty by a first inspection of the score plot of Figure 4.45, the batch is not recognised as abnormal, because all of its scores are inside the confidence limits, also if the trajectory is not similar to the normal one and scores are spread around the confidence region, in particular in the last part of the process.

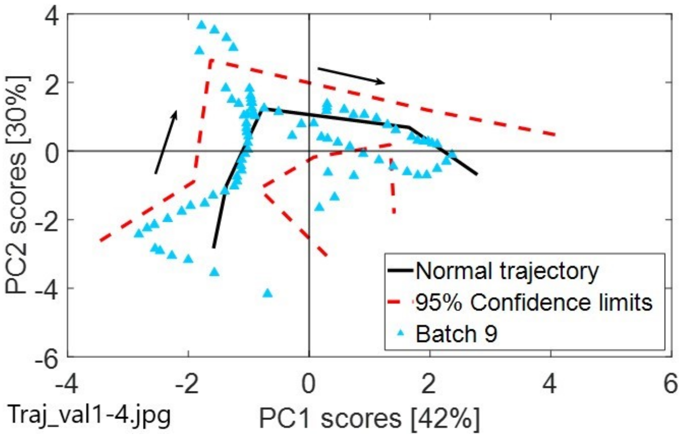


Figure 4.45. *Dataset 4. Projection of batch no.9 onto the model. Normal trajectory and confidence limits are represented, respectively, by the solid line and the dashed lines. Triangles represent scores of the new batch (batch no.9)*

The Q residual plot of Figure 4.46 confirms that batch no.9 is faulty, however the fault occurs too late to be detected by the alarm, that is calibrated on 24 consecutive residuals over the limit.

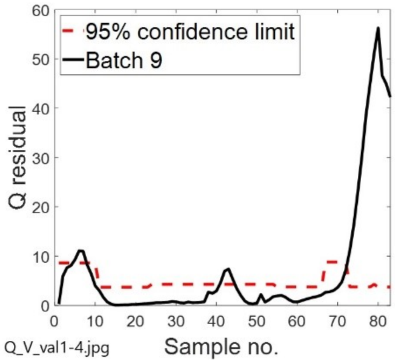


Figure 4.46. *Dataset 4. Q residuals along the sample number of batch no.9. The 95% confidence limit for residuals is reported (dashed line)*

Two limitations of the assumption-free model are highlighted by this case study: 1) the model is not able to detect the anomaly if scores are inside confidence limits but in the wrong location, because the alarm is calibrated only on the number of consecutive scores out of confidence limits, but does not consider their overall path; 2) if alarms have low sensitivity (i.e., if they activate after a large number of consecutive scores or values out of confidence limits), the fault may not be detected before the process end.

4.4.2 Monitoring with a batch-wise unfolded MPCA model

All 16 calibration normal batches have the same number of samples (83 samples) and no alignment is needed to organize data into a *I*×*JK* batch-wise unfolded matrix. Adopting the RMSECV criteria for the selection of the number of principal components to retain into the model, a PCA is performed, then results are summarized in Table 4.19.

Table 4.19. *Dataset 4: batch-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	175.47	30.20	30.20	2.23
2	89.80	15.46	45.66	2.26
3	77.05	13.26	58.92	2.25
4	61.79	10.63	69.55	2.23
5	50.53	8.70	78.25	2.2
6	34.38	5.91	84.17	2.22

Five principal components are used to build the PCA model, which is able to capture a consistent amount of variance (78%) of the initial dataset. At this number of principal components, the RMSECV is not at its minimum, but at its "elbow" point, as discussed at §1.1. All samples of 16 calibration normal batches are projected onto the score space (Figure 4.47) considering the first two principal components.

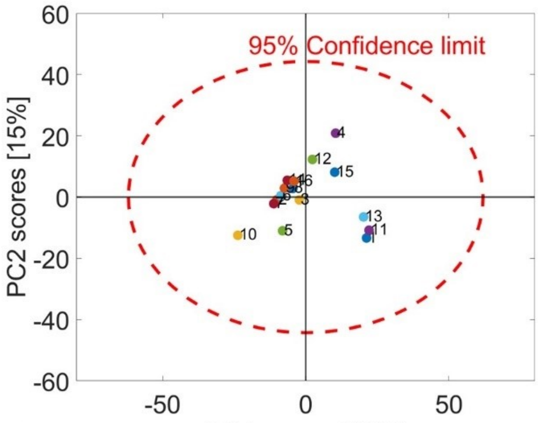
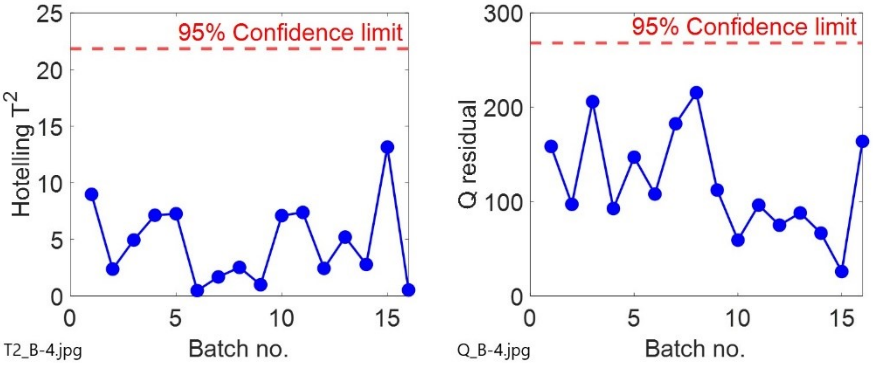


Figure 4.47. *Dataset 4. Score plot resulting from the batch-wise unfolded PCA. All 16 batches are reported with the 95% confidence limit (dashed line). Percentages in squared brackets represent the variance captured by the corresponding principal component*

The scores of Figure 4.47, each one representing one calibration batch, are not randomly distributed around the origin the coordinate system: a diagonal cluster can be identified close to the axes origin, while three batches (batches no.1, no.11 and no.13) are located in the fourth quadrant, further from others. The assumption of random distribution is not properly verified also looking at the plot of residuals in Figure 4.48 (b): batches in the second half of the dataset (batches no.9-16) have a residual smaller than the one of batches in the first half of the dataset (batches no.1-8).

In the T^2 plot in Figure 4.48 (a), no path along the sample number can be identified; however, saying that the assumption of random distribution is verified is not correct. In both cases, all values are below the 95% confidence limit: limits are affected that the assumption of normal distribution, on which they are calculated, is not verified.



(a) (b)

Figure 4.48. *Dataset 4. (a) Hotelling T^2 and (b) Q residual. 95% confidence limits are marked with dashed line*

An example of validation is reported here considering as a new batch the same one used for validation in §4.4.1: batch no.16. Future missing time instants are computed as described in §1.2, then the entire trajectory of the batch is projected onto the score plot, together with calibration batches and the 95% confidence ellipse.

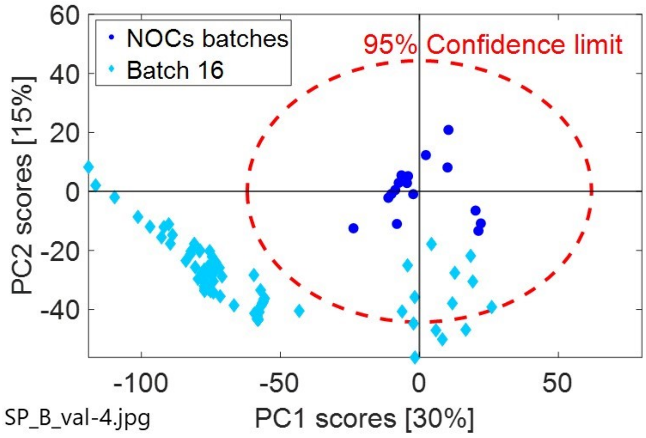
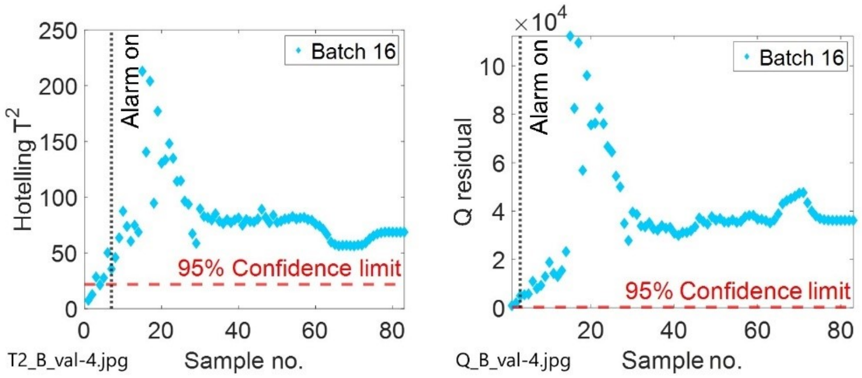


Figure 4.49. *Dataset 4. Score plot of the projection of the new batch (batch no.16) onto the model. Dots represent normal batches, while diamonds represent evolution of the new batch at every time instant. The 95% confidence limit is marked with dashed line*

From Figure 4.49, it is clear that batch no.16 (marked with diamonds) has a fault at the beginning of the process: its trajectory starts inside the 95% confidence area and evolves rapidly towards the left-direction in the score space, ending far from the 95% confidence limit (dashed line). Even if the initial scores of the new batch are inside the confidence ellipse, they are located far from scores of normal calibration batches; therefore a fault can be suspected already at the beginning of the process, before waiting for scores to go out of confidence limits. Plots of T^2 and Q residual of Figure 4.50 are useful to identify the exact time at which the fault is recognized by the batch-wise unfolding model. In the T^2 control chart, the alarm turns on at the 7th time instant, while the alarm on residuals activates immediately at the 3rd time instant: since the alarm is calibrated on 3 consecutive values over the confidence limit, residuals of the new batch are over the limit since the first sample: the fault is present at the beginning of the process.



(a) (b)

Figure 4.50. *Dataset 4. Plots of (a) Hotelling T^2 and (b) Q residuals for batch no.16. Diamonds represent samples of batch no.16, while the dashed line marks the 95% confidence limit. T^2 alarm turns on at sample no.7. In the Q residual plot, alarm starts at sample no.3*

The cause of the fault is investigated through the contribution plots of the Hotelling T^2, that can be obtained with the validation_BWU.m file reported in Appendix 2. The ethanol content, the tank level and the temperature are all below their confidence limit in the T^2 contribution plot, while the molasses flowrate is larger than expected.

Table 4.20. *Dataset 4: batch-wise model results*

Description	Value
Confidence limits	95%
Normal batches	0
Faulty batches	17
False alarms	N/A

All batches available for validation are projected onto the model to test it. At the end, the Q residual control chart results not to be reliable for fault detection, because multiple alarms occur for all abnormal and normal batches.

As mentioned in the previous section, since the identity of validation batches is not known, it is possible to report only the number of normal and abnormal batches detected by the model.

Table 4.20 reports that, according to the model, all batches available for validation are faulty, due to the fact that Q residual control charts gives alarms for all batches.

For this dataset, it is not possible to carry out a model comparison considering the number of false faulty batches and the number of missed fault detection, because the identity of the validation batches (i.e., whether they are normal or faulty) is not known. Therefore, in this case the number of batches recognised as faulty is considered. The batch-wise model detects as faulty all of the 17 batches; however, it is known that some of the validation batches are normal. The assumption-free model, instead, recognise as faulty only 3 batches; however it has been noticed that in some cases (e.g., batch no.9) it is not able to detect the anomaly, therefore the exact number of faulty batches is greater than 3, but less than 17. The batch-wise model is the one which detects the largest number of faulty batches, but several false alarms occur for sure.

4.5 Dataset 5

The available dataset contains samples of 30 normal batches for an herbicide production. 10 variables are measured for a number of time instants that varies across batches. A description of the process (García-Muñoz et al., 2003) and all details about the datasets available are provided at §2.5, while all profiles of variables, with different batch durations, are reported in at §A.1.5.

4.5.1 Monitoring with an assumption-free model

Available data are organized into a variable-wise unfolded matrix, that is used to calibrate a PCA model. According to the RMSECV criteria, which is used to choose the number of principal components to retain into the model, two PCs are selected.

Table 4.21. *Dataset 5: variable-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	5.45	54.50	54.50	0.72
2	1.27	12.73	67.23	0.72
3	1.00	10.00	77.24	0.88
4	0.80	7.96	85.20	3.11
5	0.62	6.18	91.38	3.41

As reported in Table 4.21, the minimum of the RMSECV is 0.72, while the cumulative variance explained by the model is about 67%, which means that the largest amount of information contained in the dataset is represented by the model.

The loading plot of Figure 4.51 allows to capture correlations among variables: considering the first principal component, high values of the jacket temperature, of the dryer set point temperature, of the torque and of the agitator power, are associated to low values of the collector tank level, of the dryer temperature and pressure, and of the agitator speed.

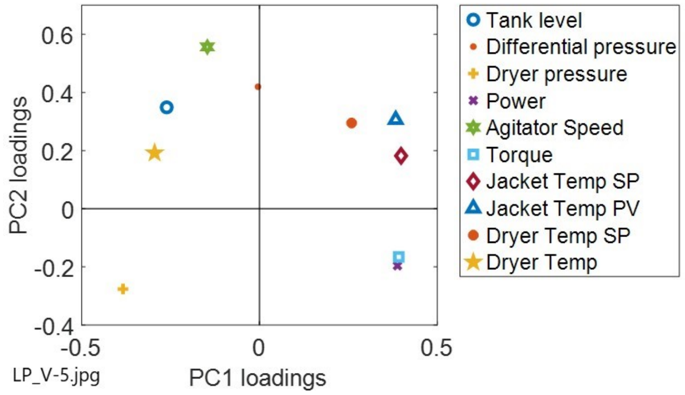


Figure 4.51. *Dataset 5. Loading plot resulting from the variable-wise unfolded PCA*

When the process starts, the level in the collector tank is zero, the temperature and pressure inside the reactor are quite low, while the power to deliver to the agitator and the torque required are very high due to the high amount of solvent contained in the cake inside the reactor. When the humidity of the product is lower, the power and torque are lower too; however, the last part of the process is the most energy expensive due to the solvent to evaporate, which is present in low concentration when the product is almost completely dry.

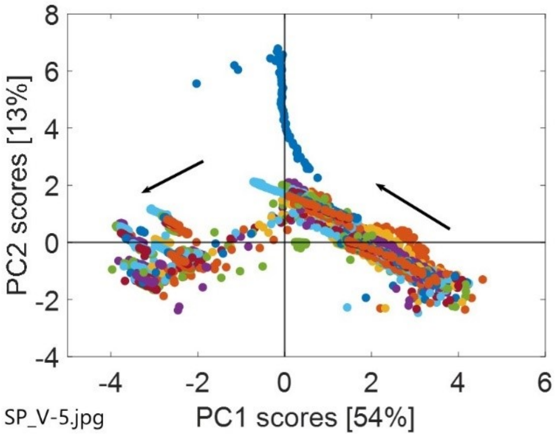


Figure 4.52. *Dataset 5. Score plot resulting from the variable-wise unfolded PCA. Arrows indicate the direction on which the process evolves. Percentages in squared brackets indicate the variance captured by the corresponding principal component*

In the score plot of Figure 4.52, all 30 calibration normal batches are reported with their trajectory, and black arrows indicate that the process evolves from right to left. The fact that the process dynamics is not linear is evident from the accumulation of points in the first part of the process, while they are less dense in the second half: this phenomena can be associated to the fact that in the first part of the process lots of variables are kept constant for almost half of the process duration, therefore the dynamics of the process is given only by variables that change continuously (e.g., the torque); then, towards the end of the process, lots of variables step-change and the trajectory of scores "jumps" from the right-side to the left one on the score space. Batch no.1 has a trajectory which deviates a lot from others, moving towards the upper part of the score plot: this has a consequence on alarm sensitivity, as will be discussed later.

Table 4.22. *Dataset 5: parameters set for grid-search algorithm*

Parameter no.	Symbol	Value	Unit
1	β	100	% of batches
2	γ	95	% of scores

Parameters set in the grid-search algorithms are kept constant also for this dataset: at least the 95% is required to the captured by the valid cells of the grid, and at least one score of each batch must be included in one cells in order to consider it as valid, as reported in Table 4.22. The initial grid resolution adopted is 15×15, which is high enough to obtain a final grid with a lower resolution.

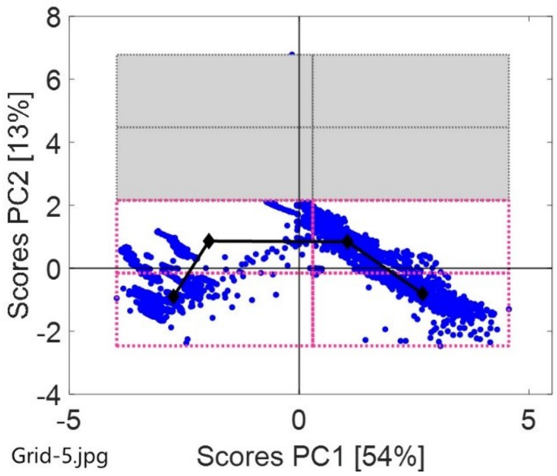
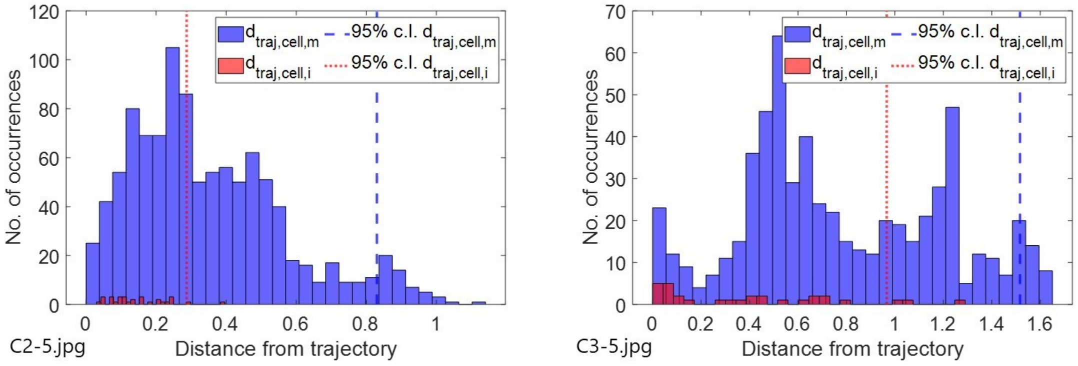


Figure 4.53. *Dataset 5. Optimal grid found by the grid-search algorithm. Valid cells are denoted with white background, while invalid cells have grey background. Normal trajectory is marked with solid line, while diamonds represent the means of scores calculated in each valid cell*

The optimal grid found by the algorithm has 4×2 cells and it is able to capture 99% of all calibration scores: this result is very similar to the one proposed by Westad et al. (2015), that aims at capturing all calibration scores; however the normal trajectory is obtained connecting only 4-points (very low trajectory resolution). The 95% confidence limits are calculated with the method of Westad et al. (2015) and with the new approach developed in this thesis, and also for this case study an example of the difference between the two limits calculated is provided in Figure 4.54.

Figure 4.54 (a) shows that distances of mean of batches from the trajectory (red area) are much smaller than distances of single scores from the trajectory (blue area). The result is that limits calculated with the Westad et al. (2015) method are more than one half smaller than limits calculated with the new approach proposed in this thesis. Same considerations are valid for Figure 4.54 (b), in which cell no.3 is considered.



(a) (b)

Figure 4.54. *Dataset 5. Plot of distances and of limits calculated for (a) cell no.2 and (b) cell no.3. Distances of single scores from trajectory are in blue, while distances of mean of scores of each batch from trajectory are in red. Dotted line marks the limit calculated with approach of* Westad et al. (2015)*, while the dashed line marks the limit calculated with the new approach*

In Figure 4.55 the normal trajectory, modelled through the grid search algorithm, is reported together with its confidence limits, calculated with the new approach for each valid cell, and plotted avoid intersections with the normal trajectory. The low resolution of the trajectory and the consequent low resolution of limits, reflects on the number of calibration scores that are actually left out of confidence limits when limits are plotted in the score space (dashed lines).

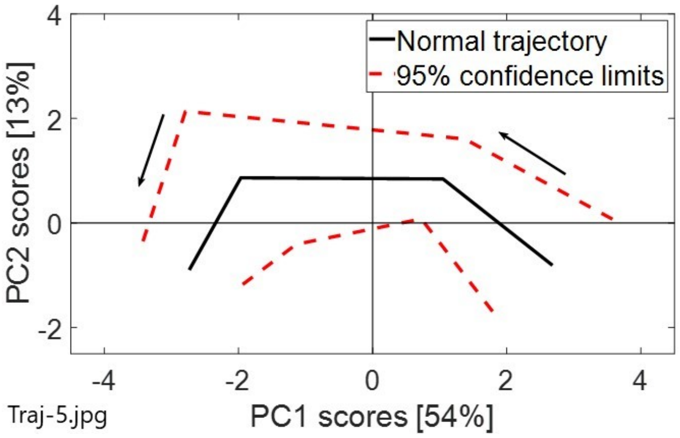


Figure 4.55. *Dataset 5. Final normal trajectory of the process obtained through the grid-search algorithm (solid line) with 95% confidence limits (dashed line)*

Despite limits are calculated considering a confidence of the 95%, the total amount of calibration scores out of confidence limits is equal to 11%. Two alarms (one in the score plot and one for residuals) are calibrate on normal batches. The deviation of batch no.1 from trajectories of the other batches, reflects on a very low sensitivity of the alarm on scores: it is needed to have 63 consecutive scores out of confidence limits before the alarms associated to the score plot turns on. Waiting for 63 consecutive scores out of confidence limits leads to a long delay in the fault detection and, in most of the cases, to a missing fault detection. The sensitivity of the alarm on residuals is higher: it turns on after 18 consecutive values over the confidence limit. All 41 batches available for validation are used to test the model. An example is reported here considering batch no.4.

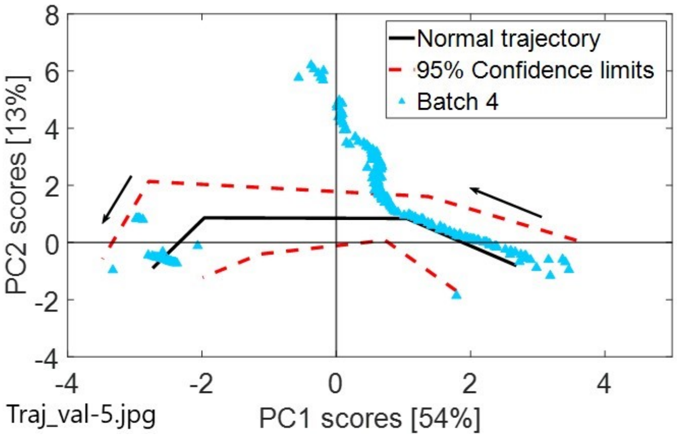
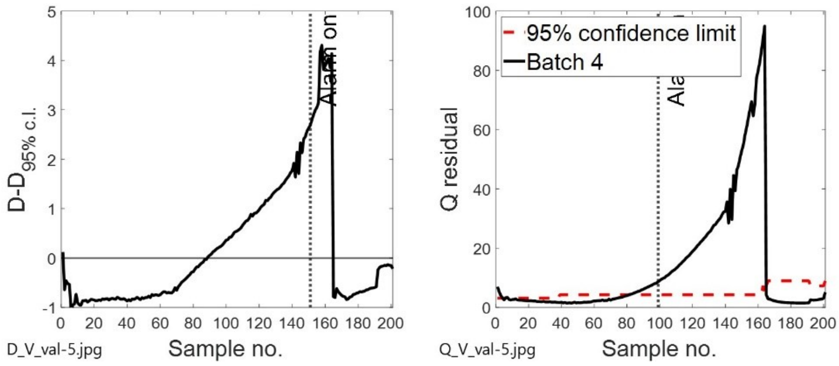


Figure 4.56. *Dataset 5. Projection of a new batch (batch no.4) onto the model. Normal trajectory and confidence limits are represented, respectively, by the solid line and the dashed lines. Triangles represent scores of the new batch*

The new batch, which is designated in the literature as a faulty batch, starts inside the confidence limits and follows a trajectory that goes straight up to the top of the plot, going out of the confidence limits for more than 62 samples.



(a) (b)

Figure 4.57. *Dataset 5. Plot of (a) the difference between the distance from trajectory of a new sample (D) and the limit distance ($D_{95\% \ c.l.}$), and of (b) Q residuals along the sample number. The 95% confidence limit for residuals is reported (dashed line)*

An alarm occurs in the score plot at sample no.151.

Despite its deviation from the normal trajectory, the new batch ends inside confidence limits in the left side of the plot. The profile of the distance of Figure 4.57 (a) is similar to the one of the residuals of Figure 4.57 (b). The fault is detected in the score plot not before the 151st instant, however from Figure 4.57 (a) is it possible to notice that the deviation of the new batch from the normal trajectory starts before the 100th sample. The alarm on the residuals has a higher sensitivity and is able to notify the anomaly at the 99th sample, that means 52 time instants earlier with respect to the alarm on the score plot.

Considering the direction on which the trajectory deviates from the normal path and the loading plot in Figure 4.51, it can be concluded that the abnormal variable is the differential pressure. This assertion is confirmed by comparing the profile of the variable of this new batch along the time with the profiles of the same variable of calibration batches (all of the profiles of variables can be obtained through the validation_VWU.m script): although batches have different length, it is possible to see that while the differential pressure of calibration batches remains close to zero for the entire process duration, in batch no.4 it starts to increase after about 70 time instants and reaches a pick at the 164th sample.

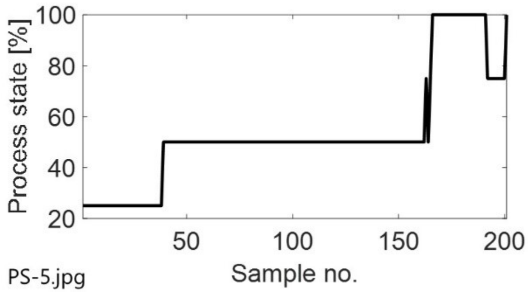


Figure 4.58. *Dataset 5. Plot of the process state (or relative time) against sample number of batch no.4. The process seems to stabilize after about 50 samples, then it reaches completion (100%) rapidly*

The profile of the process state in Figure 4.58 highlights that an anomaly occurs during the process, which seems to stabilize after about 50 time instants. The static part of the process corresponds to the period in which the trajectory of scores deviates from the normal one in the north-direction, therefore the point into trajectory which is the closest to new scores (and on which the process state is calculated) results to be always the second one (second point on which the trajectory is built by interpolation). For this reason, the process seems to be static, however it is not, and a trajectory of scores can be recognized in the score plot, instead of an accumulation (as in the case at §4.1.1).

This is the only validation case on which the fault is detected both in the residual plot and in the score plot: in all other cases, because of the very low sensitivity of the alarm on scores, the fault can be detected only through the residuals.

Table 4.23 summarizes the results of the model validation with all 41 batches available. Due to the low resolution of the trajectory (4-point trajectory) and the rough approximation of the confidence limits (calculated only at 4 points), 11% of the calibration scores remains out of the 95% confidence limits. The presence of a batch that deviates from the mean trajectory is the main cause of the low sensitivity of the alarm on scores, and the alarm on residuals is able to recognise the fault only in 25 out of 38 cases. Since it is not known neither the time at which the fault occurs in each abnormal batch, nor its cause, it is not possible to determine if any false alarms occur.

Table 4.23. *Dataset 5: assumption-free model results*

Description	Value
Confidence limits	95%
Calibration scores out of c.l.	11%
Consecutive scores out of c.l. for alarm on scores	63
Consecutive scores out of c.l. for alarm on residuals	18
False faulty batches	0/3
Missed fault detection	25/38
False alarms	N/A

This is a good example to explain the reason why it is not possible to capture all calibration batches for trajectory modelling with the grid-search algorithm. Since one grid cell is considered valid for an overall mean of scores calculation (see §3.2) only if it contains at least one score of each calibration batch, in order to capture the 100% of calibration scores, and so also scores of batch no.1 that deviate towards the upper part of the score space, and that are much further than others, cell dimension should increase dramatically. The number of valid cells in this case would be only three, therefore the normal trajectory would very coarse: the performance of the model would decrease dramatically because only batches out of 38 would be detected as faulty.

An attempt to improve the performances of the alarm has been done by excluding batch no.1 from the calibration dataset. After the assumption-free model calibration and validation, the number of faulty batches detected was only 15 out of 38. Although the sensitivity of the alarm improves in this case, and the number of the consecutive scores out of confidence limits for fault detection is equal to 39 (instead of 63), the overall model performance does not improve significantly.

4.5.2 Monitoring with a batch-wise unfolded MPCA model

The aligned dataset, containing all calibration batches with the same number of samples, is available in Aspen ProMV and can be downloaded following the indications reported at §2.5. Data are organized into a I×JK matrix, as discussed in §1.2, that is used to calibrate a PCA model. Table 4.24 reports that the RMSECV is at its minimum with one principal component, however the cumulative variance explained is only 24% of the total one. Since the RMSECV does not increase consistently, three principal components are chosen to be retained into the model: in this way at least the 50% of cumulative variance is explained. After 3 principal components, the RMSECV increases.

Table 4.24. *Dataset 5: batch-wise unfolded PCA model summary*

PC no.	Eigenvalue	% variance captured	% cumulative variance captured	RMSECV
1	775.69	23.87	23.87	46.98
2	527.78	16.24	40.12	46.98
3	360.51	11.10	51.22	46.99
4	311.06	9.57	60.79	47.01
5	197.12	6.07	66.79	47.01

Scores of all 30 calibration batches, obtained after the model calibration, are reported in the score plot of Figure 4.59. Batches are not randomly distributed: a dense cluster can be identified close to the centre of the axes origin, while a smaller cluster made of 6-7 batches is located in the right-side of the score space. Only batch no.20 is out of the confidence area, marked with a dashed line.

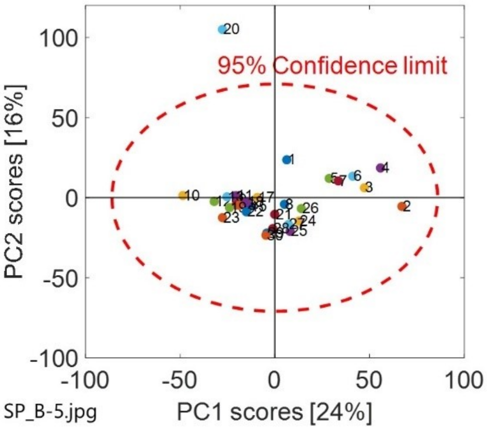
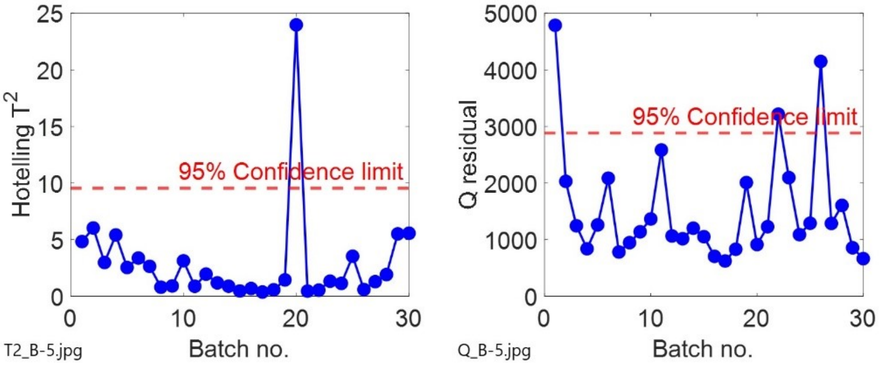


Figure 4.59. *Dataset 5. Score plot resulting from the batch-wise unfolded PCA. All 30 batches are reported with the 95% confidence limit (dashed line). Percentages in squared brackets represent the variance captured by the corresponding principal component*

The assumption of random distribution of scores is not properly verified, and this is more evident in the plot of the Hotelling T^2 of Figure 4.60 (a): first and last batches have values of T^2 that are larger than the ones of batches in the middle of the dataset, and corresponds to batches that in the score plot are the furthest from the main batch cluster (e.g., batches from no.1 to no.7). Consistently with the score plot, only the T^2 value of batch no.20 is over the 95% confidence limit.

Values of residuals of Figure 4.60 (b), instead, are randomly distributed.



(a) (b)

Figure 4.60. *Dataset 5. (a) Hotelling T^2 and (b) Q residual plots for an herbicide production dataset. 95% confidence limits are marked with dashed line. The assumption of random distribution is not verified for the Hotelling T^2*

Similarly to previous cases, an example of validation is provided here considering the same batch used for the assumption-free model validation at §4.5.1. Batch no.4 is projected onto the model carrying out an online monitoring: the score, the residual and the Hotelling T^2 are calculated at each time instant with the procedure reported at §1.2.

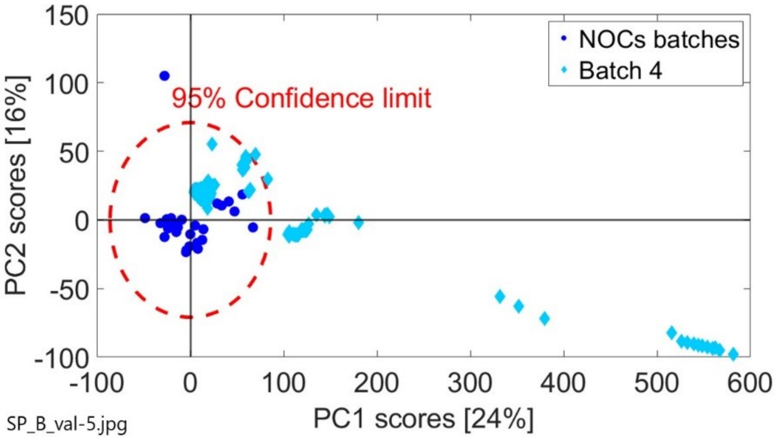
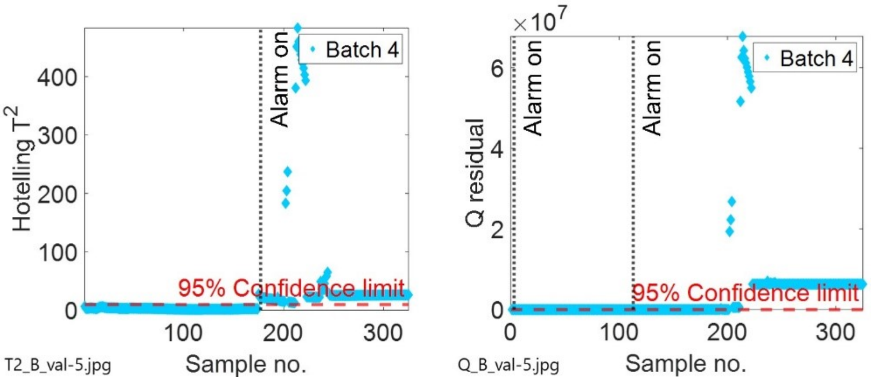


Figure 4.61. *Dataset 5. Score plot of the projection of the new batch (batch no.4) onto the model. Dots represent normal batches, while diamonds represent evolution of the new batch at every time instant. The 95% confidence limit is marked with dashed line*

Score plot of Figure 4.61 shows the evolution of the new batch projected at every time instant. The process starts inside the confidence ellipse, but then it starts to deviate on the right-direction on the score space, ending in the fourth quadrant region, very far from the scores representing normal batches (dots). From a preliminary qualitatively evaluation of the trajectory of scores of batch no.4, it can be supposed that a fault occurs at about half-way of the process evolution. A more precise inspection of the time at which the anomaly takes place can be done through the T^2 and Q control charts, reported in Figure 4.62. In the T^2 control chart, only one alarm occurs at sample no.177: values of Hotelling T^2 overcome the confidence limit for some time instants, then they increase dramatically and become smaller in the last part of the process. Considering the Q residual control chart, the first alarm starts immediately at the 3rd sample, however the real fault probably takes place later, when a rapid increasing in the values of residuals occurs.



(a) (b)

Figure 4.62. *Dataset 5. Plots of (a) Hotelling T^2 and (b) Q residuals for batch no.4. Diamonds represent samples of batch no. 4, while the dashed line marks the 95% confidence limit. T^2 alarm turns on at sample no.177. In the Q residual plot, the first alarm starts at sample no.3 and the second one at sample no.113*

Considering that values of residuals remain close to the 95% confidence limit also when the alarms turn on, and that a real increasing in both T^2 and Q residual happen after sample no.177, the fault probably does not take place before the second half of the process.

The fault diagnosis is carried out through the contribution plots. In this case, it is very difficult to identify one or more variables that are clearly responsible of the anomaly: almost all of the variables profiles goes out of confidence limits in the contribution plots, however a pick clearly abnormal can be identified in the Q contribution profile of the differential pressure. All contribution plots of T^2 and residuals can be obtained with the validation_BWU.m file reported in Appendix 2.

Table 4.25. *Dataset 5: batch-wise model results*

Description	Value
Confidence limits	95%
False faulty batches	3/3
Missed fault detection	5/38
False alarms	N/A

Results obtained using all 41 validation batches for model testing are summarized in Table 4.25. Similarly to cases §4.3.2 and §4.4.2, also in this case all normal batches are recognized as faulty by the model. This wrong classification is due to multiple alarms occurring in the control chart of residuals for almost all validation batches, both normal and abnormal. In order to avoid discarding normal batches in real industry, it is advisable to check the magnitude of the residuals: in faulty batches, values of residuals are much larger than the confidence limit value, while in the case of normal batches, residuals are over the confidence limit, but close to it. In 5 out of 38 faulty batches, the anomaly is not recognized, either in the T^2 plot, or in the Q residual plot. The cause of the fault and the time at which the anomaly occurs are not known, therefore it is not possible to determine the exact number of false alarms.

For this dataset, the high sensitivity of the batch-wise model allows to detect almost all faulty batches. However, also all of the 3 normal batches are classified as abnormal. The assumption-free model, instead, does not give false alarms for normal batches, but only 13 out of 38 abnormal batches are recognized. Considering the good performances in the fault detection of the batch-wise model, an adjustment in the confidence limit for residuals would probably improve the performance of the model, decreasing the number of false alarms during validation with normal batches.

Conclusions

The objective of this thesis was to investigate the batch process monitoring methodology proposed by Westad et al. (2015), also called an assumption-free methodology, and to compare its performance to the one of a standard monitoring approach based on a batch-wise unfolding of the 3D process data matrix. The monitoring performances have been tested on five benchmark batch processes.

Each assumption-free model was developed by unfolding the available data in a variable-wise form; then, a grid-search algorithm was applied to the score plot to model a normal trajectory of the process. The confidence limits around this trajectory and for the *Q* residuals have been calculated at each trajectory point, and two alarms have been calibrated on normal calibration batches, with the purpose to discriminate normal from abnormal validation batches, and to identify the fault time. Each batch-wise model, based on a batch-wise unfolded calibration matrix, was developed by estimating at each time instant the missing future values of variables, with the purpose of carrying out an online monitoring. In this case, confidence limits have been calculated for the Hotelling T^2 and *Q* residuals control charts and alarms have been set to start after 3 values exceeding the relevant confidence limit.

The main difficulty that has been faced in this thesis is related to the assumption-free model: the grid-search algorithm is not described in detail in the paper of Westad et al. (2015), therefore some assumptions and criteria have been adopted to develop the algorithm used in the thesis. Moreover, the criteria adopted by Westad et al. (2015) for the calculation of the confidence limits revealed not entirely appropriate, therefore a new approach has been proposed, aiming at modelling the entire process dynamics. Another aspect that required investigation, is the amount of calibration scores captured by the grid-search algorithm. Capturing 95% of calibration scores was sufficient to model a normal trajectory of the process, while capturing 100% of the scores (as suggested by Westad et al., 2015) would lead to a coarser trajectory and a worse monitoring performance.

Considering the results obtained after model testing, it can be concluded that the assumption-free methodology raises fewer false alarms, but it also has the highest rate of missed fault detection. The batch-wise methodology is able to recognise faulty batches in almost all of the cases; however the number of occurrences of false faulty batches is very high: considering datasets no.3, no.4 and no.5, all normal batches are recognised as faulty by the batch-wise model. The cause of the false faulty batches in the batch-wise model is the *Q* control chart, on which several false alarms occur. The performance of the control chart might be improved if the alarm was calibrated not only on the number of consecutive residuals exceeding the confidence limits, but also on the magnitude of the residuals: despite residuals are often out of confidence limits, faulty batches have residuals that are much larger than the ones of normal batches.

One of the limits of the study is related to the fact that although both models are designed for an online monitoring, in the assumption-free model each time instant is based on the conditions of the process at that time instant only, while in the batch-wise unfolded model, each projected scores embeds information on the process at previous and future time instants. Concerning the assumption-free model, the similarity between calibration batches is fundamental: all calibration batches, in fact, are considered to have the same importance in process modelling and alarm calibration, which means that it is sufficient to have also only one calibration batch which deviates from the others to have a very low alarm sensitivity and a rapid decreasing of the model performance. Another weakness of the model is related to the fact that the alarm calibration is based only on the number of consecutive scores exceeding the confidence limits, and it does not consider their overall path: if the limits around a normal trajectory are very wide (cases §4.4.1 and §4.5.1) and scores follow a different trajectory with respect to the normal one, but they remain inside confidence limits, the batch is not recognised as faulty. At the end of the study, it can be concluded that the assumption-free model has good performances if calibration batches are very similar to each other, their trajectories do not have rapid changings in directions (sharp corners), the normal trajectory is modelled with a reasonable number of points, and faulty batches are very different from normal calibration batches.

An improvement in the alarm calibration, might be obtained by excluding some calibration batches when the maximum number of scores allowed to be out of the confidence limits for normal batches is computed: if there is one calibration batch whose trajectory is much different with respect to the others, and its maximum number of scores out of the confidence limits is much greater than the others, then it should be excluded from alarm calibration. In this model, in fact, all batches are considered to have the same importance; however, it may happen that if one batch deviates a lot from the others (see case §4.5.1), the alarm sensitivity decreases dramatically, and the model is not able to recognise faulty batches during validation.

A further improvement of the model may be realized by considering the overall path of scores in the score plot, in addition to their position being in or out of the confidence limits.

Eventually, it can be concluded that an assumption-free model is able to recognise faulty batches; however, its performance strongly depends on the quality of calibration batches and the shape of the trajectory of their scores. A batch-wise model, instead, is always able to recognise faulty batches, however several false alarms may occur in the case of normal batches.

Nomenclature

$\mathbf{X}_{\mathbf{3D}}$	=	three-dimensional matrix
I	=	number of batches
i	=	batch i
J	=	number of variables
K	=	number of time instants sampled
k	=	time instant k
$\mathbf{X}$	=	unfolded matrix
m	=	number of samples (rows) in the unfolded matrix
N	=	number of principal components retained into the model
$\mathbf{p}_n$	=	loading. Eigenvector associated to the eigenvalue λ_n
$\mathbf{P}_{all}$	=	matrix containing all eigenvectors
$\mathbf{T}$	=	score matrix
$\hat{\mathbf{X}}$	=	matrix of data represented by the model
$\mathbf{P}$	=	truncated matrix of loadings
$\mathbf{E}$	=	error matrix
$\mathbf{t}_i$	=	score of sample i
$\hat{y}_l$	=	predictions for samples that are not included in model formulation
y_l	=	real samples not included in model formulation
Z	=	number of samples that are not included in model formulation
$\mathbf{\Lambda}$	=	diagonal matrix containing eigenvalues up to principal component n
T^2	=	Hotelling T^2 statistic
T_i^2	=	Hotelling T^2 statistic for sample i
Q	=	residuals
Q_i	=	residuals for sample i
$\mathbf{e}_i$	=	vector of errors for sample i
$\mathbf{I}$	=	identity matrix
$t_{n,\alpha}$	=	confidence limit for scores on principal component n
$t_{m-1,\alpha/2}$	=	probability point on the single-sided t-distribution with m-1 degrees of freedom and area α/2
c.l.	=	confidence limit
$T^2_{N,m,\alpha}$	=	confidence limit for Hotelling T^2
$F_{N,m-N,\alpha}$	=	F-distribution
Q_α	=	confidence limit for Q statistic
$\mathbf{X}_{new}$	=	unfolded matrix of new sample
$\mathbf{t}_{new}$	=	scores of the new sample

T^2_{new}	=	Hotelling T^2 of the new sample
Q_{new}	=	residuals of the new sample
$\mathbf{e}_{new}$	=	vector of errors of the new sample
$\mathbf{t}_{con,i}$	=	vector of contributions of all variables to the score of batch *i*
$\mathbf{T}^{\mathbf{2}}_{con,i}$	=	vector of contributions of all variables to the T^2 of batch *i*
$\mathbf{Q}_{con,i}$	=	vector of contributions of all variables to the Q of batch *i*
a	=	score component
$cell$	=	cell of the grid in the grid-search algorithm
$t_{a,cell,m}$	=	component *a* of score *m* in one cell
$\bar{\bar{t}}_{a,cell}$	=	component *a* of the overall mean of scores in one cell
$\bar{\bar{\mathbf{t}}}_{cell}$	=	overall mean of scores in one cell
M	=	total number of scores inside one cell
$t_{a,cell,m\perp traj}$	=	component *a* of the projection into trajectory of score *m* in one cell
$d_{traj,cell,m}$	=	distance from the trajectory of sample *m* in one cell
$d_{(1-\alpha)\%\ c.l.,cell}$	=	distance for the $(1-\alpha)\%$ confidence limit for one cell
L	=	total number of scores of one batch inside one cell
$t_{a,cell,i,l}$	=	component *a* of the score *l* of batch *i* in one cell
$\bar{t}_{a,cell,i}$	=	component *a* of the of the mean of scores of batch *i* in one cell
$\bar{\mathbf{t}}_{cell,i}$	=	mean of scores of batch *i* in one cell
$d_{traj,cell,i}$	=	distance of the mean of scores of batch *i* from trajectory
$d_{(1-\alpha)\%\ c.l.\mathrm{W},cell}$	=	distance for the $(1-\alpha)\%$ confidence limit for one cell according to Westad

Greek letters

λ_n	=	eigenvalue of principal component *n*
α	=	1-c.l./100. Example: if c.l.=95%, then α=1-0.95=0.05
β	=	fraction of batches in a valid cell in the grid-search algorithm
γ	=	total fraction of scores captured by the grid in grid-search algorithm

Acronyms

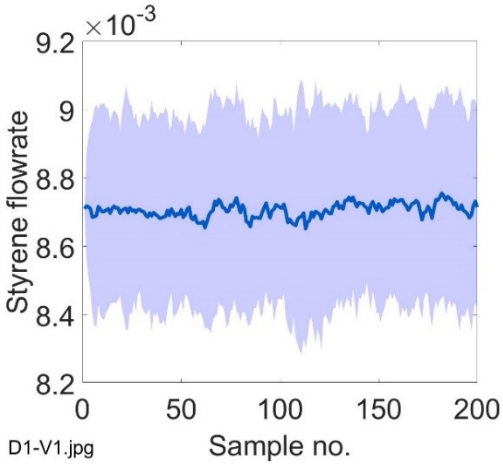
cov(**X**)	=	covariance matrix of **X**
RMSECV	=	root-mean-square-error in cross validation
RMSECV$_n$	=	root-mean-square-error in cross validation retaining *n* principal components

Appendix 1

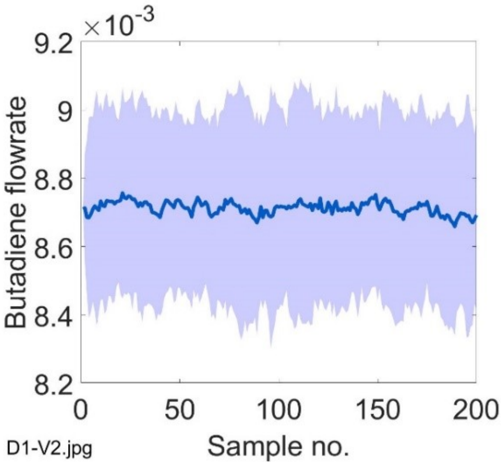
All variables profiles of the calibration datasets are reported here. In the case of calibration datasets containing batches all with the same duration, the mean profile (solid line) with its variability across batches (shaded area) is reported. In the case of calibration datasets with a varying process duration across batches (dataset no.3 and dataset no.5), all profiles of variables for all batches are reported.

All figures are available at: Master_thesis_Alice_Fracassetto\Figures\Variables_profiles

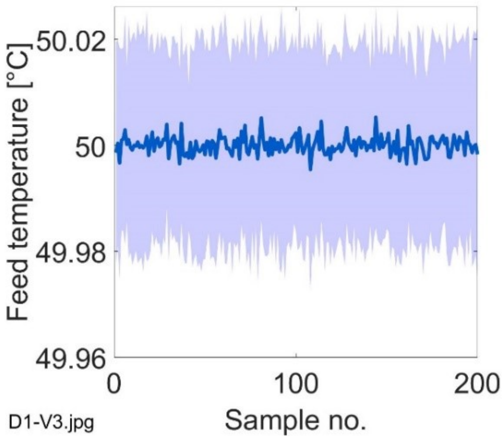
A.1.1 Dataset 1 – Variables mean profile along the time



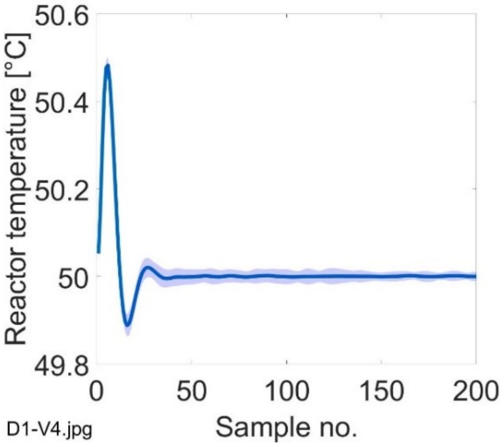
D1-V1.jpg



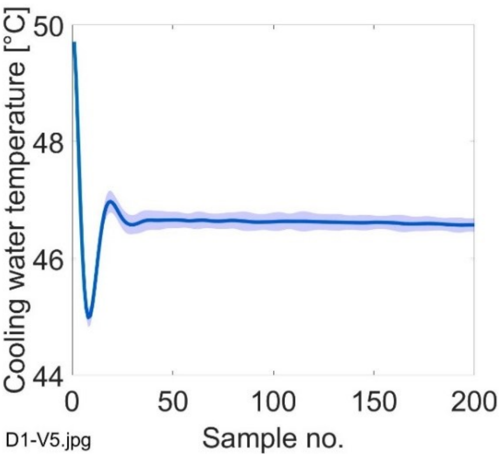
D1-V2.jpg



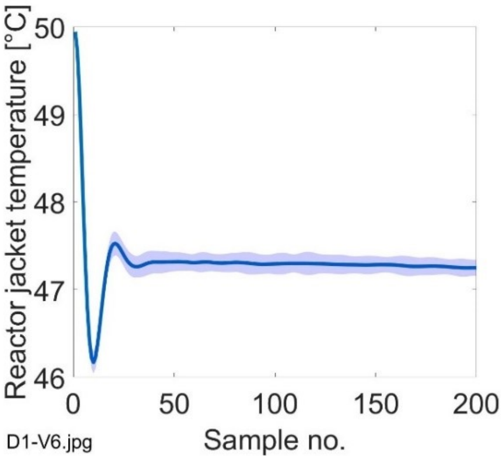
D1-V3.jpg



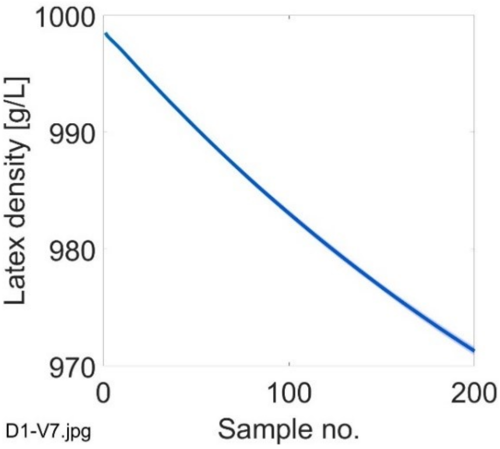
D1-V4.jpg



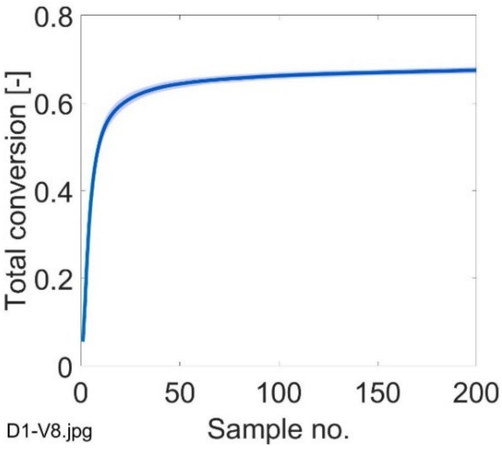
D1-V5.jpg



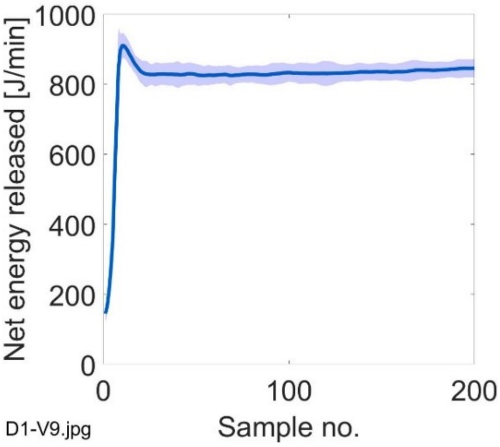
D1-V6.jpg



D1-V7.jpg

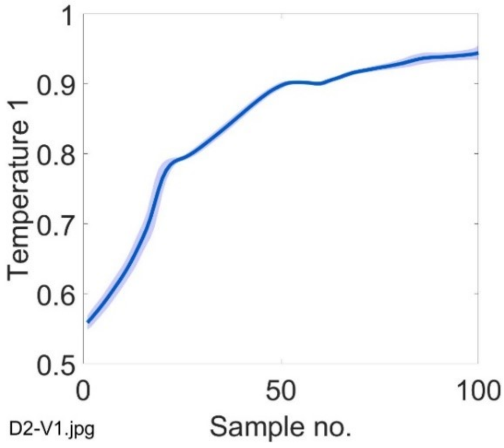


D1-V8.jpg

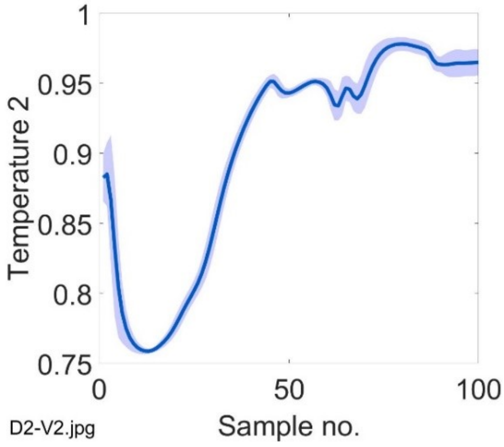


D1-V9.jpg

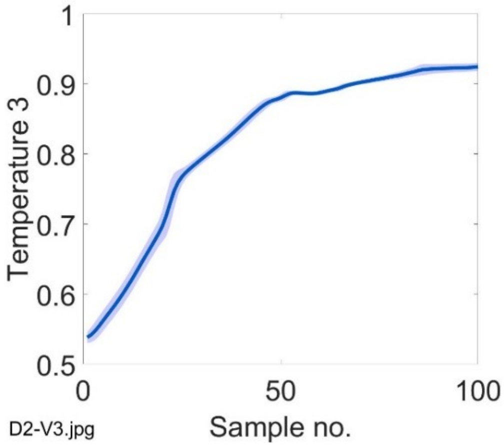
A.1.2 Dataset 2 – Variables mean profile along the time



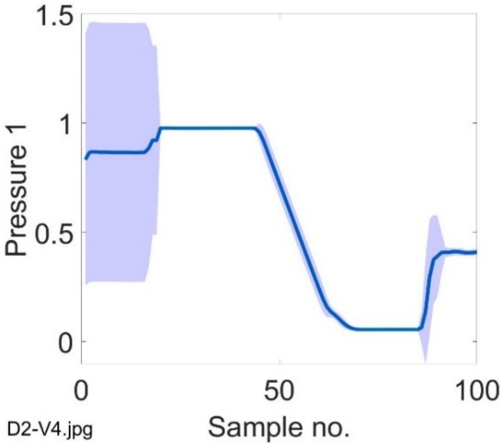
D2-V1.jpg



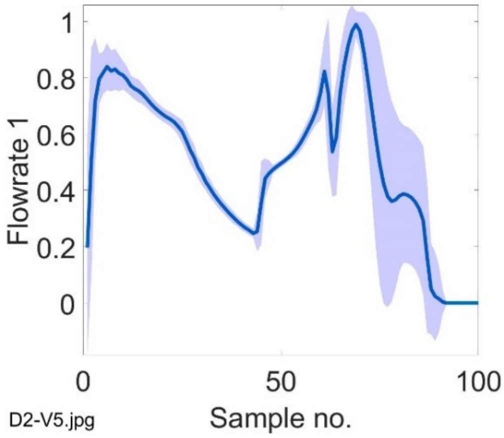
D2-V2.jpg



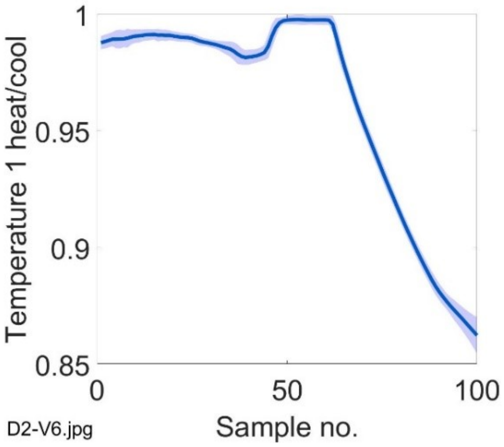
D2-V3.jpg



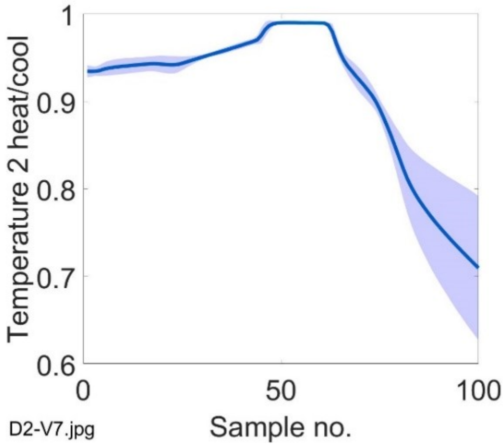
D2-V4.jpg



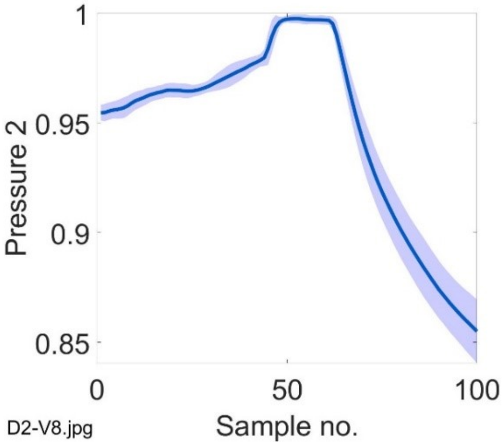
D2-V5.jpg



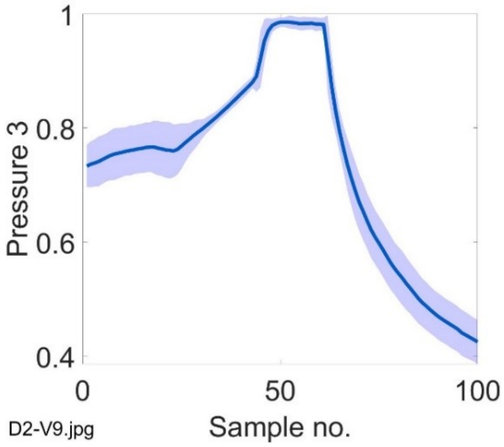
D2-V6.jpg



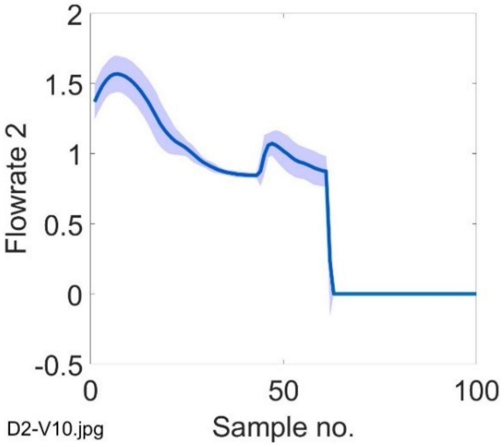
D2-V7.jpg



D2-V8.jpg

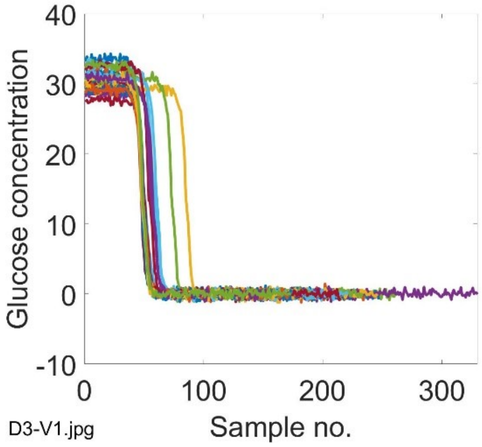


D2-V9.jpg

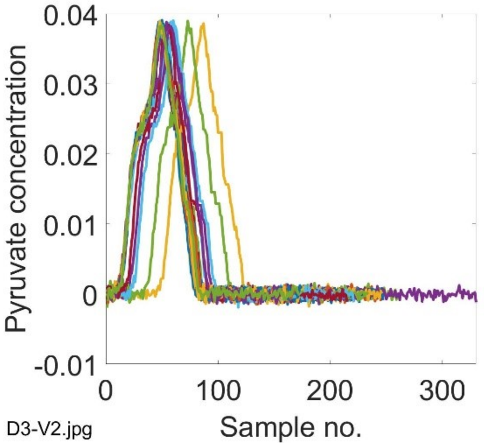


D2-V10.jpg

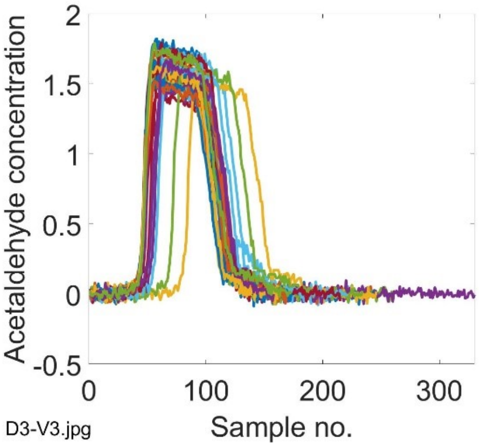
A.1.3 Dataset 3 – Variables profiles along the time



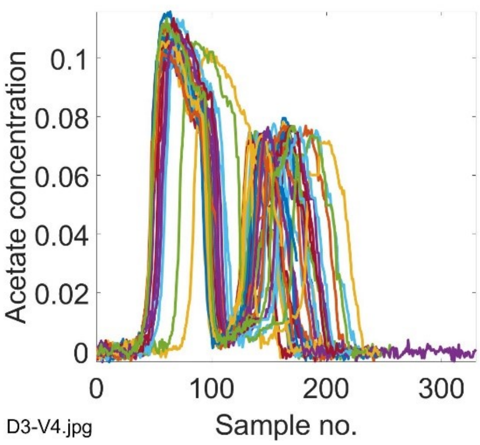
D3-V1.jpg



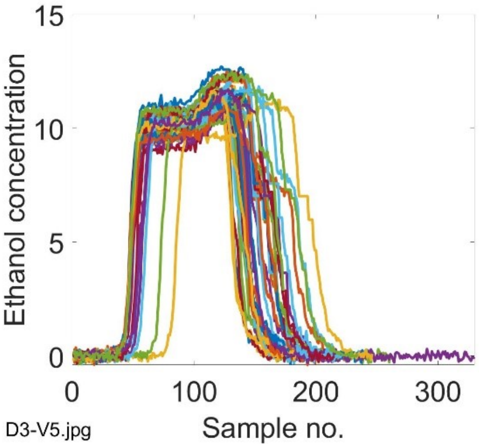
D3-V2.jpg



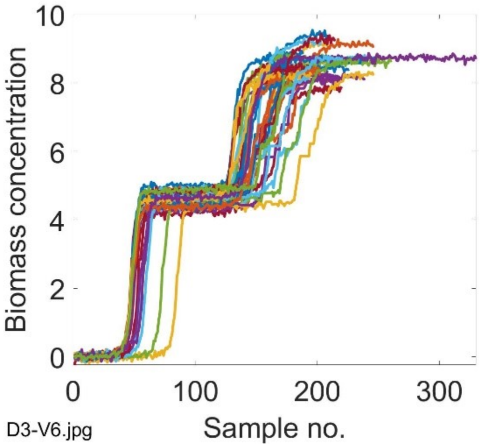
D3-V3.jpg



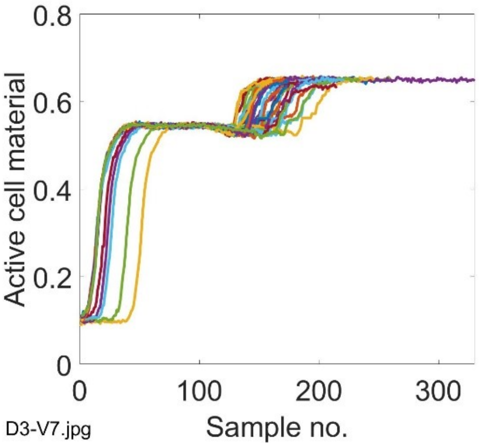
D3-V4.jpg



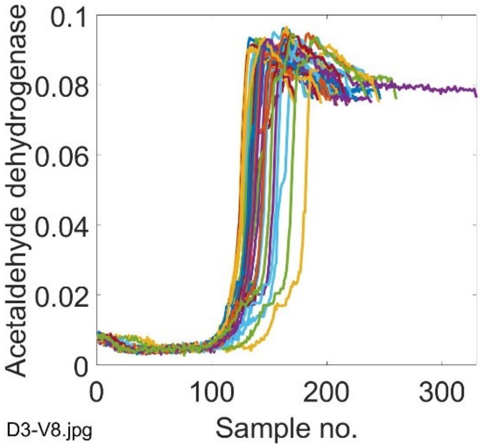
D3-V5.jpg



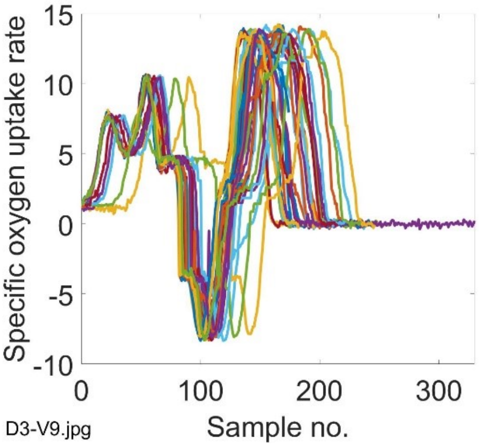
D3-V6.jpg



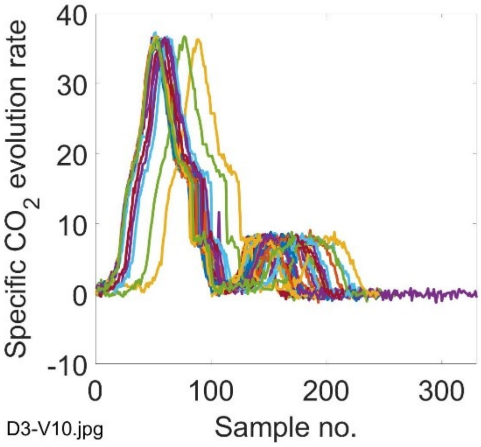
D3-V7.jpg



D3-V8.jpg

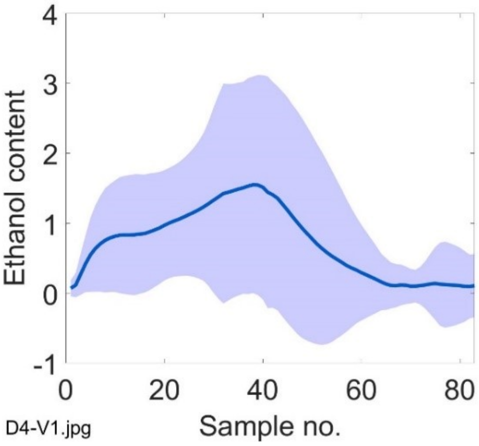


D3-V9.jpg

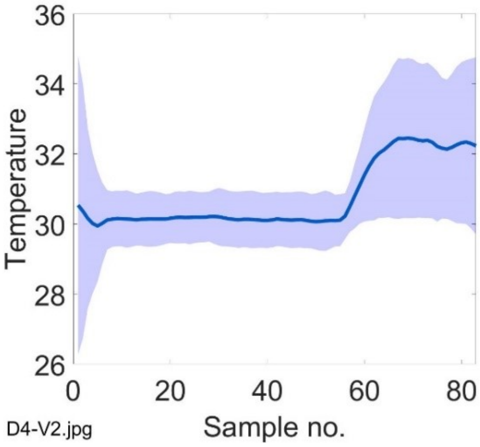


D3-V10.jpg

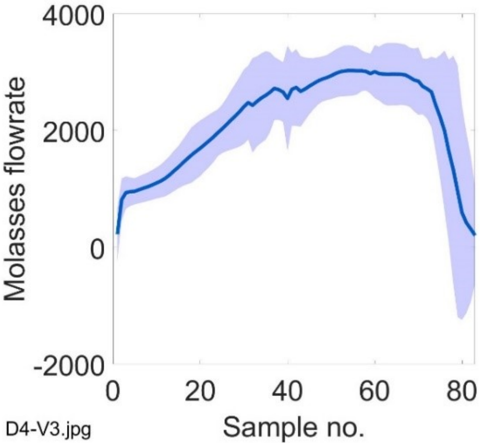
A.1.4 Dataset 4 – Variables mean profile along the time



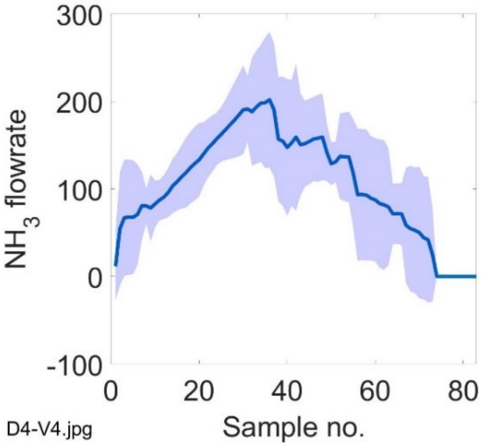
D4-V1.jpg



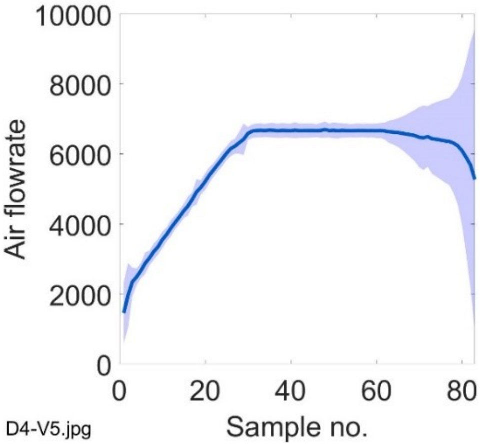
D4-V2.jpg



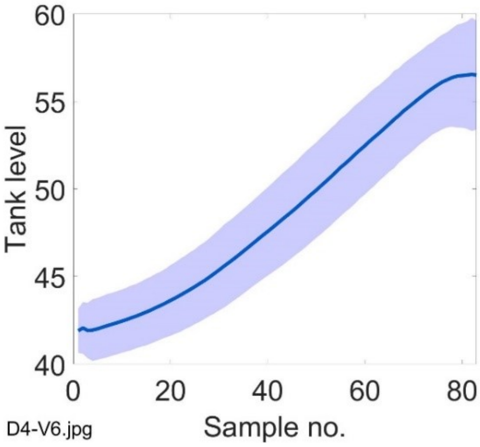
D4-V3.jpg



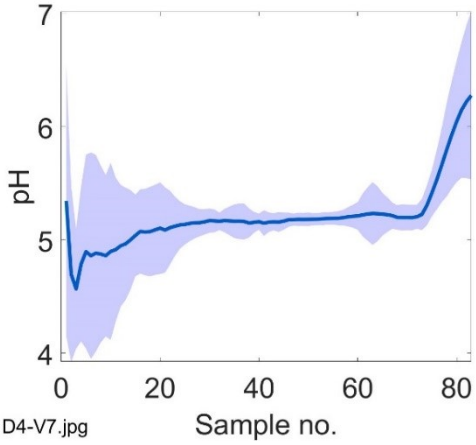
D4-V4.jpg



D4-V5.jpg

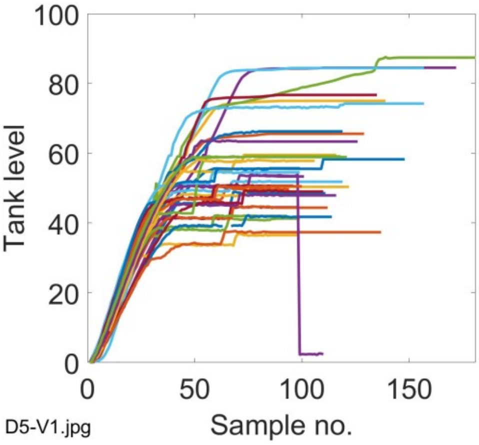


D4-V6.jpg

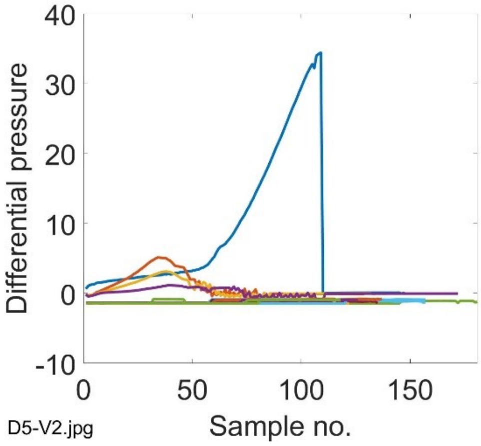


D4-V7.jpg

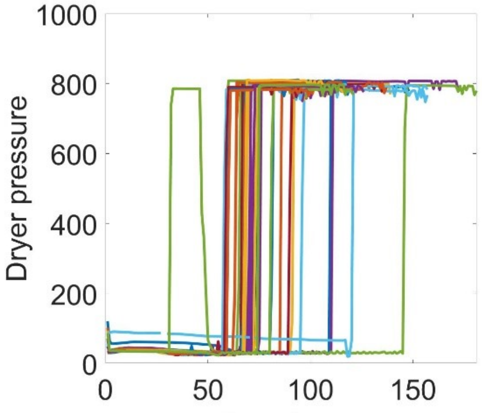
A.1.5 Dataset 5 – Variables profiles along the time



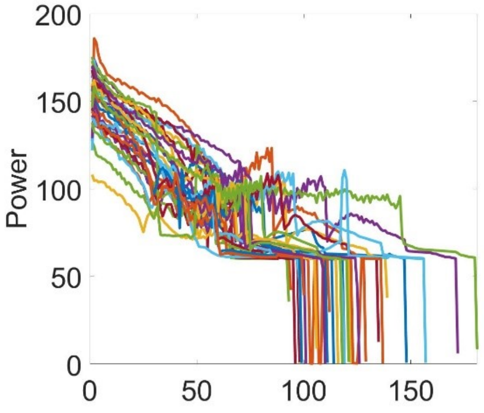
D5-V1.jpg



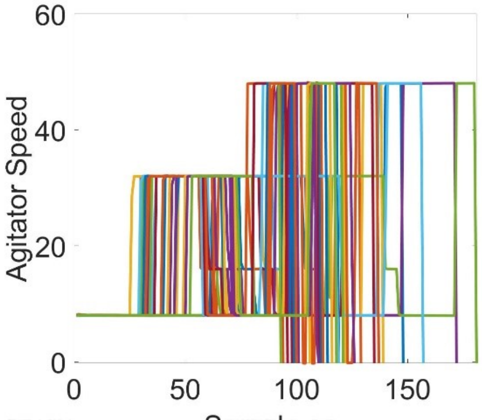
D5-V2.jpg



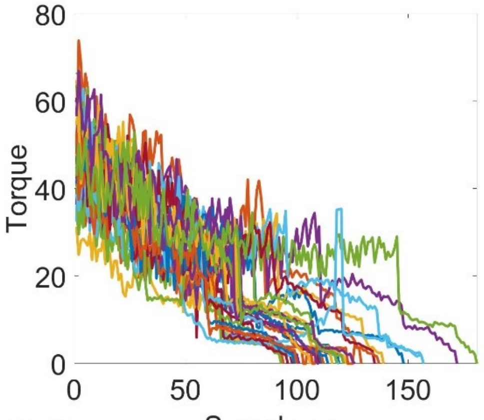
D5-V3.jpg



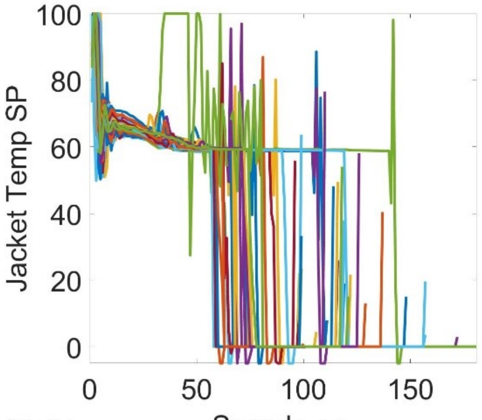
D5-V4.jpg



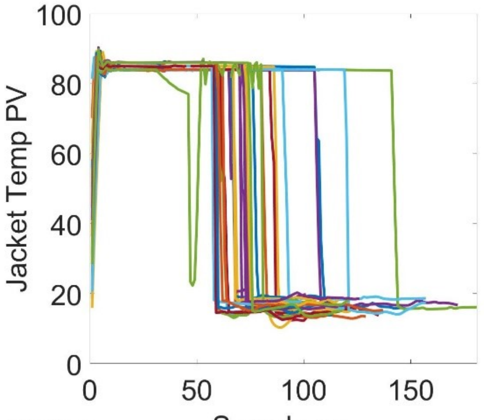
D5-V5.jpg



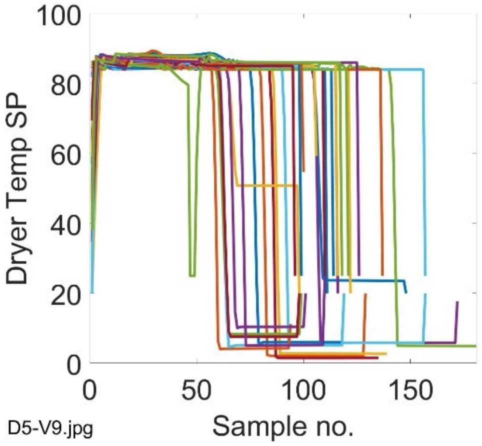
D5-V6.jpg

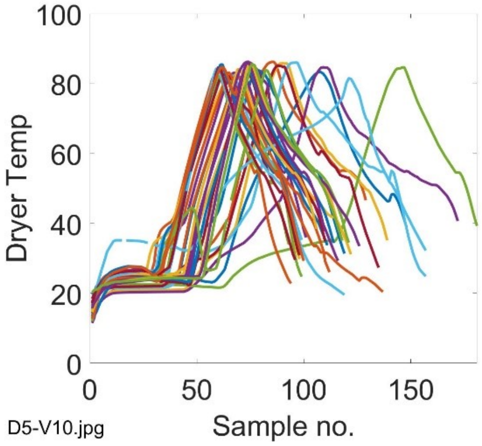


D5-V7.jpg



D5-V8.jpg

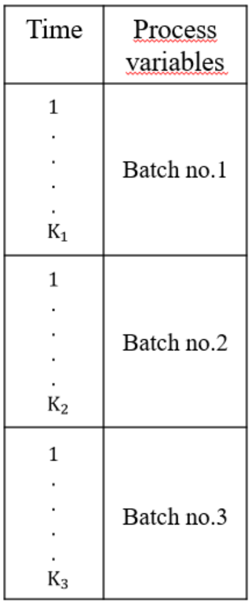

D5-V9.jpg


D5-V10.jpg

Appendix 2

A.2.1 User's guide - Input data structure

1. The MVBatch Toolbox for Matlab (used for batch synchronization of dataset no.3) can be found at the path: Master_thesis_Alice_Fracassetto\MVBatch-1.1
2. The variable-wise unfolded calibration matrix for the assumption-free model must have as first column the "time" variable, whose values (in ascendent order) refer to the sample number of the corresponding batch.



Time	Process variables
1 . . . . K_1	Batch no.1
1 . . . . K_2	Batch no.2
1 . . . . K_3	Batch no.3

3. The PCA models must be "evrimodel" or "structure" array (it is recommended to use the PLS Toolbox for Matlab to calibrate the PCA model).
4. The validation dataset must be a cell array containing: 1) no. of the batch in the first column; 2) batch data in a variable-wise form in the second column.
   Refer to the structure of the datasets available for a better understanding.
5. The matrix containing the names of the variables must be a cell array containing: 1) the name of the variables in the first column; 2) the units of measure (if available) in the second column. If units of measure are not available, the second column is empty.

A.2.2 User's guide - Assumption-free model

1. The assumption-free model can be found at the path: Master_thesis_Alice_Fracassetto\Process_monitoring_models\Assumption-free_model
2. Open main_script_VWU.m, fill the "Input" section with: the variable-wise unfolded matrix, the PCA model, the validation dataset and the matrix containing the names of variables.
3. Modify (eventually) the grid-search parameters and the initial grid resolution:
   - To modify the γ parameter, open Gridsearch_algorithm.m and modify the variable "gamma" in the "Initialization" section.

     If γ =95%, then gamma=0.95; if γ =90%, then gamma=0.90.
   - To modify the β parameter, open grid_search.m and modify the variable "beta" in the "Initialization" section.

     If β=100%, then beta=1; if β=95%, then beta=0.95.
   - To modify the initial grid resolution, open Gridsearch_algorithm.m and modify the variables "yin" and "xin", corresponding to the number of rows and columns respectively, in the "Initialization" section.
4. Run main_script_VWU.m, keeping in the same folder all of the scripts reported in Table A2.1.

Table A2.1. *List of the scripts of the assumption-free model*

Script name	Function
main_script_VWU.m	Main script for calibration of assumption-free model
Gridsearch_algorithm.m	Grid-search algorithm
grid_search.m	Perform all calculations in a grid
chrono_order.m	Sort trajectory points in chronological order
traj_morepoints.m	Find more points into trajectory for limits calculation
traj_limits.m	Calculation of limits with new approach
distance_traj.m	Calculation of limits with Westad approach
plot_lim.m	Plot limits avoiding intersections
proj_newbatch1.m	Project new batch onto the model
validation_VWU.m	Main script for validation of assumption-free model

5. A window appears: choose the confidence limit.
6. If it is necessary, adjust the plot of the confidence limits around trajectory with the plot_lim.m, then run main_script_VWU.m from the section "Plot confidence limits".
7. After the first validation, the system asks for another one: answer "yes" or "no".

All results obtained after validation can be found in the "Validation_all_results.xls" at Master_thesis_Alice_Fracassetto\Process_monitoring_models

A.2.3 User's guide - batch-wise unfolded model

1. The batch-wise unfolded model can be found at the path: Master_thesis_Alice_Fracassetto\Process_monitoring_models\Batch-wise_model
2. Run the main_script_BWU.m, keeping in the same folder the two scripts reported in Table A2.2.

Table A2.2. *List of the scripts of the batch-wise unfolded model*

Script name	Function
main_script_BWU.m	Main script for calibration of batch-wise model
validation_BWU.m	Main script for validation of batch-wise model

3. A window appears: choose the confidence limit.
4. After the first validation, the system asks for another one: answer "yes" or "no"

All results obtained after validation can be found in the "Validation_all_results.xls" at Master_thesis_Alice_Fracassetto\Process_monitoring_models

References

Aspen ProMV Getting Started Guide (2017). Aspen Technology. p.31

Broadhead, T. O. (1984). Dynamic modelling of the emulsion copolymerization of strene/butadiene. *Master Thesis in Chemical Engineering*, McMaster University (Hamilton, Ontario)

Broadhead, T. O., Hamielec, A. E., & MacGregor, J. F. (1985). Dynamic modelling of the batch, semi-batch and continuous production of styrene/butadiene copolymers by emulsion polymerization. *Makromolekulare Chemie Supplement*, **10**, 105–128.

Camacho, J., Picó, J., & Ferrer, A. (2008). Bilinear modelling of batch processes. Part I: Theoretical discussion. *Journal of Chemometrics*, **22(5)**, 299–308.

Camacho, J., Picó, J., & Ferrer, A. (2009). The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter? *Analytica Chimica Acta*, **642(1–2)**, 59–68.

Chai, Y., Yang, H., & Zhao, L. (2013). Data unfolding PCA modelling and monitoring of multiphase batch processes. *13th IFAC Symposium on Large Scale,* Shanghai (China), 7-10 July, pp.569-574

Eriksson, L., Byrne, T., Johansson, E., Trygg, J., & Vikström, C. (2013). *Multi- and Megavariate Data Analysis: Basic Principles and Applications* (3rd ed.). MKS Umetrics AB, Umeå (Sweden), p.290-299.

García-Muñoz, S., Kourti, T., MacGregor, J. F., Mateos, A. G., & Murphy, G. (2003). Troubleshooting of an industrial batch process using multivariate methods. *Industrial and Engineering Chemistry Research*, **42(15)**, 3592–3601.

George, S., Larsson, G., Olsson, K., & Enfors, S.-O. (1998). Comparison of the Baker's yeast process performance in laboratory and production scale. *Bioprocess Engineering*, **18**, 135–142.

González-Martínez, J. M., Camacho, J., & Ferrer, A. (2018). MVBatch: A matlab toolbox for batch process modeling and monitoring. *Chemometrics and Intelligent Laboratory Systems*, **183**, 122–133.

Jeffy, F. J., Gugaliya, J. K., & Kariwala, V. (2018). Application of Multi-Way Principal Component Analysis on Batch Data. *12th International Conference on Control*, Sheffield (UK), 5-7 September, pp. 414–419.

Kosanovich, K. A., Dahl, K. S., & Piovoso, M. J. (1996). Improved Process Understanding Using Multiway Principal Component Analysis. *Industrial and Engineering Chemistry Research*, **35(1)**, 138–146.

Lei, F., Rotboll, M., & Jorgensen, S. B. (2001). A biochemically structured model for Saccharomyces cerevisiae. *Journal of Biotechnology*, **88**, 205–221.

Nomikos, P., & MacGregor, J. F. (1994). Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, **40(8)**, 1361–1375.

Nomikos, P., & Macgregor, J. F. (1995). Multivariate SPC Charts for Monitoring Batch Processes. *Technometrics*, **37(1)**, 41–59.

Wang, D. (2015). Data analytics in semiconductor industry. *Joint E-Manufacturing and Design Collaboration Symposium*, pp.1–4.

Westad, F., Gidskehaug, L., Swarbrick, B., & Flåten, G. R. (2015). Assumption free modeling and monitoring of batch processes. *Chemometrics and Intelligent Laboratory Systems*, **149**, 66–72.

Wise, B. M., Gallagher, N. B., Bro, R., Shaver, J. M., Windig, W., & Koch, S. R. (2006). *Chemometrics Tutorial for PLS_Toolbox and Solo*. Eigenvector Research, Inc., 99-105

Acknowledgements

First of all, I would like to thank prof. Massimiliano Barolo, for giving me the opportunity of this thesis. Thanks for your fundamental suggestions, and for always supporting me during these months of challenging work. The passion and dedication you put into your work, and the respect and attention you have for students, make the difference.

A big thanks goes also to prof. Pierantonio Facco, for his support in the moments of discouragement and for his precious help for this thesis. You have an incredible ability of making complicated concepts easy, and the way you approach students is simply unique.

Thanks to all of my CAPE-Lab friends, that welcomed me as one of them since the first day at the laboratory. You’re a fantastic group, and I was lucky to share with you the last part of my university career. Thanks for the coffees and the laughs together, and for making these months less tough.

A special mention for my best friend Anita, the kind of friend that everybody should have. Thanks for always being the shoulder to cry on when I needed, and the person with whom to celebrate happy moments. I wish you to achieve your goals, as I have achieved mine.

Thanks to my parents, for teaching me the values of work and sacrifice, for always leaving me free to make my choices and to take my responsibilities. Thanks for always supporting me in following my dreams and for never letting me miss anything. I hope you’re proud.

Finally, thanks to my sister Evelin, because also if we’re not the perfect sisters, I know she’s one of my greatest supporters. Have more confidence in yourself and don’t let anyone tell you that you can’t do something.