UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN CHEMICAL AND PROCESS ENGINEERING

Tesi di Laurea Magistrale in
Chemical and Process Engineering

On the implementation and performance assessment of an assumption-free methodology for batch process monitoring

Relatore: Prof. Massimiliano Barolo

Correlatore: Prof. Pierantonio Facco

Laureando: GIULIO DI CARLO

ANNO ACCADEMICO 2023 - 2024

Abstract

Batch process monitoring is a challenging task due to the high variability of these type of processes. In order to ensure that the final product is of prescribed quality, statistical techniques have been developed for monitoring the process. Among these, multi-way principal component analysis is the most used. The standard methodology for monitoring batch processes requires that the data have the same number of samples. However, this is often not the case. Alignment methods for reaching this goal exist, but they are known for generating artifacts and being computationally demanding. Westad et al. (2015) proposed an assumption-free methodology that does not require any kind of data alignment. However, limited details were provided on the design and use of this methodology to perform process monitoring. Previous studies (Fracassetto, 2022; Sartori, 2023) have been done to understand how to exploit an assumption-free model for process monitoring. In this thesis, further improvements on the topic have been carried out by providing an extensive set of guidelines on how to design the monitoring model. Furthermore, the assumptions made in the previous studies have been verified and a new methodology to build the control chart on the squared prediction error has been developed. In order to assess the monitoring performances of the assumption-free model, the obtained results have been compared to the ones reported by Sartori (2023) using a standard monitoring method on the same datasets. The comparison indicated that, on data which are already aligned, there is no clear evidence that a model performs better than the other. However, the assumption-free modelling outperformed the standard methodology on unaligned data in terms of both detection strength and detection speed.

Table of contents

Introduction

Batch processes are employed in the manufacturing of low-volume of high-added value products including pharmaceuticals, polymers, food and semiconductors. These processes are run in accordance with a recipe which is made of serial and defined steps. The principal stages are charge of the reactor, holding phase and discharge. Typically, the recipe has a predefined time duration. However, in order to ensure the desired product quality, modifications may be necessary.

Controlling batch processes is a challenging task, due to their flexibility, finite duration and non-linear behaviour. Considering this, feedback control may be limited to few variables such as temperature and pressure as mentioned by Kosanovich et al. (1996). Nevertheless, being able to monitor a batch process is essential considering that a prompt detection of a fault allows one to save raw materials, energy, time and money. Furthermore, it allows one to increase productivity and average quality of the product. Moreover, in chemical plants many process variables are measured at a high sampling rate. Therefore, combining the necessity of monitoring a process and the capability of computers to perform advanced calculations in a relatively short time has led to the proliferation of statistical process monitoring (SPM) in the process industry, particularly in the context of batch processes. SPM is a technique that uses data collected in a plant to assess whether the process is running in a state of normal operating conditions (NOC) or not and allows one to have a better process understanding and to capture the correlation between variables (Nomikos and MacGregor, 1994). This technique constructs a model based only on the historical data of NOC batches, therefore no knowledge of the physical and chemical phenomena is needed, and then, once the model has been trained, new observations are projected into it to determine if they are regarded as faulty or not.

Batch processes are usually monitored using multi-way principal component analysis (MPCA) which is a method that performs two main tasks: data compression and fault detection. The first task is accomplished by projecting the observation of the high-dimensional space of the original variables to a lower-dimensional space of few latent variables. The second task is performed by comparing the projection, Hotelling's T^2 and square prediction error (*SPE*) to statistical confidence limits evaluated from the NOC data used to calibrate the model (Wise, 1996).

Before applying principal component analysis (PCA), the unfolding of the three-dimensional matrix containing the data to a two-dimensional matrix is necessary. Many methods to perform the unfolding have been highlighted in literature by Chamaco et al. (2008) and by Chamaco et al (2009) but the two main approaches are batch-wise unfolding (BWU) and variable-wise unfolding (VWU).

The first one is the most used unfolding strategy to achieve process monitoring, however it requires that all the batches are aligned (Nomikos and MacGregor, 1994). Having aligned batches means that all must have the same starting and ending point and the same number of samples. Many techniques exist to perform alignment, such as dynamic time warping (DTW) (Kassidas et al., 1998) and relaxed greedy time warping (RGTW) (Gonzàlez-Martínez et al., 2011) which is used for on-line process monitoring. After the application of these methods the trajectories are aligned, and batch-wise unfolding can be applied with a subsequent MPCA model building. The problem with these methods is that: artifacts are created when some batches are significantly shorter than the reference batch used to align the data, thus limiting the monitoring performance of the model as pointed out by Sartori et al. (2023). Moreover, depending on the dataset size, the process could be computationally demanding.

On the other hand, a PCA model built on a variable-wise unfolded matrix does not require any alignment, indeed PCA can be built immediately. The application of this type of unfolding has the advantage that the calculations are less intensive, the drawback is that a PCA performed on this unfolded matrix is the study of the dynamic behaviour of the process around the overall mean of each variable and therefore a mean correlation between the variables is forced for all the duration of batches (Kourti, 2003)

In order to overcome the drawback of batch-wise unfolding while exploiting the advantages of variable-wise unfolding, the assumption-free modelling was proposed by Westad et al. (2015). The proposed method consists in the reconstruction of the mean batch run, from the NOC data, followed by the monitoring of a new batch, comparing its trajectory to the one evaluated during the calibration. The first part is accomplished by partitioning the score plot, to identify the points of the trajectory, and reconstructing the mean run by interpolating those points. However, no details are given by Westad et al. (2015) about the algorithm used to find such trajectory, nor the approach used to perform fault identifications. Moreover, no comparison was carried out with a state-of-the-art methodology in order to assess whether the assumption-free modelling performs better and under which assumptions.

An analysis on how to carry out the grid-search algorithm and a preliminary comparison with the batch-wise unfolding MPCA were performed by Fracassetto (2022) and Sartori (2023); however no detailed guidelines have been drawn up on how to apply the assumption-free modelling and the assumptions behind the application of the model were not verified.

In this Master Thesis these issues are addressed and a comparison between the two models will be performed. The Thesis is divided in 4 chapters. Chapter 1 introduces the process monitoring theory and the statistics that will be used. Chapter 2 introduces the datasets that will be used for the evaluation of the monitoring performances. Chapter 3 describes in detail how the assumption-free model is implemented. Chapter 4 shows the results of the monitoring using the assumption-free modelling and its performances are compared to a batch-wise unfolding MPCA.

Chapter 1

Background on process monitoring

In this Chapter the theory behind process monitoring will be discussed. Principal component analysis (PCA) will be first introduced and then extended to batch processes. Moreover, two types of data unfolding and their respective characteristics will be considered: batch-wise unfolding and variable-wise unfolding: Lastly, the features of an assumption-free modelling and its advantages and drawbacks are shown.

1.1 Principal component analysis

Principal component analysis is an unsupervised machine learning methodology which performs data grouping by uncovering hidden relations between variables; indeed it is not a model able to predict a quality variable, but it is able to assess whether an observation is similar to the ones used to train the model.

The main tasks of PCA are:

- Data compression
- Clustering
- Finding correlation between variables and observations

It completes these tasks by finding the directions of maximum variability of the data relying on the eigenvector decomposition of the correlation matrix. The directions of maximum variability will define the space of the new latent variables.

The data are contained in the **X** matrix whose dimensions are $N \times V$. Prior to proceeding with the PCA, the data contained in the **X** matrix are autoscaled. It is done by subtracting the mean and dividing by the standard deviation each column of **X**; performing autoscaling allows one to give the same importance to each variable regardless of its unit of measure and its range.

To extract the directions of maximum variability of the data, the correlation matrix is calculated according to Wise et al. (1996) as:

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

where m is the number of rows of the data matrix.

Each eigenvector of the covariance matrix of $\boldsymbol{X}$ is then calculated according to Wise et al. (1996)

$$\mathrm{cov}(\mathbf{X})\mathbf{p}_a = \lambda_a \mathbf{p}_a \quad , \tag{1.2}$$

where λ_a is the eigenvalue associated with the eigenvector and it is a measure of the variance explained by each principal component while $\mathbf{p_a}$ $(1 \times V)$ is called "loading" and contains the information on how the variables are related to each other on the a^{th} principal component. The loadings are orthonormal, meaning that each pair is orthogonal and of unit length.
The coordinates of the point in the space of the principal component can be calculated from

$$\mathbf{T} = \mathbf{XP} \quad . \tag{1.3}$$

The score matrix $\mathbf{T}$ $(N \times A)$ represents the coordinates of each observation in the space of the A principal components, which are a linear combination of the original variables contained in the matrix $\mathbf{X}$. The matrix $\mathbf{P}$ $(A \times V)$ contains the loadings related to all the A principal component.
In order to reduce the dimension of $\mathbf{X}$ and give a better understanding of the correlation between data, only the significant part of the measurement should be retained, this can be accomplished by selecting few principal components. PCA decomposes $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{TP^T} + \mathbf{E} \quad . \tag{1.4}$$

$\mathbf{E}$ is the residual matrix which contains the variance that is not captured by the model. This is the non-systematic part of the measurements, namely noise.
Each row $\mathbf{t_n}$of the score matrix represents the coordinates in the latent space for the n^{th} observation. This information is displayed in the score plot where points which are close have similar characteristics. In the loadings, information of the correlation between variables are stored. Loadings have decreasing importance, indeed the loading related to the first principal component is the one that explains more variance. In the loading plot, the relation between variables represented by each PCs can be seen.
The choice on how many principal components have to be included into the model is made considering the root mean square error of cross-validation (*RMSECV*) which is calculated, in accordance with Wise et al. (1996) as

$$RMSECV_a = \sqrt{\frac{1}{L}\sum_{l=1}^{L}(\hat{y}_l - y_l)} \quad , \tag{1.5}$$

where y_l are the observation not included into the model while $\hat{y}_l$ are the prediction for those observations and L is the number of samples not included into the model. A sample corresponds to a measurement of all the V variables. The $RMSECV_a$ is referred to a PCs and the number of retained principal component is the one for which $RMSECV_a$ reaches a minimum or where an "elbow" is present, meaning that by increasing the number of principal components only noise will be included.
In order to assess the similarity of an observation with respect to the mean, the Hotelling's T^2 statistic is used. It is defined as

$$T_n^2 = \mathbf{t}_n \mathbf{\Lambda}^{-1} \mathbf{t}_n^{\mathbf{T}} \quad , \tag{1.6}$$

where $\mathbf{\Lambda}^{-1}$ is the inverse of the diagonal matrix of the eigenvalues. Therefore T_n^2 is the sum of normalized square of scores (Wise et al., 1996). From a geometrical point of view, it measures the distance between the origin of the space of the latent variable and the n^{th} observation. High value of T^2 means that an observation is far from the average behaviour.

In order to understand if an observation is fitted well by the model, the square prediction error of the observation *n* (*SPE*) is used.

$$SPE_n = \mathbf{e}_n \mathbf{e}_n^{\mathbf{T}} \quad . \tag{1.7}$$

It is the sum of squares of the residual of each variable for the n^{th} observation, indeed $\mathbf{e_n}$ is a row of the **E** matrix. Graphically, it is the orthogonal distance between the point in the space of the original variables and the projection of that point into the model space. High values of *SPE* are indicators of an observation which is not well described by the model, meaning that the correlation between variables in that sample differs from the one found in the historical data used to train the model.

These two statistics are used to understand if a new observation can be considered normal or abnormal. To perform this task some statistical confidence limits are needed. These limits are calculated from the inverse of a distribution with a certain degree of confidence α.

The Hotelling's T^2 statistic is calculated from the ratio of sum of squares of multi-normally distributed scores and a variance, therefore it is the ratio of two χ^2 distributed variables. This is approximated by an *F* distribution. Indeed, the confidence limit on the T^2 statistic is calculated with α confidence level according to Wise et al. (1996) from

$$T_{lim}^2 = \frac{A(N-1)}{(N-1)} F_{A,N-1,\alpha} \quad . \tag{1.8}$$

The *SPE* is a sum of squared variables which are normally distributed, therefore is χ^2 distributed and its limits is

$$SPE_{lim} = \frac{\sigma^2}{2\mu} \chi^2_{\frac{2\mu^2}{\sigma^2},\alpha} \quad . \tag{1.9}$$

Where σ^2 is the standard deviation and μ is the mean of the population of the *SPE*.

Both statistics and their relative limit can be used to detect a fault during a process. If a new observation falls outside one or both statistical limits an alarm is triggered, and the observation is identified as abnormal.

1.2 Multi-way principal component analysis (MPCA)

The extension of PCA to batch process is called multi-way PCA because the matrix containing data has one more dimension, time. Batch process data are stored in a three-dimensional matrix $\boldsymbol{X_{3D}}$ ($N \times V \times K$) like reported in Figure 1.1.

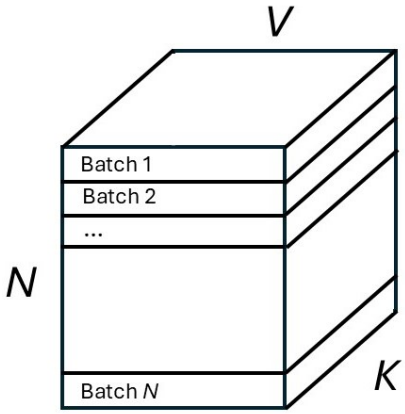


Figure 1.1. *A typical three-dimensional matrix that contains batch process data.*

In Figure 1.1, N is the number of batches, V the number of variables and K the number of samples. Prior to the application of PCA, $\boldsymbol{X_{3D}}$ must be unfolded, meaning that it must be transformed into a two-dimensional matrix. Chamaco et al. (2008) highlighted that many possible strategies exist to perform the unfolding and to build the corresponding PCA model, depending on the characteristics of the batch process. In this Thesis batch-wise unfolding and variable-wise unfolding will be considered.

1.2.1 Batch-wise unfolding MPCA

Using batch-wise unfolding, $\boldsymbol{X_{3D}}$ is unfolded in the variable direction (Figure 1.2), indeed the matrix is decomposed into K matrixes of dimensions $N \times V$. The resulting unfolded matrix is **X** with N rows and VK columns.

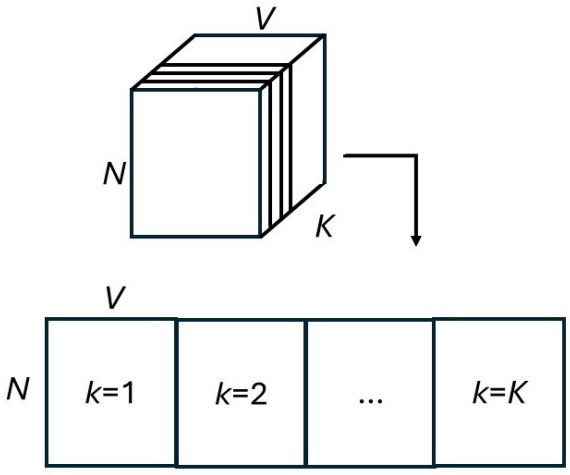


Figure 1.2 *Graphical representation of the batch-wise unfolding.*

A row of the resulting matrix represents the entire history of a single batch, while a column represents the same variable for all the batches at a specific time instant. To perform PCA, $\boldsymbol{X_{3D}}$

must have the third dimension equal for all the batches, namely the matrix must be aligned. This can be done by applying dynamic time warping (Kassidas et al. 1998) and represents one of the downsides of batch-wise unfolding because, not only it may generate artifacts when a batch is significantly shorter than a reference batch (Sartori et al. 2023), but also this process may be computationally demanding, depending on the size of the dataset. Once the data are aligned and the matrix has been unfolded, PCA can be applied, choosing A PCs by looking for the minimum of *RMSECV*.

Now the scores **T** and loadings **P** can be obtained, and the observation can be projected into the core plot (Figure 1.3).

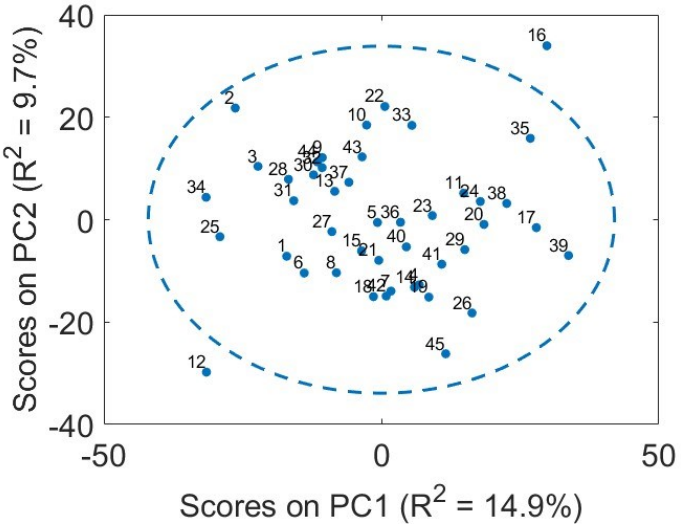


Figure 1.3. *Score plot obtained after performing a batch-wise unfolding MPCA. Each points represents a batch, the dotted line is the 95% confidence interval on the multivariate distribution of the scores. This dataset is the one indicated as Dataset 1 in Chapter 2.*

The score matrix has N rows and A columns therefore a point in the score plot is the summary of the entire batch run and differences and similarity can be identified.

The loadings on the other hand contain the time information of the process, indeed **P** has A columns and VK rows. Each column therefore contains information on how the variables are related in each time instant. In batch-wise unfolding, the correlation between variables is updated every samples. Indeed, this type of unfolding is able to maintain dynamics of the process

Both the T^2 statistic and the *SPE* statistic, with their respective confidence limits can be evaluated. A statistic for each batch is calculated. However, by looking at the entire history of the batch, information of a local departure from the average behaviour might be lost. Therefore, for each sample both statistics can be computed.

Firstly, the scores for the time instant k are evaluated as

$$\mathbf{t}_{n,k} = \left(\mathbf{P}_k^{\mathbf{T}}\mathbf{P}_k\right)^{-1}\mathbf{P}_k^{\mathbf{T}}\mathbf{x}_{n,k} \quad , \tag{1.10}$$

where the loadings $\mathbf{P}_k$ are the ones from time instant 1 to time instant *k*. From $\boldsymbol{t_{n,k}}$ is possible to evaluate the Hotelling's

$$T^2_{n,k} = \mathbf{t}_{n,k}\boldsymbol{\Lambda}^{-1}\mathbf{t}^{\mathbf{T}}_{n,k} \quad , \tag{1.11}$$

Where $\mathbf{t}_{n,k}$ is the row vector of the scores from time instant 1 to time instant *k*.
Similarly, *SPE* is calculated for each time instant starting from the reconstructed value of the original variables until time instant *k* ($\widehat{\mathbf{X}}_k$).

$$\widehat{\mathbf{X}}_k = \mathbf{T}_k\mathbf{P}_k^{\boldsymbol{T}} \quad . \tag{1.12}$$

Then the residuals are evaluated as

$$\mathbf{E}_k = \widehat{\mathbf{X}}_k - \mathbf{X}_k \quad , \tag{1.13}$$

and the SPE for the time instant *k* is

$$SPE_{n,k} = \mathbf{e}_{n,k}\mathbf{e}^{\mathbf{T}}_{n,k} \quad . \tag{1.14}$$

Since *SPE* has a value for each time instant, the confidence limit for each sample can be evaluated in order to obtain a control chart that embeds the time dependency of batches. The instantaneous confidence limit is evaluated in accordance with Nomikos and MacGregor (1995) as:

$$SPE_{lim,k} = \frac{{\sigma_k}^2}{2\mu_k}\chi^2_{\frac{2{\mu_k}^2}{{\sigma_k}^2},\alpha} \quad . \tag{1.15}$$

rom the control charts obtained (Figure 1.4) a fault is detected if a new observation has a certain number of consecutive points out of the confidence limit. This number of points is evaluated from the NOC data used to calibrate the model.

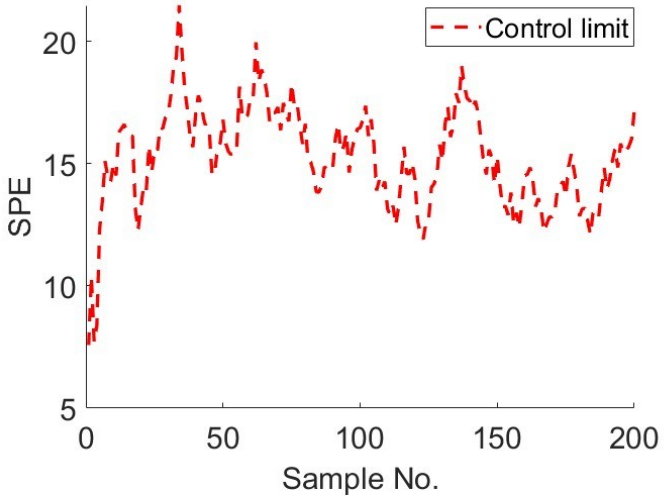


Figure 1.4 *SPE control chart built on a batch-wise unfolding MPCA obtained using (1.13). This dataset is the one indicated as Dataset 1 in Chapter 2.*

Once the model is trained and the control charts have been constructed is possible to project a new observation into the model to assess if it is faulty or not. The monitoring can be performed by following these steps

1. A new observation $\mathbf{x}_{new}$ is measured and it is synchronized by applying relaxed greedy time warping (Gonzàlez-Martínez et al., 2011), which is the counter part of DTW used for on-line monitoring. The synchronization is performed only if the dataset has uneven length. The synchronized observation obtained is then autoscaled using the mean and the standard deviation coming from the NOC data.
2. The scores are obtained using equation (1.8) and the Hotelling's T^2 and the *SPE* are evaluated using equations (1.9) and (1.12).
3. The statistics obtained are than projected into the control charts and compared to the limits evaluated with equations (1.6) and (1.13). If the point is outside the limits, a counter for the respective statistic is increased by one.

These three steps are iterated either until the batch is completed, meaning that no alarm arose and the batch is in a state of statistical control, or until the counter exceeds the limit for a consecutive number of points greater than the one found in the dataset used to train the model. If the second condition is verified on one of the two statistics the batch is regarded as faulty.

1.2.2 Variable-wise unfolding MPCA

The other unfolding strategy is variable-wise unfolding, it consists in unfolding the three-dimensional matrix $\mathbf{X}_{3D}$ into the batch direction to obtain a matrix $\mathbf{X}$ with $N \times K$ rows and V columns (Figure 1.5).

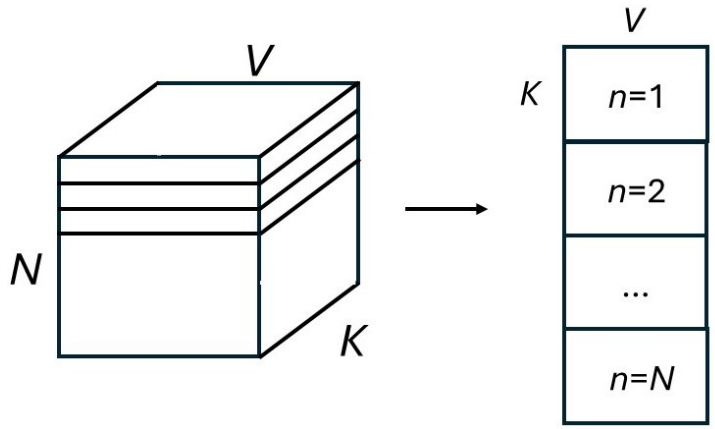


Figure 1.5 *Variable-wise unfolded matrix. Each row of the unfolded matrix represents a sample from a batch. A column contains all the samples for all the batches of a single variable.*

This type of unfolding does not require that the trajectories be aligned. Therefore data can be immediately used, and is a great advantage as the PCA building is significantly less computationally demanding and a new observation can be directly projected without the need of using RGTW. A row of $\mathbf{X}$ corresponds to the summary of a time instant of a single batch, while a column represents the entire history of the V variable. When autoscaled, the grand average of each variable is subtracted from the corresponding columns of the unfolded matrix

and each column is divided for the entire variability of that variable. Therefore, unlike the batch-wise unfolding, a mean behaviour of all the batches in all the time instant is taken into account (Kourti, 2003).

After the unfolding, a PCA model can be applied, and the loadings are obtained according to equation (1.2). The scores are subsequently evaluated using equation (1.3). In contrast to batch-wise unfolding, the intrinsic time behaviour of the batches is stored in the scores **T**, indeed this matrix has $N \times K$ rows and A columns.

Therefore, a trajectory for each batch can be identified in the score plot (Figure 1.6)

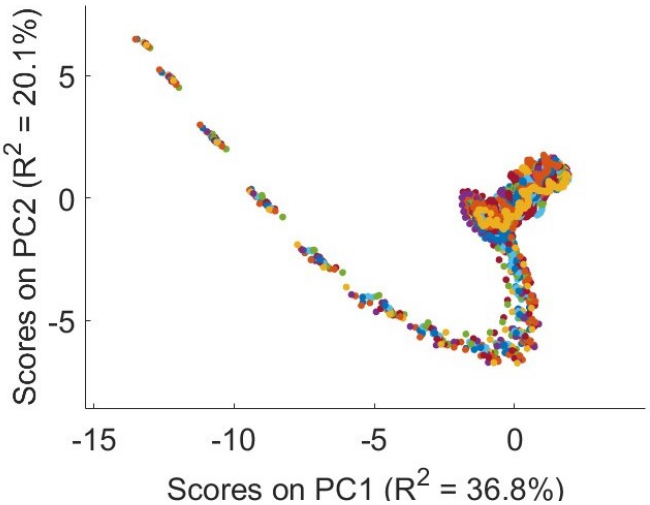


Figure 1.6 *Score plot obtained after applying variable-wise unfolding MPCA. Each point represents the summary of a time instant of a batch. This dataset is the one indicated as Dataset 1 in Chapter 2.*

On the other hand, the loadings contain only the mean correlation between variables, not the instantaneous one, leading to a loss in the capability of the model to capture auto- and cross-correlation between observations.

However, performing SPM using a variable-wise unfolding MPCA is not an easy task. Indeed, following the projection of a new observation into the score plot, it is not possible to assess whether the obtained point is similar to the one used to calibrate the model or not; moreover, lots of correlation is not captured by the model and remains in the residuals therefore their distribution is not normal and a control chart built using equation (1.7) is not theoretically correct.

1.2.3 Assumption-free modelling

To overcome the clear problems of the variable-wise unfolding MPCA, Westad et al. (2015) published a paper in which it was explained how to perform SPM starting from a variable-wise unfolding MPCA. Their idea consists in partitioning the score plot in order to reconstruct a mean batch trajectory and, once identified, build statistical control interval around it. The monitoring should be performed by projecting a new observation onto the score plot and comparing it to the control limits that have been found.

The main steps are reported as follows.

Calibration section.

1. Variable-wise unfold the matrix $\mathbf{X_{3D}}$ and autoscale the data.
2. Build a PCA model on the unfolded matrix **X**.
3. Partition of the score plot using a "grid search algorithm" in the score plot to obtain a grid that gives the most "grid elements".
4. Calculate the mean of each "grid element" and the mean of each batch inside the "grid element".
5. Interpolate the overall mean of each "grid element" to obtain the common batch trajectory.
6. Orthogonally project the means of the batches (calculated at step 4) on the common trajectory and evaluate their "relative time", which is the ratio between the number of points of the common trajectory before the projection and the overall number of points in the trajectory. Save the distance of each mean from the common trajectory. Calculate the SPE
7. From the distances estimate the standard deviation around the common trajectory and build the control interval.
8. For each grid element a limit on SPE is identified with the relative time associated.

Monitoring section.

1. Preprocess and autoscale the new observation.
2. Evaluate the scores using equation (1.3)
3. Project the scores onto the score plot to estimate the distance from the trajectory.

However, no details have been reported by Westad et al. (2015) on how to perform these steps. Particularly, no clear instructions were given on how to perform the grid search algorithm and on how to build the control intervals. Moreover, no guidelines on how to assess if a new batch is faulty or not were explained.

Fracassetto (2022) and Sartori (2023) investigated the assumption-gree method and developed a code used to perform the grid search algorithm and carry out the monitoring. However, no precise instructions were given on how to build the model and some of the assumptions were not proven. In the following chapters the guidelines on the procedure to apply the assumption-free model will be given and its monitoring performance will be assessed.

Chapter 2

Datasets used

In order to test the effectiveness of the assumption-free monitoring approach, five datasets coming from both industrial and simulated processes have been used. A summary of the datasets is given in Table 2.1. Not all datasets had aligned data.

Table 2.1 *Datasets summary*

Dataset No.	Description	Aligned/Not Aligned	Industrial/Simulated	Reference
1	Styrene-Butadiene polymerization	Aligned	Simulated	Nomikos and MacGregor (1994)
2	Low density polyethylene polymerization	Aligned	Industrial	Nomikos and MacGregor (1995a)
3	Saccharomyces cerevisiae production	Not aligned	Simulated	González-Martínez et al. (2018)
4	Penicillin production	Not Aligned	Simulated	Birol et al. (2002)
5	Herbicide drying	Not aligned	Industrial	García-Muñoz et al. (2003)

The datasets are divided into two blocks, the first one is used to calibrate the model (calibration set) the second one to assess the monitoring performances (validation set).
Each process and its dataset structure will be described in the following paragraphs

2.1 Dataset 1: Simulated styrene-butadiene polymerization

This dataset comes from the paper of Nomikos and MacGregor (1994). It contains data from a simulated polymerization between styrene and butadiene. The simulation has been carried out using the model developed by Broadhead et al. (1985) for the production of a styrene-butadiene rubber latex (SBR). According to the model, the process starts with the charge of the jacketed reactor with: SBR particles, initiator ($S_2O_8^{2-}$), chain transfer agent, emulsifier (fatty acid soap), water and small quantities of monomers of styrene (S) and butadiene (B). The monomers are then fed to the reactor at an almost constant rate to continue the polymerization. The reactor is assumed to be cylindrical and perfectly mixed. The reactions are exothermic, and the temperature is controlled by adjusting the flow of cooling water in the reactor's jacket.
The polymerization starts from to the decomposition of the initiator into radicals.

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

Once the radicals are formed they react with one of the two monomers that are charged in the reactor according to the following stoichiometry.

$$SO_4^- \cdot + S \rightarrow SO_4^- + S \cdot \quad (2.2)$$

$$SO_4^- \cdot + B \rightarrow SO_4^- + B \cdot \quad (2.3)$$

The radicals of the monomers that have been obtained can propagate the reaction lengthening the polymer chain. Radical polymerizations are non-specific, therefore radicals can react in random order.

$$\sim S \cdot + S \rightarrow \sim SS \cdot \quad (2.4)$$

$$\sim S \cdot + B \rightarrow \sim SB \cdot \quad (2.5)$$

$$\sim B \cdot + S \rightarrow \sim BS \cdot \quad (2.6)$$

$$\sim B \cdot + B \rightarrow \sim BB \cdot \quad (2.7)$$

The prolongation of the polymer chain stops when a reaction between two radicals occurs.
Batch-to-batch variability was introduced by considering impurities in the initial charge of the reactor. Moreover, noise was added to the monomer's feeds and to their temperature measurements. Each batch lasted 1000 min and samples were taken every 5 min, therefore each batch has 200 samples. The variables measured are shown in Table 2.2, units of measurement of flowrates are not available.

Table 2.2 *Dataset 1: variables measured in the process.*

Variable number	Name	Units
1	Time	min
2	Styrene flowrate	N.A
3	Butadiene flowrate	N.A
4	Feed temperature	°C
5	Reactor temperature	°C
6	Cooling water temperature	°C
7	Reactor jacket temperature	°C
8	Latex density	g/L
9	Total conversion	[-]
10	Net energy released	J min^{-1}

The reaction rate is high at the beginning of the process. Indeed, the plots of the variable's time evolution show how most of the dynamics occur in the first 20 samples (the first 100 minutes). As a matter of fact, the total conversion (Figure 2.1a) reaches an almost constant value after 50 samples, which is a fourth of the entire duration of the process. The same behaviour can be seen for the net energy released (Figure 2.1b). The rate at which the energy is released is extremely high at the beginning, meaning that the reaction is fast and after 200 minutes it settles at about 820 J/min.

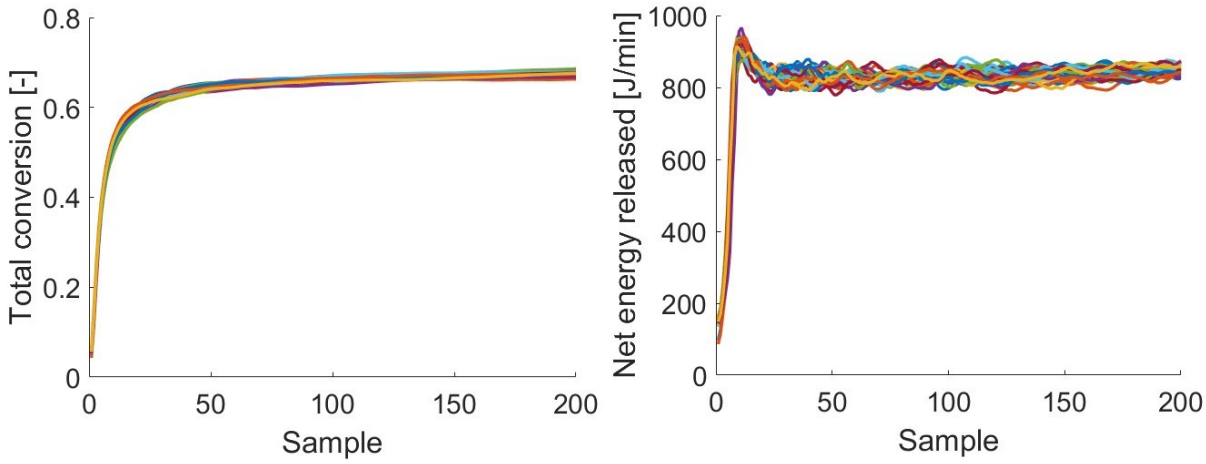


Figure 2.1. *Dataset 1, total conversion of the reactants (a) and net energy released (b) through all the process. Each line represents a batch of the calibration set.*

The calibration set of this process contains 45 NOC batches, while the validation set contains 8 batches, 6 of them are NOC and 2 of them are faulty (Table 2.3).

Table 2.3 *Dataset 1: summary of calibration and validation set*

Set	NOC	Faulty	Aligned/unaligned	Duration [min]	Samples
Calibration	45	0	Aligned	1000	200
Validation	6	2			

The cause of the fault is the same in both batches but differs in intensity and timing. Indeed, the first faulty batch had an impurity in the butadiene feed 30% larger than the one of the base case and the disturbance occurred at the beginning of the process. The second faulty batch had an impurity in the butadiene feed grater then 50% which happened halfway through the process.

2.2 Dataset 2: Industrial polymerization

This dataset comes from the paper of Nomikos and MacGregor (1995a). The process polymerization consists of two steps. During the first step the flows of the heating medium are adjusted to establish proper control over pressure and temperature changes. The solvent used to load the reactants into the reactor is vaporized and removed from the pressure vessel; due to the intensity of the vaporization stirring is not required. After about one hour the solvent is removed, and the second step begins. In this step the reaction is completed, and the polymer is formed, in the meantime pressure and temperature are still controlled. The process ends by pumping the polymer obtained to a downstream unit after about two hours of processing.

The 10 measured variables are reported in Table 2.4 and no units of measurements are available due to confidentiality reasons.

Table 2.4 *Dataset 2: variables measured in the process.*

Variable number	Name
1	Reactor temperature 1
2	Reactor temperature 2
3	Reactor temperature 3
4	Pressure 1
5	Flowrate 1
6	Heating/cooling medium temperature 1
7	Heating/cooling medium temperature 1
8	Pressure 2
9	Pressure 3
10	Flowrate 2

Table 2.5 *Dataset 2: summary of calibration and validation set*

Set	NOC	Faulty	Aligned/unaligned	Duration [h]	Samples
Calibration	50	0	Aligned	≃2	100
Validation	4	1			

The data are aligned and the number of samples is 200 for each batch. The calibration set contains 50 NOC batches while the validation set contains 4 NOC batches and 1 faulty batch (Table 2.5) which was upset since the beginning and yielded to a polymer of marginal quality.

2.3 Dataset 3: Simulated saccharomyces cerevisiae production

This dataset is included in the MVBatch toolbox developed by González-Martínez et al. (2018) and is available at https://github.com/jogonmar/MVBatch/releases. The data comes from a simulated process for the production of *Saccharomyces Cerevisiae* and is based on the model developed by Lei et al. (2001) of the fermentation of this yeast on a glucose limited medium. The production is divided into four phases: lag phase, first exponential growth, second exponential growth and stationary phase. In the lag phase, the yeast becomes acclimated to the culture medium before starting its reproduction. In the second phase the glucose in excess is consumed and ethanol is produced together with pyruvate and acetate, this phase ends once the glucose is completely consumed by the cells. In the third phase the cells start growing using ethanol as substrate and producing acetate. In the development of the model, the assumptions of a perfect abiotic subsystem had been considered (Lei et al. 2001). Among these assumptions the most relevant are perfect mixing of the reactor and perfect control with respect to oxygen, temperature and pH inside the reactor.

The variables taken into account in the model are shown in Table 2.6.

Table 2.6 *Dataset 3: variables measured in the process*

Variable number	Name	Units
1	Glucose concentration	$g\ L^{-1}$
2	Pyruvate concentration	$g\ L^{-1}$
3	Acetaldehyde concentration	$g\ L^{-1}$
4	Acetate concentration	$g\ L^{-1}$
5	Ethanol concentration	$g\ L^{-1}$
6	Biomass concentration	$g\ L^{-1}$
7	Active cell material	[-]
8	Acetaldehyde dehydrogenase	[-]
9	Specific oxygen uptake rate	$mmol\ g^{-1}\ h^{-1}$
10	Specific CO_2 evolution rate	$mmol\ g^{-1}\ h^{-1}$
11	Simulation time	h

Each batch has been processed for about 34 hours. However data are not aligned, therefore, both the number of samples and the sampling rate vary from one batch to the other. The average number of samples is 211 with a standard deviation of 32 samples. To generate batch-to-batch variability, gaussian noise of low magnitude was added to the initial condition (standard deviation of 10%) and to the measurements (standard deviation of 5%) to simulate typical sensor errors.

As summarized in Table 2.7 the calibration set contains 40 NOC batches, while the validation set contains 45 batches. 5 of them are NOC, the remaining 40 are faulty.

Table 2.7 *Dataset 3: summary of calibration and validation set*

Set	NOC	Faulty	Aligned/Not aligned	Duration [h]	Mean number of Samples
Calibration	40	0	Not aligned	≃34	211
Validation	5	40			

Two types of faults have been simulated. The first one is obtained by modifying an internal rate constant associated with the glucose consumption which leads to a higher utilization of the substrate with respect to the NOC. The second type of fault is generated by adding a bias in the biomass concentration probe. Both faults have been considered with different magnitude at the beginning and halfway through the process.

2.4 Dataset 4: simulated penicillin production

The data have been obtained using Pensim, a simulator developed by Birol et al. (2002). This software simulates a fed-batch fermentation to produce penicillin. A simplified P&I of the process is shown in Figure 2.2.

The penicillin production is arranged in two steps.

The process begins with a batch operation that involves the charge of the reactor with the bacteria *Penicillum Chrysogenum* and with glucose in order to promote biomass growth, this step ends when the glucose concentration drops below a certain threshold. On the other hand, this step is a fed-batch operation in which glucose and air are continuously fed to the reactor at a constant rate, in the meantime glucose is consumed to produce penicillin. It is assumed that the process ends when 14 L of substrate have been added to the reactor (Sun et al., 2011). During the process temperature and pH are controlled.

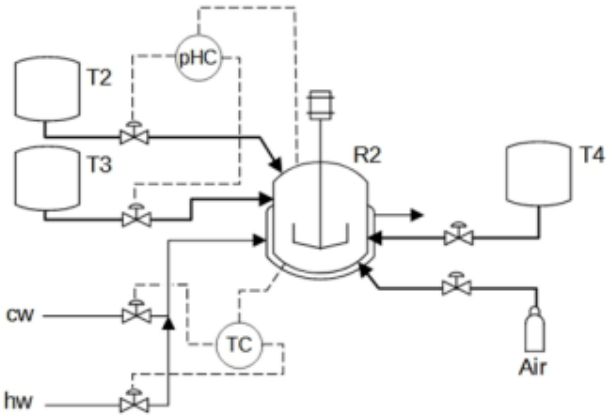


Figure 2.2 *Dataset 4: simplified P&I of the simulated process (Sartori, 2023)*

The data that will be used have been retrieved by Sartori (2023), who created batch-to-batch variability by adding noise in the form of additive random numbers sampled by a normal distribution with zero mean and standard deviation σ. The variables measured are shown in Table 2.8.

Table 2.8 *Dataset 4: variables measured in the process; σ is the standard deviation of the distribution*

Variable No.	Name	Units	σ
1	Time	h	0
2	Aeration rate	L h^{-1}	0.083
3	Agitator power	W	0.167
4	Glucose feed rate	L h^{-1}	0.00083
5	Glucose feed temperature	K	0.167
6	Glucose concentration	g L^{-1}	0
7	Dissolved O_2	g L^{-1}	0.0067
8	Biomass concentration	g L^{-1}	0
9	Penicillin concentration	g L^{-1}	0
10	Bulk volume	L	0.033
11	Dissolved CO_2	mmol L^{-1}	0
12	pH	[-]	0.0167
13	Fermentor temperature	K	0.167
14	Generated heat	cal	0
15	Acid flowrate	L h^{-1}	$3.3 \cdot 10^{-7}$
16	Base flowrate	L h^{-1}	$3.3 \cdot 10^{-6}$
17	Cooling/heating water flowrate	L h^{-1}	0.83
18	Cumulated acid flowrate	L	0
19	Cumulated base flowrate	L	0
20	Cumulated glucose feed rate	L	0

Process variability has been introduced by randomly changing the initial condition (Table 2.9) by sampling ε from a standard normal distribution (mean 0 and standard deviation equal to 1) and by randomly varying the threshold that determines the end of the first processing step between 0.3 and 7 g L^{-1}.

Table 2.9 *Dataset 4: initial conditions and operating variables; ε is a random number sampled from a standard normal distribution with mean 0 and standard deviation equal to 1*

Initial condition	Units	Nominal value
Glucose concentration	g L^{-1}	15 + ε
Dissolved oxygen	%	1.16
Biomass concentration	g L^{-1}	0.1
Penicillin concentration	g L^{-1}	0
Culture volume	L	150 + 10ε
CO_2 concentration	mmol L^{-1}	0.75 + 0.05ε
Hydrogen ions concentration	mol L^{-1}	$10^{-5+0.1\varepsilon}$
Fermentor temperature	K	298
Generated heat	kcal h^{-1}	0
Operating variable	**Units**	**Nominal value**
Aeration rate	L h^{-1}	8
Agitator power	W	30 + ε
Glucose feed rate	L h^{-1}	0.04 + 0.0025ε
Glucose feed temperature	K	296
Culture volume	L	150 + 10ε
pH	[-]	5
Fermentor temperature	K	298

As summarized in Table 2.10 the calibration set is made of 30 NOC batches with a mean duration of 200 h. The variables are measured every 15 min leading to an average number of 800 samples. The validation set contains 39 batches, 9 of which are NOC, the remaining 30 are faulty.

Table 2.10 *Dataset 4: summary of calibration and validation set*

Set	NOC	Faulty	Aligned/Not aligned	Duration [h]	Mean number of Samples
Calibration	30	0	Not aligned	≃200	800
Validation	30	9			

The types of simulated faults are difference in the aeration rate and difference in the substate feed, with respect to the NOC, these faults are simulated with different magnitude and at the beginning and halfway through the fermentation.

2.5 Dataset 5: industrial herbicide drying

This industrial herbicide drying batch process is described by García-Muñoz et al. (2003). The process goal is to remove and collect all the solvent contained in the initial wet cake and reduce its content to a target level. During the process some chemical structural changes may happen, leading to unacceptable product quality. A schematic representation of the process is shown in Figure 2.3.

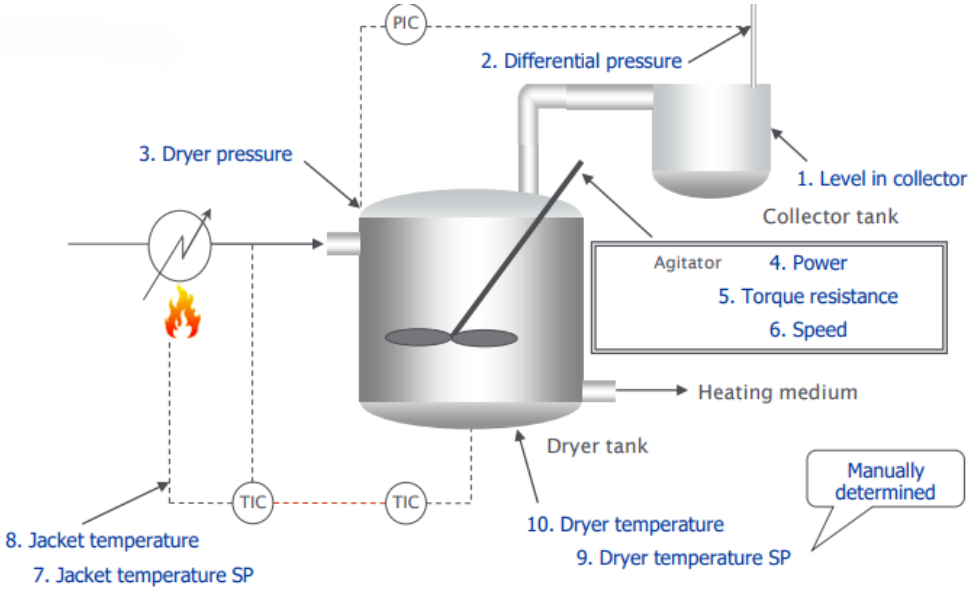


Figure 2.3 *Dataset 5: schematic representation of the drying process. (Aspen ProMV getting started guide, 2017)*

The process is divided into 4 steps.

1. The dryer tank is charged with the wet cake of known mass and unknown solvent content, these two variables vary from a batch to another.
2. Agitation is turned on at about 8 rpm and the heating medium starts flowing in the jacket leading to a slow temperature increase inside the dryer.
3. After the temperature inside the dryer has reached a predefined level, the agitator is turned to high speed (about 30 rpm) and the temperature increase. The end of this processing step is reached when the temperature has reached its maximum, when this occurrence is encountered the agitator is turned back to low speed.
4. The product is cooled and, once the process is complete the agitator is turned again to high speed.

As the process keeps running the evaporated solvent is collected in the collector tank that is emptied at the end of each run. Variability between batches exists because the time at which the agitator is turned from low to high speed (and form high to low speed) may change. Moreover, the peak temperature is not the same for every batch because an operator adjusts the temperature setpoint in order to obtain the desired product quality.

The measured variables are numbered in Figure 2.3 and listed in Table 2.11 The units of measurement are unknown

Table 2.11 *Dataset 5: measured variables*

Variable number	Name
1	Collector tank level
2	Dryer differential pressure
3	Dryer pressure
4	Power to the agitator
5	Torque resistance for the agitator
6	Agitator speed
7	Heating medium temperature set point
8	Jacket temperature
9	Dryer temperature setpoint
10	Dryer temperature

The dataset is taken form the Aspen ProMV getting started guide (2017) at the path C:\ProgramData\AspenTech\AspenProMVDesktop\Examples and contains 69 batches, 30 of them are NOC. However, from a preliminary data visualization of the NOC batches, it was noticed that three of them (Batch No. 1, 2 and 18) had a behaviour which strongly differ from the mean one even though have been classified as NOC. Indeed, the evolution of the differential pressure (Figure 2.4) indicates that two batches (Batch No. 1 and 2) have a completely different behaviour with respect to the average.

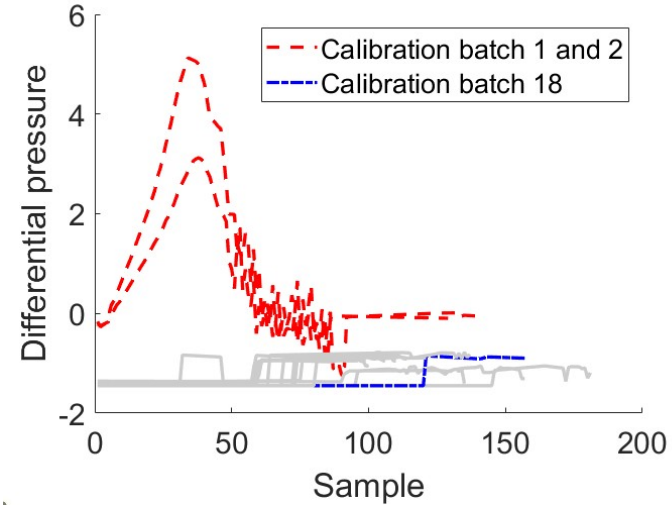
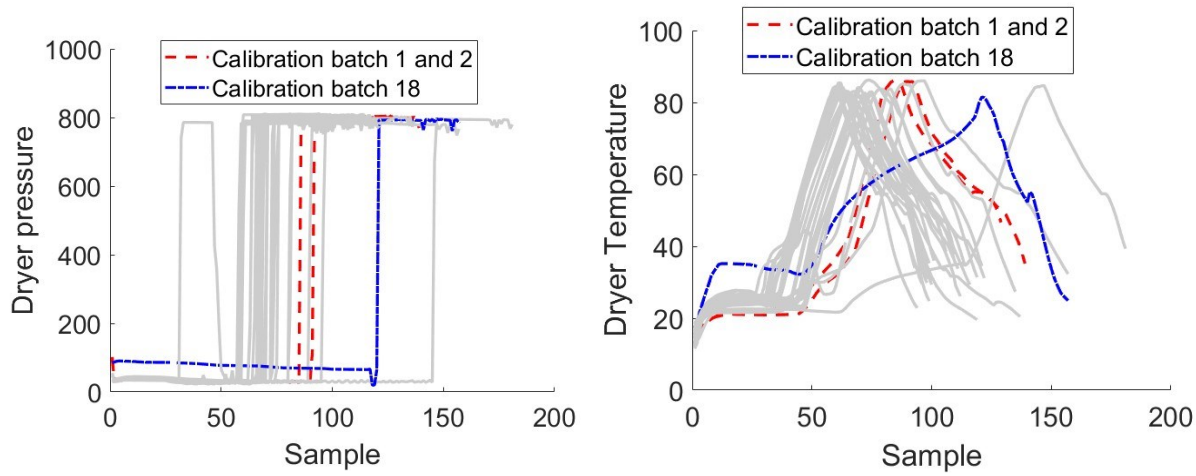


Figure 2.4 *Dataset 5: differential pressure profile*

In Particular, the differential pressure increases until a maximum and only later decreases and fluctuates around zero, instead of remaining negative for the whole duration of the process.
Due to this difference, these batches have been discarded from the dataset in order to avoid bad model calibration.

The other batch that exhibits an unusual behaviour is batch No. 18. The departure from the average is shown in the plot of the dryer pressure (Figure 2.5).



(a) (b)

Figure 2.5 *Dataset 5: (a) dryer pressure profile (b) and dryer temperature (b).*

Batch No. 18 has a higher pressure with respect to all the other batches for most part of the run. Moreover, the dryer temperature rises above the others during the beginning of the batch and has a different trajectory with respect to the others. Therefore, it has been decided to remove this batch.

The dataset is available in both aligned and not aligned form. For the purpose of this study it has been decided to use the unaligned one. The mean number of samples is 129 with a standard deviation of 24. The calibration set of this dataset contained 28 NOC batches which has been reduced to 25 after removing batches No. 1, 2 and 18. The validation set, instead, contains 41 batches, 3 of them are NOC while the remaining 38 are faulty (Table 2.12).

Table 2.12 *Dataset 5: summary of calibration and validation set*

Set	NOC	Faulty	Aligned/Not aligned	Duration [h]	Mean number of Samples
Calibration	25	0	Not aligned	N.A.	129
Validation	41	38			

The faults are due to either high solvent content at the end of the batch or because the product was out of specification.

Chapter 3

Assumption-free modelling implementation

In this Chapter the guidelines for the implementation of the assumption-free modelling are presented. Moreover, the assumptions made in previous studies (Fracassetto, 2022; Sartori, 2023) will be verified. To develop these guidelines, the dataset of the simulated polymerization of styrene-butadiene (Nomikos and MacGregor, 1994) described in §2.1 is considered.

3.1 Inputs to the model

In order to perform process monitoring using an assumption-free modelling, it is not sufficient to use data to calibrate the model and to test the performances. Indeed, it is also necessary to set some hyperparameters that are used to define the grid.

3.1.1 Dataset

The dataset, as discussed in Chapter 2, is composed of two separate blocks. The first one is the calibration set which is used to create the PCA model, to reconstruct the average batch run and to find the control limits. It contains only the NOC batches in order to build a reliable model based only on batches of acceptable quality. The second one is the validation set, which contains both NOC and faulty batches and is used to assess the performances of the monitoring model in terms of detection strength and detection speed.

3.1.2 Hyperparameters

In order to better understand how the assumption-free model is calibrated, some definitions are required before proceeding with the study:

- Grid: partitioned score plot, created by dividing the 2-dimensional score plot into rectangular elements of equal size.
- Grid configuration: division of the grid with a certain amount of cell.
- Cell: an element of the grid.
- Valid cell: a cell where at least β % batches are present. In order to consider a batch present in a cell, at least a score of that batch must be inside the cell taken into account.
- Valid grid: a grid whose valid cells contains at least γ % of all the scores.

- Best grid: the grid, among the valid ones, that contains the highest number of valid cells and the highest percentage of all the scores in the calibration set included.

β and γ, alongside with n_{PC1}^{max} and n_{PC2}^{max} (to be defined below) are the hyperparameters necessary to calibrate the model. Those parameters will influence the grid site, the control limits and the reconstructed mean trajectory which will have a strong impact on the monitoring performance of the model.

The parameters' meaning is listed here.

- β: is the fraction of batches that need to be present in a cell in order to consider it valid.
- γ: is the fraction of the total scores that need to be present in all valid cells in order to consider the grid valid.
- n_{PC1}^{max}: is the maximum number of cells into which the score plot will be divided in the direction of the first principal component.
- n_{PC2}^{max}: is the maximum number of cells into which the score plot will be divided in the direction of the second principal component.

Note that in the discussion to follow we assume that two PCs are used.

For the dataset used in this chapter the hyperparameters values are summarized in Table 3.1.

Table 3.1 *Dataset 1: Hyperparameter values used during the calibration of the assumption-free model.*

Hyperparameter	Value	Units
γ	0.95	Fraction of scores
β	0.90	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

Both γ and β are fractions, therefore their range is between 0 and 1. Westad et al. (2015) suggests setting γ equal to 1. However, it is not recommended to set neither of these parameters to 100% because it will lead to detrimental monitoring performances. n_{PC1}^{max} and n_{PC2}^{max} can theoretically be set arbitrarily to any value. Nevertheless, is not advisable to set one (or both) values too high because it slows down the calibration of the model and it may worsen the representation of the mean batch trajectory. Some preliminary trials suggested setting both parameters to 12.

3.2 PCA model building

Once the hyperparameters are set to the desired value, the calibration of the assumption-free model begins. First of all, the matrix containing the data array is unfolded in the batch direction, autoscaled and then a PCA model is built as described in §1.2.2.

The number of principal components that has been chosen is 2. This choice has not been done because the minimum of *RMSECV* was found as described in §1.1. It has been decided to limit the number of PCs to 2 due to the algorithm used to create the grid. Indeed, the algorithm used

to partition the score plot is not able to work on a dimension greater than 2. This is one of the most important limitations of the assumption-free modelling, as other correlation between variables may be considered during the calibration by adding more PCs, leading to a better description of the process.

Once the PCA model is built, the data matrix can be decomposed according to (1.4) and the score and loading matrixes are obtained.

The scores can now be projected onto the score plot as reported in Figure 3.1.

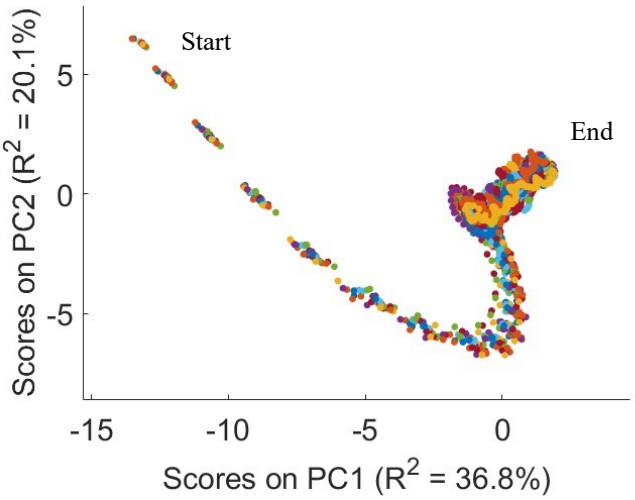


Figure 3.1 *Dataset 1: score plot obtained after variable-wise unfolding the calibration set of Dataset 1 described in §2.1. Each point represents a time instant of a single batch. Each colour represents a batch of the calibration set.*

After the scores are plotted the grid search algorithm can be performed to reconstruct the average batch run.

3.3 Grid search algorithm

The grid search algorithm is a procedure that allows for the identification of the best grid. It consists of two steps: the definition of the grid limits and an iterative procedure that analyses and saves all the information related to a specific grid configuration. Once all the possible configurations have been considered, the best grid is chosen.

3.3.1 Grid limits

The grid limits define the zone of the score plot where the iterative part of the grid search algorithm will be performed. It is carried out by identifying the lowest and highest value of scores on both PCs. The limits are defined as:

$$m_{PC1} = \min\left(\mathbf{t}_{PC1}\right) \quad , \tag{3.1}$$

$$M_{PC1} = \max(\mathbf{t}_{PC1}) \quad , \tag{3.2}$$

$$m_{PC2} = \min(\mathbf{t}_{PC2}) \quad , \tag{3.3}$$

$$M_{PC2} = \max(\mathbf{t}_{PC2}) \quad , \tag{3.4}$$

where $\mathbf{t}_{PC1}$ and $\mathbf{t}_{PC2}$ are the vectors containing respectively the scores on the first and on the second principal component.

Figure 3.2 shows the grid limits for Dataset 1.

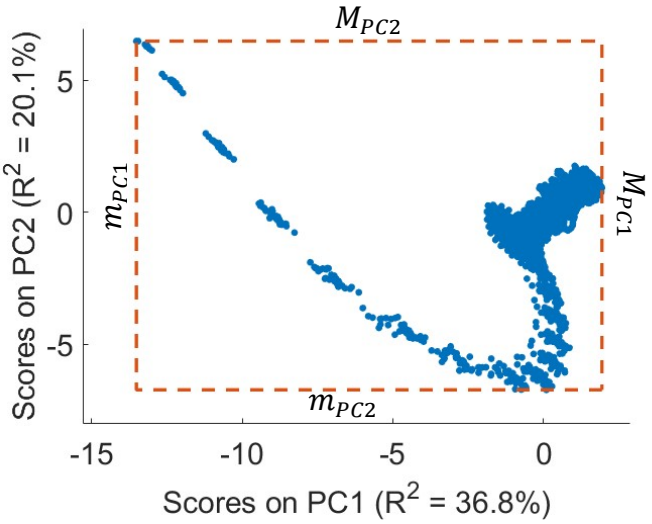


Figure 3.2 *Dataset 1: score plot with grid limits.*

By creating the grid limits in this manner, all the scores are included in the iterative part of the grid search algorithm.

3.3.2 Valid cell identification

In order to reconstruct the average batch run, it is important to identify the valid cells (defined in §3.1.2). Valid cells are identified by partitioning the grid and by analysing the scores that are present in each cell. If the fraction of batches that are present in the considered cell is greater than γ, the cell is classified as valid. For a batch to be present in a cell at least 1 score points related to that batch must be inside the cell.

The algorithm starts working by initially considering a grid with the lowest number of cells, which are increased at each iteration in the direction of the second PC. At each iteration, the cell dimension is fixed and is evaluated as:

$$l_{PC1,w} = \frac{M_{PC1} - m_{PC1}}{n_{PC1,w}} \quad , \tag{3.5}$$

$$l_{PC2,w} = \frac{M_{PC2} - m_{PC2}}{n_{PC2,w}} \quad , \tag{3.6}$$

where $n_{PC1,w}$ and $n_{PC2,w}$ are the number of cells into which the grid area will be partitioned at iteration w.

In the first iteration, both $n_{PC1,w}$ and $n_{PC2,w}$ are equal to 1 and the there is only one cell which corresponds to the grid area show in Figure 3.2. The cell is considered valid because all the scores are present in that area. In this thesis, valid cells will be identified in the plots with green borders.

For each valid cell, the mean of the all the scores on each PC contained in that cell is evaluated as

$$\bar{t}_{a,u} = \frac{1}{M}\sum_{m=1}^{M} t_{a,m} \qquad a = (1,2) \quad , \tag{3.7}$$

where a is the considered principal component $a = (1,2)$ and M is the total number of scores present in the valid cell u. These scores are the nodes that will be used to reconstruct the common trajectory.

In the following iteration, $n_{PC2,w}$ is increased by one and the cell dimensions are evaluated again according to (3.5) and (3.6). Both cells are identified as valid because at least 90% of the batches are present in the cells.

The grid obtained is shown in Figure 3.3.

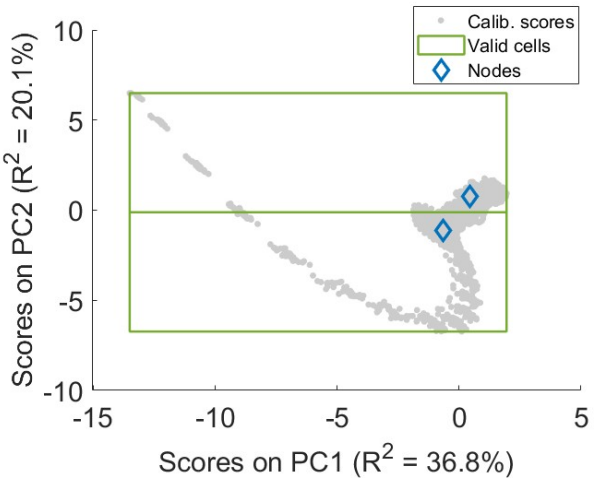


Figure 3.3 *Dataset 1: iteration of the grid search algorithm for the calibration batches.* ($n_{PC1,w} = 1$ *and* $n_{PC2,w} = 2$). *The blue diamonds represent the means of the scores evaluated by (3.7).*

After some iterations the maximum number of cells along the second PC is reached (Figure 3.4).

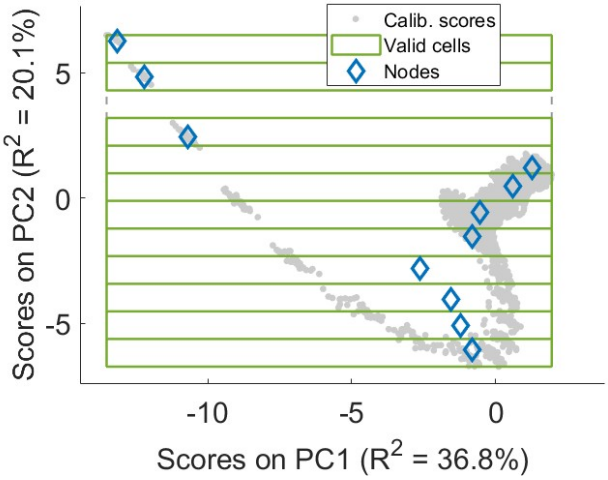


Figure 3.4 *Dataset 1: iteration of the grid search algorithm for the calibration batches.* ($n_{PC1,w} = 1$ *and* $n_{PC2,w} = 12$). *The blue diamonds represent the mean of the scores evaluated by (3.7).*

It is shown that not all cells are identified as valid, indeed the mean of the scores is evaluated only in 11 out of 12 cells.

The grid search algorithm has been developed increasing until its limit the number of cells along the direction of the second principal component. However, doing the opposite lead to the same results.

As the number of cells along the direction of the second principal component has reached its maximum, the next iteration will be performed with $n_{PC1,w} = 2$ and $n_{PC2,w} = 1$, as shown in Figure 3.5.

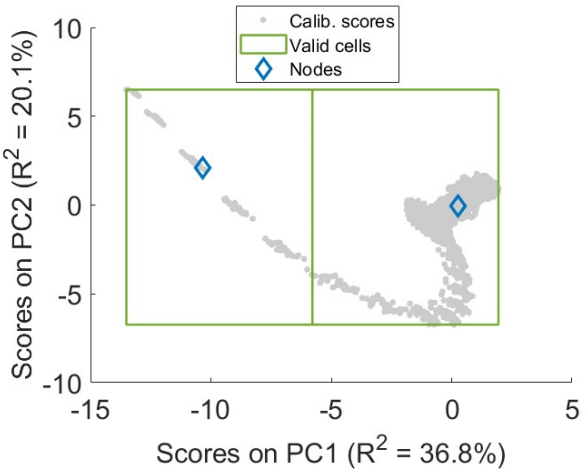


Figure 3.5 *Dataset 1: iteration of the grid search algorithm.* $n_{PC1,w} = 2$ *and* $n_{PC2,w} = 1$. *The blue diamonds represent the mean of the scores evaluated using equation (3.7.)*

The grid search algorithm continues by adding a cell in the direction of the second principal component until the maximum number of cells in that direction is reached. The next iteration will start by further dividing the grid along the first PC and by setting $n_{PC2,w} = 1$. The algorithm proceeds until all the grids have been considered. The last iteration of the grid search algorithm is performed with $n_{PC1,w} = 12$ and $n_{PC2,w} = 12$

A flowchart representation of the grid search algorithm is shown in Figure 3.6.

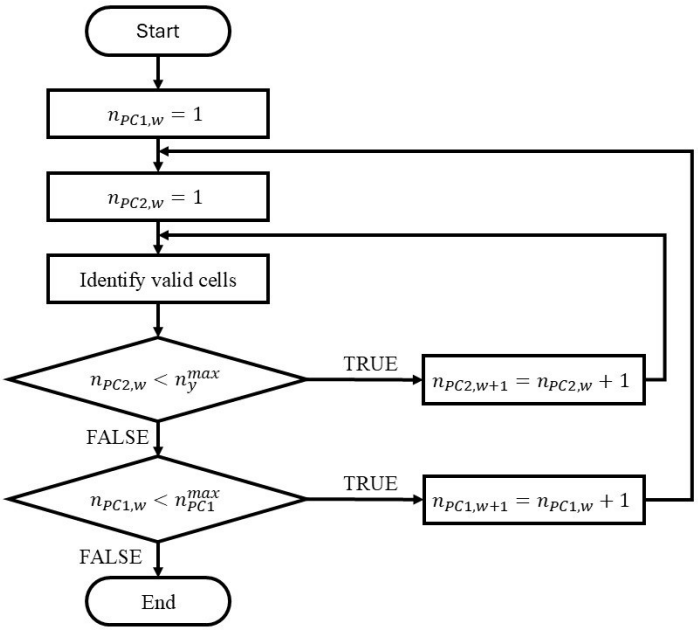


Figure 3.6 *Flowchart diagram representation of the grid search algorithm.*

Moreover, in each valid cell u the mean of scores of batch n is calculated similarly as equation (3.7).

$$\bar{t}_{a,n,u} = \frac{1}{M}\sum_{m=1}^{M} t_{a,m,n} \quad , \tag{3.8}$$

The difference is that the scores considered are those of batch n and not all the scores inside the valid cell u.

At the end of the iterative section of the grid search algorithm, the number of valid cells is calculated for each possible grid configuration.

3.3.3 Grid selection

In the previous section of the algorithm, the number of valid cells has been found for each possible grid configuration. In order to choose the correct grid configuration, parameter γ is taken into account.

In order to identify a valid grid two conditions must be satisfied:

1. The grid has to contain at least γ% of all the scores inside the valid cells.
2. The grid has to be the one with the highest number of valid cells (n_{valid}^{max}).

The grid obtained for dataset 1 is shown in Figure 3.7.

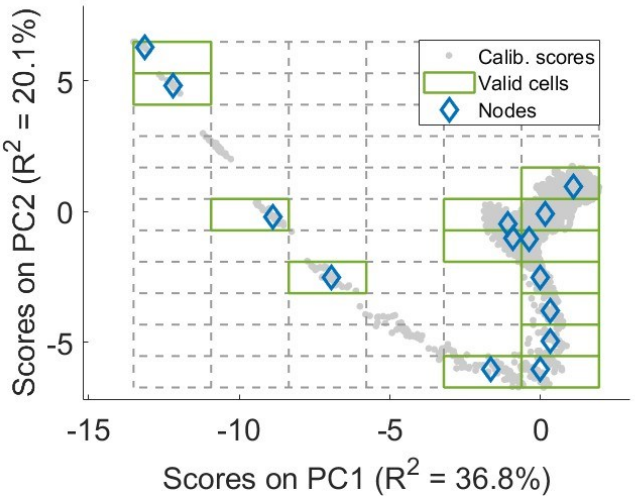


Figure 3.7 *Dataset 1: best grid found by the algorithm for the calibration batches. ($n_{PC1} = 6$ and $n_{PC2} = 11$).*

The best grid has a dimension of 6×11, with 14 valid cells and a percentage of included scores of 98.6%.

3.3.3.1 Particular case of the grid selection

There may be cases in which the highest number of valid cells is the same in two or more configurations. If this condition is met, the grid with the greatest number of scores included in

the valid cells. This situation does not occur in Dataset 1. However, it happens when Dataset 3 (described in §2.3) is used. Therefore, the following example is done considering this dataset. In Table 3.2 the values of the hyperparameters used to calibrate the assumption-free model for Dataset 3 are shown.

Table 313.2 *Dataset 3: Hyperparameter values used during the calibration of the assumption-free model.*

Hyperparameter	Value	Units
γ	0.90	Fraction of scores
β	0.90	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

After applying the grid search algorithm, 3 valid grids have been identified. The grids have different configurations (Figure 3.8), but the same number of valid cells, $n_{valid}^{max} = 24$.

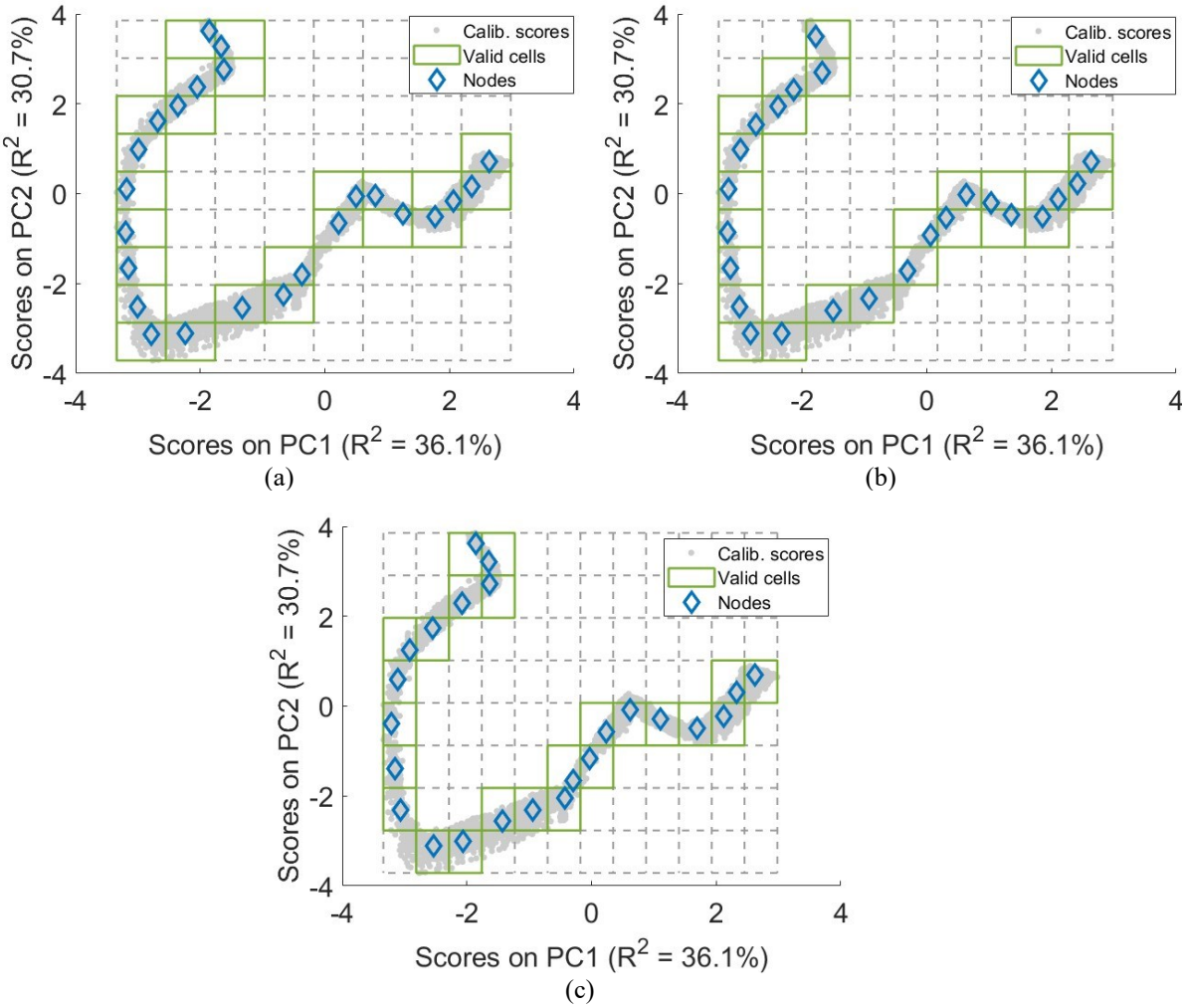


Figure 3.8 *Dataset 3: valid grids identified by the grid search algorithm for the calibration batches. (a) .* $n_{PC1} = 8$ *and* $n_{PC2} = 9$. *(b) .* $n_{PC1} = 9$ *and* $n_{PC2} = 9$. *(c)* $n_{PC1,w} = 12$ *and* $n_{PC2} = 8$. *In all the case the number of valid cells is 24.*

Table 3.3 summarizes the grid dimensions of each one of the possible configurations and the percentage of scores included in the valid cells.

Table 3.3 *Dataset 3: summary of the three possible configurations identified by the grid search algorithm.*

Configuration	n_{PC1}	n_{PC2}	n_{valid}^{max}	Scores included in the valid cells	Figure
1	8	9	24	97.3 %	3.8a
2	9	9	24	96.0 %	3.8b
3	12	8	24	91.1 %	3.8c

Among the identified grids, the one that is chosen is configuration no. 1, as it has (among those with the highest number of valid cells) the highest percentage of scores included in the valid cells.

3.4 Chronological ordering and common trajectory construction

The identification of the valid cells does not give any information on how the nodes must be connected. Indeed, the common trajectory must be coherent with the time evolution of the batches. In order to order the points of the common trajectory, the sampling number of each score is used. The sampling number is a sequential number assigned to each sample. Indeed, if the fifth score of a batch is taken into account, the sampling number would be 5. The idea is to find the maximum sampling number of each batch in each valid cell. Then, consider the mean of the maximum sampling number in each cell and order the valid cell from the one that has the lowest sampling number to the one that contains the highest.

Being $\mathbf{s}_{k,u}$ the vector containing the sampling number of batch k in cell u, the maximum of this vector is found as:

$$s_{k,u}^{max} = \max\left(\mathbf{s}_{k,u}\right) \quad . \tag{3.9}$$

This value is calculated for all the batches in each valid cell. Subsequently the mean of the maximum sampling number in each valid cell is calculated as:

$$\bar{s}_u = \frac{1}{K}\left(\mathbf{s}_u^{max}\right) \quad , \tag{3.10}$$

where $\mathbf{s}_u^{max}$ is the vector containing all the $s_{k,u}^{max}$ for valid cell u and for all K batches. The corresponding valid cell is stored along with this value the position of the valid cell considered. The values obtained in equation (3.10.) are stored in the vector $\bar{\mathbf{s}}$ which contains n_{valid}^{max} elements. Its rows are then sorted with respect to $\bar{s}_u$ and the corresponding chronological order of the valid cell is found.

Figure 3.9 shows the chronological order of the nodes of the common trajectory of Dataset 1. It is easily shown that the ordering is correct because the first point corresponds to the start that has been seen in Figure 3.1., while the end coincides with the 14th point of the common

trajectory. Once the order is found, the nodes can be interpolated linearly to build the common trajectory.

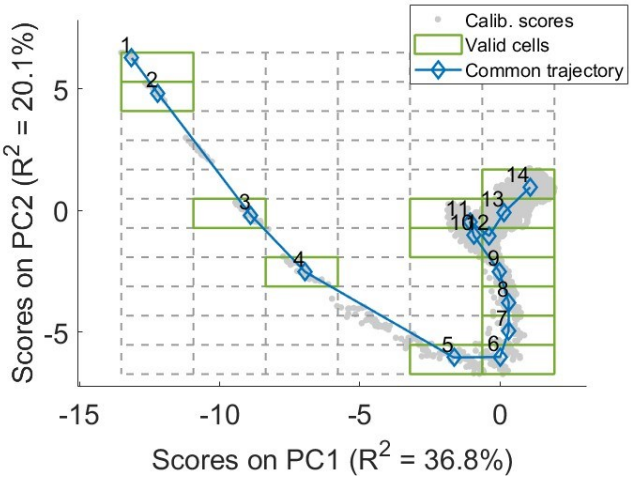


Figure 3.9 *Dataset 1: ordered common trajectory.*

Once the common trajectory has been built it is possible to proceed with the calibration by calculating the relative time of the batches and of the common trajectory.

3.5 Relative time estimation

Relative time (r_t) is a measurement of the progress of the physical and chemical phenomena of the process. Thanks to this marker, the assumption-free modelling is able to perform an internal alignment of the trajectories.

The relative time is evaluated for each sample of the calibration dataset and for each point of the common trajectory. In order to evaluate the relative time, it is necessary to subdivide the common trajectory into a high number of equally spaced points. In this thesis $n_{int} = 15000$ points will be used. The points sampled from the common trajectory are progressively numbered and the relative time is then defined as the ratio of the number of points before (i) the considered score over the total number of points into which the common trajectory has been divided into.

$$r_{t,i} = \frac{i}{n_{int}} * 100 \tag{3.11}$$

Estimating the relative time for the calibration batches is less straightforward. Indeed, the common trajectory must be divided into segments. A segment is that part of the common trajectory comprised between nodes. Therefore, the number of segments into which the common trajectory will be divided is equal to the number of valid cells minus one. For Dataset 1, the number of segments is 13.

In order to estimate the relative time of a score, its orthogonal projection onto the common trajectory is considered (Figure 3.10a). For the points that cannot be orthogonally projected, the closest node will be taken instead (Figure 3.10b). In the cases in which the point can be

orthogonally projected onto two different segments, the smaller distance is taken into account (Figure 3.10c).

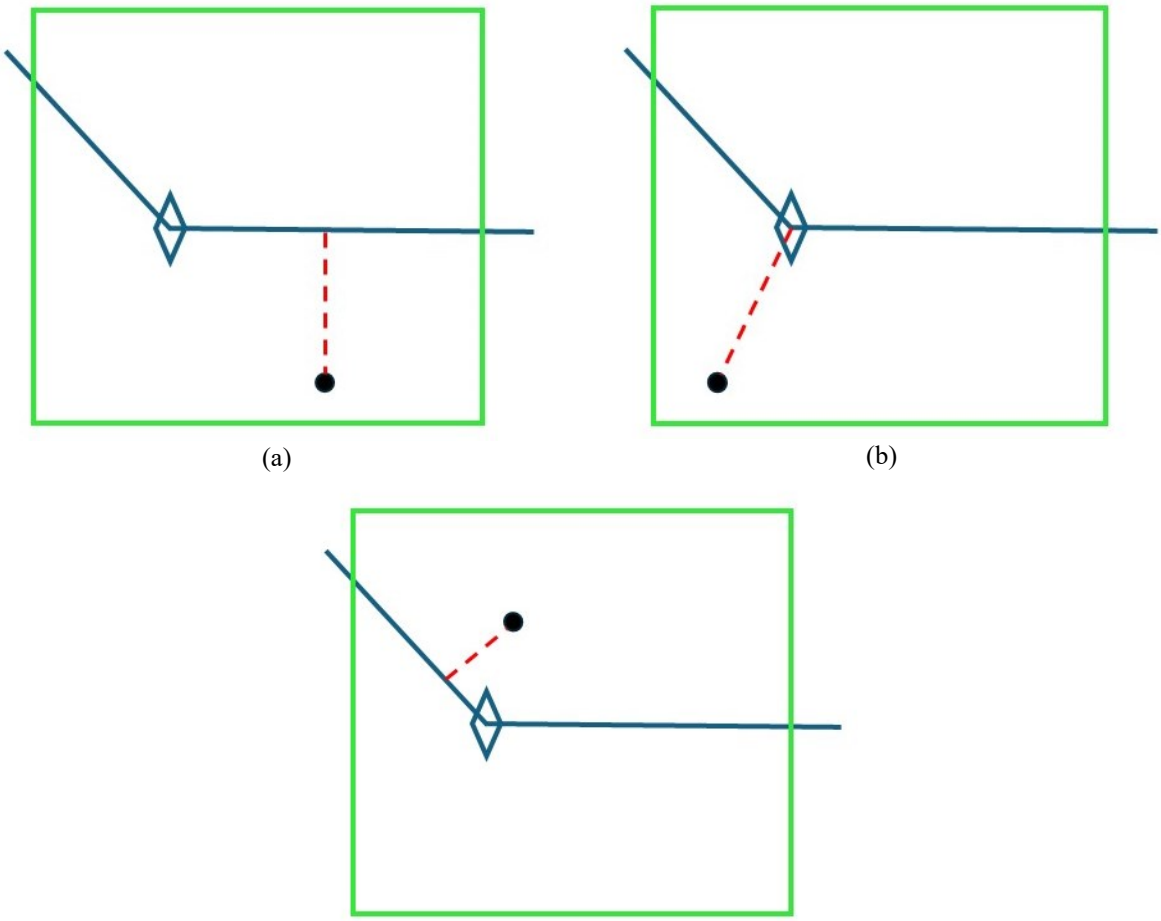
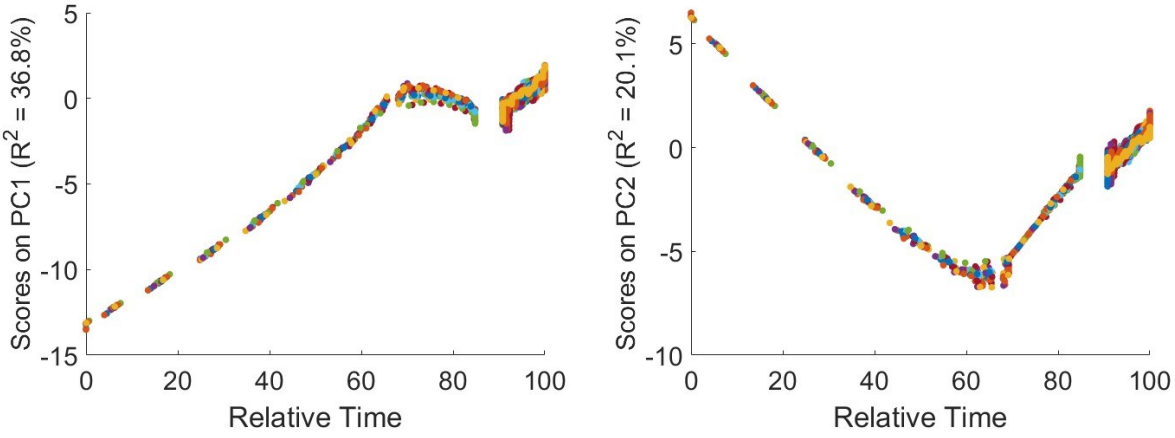


Figure 3.10 *Possible projections of a score onto the common trajectory. (a) The score can be orthogonally projected onto a single segment of the common trajectory. (b) The score cannot be orthogonally projected onto the common trajectory. (c) The score can be orthogonally projected onto two segments of the common trajectory.*

In order to assess which is the case, the classical problem of analytical geometry of the shortest distance between a point and a segment is exploited. Indeed, are evaluated the shortest distances between the point and the segments and, between the two the smaller is taken into account.
The projection is evaluated for all the points in the calibration batches and a relative time is assigned to each score. Figure 3.11 shows the evolution of the scores of the first PC (Figure 3.11a) and the one of the second PC (Figure 3.11b) with respect to the relative time for all the calibration batches of dataset 1.



(a) (b)

Figure 3.11 *Dataset 1: Relative time of the calibration batches of (a) PC1 and (b) PC2.*

It can be noticed that in some cases the relative time remains constant as the process progresses as can be seen from Figure 3.11 at $r_t = 90$. This occurs when the projection of the point has a lower relative time than the previously considered score. Therefore, in order to avoid situations in which the relative time decreases as the batch run continues, it has been decided to set the relative time equal to the one of the previous sample.

3.6 Control interval around the common trajectory

Once the common trajectory and the relative time associated to the nodes have been calculated, it is possible to build one of the two control charts that will be used to monitor the process. This control chart is used to understand the deviation of the batches from the common trajectory, therefore it exploits control limits built around it. In order to build, it the distances of the means of the batches (evaluated using (3.8)) from the common trajectory are used. Indeed, these points have been evaluated for each valid cell in §3.3.2 and, as suggested by Westad et al. (2015) their mean and standard deviation can be used to evaluate the control limits around the common trajectory. In the previous study of Sartori (2023) the control limits have been evaluated from the inverse of a normal distribution, however it was not verified if the data used to evaluate the limit were normally distributed. The following paragraphs will describe how the distances of the means of the batches are calculated and how the control limits are evaluated.

3.6.1 Distance from the common trajectory

In order to build the control limits around the common trajectory, Westad et al. (2015) suggest to use the standard deviation of the orthogonal distances between the means of the batches (evaluated using (3.8)) and the common trajectory. This leads to having a different value for each valid cell. However, no other details are given on how to construct those limits. Sartori (2023) hypothesized that the distances are normally distributed around their mean, therefore the limits can be evaluated from the inverse of this distribution. However, no proof was given that

the sample can be represented with such distribution. The method used in this Thesis to evaluate orthogonal distance of the mean of the batch evaluated by (3.8) from the common trajectory is similar to the one used in §3.5. Two cases can be highlighted to evaluate the distance as shown in Figure 3.12.

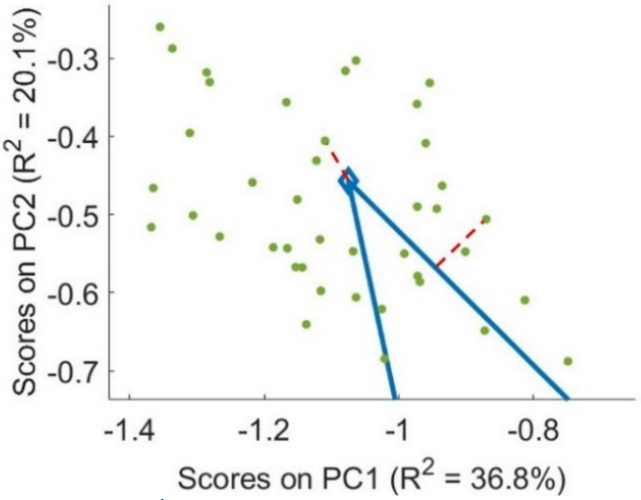


Figure 3.12 *Dataset 1: detail of the 11th node of the common trajectory and two possible projections of the means of the batches in that cell.*

If the mean can be orthogonally projected onto the common trajectory, the Euclidean distance between the considered point and the projection ($t_{a,u,\perp}$) is calculated according to:

$$d_{n,u,\perp} = \sqrt[2]{\sum_{a=1}^{2}\left(t_{a,u,\perp} - \bar{t}_{a,n,u}\right)^2} \quad , \tag{3.12}$$

if it is not the case, the distance between the considered point and the node of that cell is calculated:

$$d_{n,u} = \sqrt[2]{\sum_{a=1}^{2}\left(\bar{t}_{a,u} - \bar{t}_{a,n,u}\right)^2} \quad . \tag{3.13}$$

The number of distances evaluated for the valid cell *u* is equal to the number of batches which are present in that cell.

3.6.2 Distance distribution

Before building the control limits, it is necessary to assess whether the distances of the means of the batches inside a cell are normally distributed or not. The distances are always positive, and their typical distribution is shown in Figure 3.13.

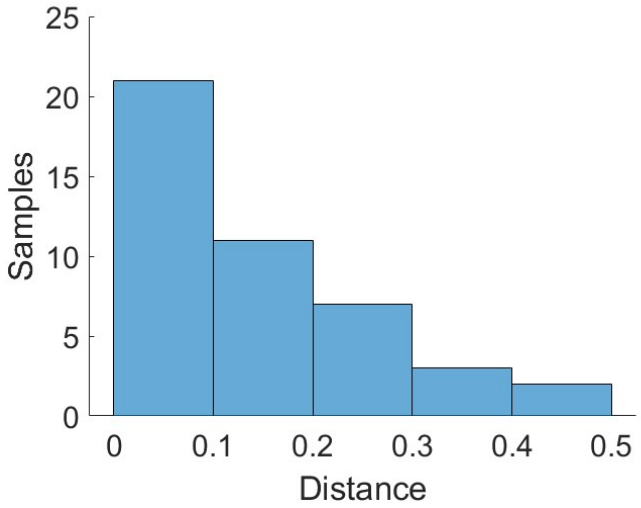


Figure 3.13 *Dataset 1: distribution of the distances of the mean of the calibration batches from the common trajectory in the 11th valid cell*

It is not possible to assess whether the sample is normally distributed with such shape. Therefore, a criterion to give a sign to the distances is needed in order to recreate the typical shape of a Gaussian distribution. In order to give a sign to the distances, the position of the points with respect to the common trajectory is considered. Indeed, a polygon is created closing the common trajectory by joining the first and the last node (Figure 3.14).

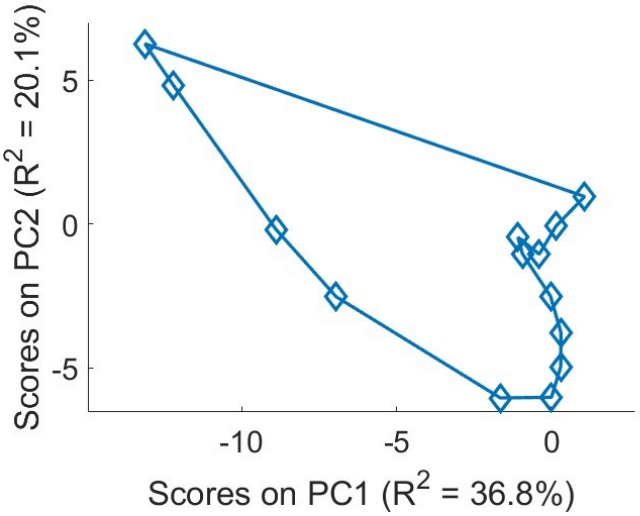


Figure 3.14 *Dataset 1: polygon created by joining the first and the last node of the common trajectory.*

Once the polygon is created the signs are given. If the points are inside the polygon the distance is considered positive, otherwise negative (Figure 3.15).

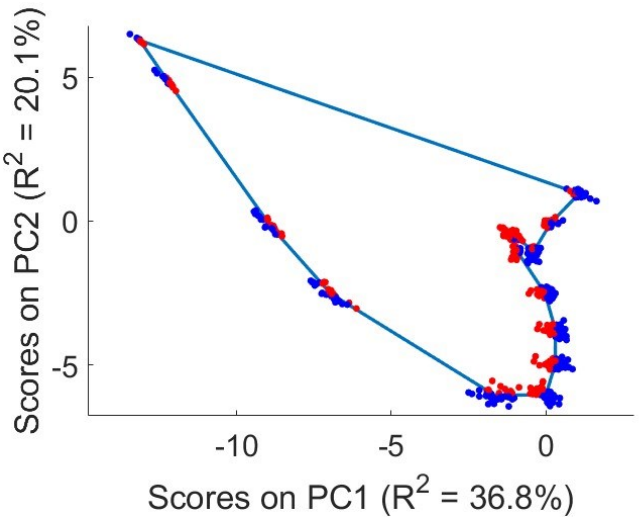


Figure 3.15 *Dataset 1: polygon used to give a sign to the distances depending on the position of the points. To the points outside the polygon (blue) a negative sign is given; to the points inside (red) a positive sign is given.*

After signs are attributed to each distance, the distribution also assumes negative values (Figure 3.16) and an Anderson-Darling normality test (Anderson and Darling 1954) with 95% significance level is applied to the sample of each valid cell.

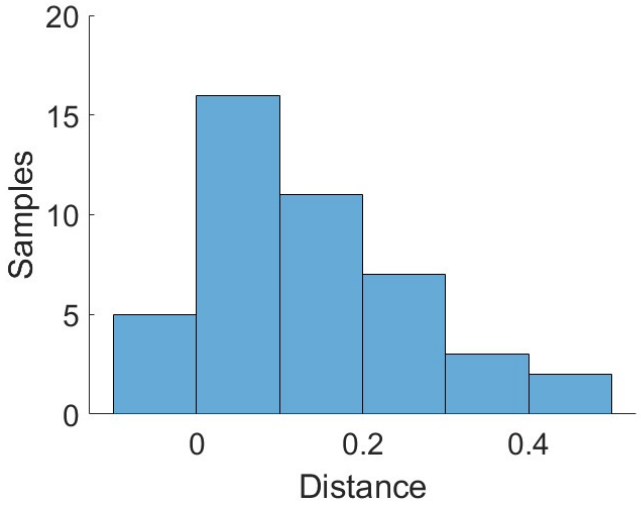


Figure 3.16 *Dataset 1: histogram of the distances of the 11th valid cell after giving a sign to each distance.*

The normality test on Dataset 1 indicates that the distances are normally distributed in 6 cells out of 14, which amounts to 42.9% of the total. This result may not seem satisfactory; however, in some cells where the test fails to reject the null hypothesis, the p-value is close to the limiting value (0.05), meaning that the sample is almost normally distributed. Moreover, in the first and in the last valid cell it is difficult to create a criterion that gives the sign to the distances. Therefore, it has been decided to not account for these cells when evaluating the percentage of normally distributed distances.

Table 3.4 summarizes the results obtained for all the dataset used in this thesis.

Table 3.4 *Summary of the distances distribution for all the datasets used.*

Dataset No.	Number of valid cells	Cells with a normal distribution	Percentage of valid cell with a normal distribution	Percentage of valid cell with a normal distribution (excluding first and last valid cell)
1	14	6	42.9 %	50 %
2	21	7	33.3 %	36.8 %
3	24	22	91.7 %	95.5 %
4	13	6	46.2 %	33.3 %
5	8	7	87.5 %	83.3 %

The following paragraph shows how the control intervals are actually evaluated.

3.6.3 Calculation of the control interval

From the obtained results, it has been decided to evaluate the confidence limits from the inverse of a normal distribution with mean μ_u and standard deviation σ_u of the valid cell u and confidence level α as:

$$d_u^{c.i.} = \mu_u + Z_\alpha \sigma_u \quad , \tag{3.14}$$

Where $d_u^{c.i.}$ is the distance of the control limit in the valid cell u from the common trajectory, and Z_α is the value of a standard normal distribution with α confidence level.

The evaluated distance is taken from the respective node of the common trajectory on both sides. Two possibilities arise:

1. For the first and for the last node, the distance is taken on the line perpendicular to the common trajectory (Figure 3.17a).
2. For all the other cells the distance is taken on the bisector of the angle formed by two subsequent segments of the common trajectory (Figure 3.17b).

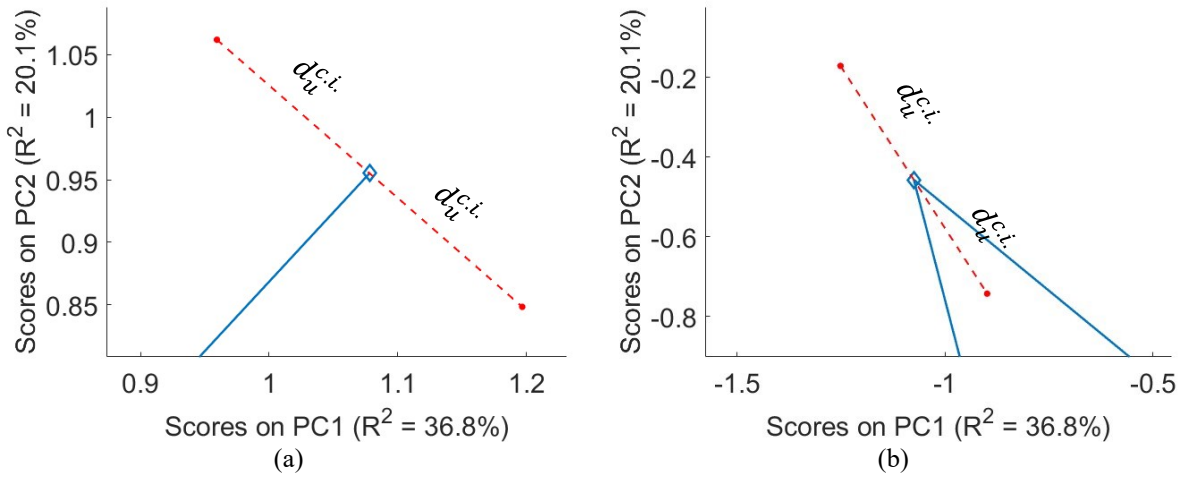


Figure 3.17 *Dataset 1: possible control limits (a) close- up of the last node, the distance of the control limit is taken on the line perpendicular to the segment of the common trajectory. (b) Close-up of the 11th node, the distance of the control limit is taken on the bisector of the angle between two segments of common trajectory*

In Figure 3.18 the control interval evaluated for Dataset 1 using a 95% confidence level are shown.

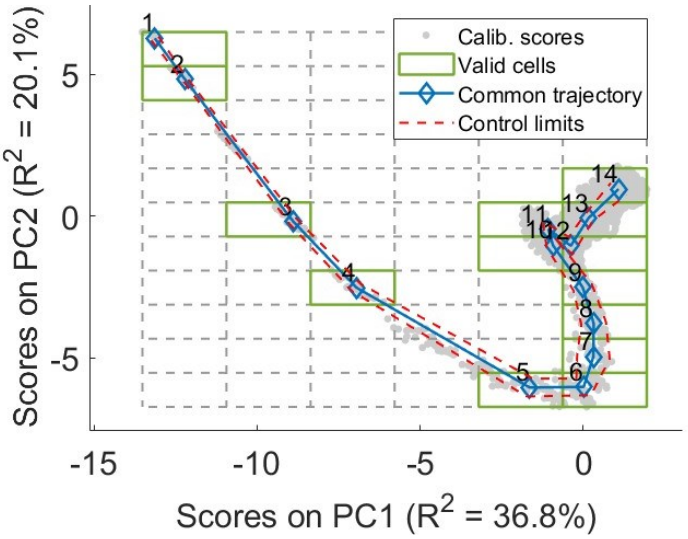


Figure 3.18 *Dataset 1: reconstructed common trajectory and control limits around it for the calibration batches.*

The limits are constructed on both sides of the common trajectory. These limits will be useful to establish if a batch can be considered NOC or faulty. The percentage of the means of the batches (points used to evaluate the control limit) out of the limits is 13.2%. This value is clearly higher than the 5% that it should be, however it is still acceptable considering that the distances are not normally distributed in all the cells. Moreover, as already mentioned, the method used to evaluate the sign of the distances does not capture well the position of the means of the batches in the first and in the last valid cell.

3.7 SPE control chart.

The second control chart is used monitor the behaviour of the *SPE.* This statistic, as explained in §1.1, is the squared sum of residuals. Residuals are usually normally distributed; however, when using a variable-wise unfolding PCA, the correlation structure between variables is kept constant for the whole duration of the process. This leads to residuals which are auto-correlated and therefore that are not normally distributed. In order to build the control chart on *SPE,* Sartori (2023) used an approach that is similar to the one described by Nomikos and MacGregor (1995). Indeed, the control limit was evaluated for each valid cell *u* as:

$$SPE_u^{lim} = \frac{{\sigma_u}^2}{2\mu_u} \chi^2_{\frac{2{\mu_u}^2}{{\sigma_u}^2},\alpha} \quad . \tag{3.15}$$

Where ${\sigma_u}^2$ and μ_u are the standard deviation and the mean of the *SPE* in the valid cell *u*, and α is the confidence level chosen. However, this approach requires that residuals are normally distributed, which is not the case when applying a variable-wise unfolding PCA. In this section first it is shown how the residuals are distributed; then a new approach to evaluate the limit on *SPE* is introduced.

3.7.1 Residuals distribution

The residuals **E** are evaluated from:

$$\mathbf{E} = \mathbf{X} - \widehat{\mathbf{X}} \quad , \tag{3.16}$$

being $\widehat{\mathbf{X}}$ the matrix of the reconstructed data. A column of matrix **E** contains all the residual of all the batches for variable *v*. In order to assess whether the v^{th} column of the matrix is normally distributed, an Anderson-Darling normality test can be performed. For Dataset 1, it resulted that only 20% of the variables have normally distributed residuals. However, while performing SPM with the assumption-free modelling, the focus is on the valid cells. Therefore, *v* Anderson-Darling tests are run, one for each column of the matrix containing the residuals of the scores inside the valid cell *u* ($\mathbf{E}_u$). The tests are repeated for each valid cell and the percentage of columns in which the residuals are normally distributed is reported in Table 3.5.

Table 3.5 *Summary of the results of the normality test on the columns of the residual matrix.*

Dataset No.	Number of variables	No. of valid cells	% of normally distributed residuals
1	10	14	45.0 %
2	10	21	22.9 %
3	11	24	35.8 %
4	20	13	42.7 %
5	10	8	7.5 %

In most cases the residuals are not normally distributed, therefore the limits cannot be evaluated from a χ^2 distribution.

3.7.2 SPE limit evaluation

For the assumption-free modelling, a limit of the *SPE* can be evaluated for each valid cell at the relative time of the node of the considered cell. However, it is not possible to evaluate it from the inverse of a weighted χ^2 distribution of the *SPE* contained in the valid cell *u* ($\mathbf{SPE}_u$) like in batch-wise unfolding PCA. The approach that is used relies on the construction of a non-parametric cumulative density function (CDF) of the square prediction error in each valid cell. The steps to evaluate the limit on *SPE* in a valid cell (SPE_u^{lim}) are the following:

1. Obtain the matrix of the residuals in the valid cell *u* $\mathbf{E}_u$.
2. Calculate the *SPE* for all the scores contained in the valid cell using (1.7)
3. Build the non-parametric CDF of $\mathbf{SPE}_u$.
4. Calculate the SPE_u^{lim} form the CDF with 95% confidence level.

These steps must be repeated for all the valid cells in order to build the *SPE* control chart.
This approach has the advantage that no assumptions are made on the distribution of the sample used to build the CDF. Therefore, building the control limits from a non-parametric CDF allows to overcome the cases in which residuals are not normally distributed.

Figure 3.19 shows the non-parametric CDF built for valid cell No. 11 of dataset 1 together with the control limit for that cell and the empirical CDF evaluated from $\mathbf{SPE}_u$.

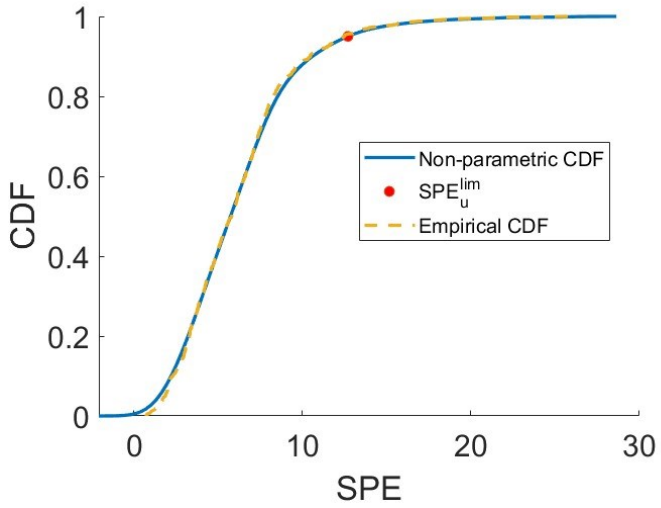


Figure 3.19 *Dataset 1: non-parametric CDF and empirical CDF of the SPE of the 11th valid cell with the relative control limit*

The empirical CDF has been evaluated to assess whether the non-parametric CDF was able to represent the population of the SPE_u or not. It can be noticed that the non-parametric function approximates the sample very well, therefore the used method is reliable.
Once the limit for each valid cell is evaluated, the control chart can be built (Figure 3.20) by plotting each limit with respect to the corresponding relative time of node.

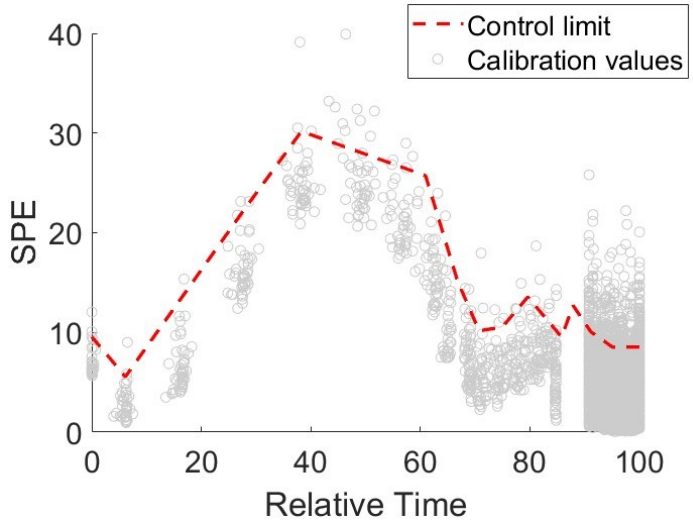


Figure 3.20 *Dataset 1 SPE control chart with the SPE of the calibration scores.*

Moreover Figure 3.20 shows the *SPE* for all the calibration values. The percentage of *SPE* out of the control limit is 5.2% which is coherent with the choice of 95% confidence level.
In order to assess the difference between this approach and the one used by Sartori (2023), the two control charts are compared in Figure 3.21.

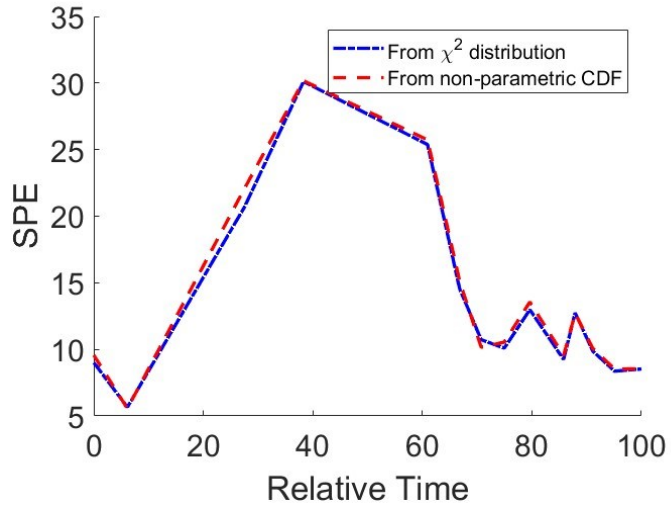


Figure 3.21 *Comparison of the control limit on SPE evaluated from the non-parametric CDF and the approach used by Sartori (2023).*

The two control charts are basically the same, however the approach that evaluates the limit from the inverse of a χ^2 distribution is theoretically wrong because it relies on the assumption that the residuals are normally distributed. Therefore, the control charts that will be used to monitor the process is the one evaluated from the non-parametric CDF.
The differences between the control charts for the other datasets are reported in Appendix 1.

3.8 Alarm calibration

In order to perform SPM, there is the need to evaluate a value of consecutive points out of the control limits that discriminate a normal batch from a faulty one. The approach that is used consists in counting the maximum number of consecutive points out of the control limit for each batch of the calibration set and choosing the highest value among them. The idea behind this approach is that all the batches in the calibration set are NOC batches, therefore a number of points equal or lower than the one found among them is still acceptable. This task is performed for both control charts.

3.8.1 Choice of C_D^{max}

C_D^{max} is the highest number of consecutive points out of the confidence limit of the common trajectory among the calibration batches. In order to count the number of consecutive points for each batch, an approach similar to the one used in §.3.6.2. has been exploited.
Indeed, a polygon has been created by joining the ends of the confidence interval to assess how many consecutive points for each calibration batch were out of the confidence limit.
Figure 3.22 shows the polygon used to discriminate if a point is inside or outside the control limit together with one of the calibration batches.

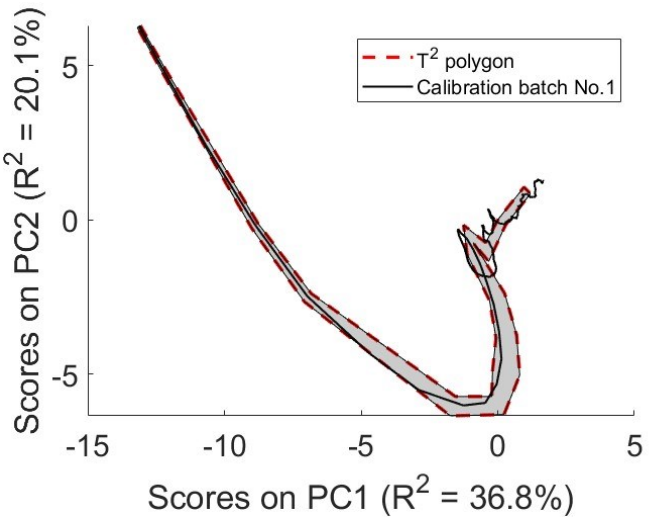


Figure 3.22 *Dataset 1: polygon used to evaluate C_D^{max} and calibration batch No.1.*

In the case reported in Figure 3.22, the maximum number of consecutive points out of the control limit $C_{D,n} = 22$. The scores are not considered if their relative time is 0 or 100 because this means that the projection of the score is not accurate, as coincides with the first or with the last node.

Among all the values found for each batch, C_D^{max} is the maximum (Figure 3.23).

$$C_D^{max} = \max(\mathbf{C}_D) \quad , \tag{3.17}$$

where $\mathbf{C}_D$ is the vector where all the $C_{D,n}$ are stored.

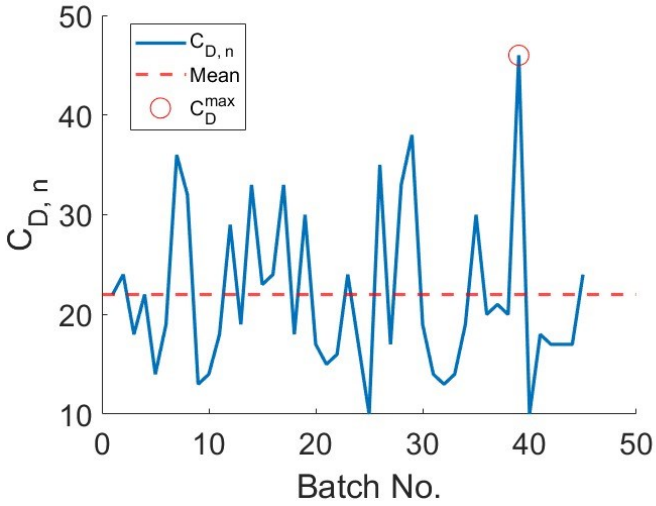


Figure 3.23 *Dataset 1: $C_{D,n}$ for all the batches in the calibration set.*

C_D^{max} in Dataset 1 is 46 and is found in batch No. 39. Therefore, in the monitoring of a new batch an alarm will be triggered when more than 46 consecutive score will be outside the confidence limits around the common trajectory.

3.8.2 Choice of C_{SPE}^{max}

C_{SPE}^{max} is the maximum number of consecutive points out of the confidence limit on *SPE* that are found in the calibration set. The approach is identical to the one used in §3.8.1 for the evaluation of C_D^{max}. Indeed, a polygon is created by closing the control limit on *SPE* and for each batch the maximum number of consecutive points out of the control limit $(C_{SPE,n})$ is found. Figure 3.24 shows the used polygon together with one of the batches in the calibration set.

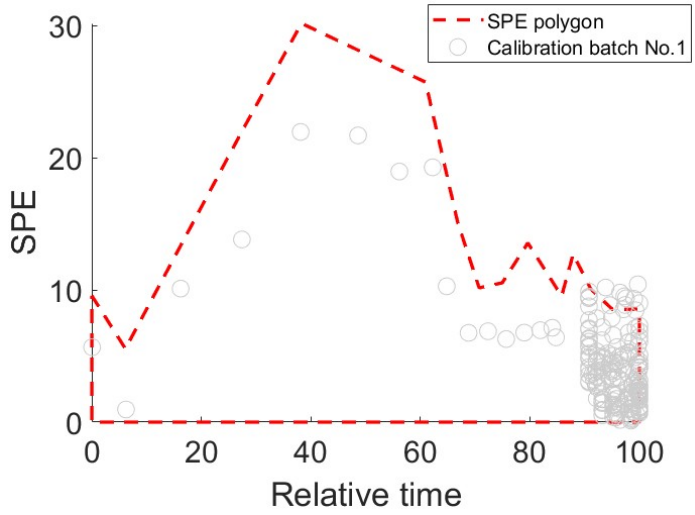


Figure 3.24 *Dataset 1: polygon used to evaluate cSPE^{max} and the SPE for calibration batch No.1.*

In the case reported in Figure 3.24, $C_{SPE,n} = 4$. The *SPE* are not considered if their relative time is 0 or 100 because this means that the projection of the score on the common trajectory is not accurate because it coincides with the first or with the last node.

Among all the $C_{SPE,n}$, the value used to trigger an alarm is the maximum (Figure 3.25).

$$C_{SPE}^{max} = \max(\mathbf{C}_{SPE}) \quad . \tag{3.18}$$

Where $\mathbf{C}_{SPE,n}$ is the vector containing all the $C_{SPE,n}$.

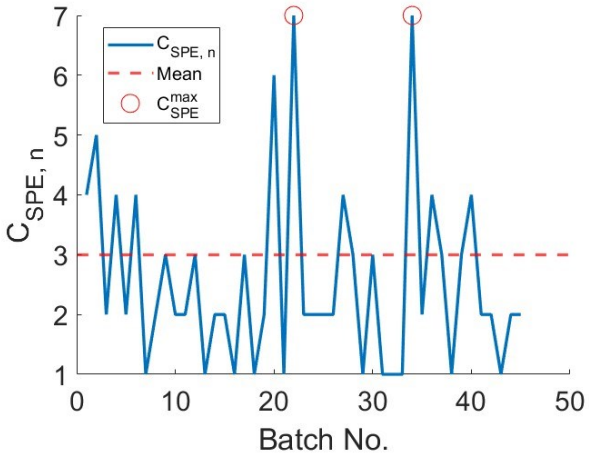


Figure 3.25 *Dataset 1: $C_{SPE,n}$ for all the batches in the calibration set*

C_{SPE}^{max} in Dataset 1 is 7 and is found in batches No. 22 and No. 34. Therefore, in the monitoring of a new batch, an alarm will be triggered when more than 7 consecutive *SPE* are out of the control limit.

3.9 Process monitoring using the assumption-free model

In order to perform SPM, the validation set, which contains both NOC and faulty batches, is used. Indeed, the presence of both types of batches is useful to assess the monitoring performances in terms of detection strength and detection speed. To assess whether a batch is faulty or not, the control charts developed in §3.6 and §3.7 are exploited.

3.9.1 Monitoring scheme

In order to monitor the behaviour of a new batch, it is necessary to obtain the scores for that batch. Therefore, when a new sample $\mathbf{x}_{new}$ is available, it is autoscaled and the scores are evaluated as:

$$\mathbf{t}_{new} = \mathbf{x}_{new}\mathbf{P}^{\mathbf{T}} \quad . \tag{3.19}$$

Once the score are obtained, it is possible to project them onto the common trajectory in order to evaluate its relative time. The estimation of the relative time is equal to that one done for the calibration scores as explained in §3.5.

The residuals for the new sample are estimated by subtracting from the sample the reconstructed value $\hat{\boldsymbol{x}}_{new}$:

$$\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new} \quad . \tag{3.20}$$

From the residuals, the *SPE* is evaluated for the new sample:

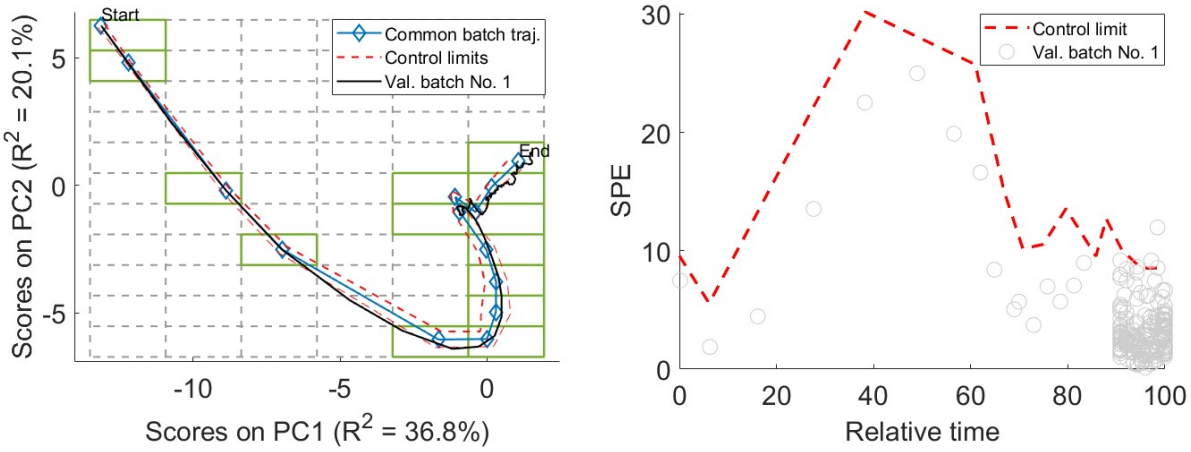
$$SPE_{new} = \mathbf{e}_{new}\mathbf{e}_{new}^{\mathrm{T}} \quad . \tag{3.21}$$

SPE_{new} is projected onto the control chart of the *SPE* with respect to the relative time of the sample.

Both $\mathbf{t}_{new}$ and SPE_{new} are compared to their control limit respectively evaluated in §3.6.3 and §3.7.2. The points can be either inside or outside the control limit and the position is identified using the same polygons described in §3.8.1 and §3.8.2 created for the calibration set. If the point is within the polygon, no action is taken and the following sample is examined.

If not, a counter is increased by one on the statistic whose limit has been violated. The counter is $C_{D,n}$ for the limits around the common trajectory and is $C_{SPE,n}$ for the limit on *SPE*. It is possible that both points are located outside the control limits, therefore both counters will be increased by one. Once the following sample is obtained, the procedure is repeated. If the point is inside a control limit, the corresponding counter is reset to zero. The monitoring continues until the batch is completed or until one of the counters overcomes the relevant limit, C_D^{max} and C_{SPE}^{max} which were evaluated in §3.8.1 and §3.8.2 If the second condition occurs, the batch is classified as faulty; otherwise, it is classified as normal.

Figure 3.26 shows the control charts on the score plot (Figure 3.26a) and on the *SPE* (Figure 3.26b) for Batch No. 1 of the validation set of Dataset 1.

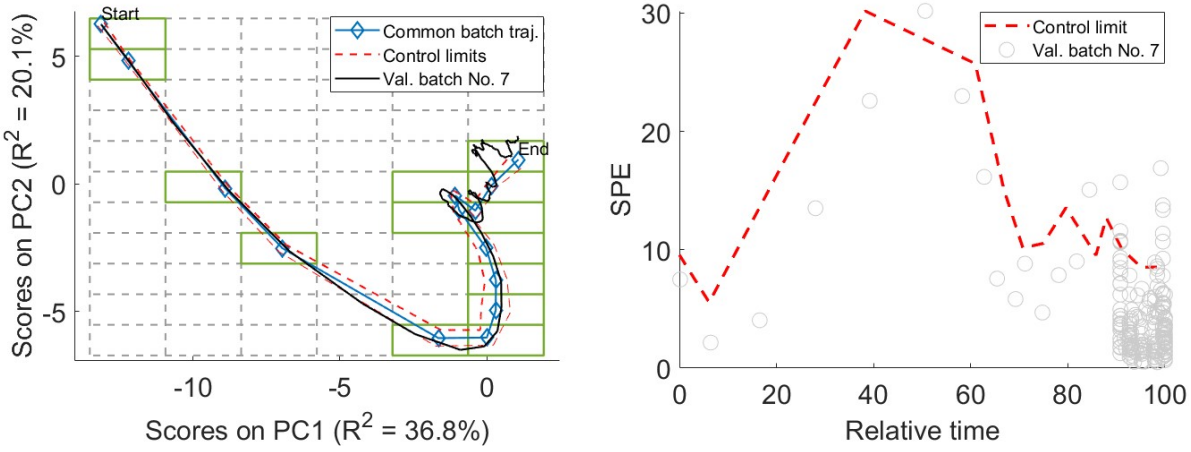


(a) (b)

Figure 3.26 *Dataset 1: monitoring of validation Batch No. 1. Control charts (a) on the score plot and (b) on the SPE*

Validation batch No. 1 is a normal batch. From the model resulted that $C_{D,n} = 27$ and $C_{SPE,n} = 2$, which are both lower than the respective limit. Therefore, the batch is correctly classified as NOC.

On the other hand, Figure 3.27 shows the control charts for Batch No. 7 of the validation set of Dataset 1, which is a faulty one.



(a) (b)

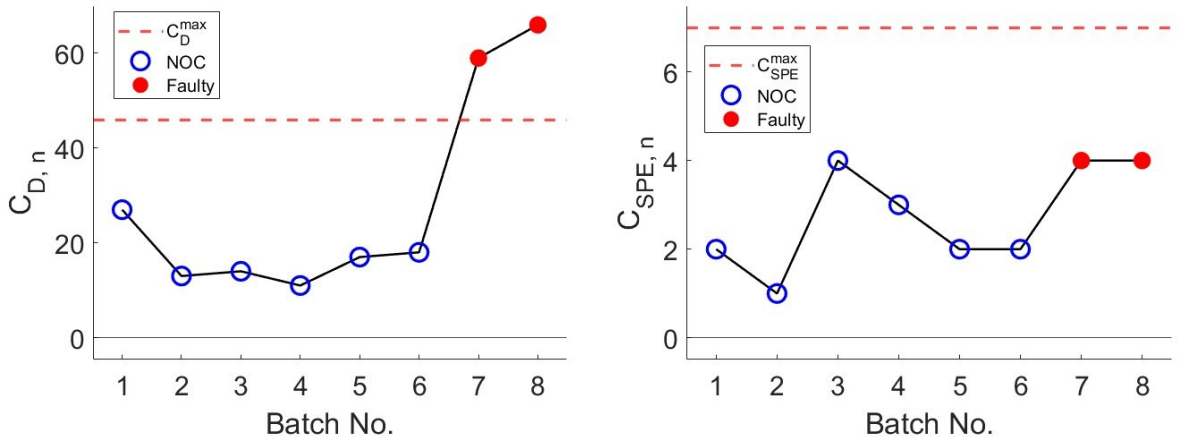
Figure 3.27 *Dataset 1: monitoring of validation Batch No. 7. Control charts (a) on the score plot and (b) on the SPE*

From the monitoring it resulted that: $C_{D,n} = 59$ and $C_{SPE,n} = 4$ and.

Although the *SPE* does not show any abnormal behaviour, from the score plot it can be noticed that the trajectory deviates from the common trajectory towards the conclusion of the batch run. Indeed, the maximum number of consecutive points is higher than C_D^{max}, therefore the batch is correctly classified as faulty.

The monitoring is repeated for all the batches in the validation set, with the model assessing whether that batch is faulty or not.

The results obtained for the monitoring of Dataset 1 are reported in Figure 3.28.



(a) (b)

Figure 3.28 *Dataset 1: monitoring of the validation set. Maximum number of consecutive points outside the control limit on (a) the score plot and on (b) the SPE The truly faulty batches are indicated in red.*

The model is capable of correctly identifying all the faulty batches due to the control limits around the common trajectory. Indeed, from the *SPE* no faulty batches have been identified.

3.9.2 Monitoring performance indicators

Once all the batches have been analysed, the monitoring performances are measured in terms of detection strength and detection speed. To determine the detection strength of the model, the true positive rate (*TPR*) and the false positive rate (*FPR*) have been used. The formulation of these parameters is described by Rato et al. (2016) as:

$$TPR = \frac{TP}{TP+FN} \quad , \tag{3.22}$$

$$FPR = \frac{FP}{FP+TN} \quad , \tag{3.23}$$

where *TP* is the number of faulty batches identified as faulty; *TN* is the number of NOC batches identified as NOC; *FP* is the number of NOC batches identified as faulty and *FN* is the number of faulty batches identified as NOC. Therefore, *TPR* is a measure of how good the model is in identifying the faulty batches, while *FPR* is a measure of how many NOC batches are misclassified.

In order to evaluate the detection speed, the average run length (*ARL*) described by Rato et al. (2016) is used. This metric measures the speed in signalling an abnormality after it occurs.

The performances of the monitoring of the Dataset 1 are reported in Table 3.6.

Table 3.6 *Dataset 1: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	100 %	% of batches
FPR	0	% of batches
ARL	81	Samples

The model built have a great ability in identifying the faulty batches and distinguish them from the NOC one. Indeed, no misclassification have been done.

Chapter 4

Results

In this Chapter the results of the calibration and of the performance assessment of the assumption-free modelling are presented. All the datasets described in Chapter 2 will be used. Moreover, the performances of the model are compared to the results obtained by Sartori (2023) after monitoring the same datasets with a model built by batch-wise unfolding the 3D data array.

4.1 Dataset 1

This dataset contains the data of the simulated SBR polymerization reaction. It was presented by Nomikos and MacGregor (1994) and as described in §2.1. Ten variables are measured, the batches are aligned and each one has 200 samples. 45 NOC batches are used to calibrate the MPCA model, 8 batches are used to validate the model.

4.1.1 Dataset 1: assumption-free modelling calibration

The matrix containing the data is variable-wise unfolded and autoscaled, then a PCA model is built using 2 PCs. Table 4.1 shows the results of the PCA model.

Table 4.1 *Dataset 1: Summary of the PCA model.*

PC No.	λ_a	R^2	$R^2_{cum.}$	*RMSECV*
1	3.68	36.8 %	36.8 %	0.89
2	2	20.1 %	56.9 %	0.75
3	1.23	12.3	69.2 %	0.68
4	1	10	79.2	1.50
5	0.99	9.9	89.1	12.98

The total captured variance is 56.9 %, which means that the model describes more than half of the correlation among the data. As already mentioned in §3.2, the choice of the number of PCs is not related to *RMSECV*, whose minimum is found for 3 PCs, but it is done for the grid search algorithm. The sum of the eigenvalues (λ_a) associated to each PCs is a measure of the variables that are captured by the model, in this case are almost six.

The scores obtained from the PCA are plotted (Figure 4.1) and the score plot confirms what was already visible from the variables' plot (Figure 2.1). Indeed, most of the scores are concentrated near the origin of the plane because most of the process is used to complete the conversion of the reactants, and the variables assume an almost stationary value.

Moreover, the process has a strong nonlinear behaviour as the scores are not evenly distributed in the score space.

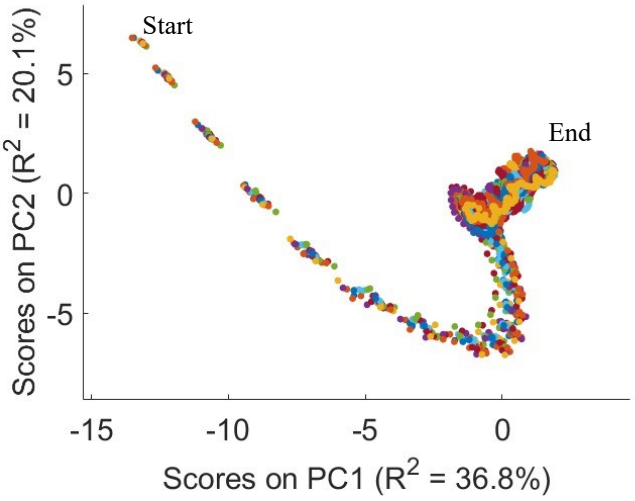


Figure 4.1 *Dataset 1: score plot obtained after variable-wise unfolding the calibration set of Dataset 1 described in §2.1. Each point represents a time instant of a single batch. Each colour represents a batch of the calibration set.*

In order to calibrate the assumption-free model, the maximum number of cells used to partition the score plot is set equal to 12 for both PCs. γ is set equal to 0.95, meaning that only the grids whose valid cells contain more than 95% of the total scores will be considered. β is chosen to be 0.9, meaning that a cell is considered valid only if 90% of the batches of the calibration set are present in that cell. The hyperparameters used to calibrate the assumption-free model are summarized in Table 4.2.

Table 4.2 *Dataset 1: Hyperparameter values used during the calibration of the assumption-free model.*

Hyperparameter	Value	Units
γ	0.95	Fraction of scores
β	0.90	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

The best grid found by the algorithm has $n_{PC1} = 6$ and $n_{PC2} = 11$ with 14 valid cells and 98.6% of all the scores included in the valid cells (Figure 4.2)

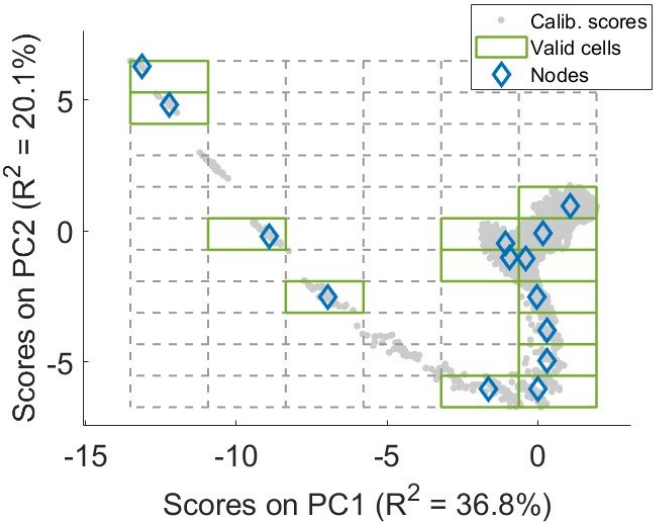


Figure 4.2 *Dataset 1: best grid found by the algorithm for the calibration batches. ($n_{PC1} = 6$ and $n_{PC2} = 11$).*

This configuration is the only one with 14 valid cells, therefore no other valid grids were taken into account for the choice of the best grid. The points of the trajectory are ordered as explained in §3.4 and the nodes are interpolated to build the common trajectory.
Figure 4.3 shows the obtained common trajectory.

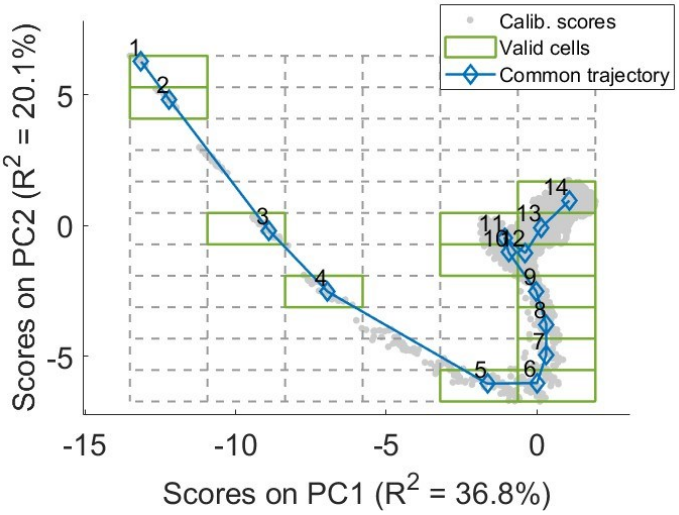


Figure 4.3 *Dataset 1: ordered common trajectory.*

The trajectory is able to satisfactorily reconstruct the average batch run. However, due to the partition of the score plot, the model does not include two clusters of points. The first one is between the 2nd and 3rd node, nonetheless the common trajectory passes through this cluster after the interpolation. On the other hand, the second cluster (between the 4th and the 5th node) is not represented by the common trajectory which passes at its side.
The relative time is estimated for all the batches in the calibration set and shown in Figure 4.4.

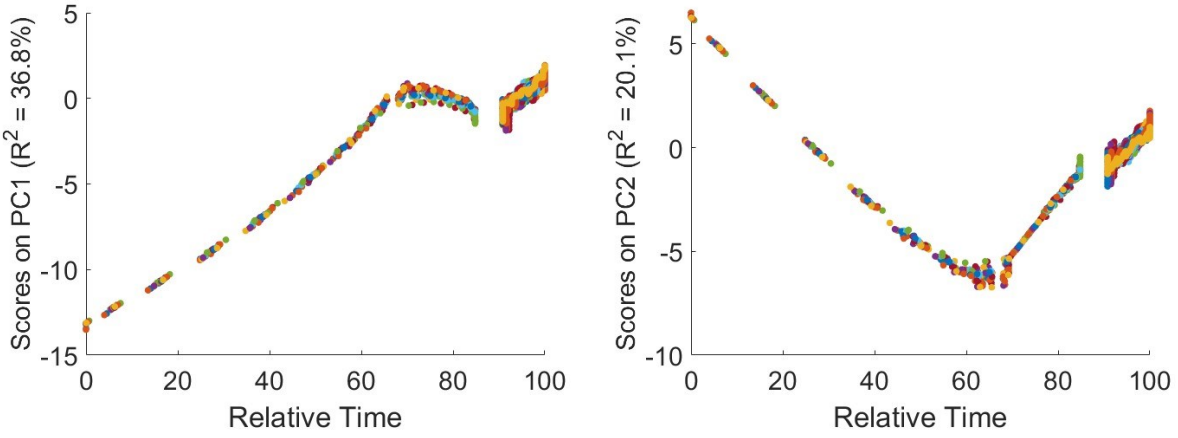


Figure 4.4 *Dataset 1: Relative time of the calibration batches of (a) PC1 and (b) PC2.*

The relative time allows to internally align events, indeed the scores have the same behaviour on all the batches when plotted against relative time. As already mentioned in §3.6 the relative time is kept constant when a subsequent score has a projection on the common trajectory which comes before the preceding score as can be noticed at $r_t = 90$.

The control limits around the common trajectory are evaluated according to §3.6, using a 95% confidence level. The resulting distances of the means of the batches are normally distributed in 50% of the valid cells after excluding the first and the last cells. The limits are shown in Figure 4.5.

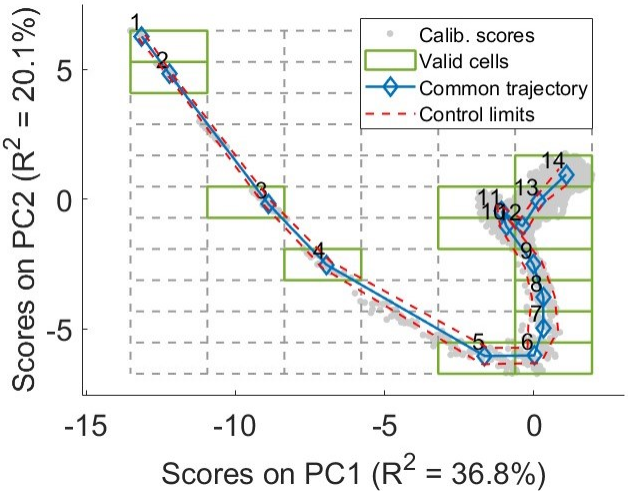


Figure 4.5 *Dataset 1: reconstructed common trajectory and control limits around it for the calibration batches.*

The control limits are narrower at the beginning of the batch due to the reduced variability, while they become broader towards the end of the process because there is more variability.

The percentage of the means of the batches out of the confidence limits is 13.2%. This value is clearly greater than the 5% that it should be, however it is still acceptable considering that the distances are not normally distributed in all the cells. Moreover, as already mentioned, the method used to evaluate the sign of the distances does not capture the position of the means of the batches well in the first and in the last valid cell. The *SPE* control chart (Figure 4.6) is built

by taking into account the distribution of the *SPE* in each valid cell as explained in §3.7, considering a 95 % confidence level.

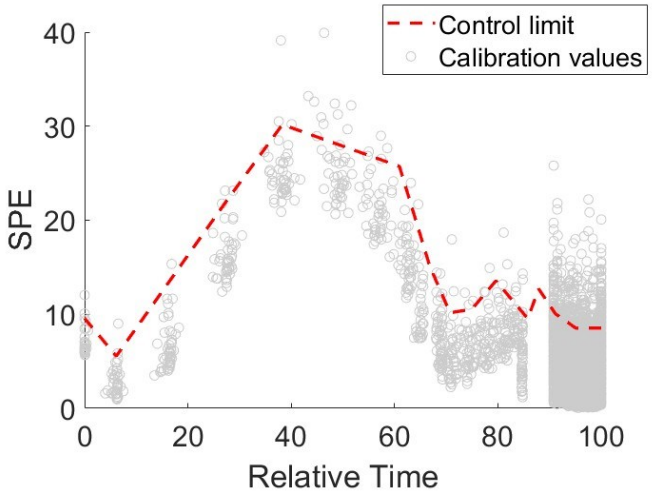


Figure 4.6 *Dataset 1: SPE control chart with the SPE of the calibration scores.*

The amount of *SPE* out of the confidence limit is 5.2%, which is coherent with the choice of 95% confidence level.

The calibration of the alarms is performed as reported in §3.8. The maximum number of consecutive points out of the confidence limit of the common trajectory is 46 and is found in the 39th batch of the calibration set. On the other hand, the maximum number out of the confidence limit of *SPE* is 7 and is found in the 22nd and 34th batch of the calibration set. Figure 4.7 shows the maximum number of consecutive points out of the confidence limit on both statistics for each batch.

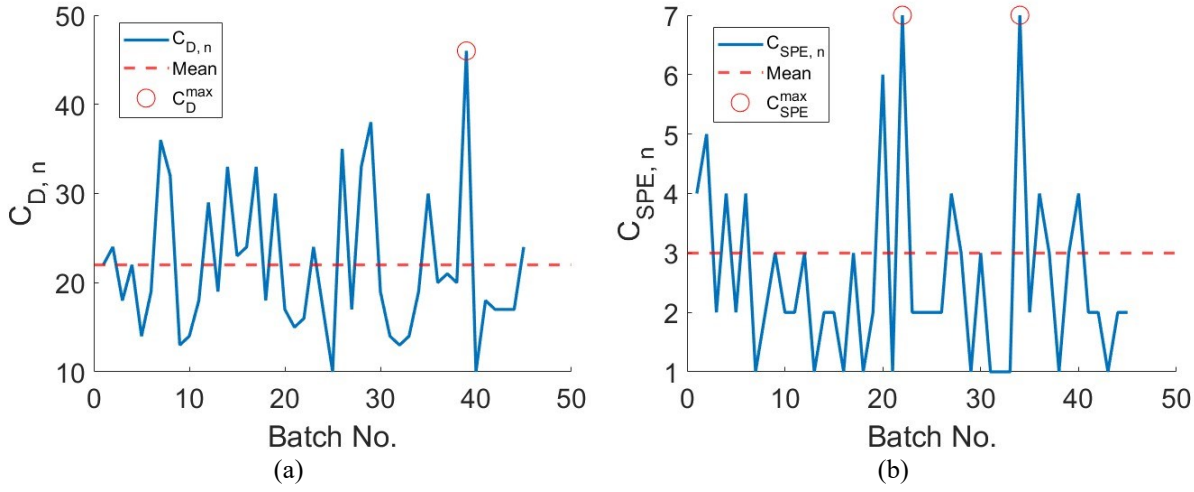


Figure 4.7 *Dataset 1: calibration set. Maximum number of consecutive points out of (a) the control limits around the common trajectory and (b) the control limit of the SPE*

It can be noticed that $C_{D,n}$ is affected by stronger variability than its counterpart of the *SPE*. Indeed $C_D^{max} = 46$, meaning that a batch which is out of the control limit for a quarter of its duration is still considerable of acceptable quality, while only a few consecutive points out of the control limit on the *SPE* are enough to classify a batch as faulty.

4.1.2 Dataset 1: monitoring using the assumption-free model

The validation set is used to assess the performance of the model in terms of detection strength and detection speed.

The monitoring is carried out as shown in §3.9, therefore the relative time, the scores and the *SPE* of each sample are estimated and compared to the limits. The 8 batches in the validation set are all correctly identified. Indeed, no false alarm rose during the monitoring. The 2 faulty batches have been identified due to the control limits around the common trajectory. From the control chart of the *SPE* no faulty batches have been found as each one had a $C_{SPE,n}$ lower than C_{SPE}^{max} . The results are reported in Figure 4.8.

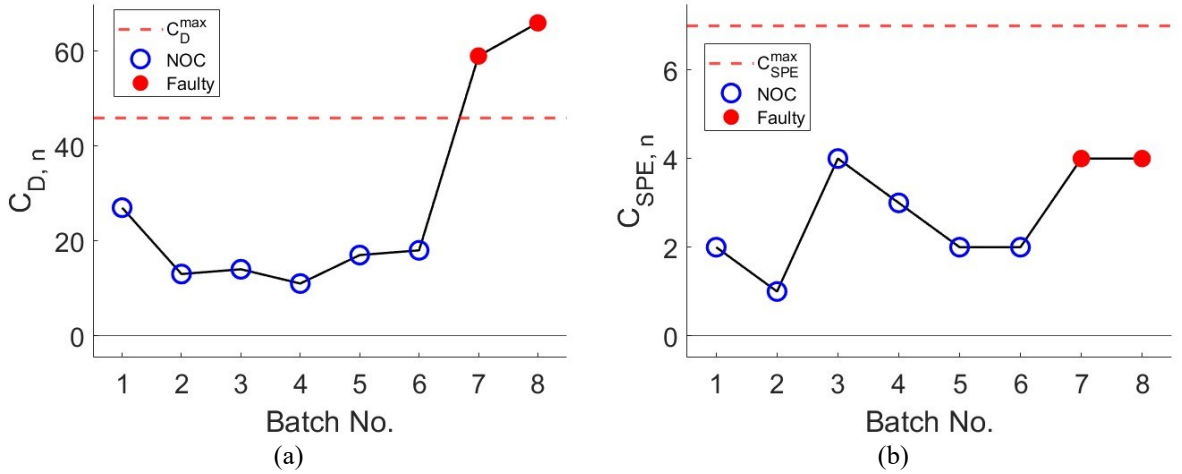


Figure 4.8 *Dataset 1: monitoring of the validation set. Maximum number of consecutive points outside the control limit on (a) the score plot and on (b) the SPE The truly faulty batches are indicated in red.*

As both NOC and faulty validation batches have been correctly identified, the developed model has a $TPR = 100\ \%$ and an $FPR = 0\ \%$.

The scores of both faulty batches have been projected onto the score plot together with the common trajectory to assess their deviation (Figure 4.9).

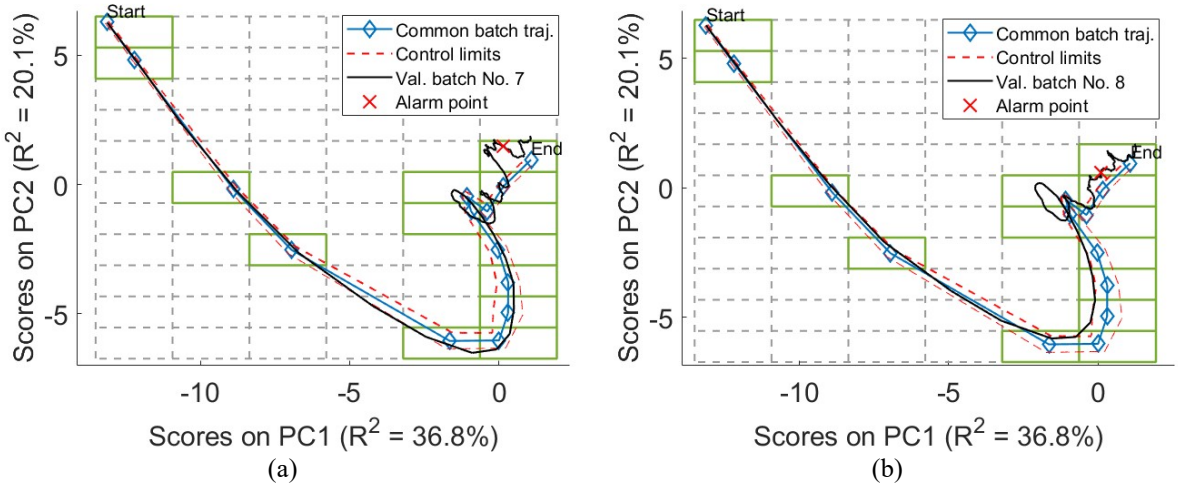


Figure 4.9 *Dataset 1: Faulty batches of the validation set on the control chart on the score plot. (a) validation batch No. 7 and (b) validation batch No. 8.*

As reported in §2.1, the two batches had the same fault, but with different magnitudes and at different times. Indeed, the first one was upset since the beginning, while the second one was upset halfway through the process. No evidence of the magnitude nor of the time of the faults is evident from the score plot of the faulty batches. Indeed, the sample that triggered and alarm in the validation batch No. 7 is the 148th, while the one of the validation batch No. 8 is the 114th. However, a prompter detection of the fault occurred in the second case. The *ARL* measured for this dataset is 81 samples.

Table 4.3 summarizes the monitoring performances of the assumption-free modelling applied to Dataset 1.

Table 4.3 *Dataset 1: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	100 %	% of batches
FPR	0	% of batches
ARL	81	Samples

Therefore, the built model is perfectly capable of distinguishing between NOC and faulty batches.

4.1.3 Dataset 1: monitoring using a standard MPCA method

The data are already aligned therefore can be directly unfolded in the variable direction and then autoscaled. A PCA model is built on the unfolded matrix. The number of PCs chosen by Sartori (2023) is 3, the total explained variability by the model is 30.6%. The scores are plotted together with the 95% confidence limit (Figure 4.10).

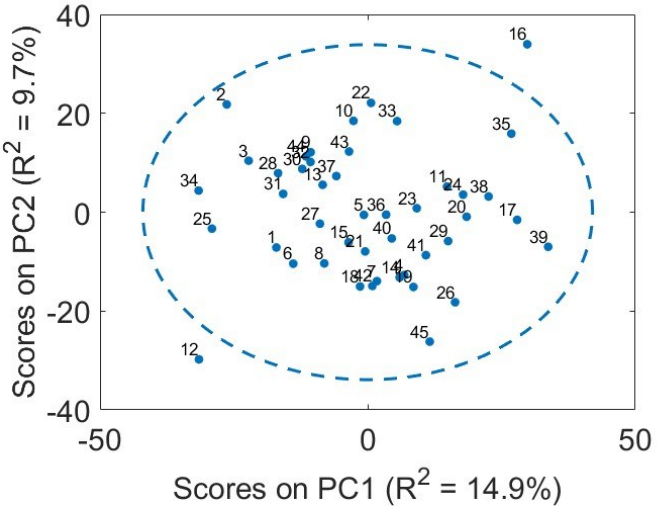


Figure 4.10 *Dataset 1: score plot obtained for a standard MPCA model. Each point represents a batch, the dotted line is the 95% confidence interval on the multivariate distribution of the scores.*

The scores result to be multi-normally distributed. Calibration batches No. 12 and 16 are outside the confidence limit, however, this is a reasonable number considering that is 4.5% of the total.

The control limit on the *SPE* is evaluated considering a 95% confidence level as shown in §1.2.1. The obtained control chart is shown in Figure 4.11.

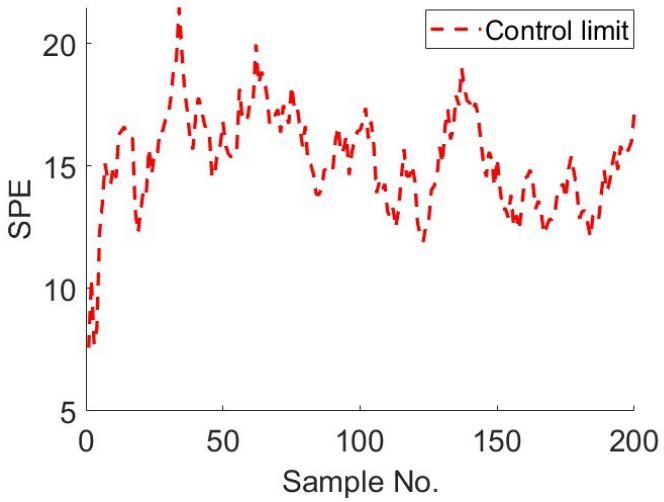


Figure 4.11 *Dataset 1: SPE control chart built on a batch-wise unfolding MPCA obtained using (1.13)*

The control limit respects the chosen confidence limit as the percentage of points out of it is 5.3%. The monitoring is performed according to section §1.2.1 and the maximum number of consecutive points out con the control limit chosen by Sartori (2023) is 2 for T^2 and 3 for the *SPE*.

The results of the monitoring are reported in Table 4.4.

Table 4.4 *Dataset 1: performance indicators of the monitoring scheme.*

Performance indicator	Value	Units
TPR	100	% of batches
FPR	16.7	% of batches
ARL	85	Samples

The results of the monitoring showed that the model is able to correctly identify all the faulty batches, indeed $TPR = 100\%$, however one normal batch has been misclassified and the $FPR = 16.7\%$. The faulty batches have been identified on average after 29 samples which translates to 145 min.

4.1.4 Dataset 1: comparison of the results

The monitoring has been carried out on the same dataset with both methodologies: the assumption-free modelling and a standard one. The results obtained with both methods are reported in Table 4.5.

Table 4.5 *Dataset 1: comparison of the performance indicators of the monitoring for both methods used*

Performance indicator	Method	Value	Units
TPR	Assumption-free model	100	% of batches
	Standard model	100	
FPR	Assumption-free model	0	% of batches
	Standard model	16.7	
ARL	Assumption-free model	81	Samples
	Standard model	85	

The results showed an equal performance when dealing with faulty batches, indeed for both methods $TPR = 100\%$. However, the monitoring performed with a batch-wise unfolding PCA, was not able to correctly classify all the NOC batches, indeed its *FPR* is higher than the one obtained from the monitoring with the assumption-free modelling. In terms of detection speed, the models have a similar behaviour. However, the assumption-free modelling, as was pointed out in §4.1.2, does not discriminate whether the disturbance occur at the beginning or halfway through the processing.

Therefore, as is noticeable, the assumption-free methodology applied to this dataset gives more robust results in terms of detection strength but has similar results in terms of detection speed.

4.2 Dataset 2

This dataset contains the data of an industrial polymerization reaction. It was presented by Nomikos and MacGregor (1995) and is described §2.2. Ten variables are measured, the batches are aligned and each one has 100 samples. fifty NOC batches are used to calibrate the variable-wise unfolding MPCA model, and 5 batches are used to validate the model.

4.2.1 Dataset 2: assumption-free modelling calibration

The matrix is variable-wise unfolded and autoscaled, then a PCA model is built using 2 PCs. Table 4.6 shows the results of the PCA.

Table 4.6 *Dataset 2: Summary of the PCA model.*

PC No.	λ_a	R^2	$R^2_{cum.}$	*RMSECV*
1	6.39	63.9 %	63.9 %	0.66
2	2.16	21.6 %	85.5 %	0.44
3	0.95	9.5 %	95 %	0.36
4	0.25	2.5 %	97.5%	0.35
5	0.11	1.1%	98.6 %	0.66

The total captured variance is 85.5%. The number of PCs chosen is 2 due to the grid search algorithm. However, from the RMSECV the optimal number of PCs is 4.

The scores obtained from the PCA are plotted and the score plot is shown in Figure 4.12.

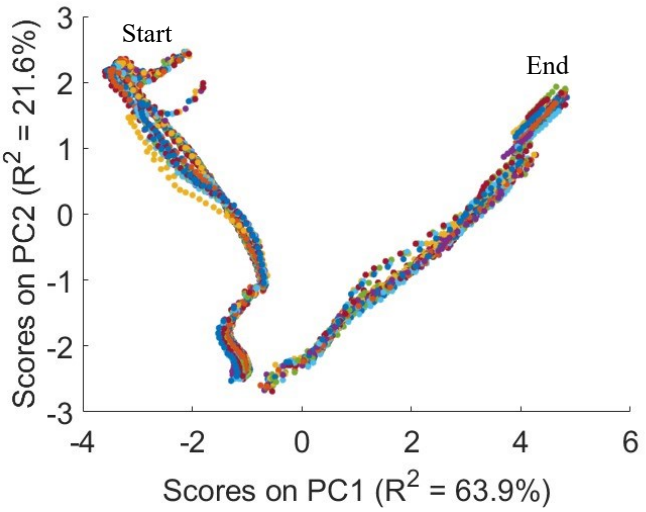


Figure 4.12 *Dataset 2: score plot obtained after variable-wise unfolding the calibration set of Dataset 1 described in §2.2. Each point represents a time instant of a single batch. Each colour represents a batch of the calibration set.*

The process evolves from the left to the right as indicated in Figure 4.12. The variability between batches is almost constant throughout the whole process.

In order to calibrate the assumption-free model, both γ and β are set equal to 0.9. Indeed, a cell must contain at least 90% of the batches in order to be valid. Moreover, no grid configuration which contains less than 90% of scores in its valid cells will be considered. The maximum number of cells into which the score plot will be partitioned by the grid search algorithm, is set equal to 12 on both PCs. Table 4.7 summarizes the settings of the assumption-free modelling.

Table 4.7 *Dataset 2: Hyperparameter values used during the calibration of the assumption-free model.*

Hyperparameter	Value	Units
γ	0.90	Fraction of scores
β	0.90	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

Using these settings, the grid that satisfied the criteria has 21 valid cells and a grid configuration of 12×6 which includes 92.0% of all the scores (Figure 4.13).

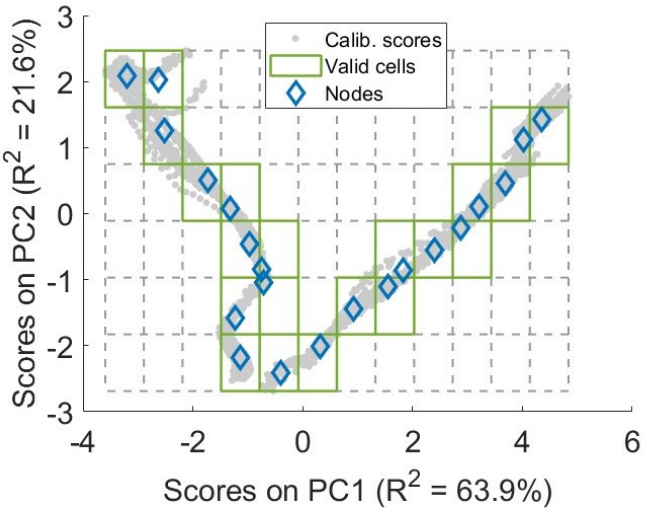


Figure 4.13 *Dataset 2: best grid found by the algorithm for the calibration batches.* ($n_{PC1} = 12$ *and* $n_{PC2} = 6$).

After the best grid has been identified, the nodes are chronologically ordered and interpolated, leading to the reconstruction of the common trajectory shown in Figure 4.14.

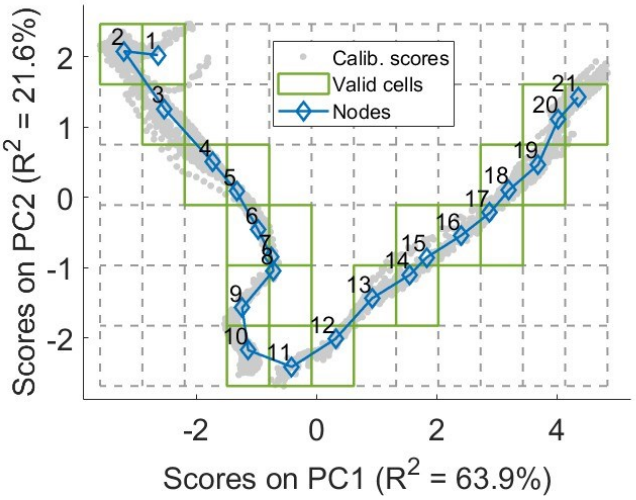


Figure 4.14 *Dataset 2: ordered common trajectory.*

The common trajectory is able to reconstruct the average batch run, indeed it approximates the path followed by the scores. Most of the scores not included in the valid cells are after the last node because the top right cell did not have more than 90% of the batches in it.

The relative time is estimated for all the batches in the calibration set and the results are reported in Figure 4.15.

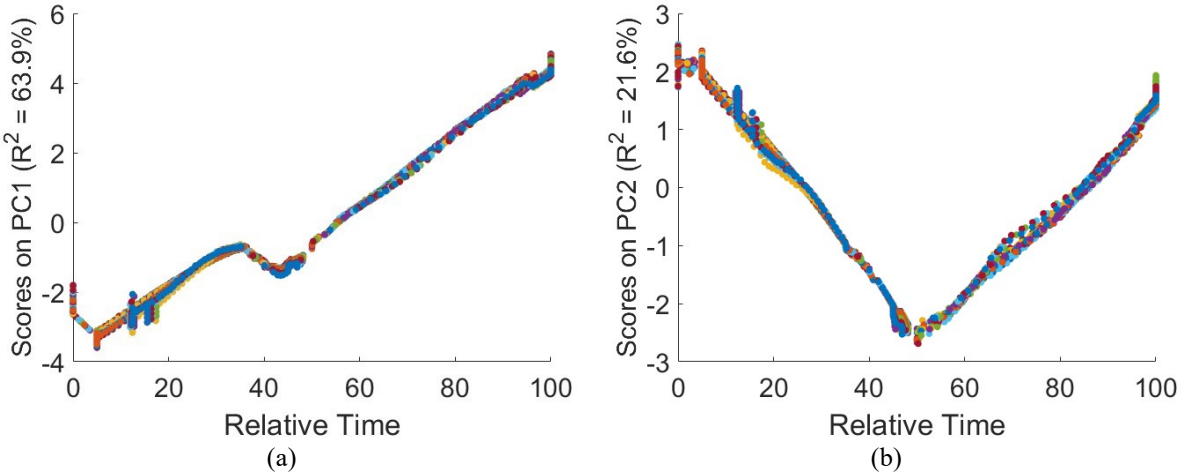


Figure 4.15 *Dataset 2: Relative time of the calibration batches of (a) PC1 and (b) PC2.*

From the relative time, it is possible to see the evolution of the batches and how the events are aligned.

The control limits are evaluated according to §3.6 with 95% confidence level. After giving a sign to each distance of the means of the batches from the common trajectory, it resulted that only 36.8% of the cells have a normal distribution of the distances from the common trajectory. However, in some cells the p-value was close to the limiting one even if the hypothesis testing rejected the null hypothesis.

Figure 4.16 shows the control limit around the common trajectory built considering a 95% confidence level.

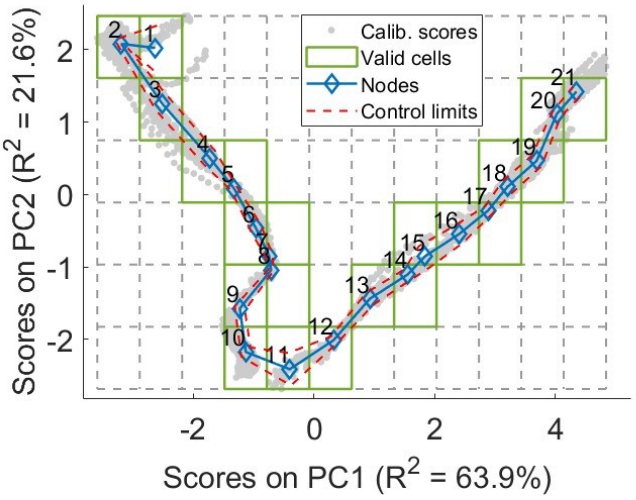


Figure 4.16 *Dataset 2: reconstructed common trajectory and control limits around it for the calibration batches.*

The control limits are broader in the first valid cell because the cell contains some scores which are far from the node. The percentage of the means of the batches out of the control limit is 14.9%, which is acceptable considering that not all cells contain distances that are normally distributed.

The control chart on the *SPE* is built using a 95% confidence level on the empirical CDF of the *SPE* in each valid cell. The control chart is shown in Figure 4.17.

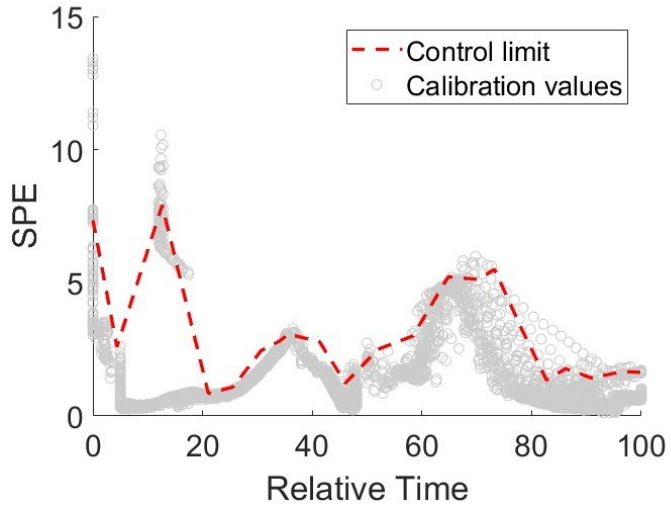
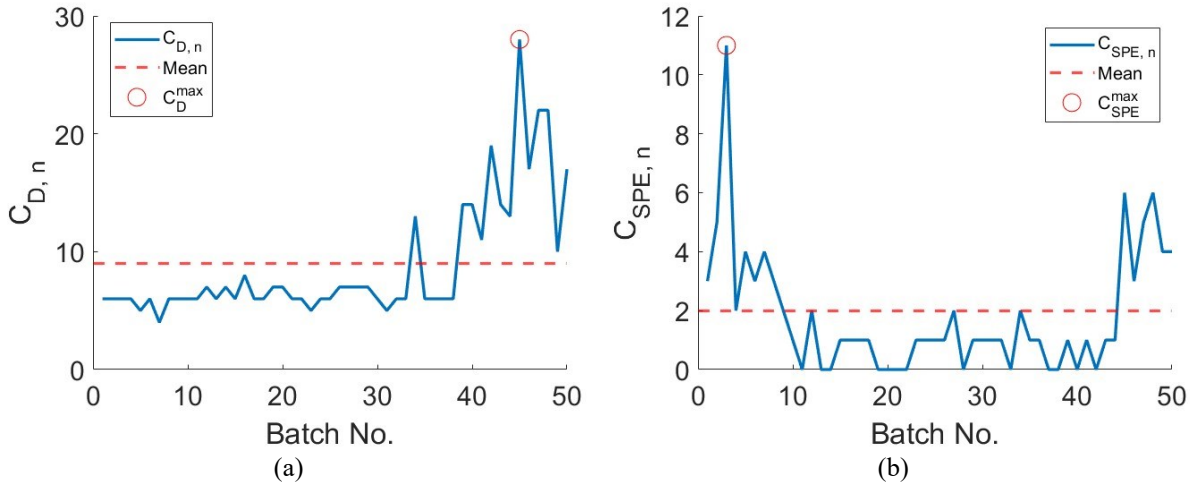


Figure 4.17 *Dataset 2: SPE control chart with the SPE of the calibration scores.*

The limit evaluated for each cell correctly represents the evolution of the *SPE*. However, the third valid cell contains some scores which have a higher value of the *SPE*; this leads to a higher value of the control limit for that cell. The percentage of the *SPE* out of the control limit is 5.0% which is coherent with the choice of 95% confidence level.

The calibration of the alarm trigger is performed as explained in §3.8 and Figure 4.18 shows the maximum number of consecutive points out of the control limit on both the charts for all the batches.



(a) (b)

Figure 4.18 *Dataset 2: calibration set. Maximum number of consecutive points out of (a) the control limits around the common trajectory and (b) the control limit of the SPE.*

$C_D^{max} = 28$ and is found in the 45th batch of the calibration set, while C_{SPE}^{max} is found in calibration batch No. 3 and its value is 11. Having $C_D^{max} = 28$ means that more than a fourth of the batch run has to be out of the control limit in order to consider the batch faulty.

4.2.2 Dataset 2 monitoring using the assumption-free model

The monitoring is performed according to §3.9, indeed all the batches of the validation set are projected onto the model to assess whether they are faulty or not. The results of the process monitoring are shown in Figure 4.19.

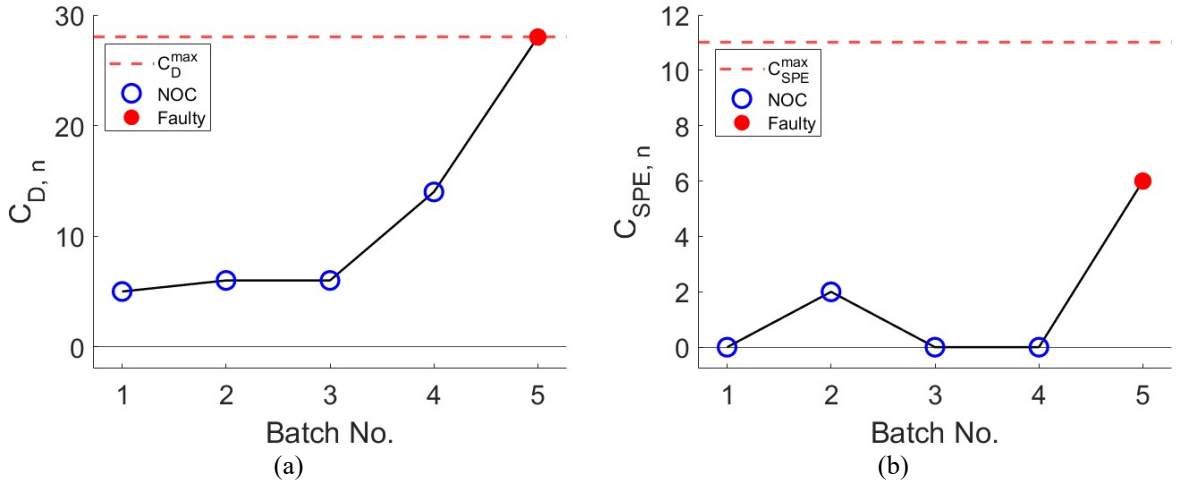


Figure 4.19 *Dataset 2: monitoring of the validation set. Maximum number of consecutive points outside the control limit on (a) the score plot and on (b) the SPE The truly faulty batches are indicated in red.*

All the NOC batches have been correctly identified, the only faulty batch (No. 5) has been identified through the control limits around the common trajectory. Indeed, it has a number of consecutive scores out of the confidence limits equal to C_D^{max}, therefore an alarm is triggered. Figure 4.20 shows the validation batch No. 5 on both the control charts.

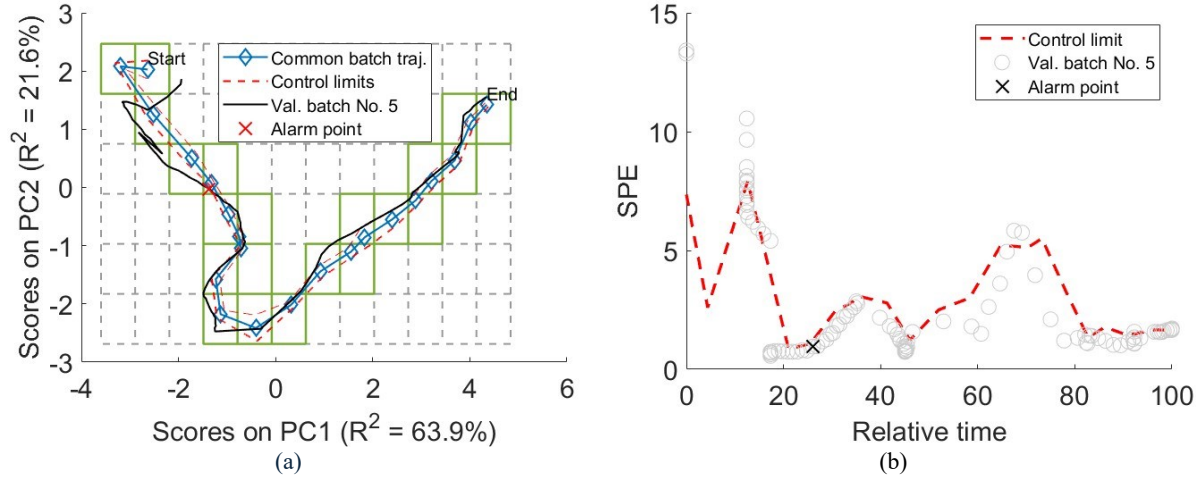


Figure 4.20 *Dataset 2: Validation batch No. 5. (a) Control chart on the score plot and (b) control chart on the SPE.*

From the control chart on the score plot it can be noticed that the batch starts far from the first node and has an unusual behaviour in the first half of the process. Indeed, the sample that triggered the alarm was the 32nd and the process was upset since the beginning. From the *SPE* control chart no unusual behaviour is noticeable.

The monitoring performances of the model are summarized in Table 4.8.

Table 4.8 *Dataset 2: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	100 %	% of batches
FPR	0	% of batches
ARL	31	Samples

The model is able to correctly identify all the batches in the validation set, therefore the $TPR = 100\%$ and the $FPR = 0\%$. The $ARL = 31$ samples because the only faulty batch has been identified at sample 32.

4.2.3 Dataset 2: monitoring using a standard MPCA method

The data are already aligned therefore can be directly unfolded in the variable direction and then autoscaled. A PCA model is built on the unfolded matrix. The number of chosen PCs by Sartori (2023) is 3 and the total explained variability by the PCA model is 64.5%.
Figure 4.21 shows the score plot of the first 2 PCs

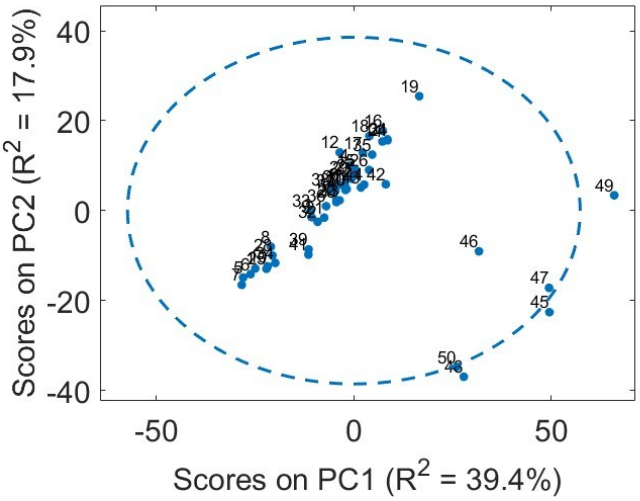


Figure 4.21 *Dataset 2: score plot obtained for a standard MPCA model. Each point represents a batch, the dotted line is the 95% confidence interval on the multivariate distribution of the scores.*

The scores are multi-normally distributed and 4 batches are out the 95 % confidence limit. This value is acceptable considering that the calibration set contains 50 batches.
The control chart on the *SPE* is evaluated as described in §1.1 and is shown in Figure 4.22.

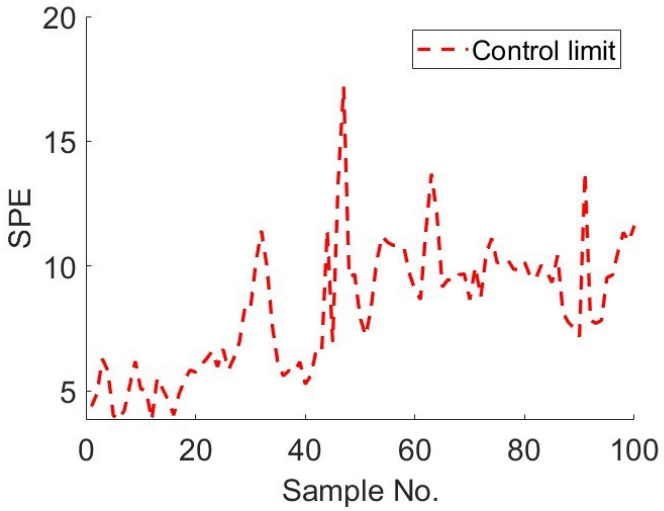


Figure 4.22 *Dataset 2: SPE control chart built on a batch-wise unfolding MPCA obtained using (1.13)*

The control limit respects the confidence limit chosen as the percentage of points out of it is 4.7%. The monitoring is performed according to §1.2.1 and the maximum number of consecutive points out con the control limit chosen by Sartori (2023) is 1 for T^2 and 3 for the *SPE*. The results of the monitoring are reported in Table 4.9.

Table 4.9 *Dataset 2: performance indicators of the monitoring scheme.*

Performance indicator	Value	Units
TPR	100	% of batches
FPR	0	% of batches
ARL	21	Samples

The model is able to identify both NOC and faulty batches correctly, indeed $TPR = 100\%$ and $FPR = 0\%$. The *ARL* is 21 samples meaning that a fault is on average detected after a fifth of the batch run.

4.2.4 Dataset 2: comparison of the results

The monitoring has been carried out on the same dataset with both methodologies: the assumption-free modelling and a standard one. The monitoring performances obtained with both methods are reported in Table 4.10.

Table 4.10 *Dataset 2: comparison of the performance indicators of the monitoring for both methods used*

Performance indicator	Method	Value	Units
TPR	Assumption-free model	100	% of batches
	Standard model	100	
FPR	Assumption-free model	0	% of batches
	Standard model	0	
ARL	Assumption-free model	31	Samples
	Standard model	21	

The results showed an equal performance when dealing with faulty batches, indeed for both methods $TPR = 100\%$. The same is noticeable when dealing with NOC batches, indeed the *FPR* of both models is 0%. In terms of detection speed, the models have a similar behaviour. However, the assumption-free modelling is slower with respect to the other method. Indeed, if the assumption-free modelling detects on average a fault after a third of the run, the monitoring performed after a batch-wise unfolding detects it after a fifth of the process.
Therefore, the monitoring performances is the same in terms of detection strength but in terms of detection speed the batch-wise unfolding PCA performs better.

4.3 Dataset 3

This dataset contains the data of a simulated production of *Saccharomyces cerevisiae*. It was presented by González-Martínez et al. (2018) and as described §2.3. Eleven variables are measured, the batches are not aligned and have a mean duration of 34 hours and a mean of 211 samples per batch. Fourty NOC batches are used to calibrate the variable-wise unfolding MPCA model, 45 batches are used to validate the model built, 40 of them are faulty.

4.3.1 Dataset 3: assumption-free modelling calibration

The matrix containing the data is unfolded in the batch direction, autoscaled and used to build the PCA model. The results of the PCA are shown in Table 4.11.

Table 4.11 *Dataset 3: Summary of the PCA model.*

PC No.	λ_a	R^2	$R^2_{cum.}$	*RMSECV*
1	3.61	36.1 %	36.1 %	0.86
2	3.07	30.7 %	66.8 %	0.68
3	1.98	19.8%	86.6 %	0.52
4	0.78	7.8%	94.4 %	0.42
5	0.38	3.8%	98.2 %	0.30

The number of PCs chosen is 2 due to the grid search algorithm, indeed the minimum of the *RMSECV* is found for 5 PCs. The scores obtained from the PCA are plotted and are shown in Figure 4.23.

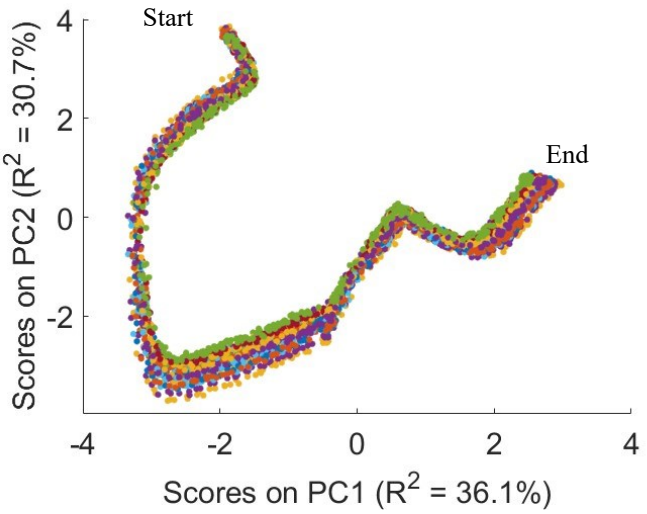


Figure 4.23 *Dataset 3: score plot obtained after variable-wise unfolding the calibration set of Dataset 1 described in §2.3. Each point represents a time instant of a single batch. Each colour represents a batch of the calibration set.*

From the score plot it is possible to notice that, even though the batches are not aligned the starting and ending point of the process are almost the same for all the batches. Moreover, the variability is smaller at the beginning, then increases when both PCs reach their minimum and finally decreases towards the end of the process.

In order to calibrate the assumption-free model, both γ and β are set equal to 0.9. Indeed, a cell must contain at least 90% of the batches in order to be valid. Moreover, no grids which contain less than 90% of scores in their valid cells will be considered. The maximum number of cells into which the score plot will be divided by the grid search algorithm, is set equal to 12 on both PCs. Table 4.12 summarizes the settings of the assumption-free modelling.

Table 4.12 *Dataset 3: Hyperparameter values used during the calibration of the assumption-free model..*

Hyperparameter	Value	Units
γ	0.90	Fraction of scores
β	0.90	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

Using these settings, three grid configurations that satisfied the criteria have been identified. The grids have 24 valid cells and are reported in Figure 4.24

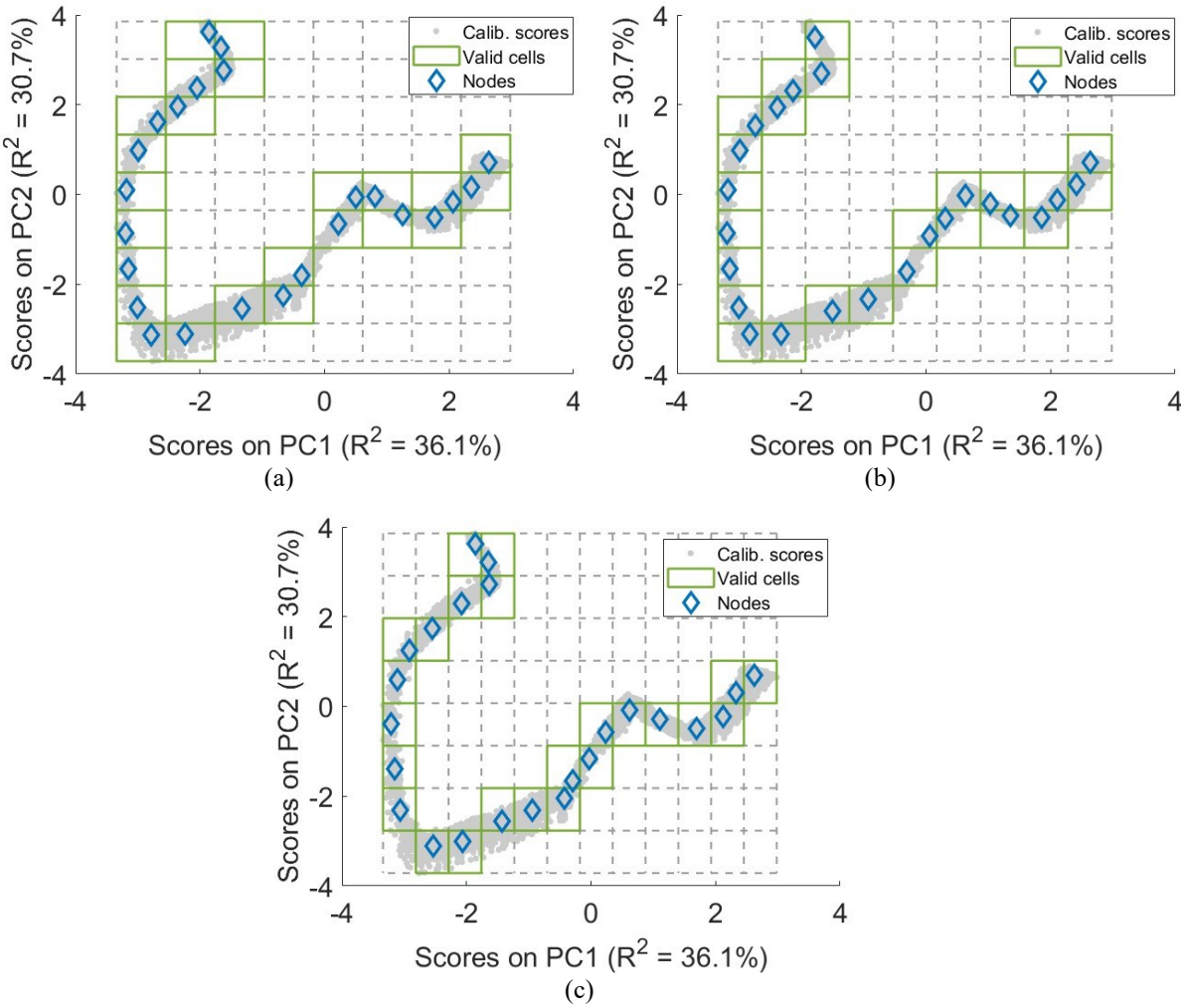


Figure 4.24 *Dataset 3: valid grids identified by the grid search algorithm for the calibration batches. (a) .* $n_{PC1} = 8$ *and* $n_{PC2} = 9$. *(b) .* $n_{PC1} = 9$ *and* $n_{PC2} = 9$. *(c)* $n_{PC1,w} = 12$ *and* $n_{PC2} = 8$. *In all cases the number of valid cells is 24.*

All the grids are satisfactorily able to approximate the common trajectory; however the one in Figure 4.24a contains more scores in the valid cells, therefore it is the one chosen by the model. The chosen grid has a 8×9 configuration and includes 97.3% of all the scores. Table 4.13 summarizes the grid characteristics of each one of the possible configurations.

Table 4.13 *Dataset 3: summary of the three possible configurations identified by the grid search algorithm.*

Configuration	n_{PC1}	n_{PC2}	n_{valid}^{max}	Scores included in the valid cells	Figure
1	8	9	24	97.3 %	4.24a
2	9	9		96.4 %	4.24b
3	12	8		91.1 %	4.24c

The nodes are chronologically ordered and interpolated in order to build the common trajectory, shown in Figure 4.25.

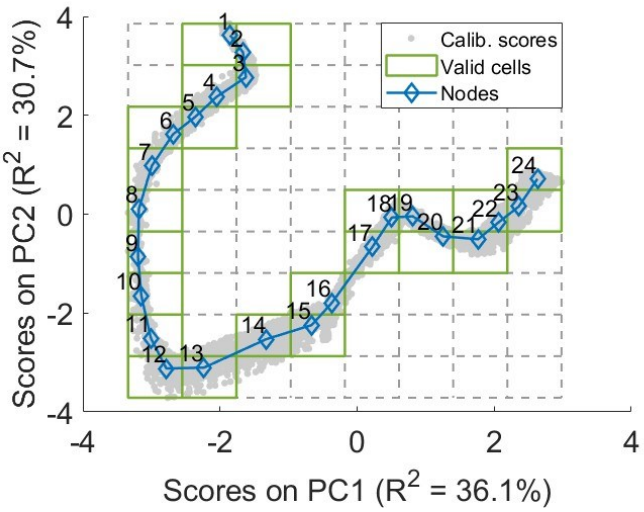


Figure 4.25 *Dataset 3: ordered common trajectory.*

The obtained common trajectory perfectly describes the evolution of the process. Once the common trajectory is available, it is possible to estimate the relative time of the batches in the calibration set as described in §3.5. The results are shown in Figure 4.26

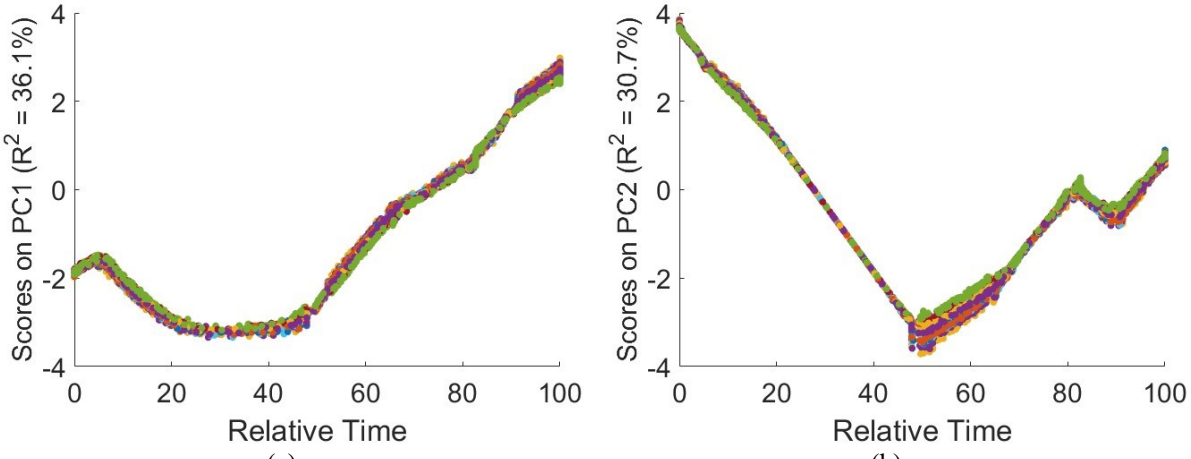


Figure 4.26 *Dataset 3: Relative time of the calibration batches of (a) PC1 and (b) PC2.*

The evolution of the scores with respect to the relative time shows the capability of the model to deal with non-aligned batches. Indeed, the main changes in the scores occur in the same manner and at the same relative time for all the batches in the calibration set even though the data were not aligned.

The control limits around the common trajectory are evaluated from the distribution of the distances of the means of the batches for each valid cell according to §3.6 using a 95% confidence level. The control limits obtained are shown in Figure 4.27.

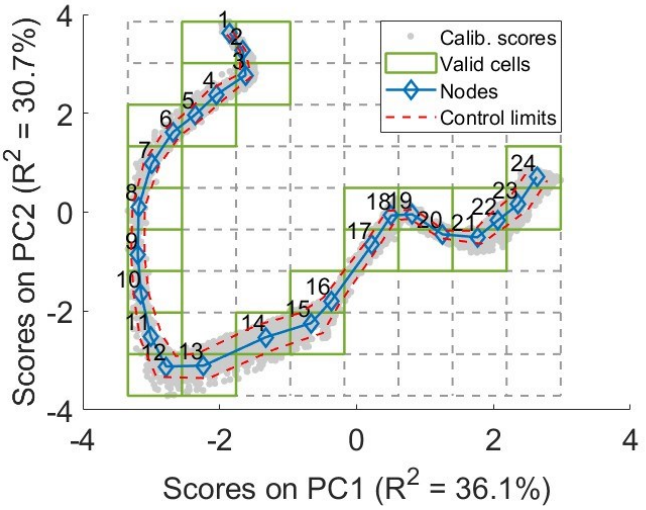


Figure 4.27 *Dataset 3: reconstructed common trajectory and control limits around it for the calibration batches.*

As expected the control limits are narrower at the beginning of the process, due to the reduced variability, and become broader after the 12th node. The control limits are not completely able to capture all the variability towards the end of the process. The percentage of the means of the batches out of the control limit is 10.6%. The percentage is higher with respect to the considered confidence level because the distances are not normally distributed in all the valid cells. Indeed, the distribution is normal in 95.8% of the cells.

In order to build the control chart on the *SPE*, the empirical distribution of the *SPE* in each valid cell is taken into account and a limit of each cell is evaluated using a 95% confidence level. The control chart obtained is shown in Figure 4.28.

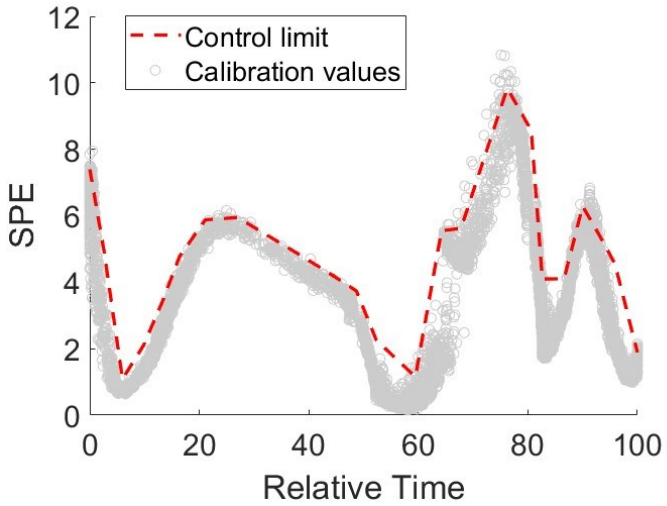


Figure 4.28 *Dataset 3: SPE control chart with the SPE of the calibration scores.*

The obtained control limit is able to represent the evolution of the *SPE* of the NOC batches. The percentage of *SPE* out of the control limit is 5.1% which is coherent with the choice of 95% confidence level.

In order to perform process monitoring there is the need to evaluate C_D^{max} and C_{SPE}^{max}. Both are evaluated considering the maximum number of consecutive points out of the control limits in the calibration set. Figure 4.29 shows $C_{D,n}$ and $C_{SPE,n}$ for all the batches.

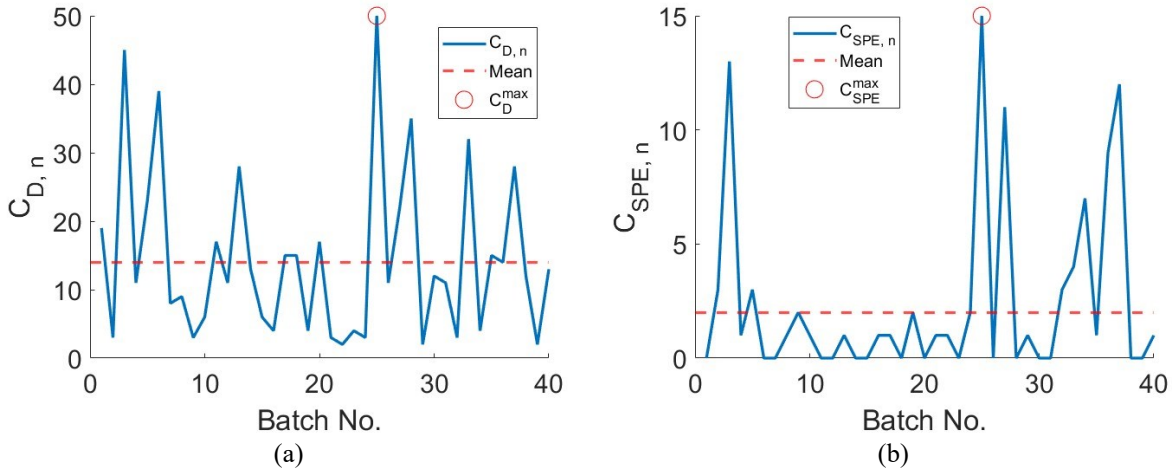


Figure 4.29 *Dataset 3: calibration set. Maximum number of consecutive points out of (a) the control limits around the common trajectory and (b) the control limit of the SPE.*

For this process $C_D^{max} = 50$ and is found in calibration batch No. 25, this value implies that about one fourth of the batches must be outside the control limit around the common trajectory in order for a batch to be classified as faulty. C_{SPE}^{max} is found in calibration batch No. 25 too and its value is 15.

4.3.2 Dataset 3: monitoring using the assumption-free model

In order to perform the process monitoring and assess the monitoring performances of the built model, the validation set is used. The scores are evaluated for each batch and projected onto the score plot to calculate their relative time. The residuals are evaluated in order to assess the behaviour of the *SPE* of the validation batches. The monitoring is carried out for all the 45 batches in the validation set and the results are reported in Figure 4.30.

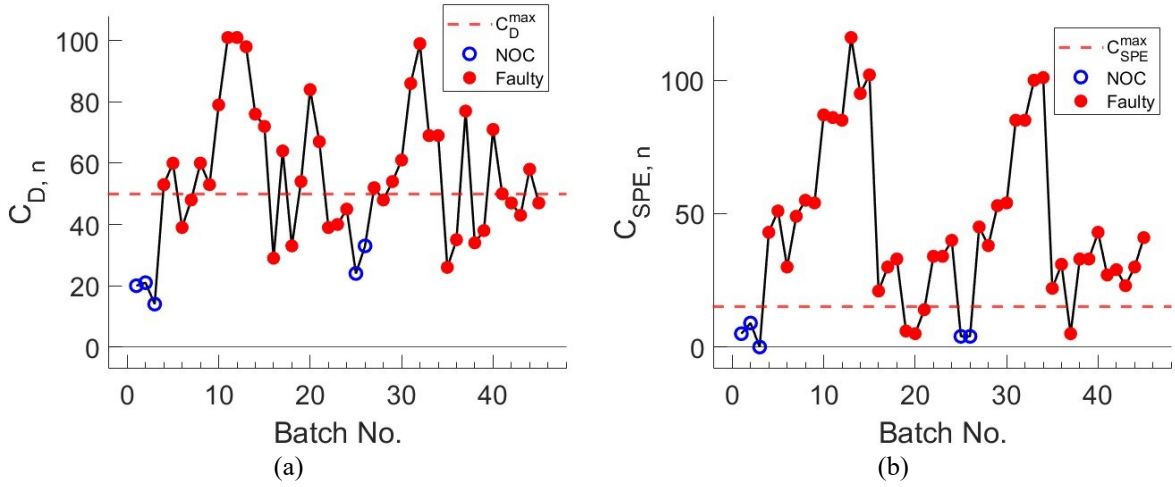


Figure 4.30 *Dataset 3: monitoring of the validation set. Maximum number of consecutive points outside the control limit on (a) the score plot and on (b) the SPE The truly faulty batches are indicated in red.*

All the NOC batches are correctly identified, indeed none of them have $C_{D,n}$ or $C_{SPE,n}$ greater than the respective limit. Also all the faulty batches have been identified, indeed, the batches that are not considered faulty from a control chart are identified by means of the other. An example are the validation batches No. 19, 20 and 21 which are normal according to the control chart on the *SPE* while resulted faulty from the analysis of the control chart on the score plot. The evolution of validation batch No. 19 is shown in Figure 4.31.

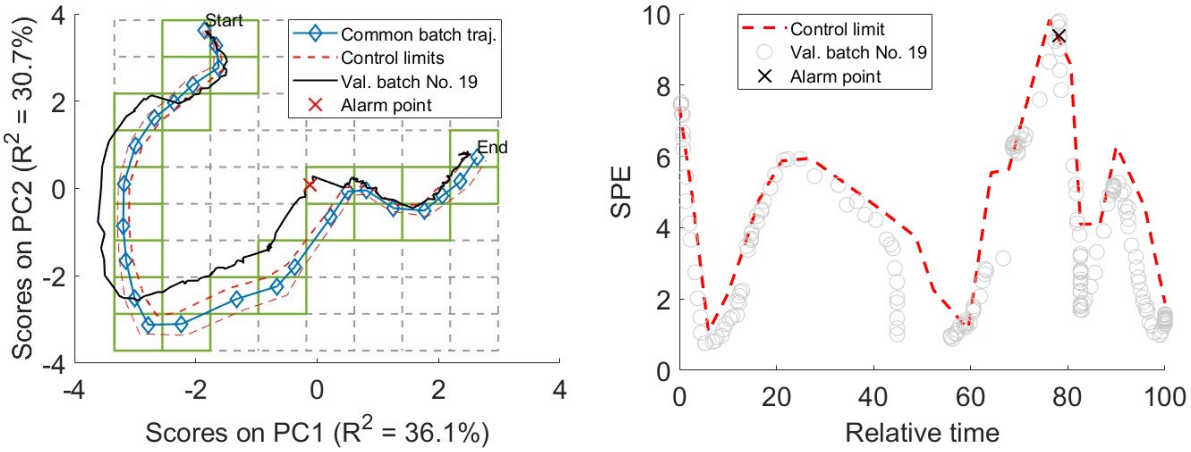


Figure 4.31 *Dataset 3: Validation batch No. 19. (a) Control chart on the score plot and (b) control chart on the SPE.*

The batch has been identified by means of the control limits around the common trajectory. The batch was upset since the beginning, indeed, between the fourth and the fifth node the batch goes out the control limits but not for a sufficient number of samples. However, after it passes again inside the limits, it departs from the common trajectory as an alarm is triggered at the 109th sample. As already mentioned, no alarms are triggered in the control chart on the *SPE*. Validation batches No. 6, 16 and 35 are identified only by means of the control chart on the *SPE*, indeed their $C_{D,n}$ is lower than the limit. Figure 4.32 shows both control charts for validation batch No. 35.

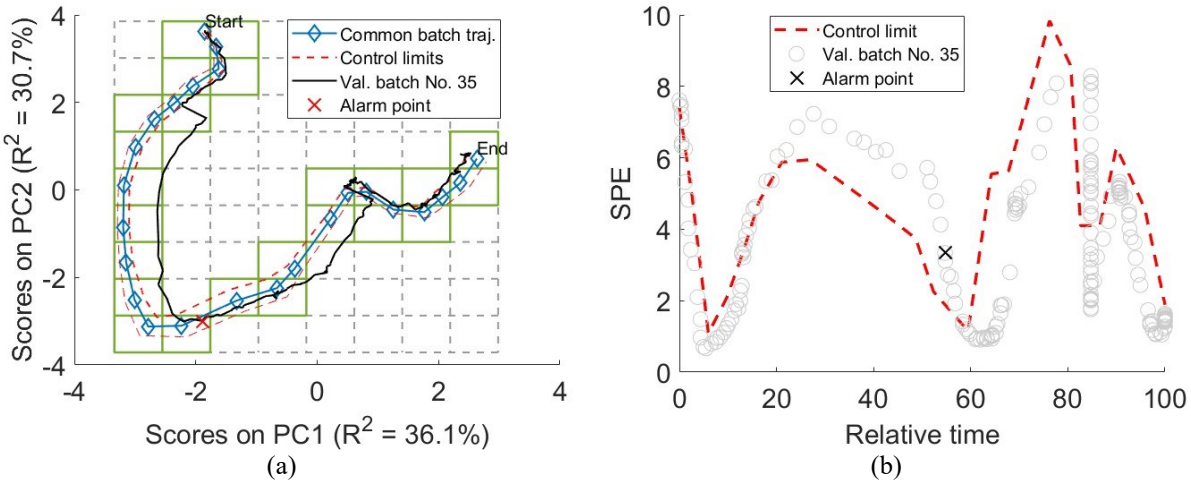
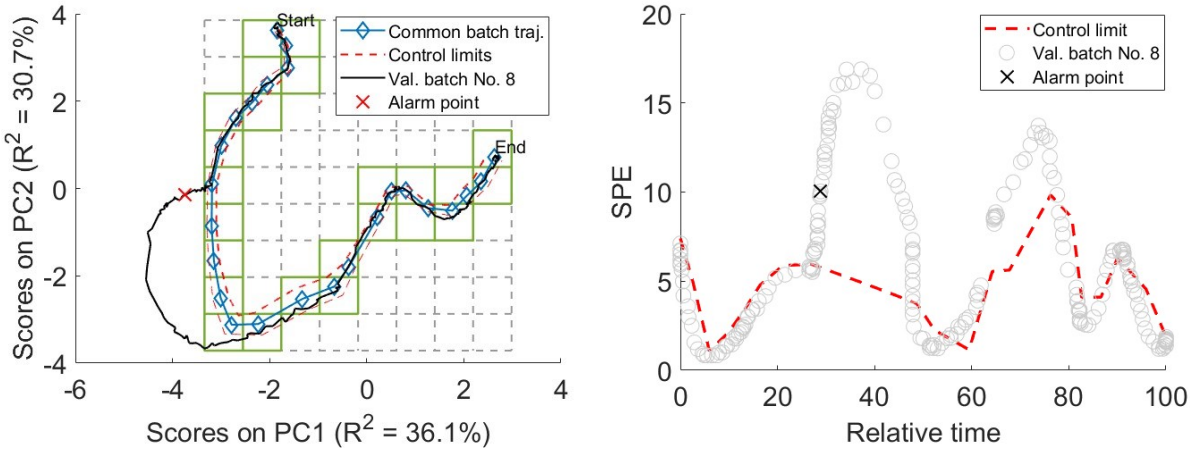


Figure 4.32 *Dataset 3: Validation batch No. 35. (a) Control chart on the score plot and (b) control chart on the SPE.*

The batch was upset since the beginning, but the *SPE* starts deviating from the 44th sample and the alarm is triggered at the 59th sample. From the score plot a deviation occurs after the 4th node of the common trajectory, however the departure is not enough big to trigger the alarm.
Most of the batches are identified by means of both control charts, however the alarm is triggered by the one that overcomes the limit faster. An example is validation batch No. 8 which is reported in Figure 4.33.



(a) (b)

Figure 4.33 *Dataset 3: Validation batch No. 8. (a) Control chart on the score plot and (b) control chart on the SPE.*

The batch was upset at sample 52 and, from the score plot, the departure from the common trajectory can be noticed after the 8th node. However, the fault was highlighted by the control chart on the *SPE* because C_{SPE}^{max} is overcome faster than the corresponding value of the other control chart. The sample that triggered the alarm was the 67th, meaning that the *SPE* got out the control limit immediately after the disturbance.
Once all the batches in the validation set have been monitored, the performance indicators of the model are evaluated, the results are reported in Table 4.14.

Table 4.14 *Dataset 3: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	100	% of batches
FPR	0	% of batches
ARL	81	Samples

As already mentioned, the model is able to correctly identify all the batches in the validation set, indeed *TPR* is 100% and *FPR* is 0%. Moreover, the model has a fast detection of the faults as the *ARL* is 54 samples, meaning that the faulty batches are identified on average after one fourth of the batch run, which translated in real time, is about 8.5 hours.

4.3.3 Dataset 3: monitoring using a standard MPCA method

The data are not aligned, therefore prior to the unfolding, both DTW according to Kassidas et al. (1998) and the one according to Ramaker et al. (2003) are performed onto the unaligned matrix. The final DTW is performed considering the geometric mean of the weights of both DTWs as suggested by González-Martínez et al. (2011). The aligned matrix, is then batch-wise unfolded and autoscaled. A PCA model is built on the unfolded matrix. The number of PCs chosen by Sartori (2023) is 4 explaining 64.5% of the total variability of the data.
The score obtained by the PCA are projected on the score plot together with the evaluated 95% confidence limit and shown in Figure 4.34.

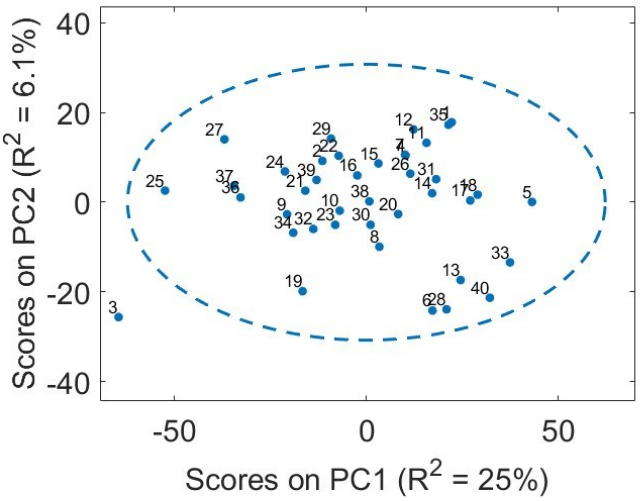


Figure 4.34 *Dataset 3: score plot obtained for a standard MPCA model. Each point represents a batch, the dotted line is the 95% confidence interval on the multivariate distribution of the scores.*

The batches are multi-normally distributed and only one of them is outside the confidence limit. The total number of batches in the calibration set is 40, therefore having only one batch out of the confidence limit corresponds to 2.5%. In order to respect the confidence level chosen the batches outside the limit should be 2, however the score plot of the first 2 PCs does not represent the entire variability captured by the model. Indeed, the number of PCs chosen is 3.
The *SPE* control chart is calibrated shown in §1.2.1 using a 95% confidence level. The control chart obtained is shown in Figure 4.35.

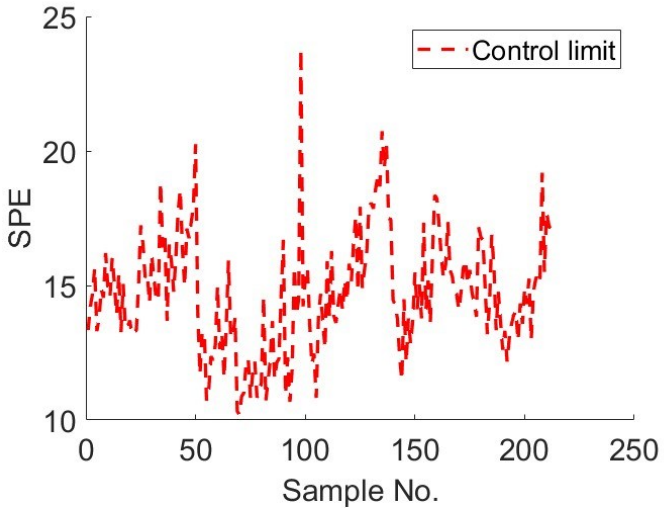


Figure 4.35 *Dataset 3: SPE control chart built on a batch-wise unfolding MPCA obtained using (1.13).*

The percentage of *SPE* of the calibration set out of the control limit is 5.1% which is coherent with the confidence level chosen.

The monitoring has been performed by Sartori (2023) applying the RGTW to every new sample and projecting the statistics onto the control chart to assess if the points were inside or outside the control limit. To assess if a batch was faulty or not the maximum number of consecutive points out of the control limit on T^2 was set to 3 while the one on the *SPE* equal to 300.

The results of the monitoring are reported in Table 4.15.

Table 4.15 *Dataset 3: performance indicators of the monitoring scheme.*

Performance indicator	Value	Units
TPR	50	% of batches
FPR	0	% of batches
ARL	83	Samples

The model built is able to correctly identify all the NOC batches, indeed $FPR = 0\%$. However not all the faulty batches have been correctly identified, indeed only half of them has been classified as normal ($FPR = 0\%$). Regarding the detection speed, $ARL = 83$ samples which translated to real time means about 13 hours.

4.3.4 Dataset 3: comparison of the results

The monitoring has been carried out on the same dataset with both methodologies: the assumption-free modelling and a standard one.

The monitoring performances obtained with both methods are reported in Table 4.16.

Table 4.16 *Dataset 3: comparison of the performance indicators of the monitoring for both methods used*

Performance indicator	Method	Value	Units
TPR	Assumption-free model	100	% of batches
	Standard model	50	
FPR	Assumption-free model	0	% of batches
	Standard model	0	
ARL	Assumption-free model	81	Samples
	Standard model	83	

The results showed an equal performance when dealing with NOC batches, indeed for both methods $FPR = 0\%$. However, the assumption-free modelling has a detection strength in identifying the faulty batches greater than the one of the batch-wise unfolding PCA. Indeed, the first one correctly identified all the faulty batches, while the second one was able to correctly classify only half of the faulty batches. In terms of detection speed, the models have a similar behaviour. Indeed, the assumption-free modelling identifies a faulty batch on average after 81 samples, while the monitoring performed with the batch-wise unfolding PCA identifies a fault after 83 samples.

Considering the obtained results, the assumption-free model has better monitoring performances. Indeed, even though the *FPR* is the same for both methods, the assumption-free modelling is able to correctly classify all the faulty batches. In terms of detection speed the methods are equivalent to each other. Therefore, for this dataset the assumption-free modelling performed better than the monitoring performed with a batch-wise unfolding PCA.

4.4 Dataset 4

This dataset contains the data coming from the simulated process of the production of penicillin. The data were obtained using the simulator Pensim developed by Birol et al. (2002), the dataset has been described in §2.2. twenty variables are measured and the batches are not aligned and have a mean duration of 8 hours and a mean of 800 samples per batch. thirty NOC batches are used to calibrate the model, while 39 batches (of which 30 are faulty) are used to assess the monitoring performance of the model.

4.4.1 Dataset 4: assumption-free modelling calibration

The matrix containing the data is unfolded in the batch direction, autoscaled and used to build the PCA model. Table 4.17 Shows the results of the PCA.

Table 4.17 *Dataset 4: Summary of the PCA model.*

PC No.	λ_a	R^2	$R^2_{cum.}$	*RMSECV*
1	9.61	48.1%	48.1 %	0.74
2	1.98	9.9%	58 %	0.69
3	1.86	9.3%	67.3 %	0.62
4	1.24	6.2%	74.5 %	0.64
5	1.01	5%	78.5 %	0.7

The chosen number of PCs is 2 due to the grid search algorithm, indeed the minimum of the *RMSECV* is found for 3 PCs.

The scores obtained from the PCA are plotted and are shown in Figure 4.36

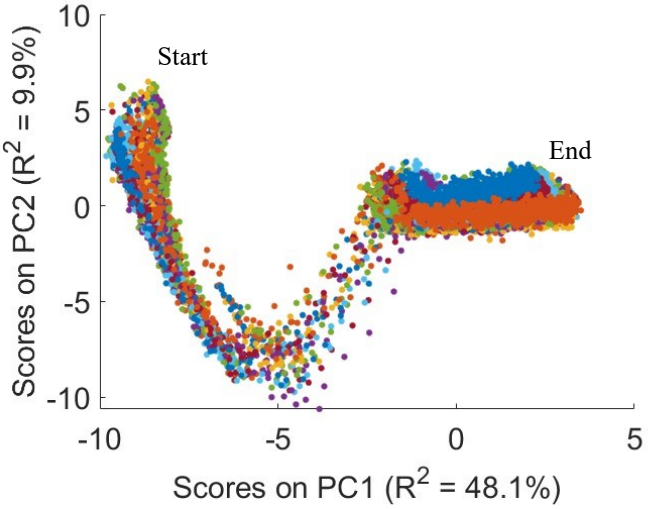


Figure 4.36 *Dataset 4: score plot obtained after variable-wise unfolding the calibration set of Dataset 1 described in §2.4. Each point represents a time instant of a single batch. Each colour represents a batch of the calibration set.*

The scores have a greater variability at the beginning and at the end of the process. Between -5 and -2, the scores on PC1 are more scattered due to the rapid change in the variables when passing from the batch to the fed-batch operation. Not all the scores start and end in the same point due to the difference in initial conditions and different values of the operating variables used to simulate the process.

The assumption-free model is calibrated considering a maximum of 12 cells in both direction of the PCs. Moreover, both γ and β are set equal to 0.95, meaning that a cell must contain at least 95% of the batches in order to considered valid and that a grid must contain minimum 95% of the total scores inside the valid cells to be considered valid. Table 4.18 summarizes the input used to calibrate the model.

Table 4.18 *Dataset 4: Hyperparameter values used during the calibration of the assumption-free model.*

Hyperparameter	Value	Units
γ	0.90	Fraction of scores
β	0.90	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

The grid search algorithm identified only one grid with the maximum number of valid cells. The grid has 13 valid cells and a configuration of 3×8 which includes 99.1% of all the scores. Figure 4.37 Shows the best grid found by the algorithm.

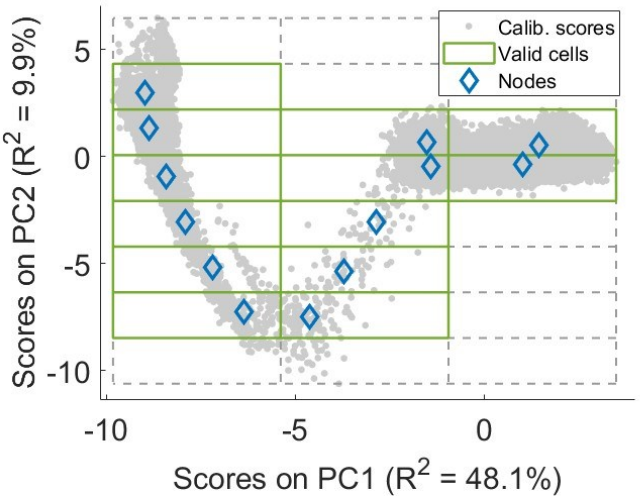


Figure 4.37 *Dataset 4: best grid found by the algorithm for the calibration batches.* ($n_{PC1} = 3$ *and* $n_{PC2} = 8$).

The trajectory is clearly recognizable until the beginning of the second processing step, where the cluster of points is characterized by 4 nodes that are difficult to order. The scores that are not included into the valid cell are the ones near the beginning of the process in the top left cell and the ones in the two cells at the bottom of the grid.

In order to identify the evolution of the common trajectory, the chronological ordering presented in §3.5 is used and the nodes are interpolated. The result is shown in Figure 4.38.

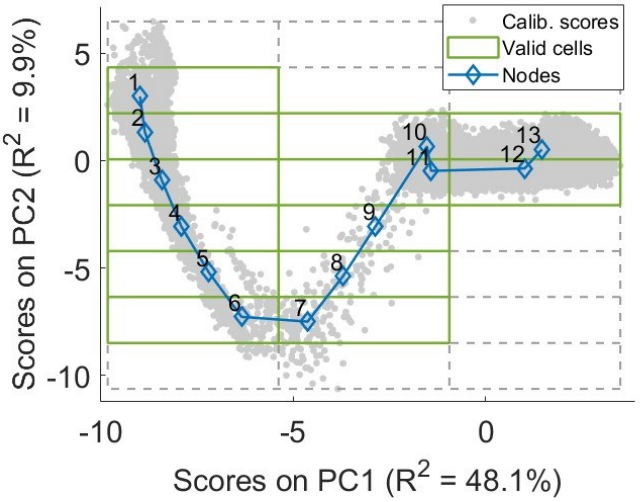


Figure 4.38 *Dataset 4: ordered common trajectory*

As already mentioned, the common trajectory was evident until the 9th node. However, the order of the last four nodes was not as evident. The points are ordered following the numbering given in Figure 4.38 which respect the evolution of the scores.

Once the common trajectory has been built, it is possible to project all the calibration scores onto it in order to estimate their relative time. The results are shown in Figure 4.39.

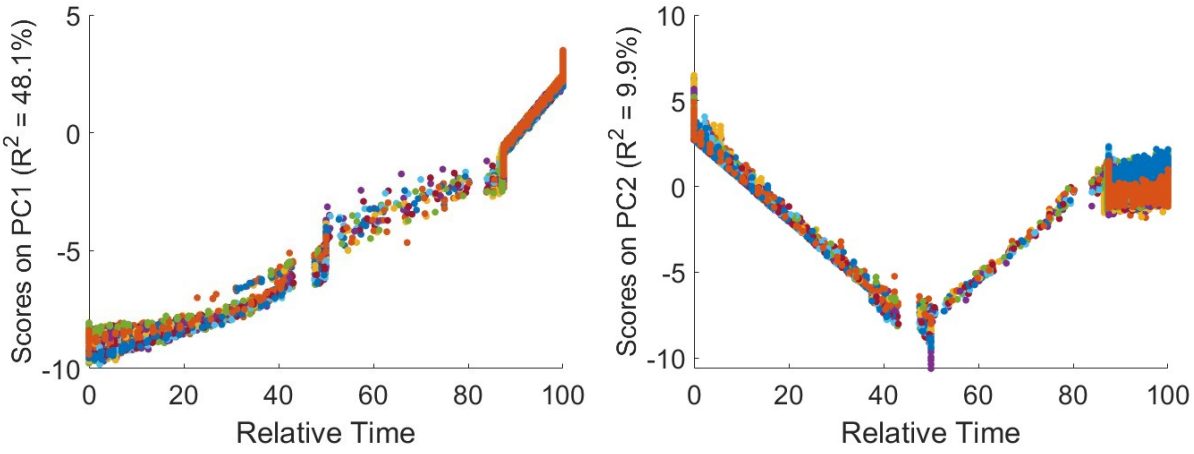


Figure 4.39 *Dataset 4: Relative time of the calibration batches of (a) PC1 and (b) PC2.*

All the scores evolve in the same manner when plotted against the relative time, this is a great advantage because it is a sort of internal alignment done by the model and allows to perform the process monitoring comparing the same state of the process. This feature is important when dealing with uneven length of batches like in this dataset.

The next step of the calibration of the assumption-free modelling is the estimation of the control limits around the common trajectory from the distribution of the means of the batches in each valid cell. For each valid cell a control limit is evaluated using a 95% confidence level and the points are interpolated in order to build the control chart. The results are shown in Figure 4.40.

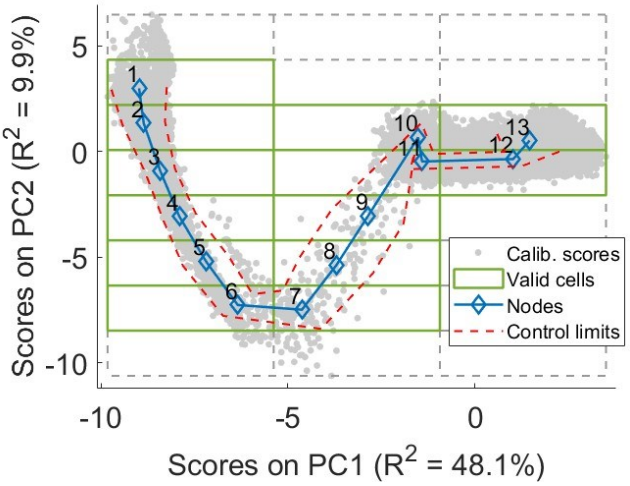


Figure 4.40 *Dataset 4: reconstructed common trajectory and control limits around it for the calibration batches*

The control limits follow the common trajectory and are able to fairly represent the variability of the process until the 9th node. After this node, the cells divide the cluster of scores in four blocks where the variability is reduced, therefore the limits are narrower with respect to the other cells.

The percentage of the means of the batches out of the confidence limit is 15.7% which is greater than the expected 5%. However, this value is reasonable considering that only a third of the valid cells had normally distributed distances of the batches from the common trajectory.

In order to build the control chart on the *SPE* the non-parametric distribution of this statistic in each valid cell is considered. A limit is evaluated for each valid cell considering a 95% confidence level and the limits are plotted against the relative time of the corresponding node. The control chart obtained is shown in Figure 4.41.

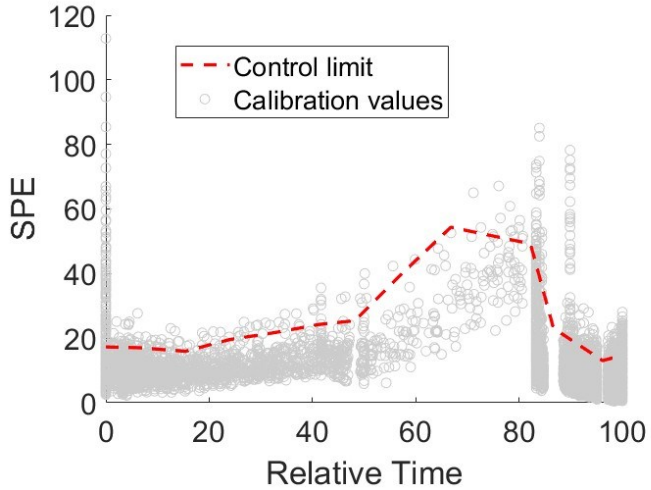


Figure 4.41 *Dataset 4: SPE control chart with the SPE of the calibration scores*

The control chart fairly represents the evolution of the *SPE* of the calibration set. The percentage of *SPE* out of the control limits is 4.8% which is coherent with the choice of 95% confidence level.

In order to perform process monitoring there is the need to evaluate the consecutive number of points out of the control limit that trigger an alarm. This is done according to §3.8 and the results are shown in Figure 4.42.

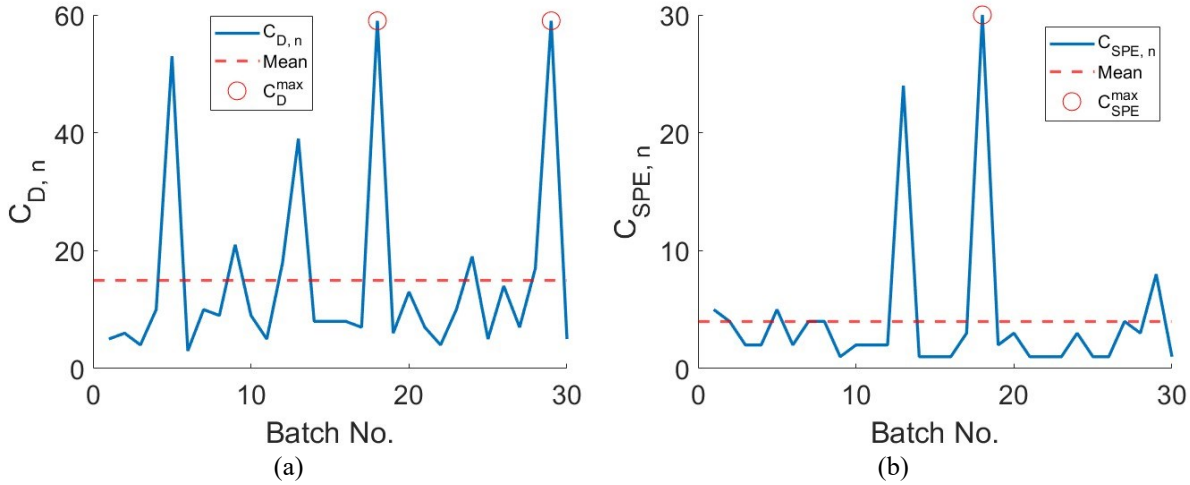


Figure 4.42 *Dataset 4: calibration set. Maximum number of consecutive points out of (a) the control limits around the common trajectory and (b) the control limit of the SPE.*

The maximum consecutive number out of the control limit around the common trajectory is 59 and is encountered in calibration batch No. 18 and 29. On the control chart on the *SPE*, $C_{SPE}^{max} = 30$ and is found in calibration batch No. 18.

4.4.2 Dataset 4: monitoring using the assumption-free model

After the model has been calibrated, it is possible to assess its monitoring performances using the validation set. It contains 39 batches, 9 of which are NOC, the remaining ones are faulty. Each sample is autoscaled and projected onto the score plot, its relative time is estimated along with the *SPE*. This procedure is repeated for all the samples in a batch run and for all the batches in the dataset. The results of the monitoring are shown in Figure 4.43.

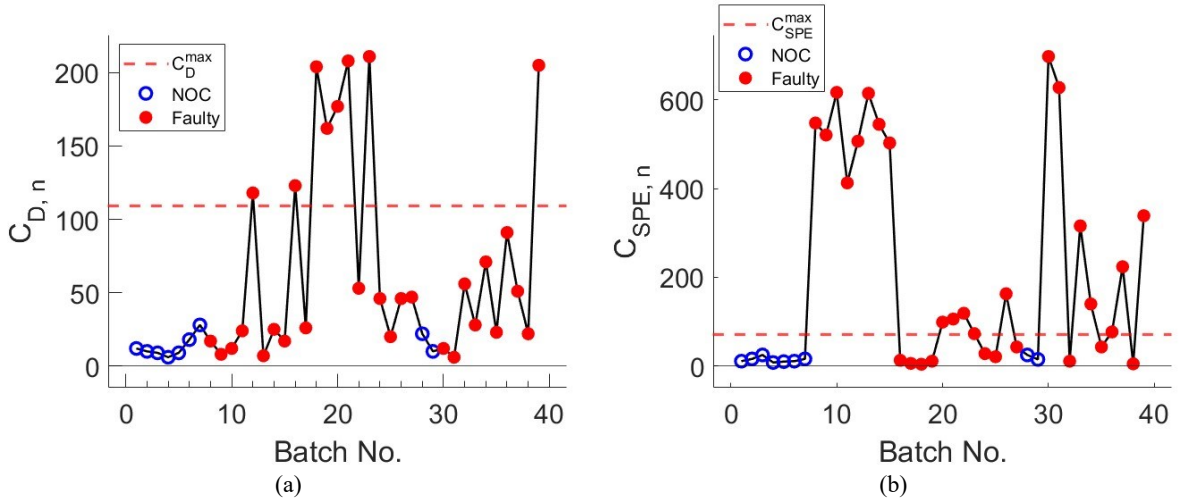


Figure 4.43 *Dataset 4: monitoring of the validation set. Maximum number of consecutive points outside the control limit on (a) the score plot and on (b) the SPE The truly faulty batches are indicated in red.*

All the NOC batches have been correctly identified, however some faulty batches were classified as NOC. Indeed, the batches that presented a disturbance in the aeration rate have been mainly identified by the *SPE* control chart. On the other hand, the batches that exhibited a disturbance in the feed rate have been identified only by means of the control limits around the common trajectory. As a matter of fact, these batches have a consecutive number of *SPE* out of the control limit similar to the one of the NOC batches.

Validation batch No. 8 has been classified as faulty only by means of the *SPE*, indeed it has only 21 consecutive scores out of the control limit around the common trajectory, but it has 388 consecutive points out of the control limit on the *SPE*.

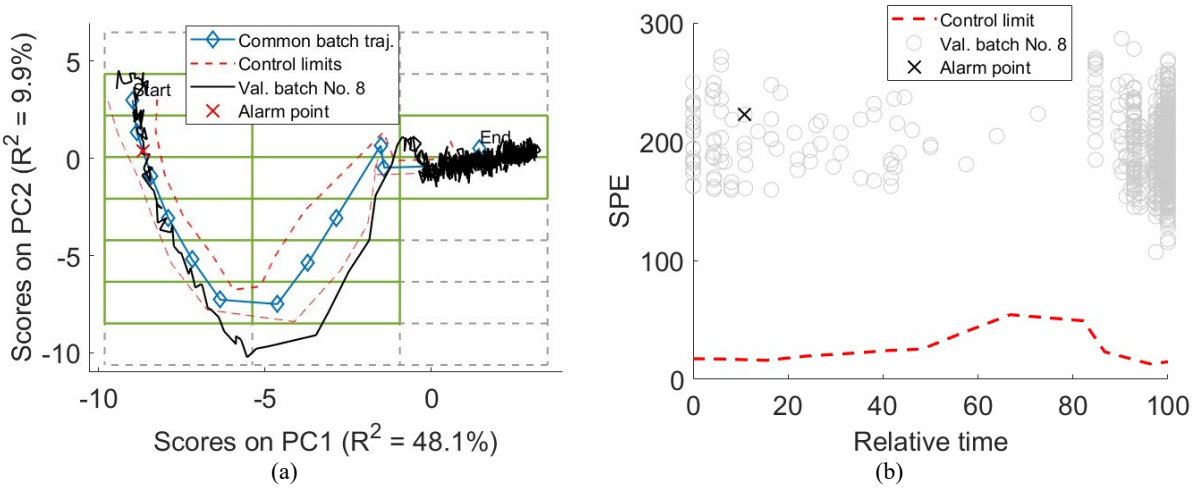


Figure 4.44 *Dataset 4: Validation batch No. 8. (a) Control chart on the score plot and (b) control chart on the SPE.*

From the score plot a fault is noticeable, indeed between the first and the second part of the process the batch goes out of the control limit. However, an alarm is triggered many samples earlier due to the control chart on the *SPE*. The sample that triggered the alarm was the 48th which means that the fault has been detected almost immediately. From Figure 4.44b it is noticeable that all the points (762 samples) are outside the control limit, even though $C_{SPE,n}$ of this batch is 388. This occurs because only points with relative time between 0 and 100 are taken into account to monitor the process. Indeed, when the relative time is evaluated, many scores are projected onto the last node and a relative time of 100 is assigned.

Validation batch No. 18 is one of the batches that triggered an alarm only in the control chart on the score plot. Indeed, the consecutive number of points out of the control limits around the common trajectory is 202, while the one on the *SPE* control chart is 1. Figure 4.45 shows both control charts for this batch.

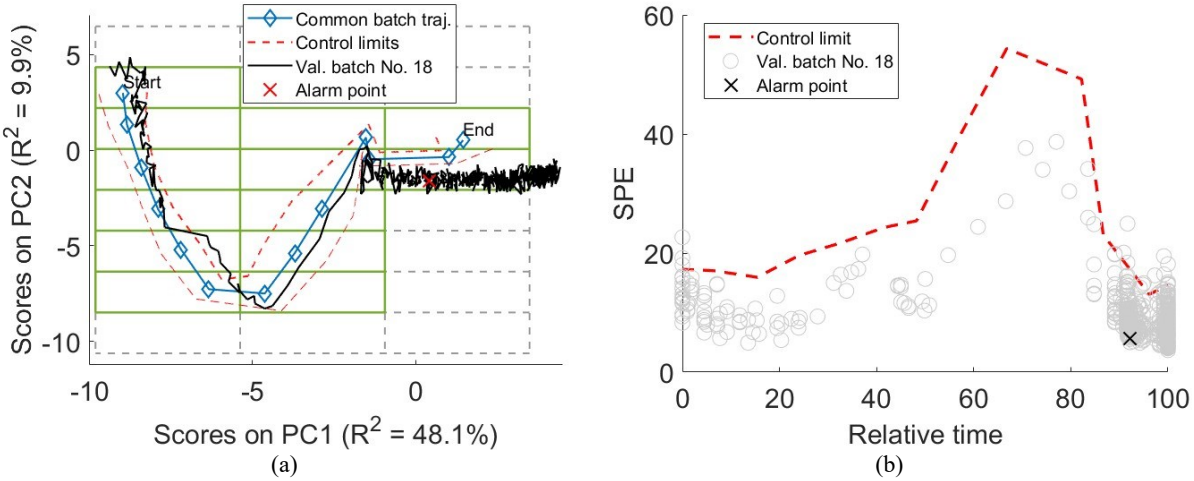


Figure 4.45 *Dataset 4: Validation batch No. 18. (a) Control chart on the score plot and (b) control chart on the SPE.*

Form the control chart on the *SPE*, no abnormal behaviour is noticeable, however from the control chart on the score plot there is a departure from the normality after the 10th node. Indeed, the scores are all outside the control limits and the alarm is triggered at sample No. 202. The batch run contains 613 samples and the disturbance in the substrate feed occurred at the beginning of the process. Therefore, the fault has been identified after a third of the run.

Unlike the batches considered above, validation batch No. 17 showed a fault on both control charts. Indeed $C_{D,n} = 103$ and $C_{SPE,n} = 388$, which are both above their limits. Figure 4.46 shows the control charts for this batch.

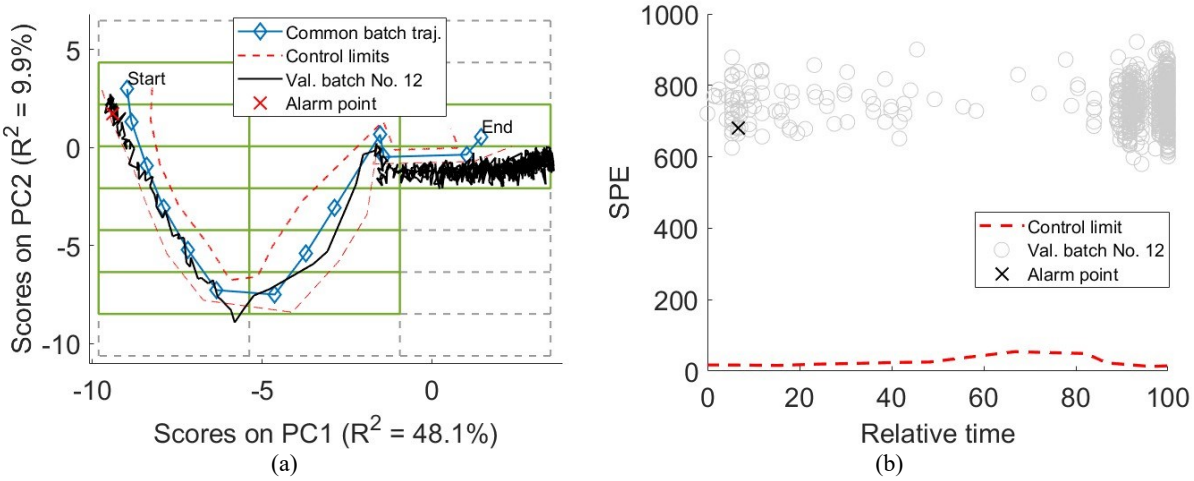


Figure 4.46 *Dataset 4: Validation batch No. 12. (a) Control chart on the score plot and (b) control chart on the SPE.*

Like in validation batch No. 18, the scores went out the control limits around the common trajectory after the 10th node. However, the batch had already been classified as faulty from the 31st sample due to the control chart on the *SPE*. The fault has been identified as soon as possible as the disturbance was present from the beginning of the process.

From the results of the monitoring on the validation set, the performance indicators are evaluated and shown in Table 4.19.

Table 4.19 *Dataset 4: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	70 %	% of batches
FPR	0	% of batches
ARL	155	Samples

As already mentioned, the model correctly identifies all the NOC batches, therefore $FPR = 0\%$, however only 21 out of 30 faulty batches have been identified with an *ARL* of 155 samples. Such an *ARL* implies that a faulty batch is identified on average after about 77 hours.

4.4.3 Dataset 4: monitoring using a standard MPCA method

The data are not aligned, therefore prior to the unfolding, both DTW according to Kassidas et al. (1998) and the one according to Ramaker et al. (2003) are performed onto the unaligned matrix. The final DTW is performed considering the geometric mean of the weights of both DTWs as suggested by González-Martínez et al. (2011). The aligned matrix, is then batch-wise unfolded and autoscaled. A PCA model is built on the unfolded matrix. The number of PCs chosen by Sartori (2023) is 2. The model built is able to explain 47.3% of the total variability of the data.

The scores obtained are shown in Figure 4.47 together with the 95% confidence limit.

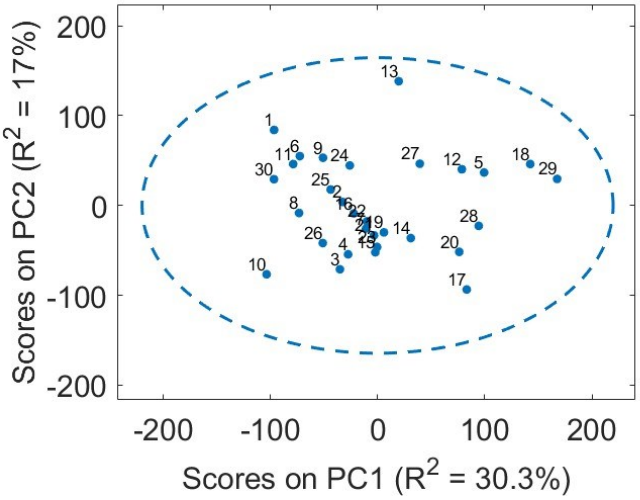


Figure 4.47 *Dataset 4: score plot obtained for a standard MPCA model. Each point represents a batch, the dotted line is the 95% confidence interval on the multivariate distribution of the scores.*

None of the scores is out of the 95% confidence limit, however the explained variability is less than a half, therefore it is possible that by adding more PCs some of the batches might be out of the confidence limit.

The *SPE* control chart is built as explained in §1.2.1. a 95% confidence level is considered to build the control chart and the result is shown in Figure 4.48.

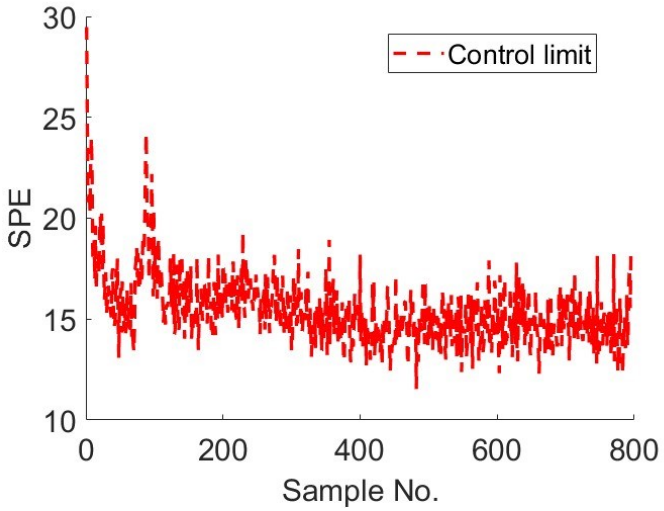


Figure 4.48 *Dataset 4: SPE control chart built on a batch-wise unfolding MPCA obtained using (1.13).*

The percentage of the *SPE* out of the control limit is 5%. This value is the expected one considering the chosen confidence level. To calibrate the alarm on both statistics, Sartori (2023) set the maximum number of points out of the control limit on T^2 equal to 3, while the corresponding value on the *SPE* equal to 700.

The monitoring is performed for all the batches in the validation set by aligning the data using the RGTW. The results of the monitoring are reported in Table 4.20.

Table 4. 20 *Dataset 4: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	62.1	% of batches
FPR	30	% of batches
ARL	400	Samples

The model built is not able to correctly identify all the faulty batches. Indeed only 62.1% of them are identified as faulty. The remaining part is wrongly labelled as NOC. When dealing with NOC batches, the model cannot classify a third of this type correctly, indeed its $FPR = 30\%$. The model is able to detect on average a faulty batch after 400 samples. Namely, after about 100 hours of processing.

4.4.4 Dataset 4: comparison of the results

The monitoring has been carried out on Dataset 4 for with both the methods considered in this study. The monitoring performances are reported in Table 4.21.

Table 4.21 *Dataset 4: comparison of the performance indicators of the monitoring for both methods used*

Performance indicator	Method	Value	Units
TPR	Assumption-free model	70	% of batches
	Standard model	62.1	
FPR	Assumption-free model	0	% of batches
	Standard model	30	
ARL	Assumption-free model	155	Samples
	Standard model	400	

From the results of both methos, it can be noticed how they have the same capability of correctly classify a normal batch (even though the assumption-free modelling performed slightly better). Moreover, the capability of the assumption-free modelling of dealing with faulty batches is superior with respect to the one of a monitoring performed using a batch-wise unfolding PCA. Indeed, for the assumption-free modelling $FPR = 0\%$ while for the batch-wise unfolding PCA $FPR = 30\%$. In terms of detection speed, the assumption-free modelling outperforms the batch-wise unfolding PCA, indeed the first one, can detect a fault 61.2% faster than the second method. As a matter of fact, the assumption-free modelling detects on average a fault after about 39 hours, while the other method after 100 hours.

From the results obtained, it can be noticed that the assumption-free model has better performances. Indeed, not only it has a *FPR* which is 0% compared to the 30% of the batch-wise unfolding PCA, but it also has an *ARL* significantly lower than the other method. Therefore, for this dataset the assumption-free monitoring outperformed the standard MPCA model.

4.5 Dataset 5

This dataset contains the data coming from an industrial herbicide drying process. The process was described by García-Muñoz et al. (2003) and presented in §2.5. The data have been retrieved from the Aspen ProMV getting started guide (2017). Ten variables are measured and the batches are not aligned and have a mean of 129 samples per batch. Twenty five NOC batches are used to calibrate the model, while 41 batches (of which 38 are faulty) are used to assess the monitoring performance of the model.

4.5.1 Dataset 5: assumption-free modelling calibration

The matrix containing the data is unfolded in the batch direction, autoscaled and then used to build the PCA model. Table 4.22 shows the results of the PCA.

Table 4.22 *Dataset 5: Summary of the PCA model*

PC No.	λ_a	R^2	$R^2_{cum.}$	*RMSECV*
1	5.77	57.7%	57.7 %	0.7
2	1.17	11.7%	69.4 %	0.71
3	0.97	9.7%	79.1 %	0.75
4	0.8	8%	87.1 %	0.79
5	0.52	5.2%	92.3 %	0.96

The chosen number of PCs is 2 due to the grid search algorithm and the total variance explained is 69.4%. The minimum of *RMSECV* is between 1 and 2 PCs therefore the number of chosen PCs is the best for this dataset. The scores obtained from the PCA are plotted and shown in Figure 4.49.

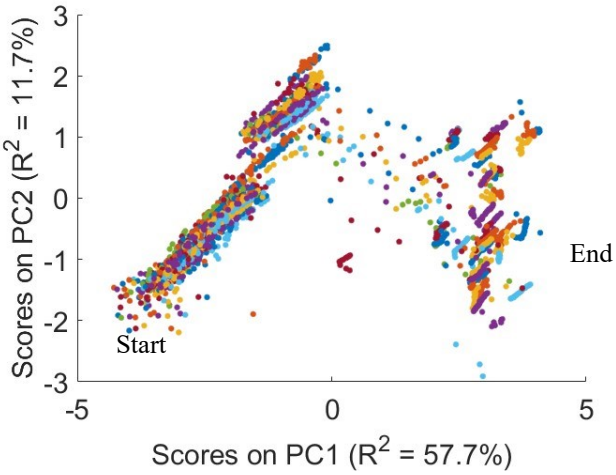


Figure 4.49 *Dataset 5: score plot obtained after variable-wise unfolding the calibration set of Dataset 1 described in §2.5. Each point represents a time instant of a single batch. Each colour represents a batch of the calibration set.*

The scores do not evolve linearly with the progression of the process. Indeed, their density is higher at the beginning of the process and lower towards the end. Moreover, there is a lower variability in the first part of the process which increases as the process progresses.

From the scores, the common trajectory is more evident until the scores on the first principal component reaches the value of 0, after that it is not easy to imagine the evolution of the common trajectory.

The assumption-free modelling is calibrated setting $\beta = 0.99$ and $\gamma = 0.6$. These choices are done for two main reasons. The first one regards the need of including as many scores as possible inside the valid cells in order to exploit all of them, hence the choice of β. The second one, regards the choice of γ. This parameter is set lower with respect to the other datasets in order to have a better description of the common trajectory; indeed, by looking at the scores there is no clear evidence of a common path of the scores. This allows the grid search algorithm to consider more valid cells with respect to the case of the parameter set to 0.95. The settings used to calibrate the model are summarised in Table 4.23.

Table 4.23 *Dataset 5: Hyperparameter values used during the calibration of the assumption-free model..*

Hyperparameter	Value	Units
γ	0.6	Fraction of scores
β	0.99	Fraction of batches
n_{PC1}^{max}	12	Cells
n_{PC2}^{max}	12	Cells

The best grid identified by the algorithm has 8 valid cells and a configuration of 9×1 which includes 99.3% of all the scores. The grid is shown in Figure 4.50.

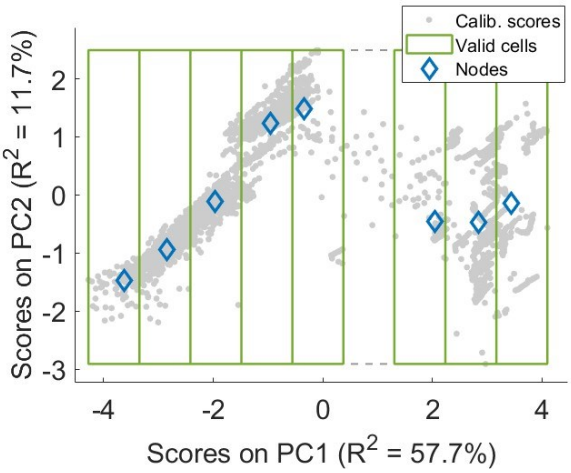


Figure 4.50 *Dataset 5: best grid found by the algorithm for the calibration batches.* ($n_{PC1} = 3$ *and* $n_{PC2} = 8$).

The evolution of the scores is well described by the first 5 nodes. However, the last 3 nodes do not describe the evolution of the process as the scores are too scattered. It is expected that in order to compensate for the high variability of the process towards the end, the control limits will be very wide in order to include as many scores as possible. The nodes are chronologically ordered and interpolated in order to reconstruct the common trajectory. The result is shown in Figure 4.51.

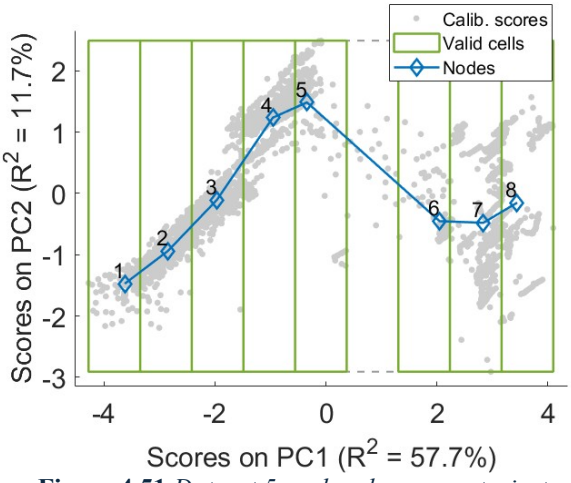


Figure 4.51 *Dataset 5: ordered common trajectory*

The last three nodes are placed where the scores are denser, therefore they are placed closer to the bottom of the cells. Once the common trajectory has been built, it is possible to evaluate the

relative time for all the batches in the calibration set. The relative time is evaluated as explained in §3.5 and the results are shown in Figure 4.52.

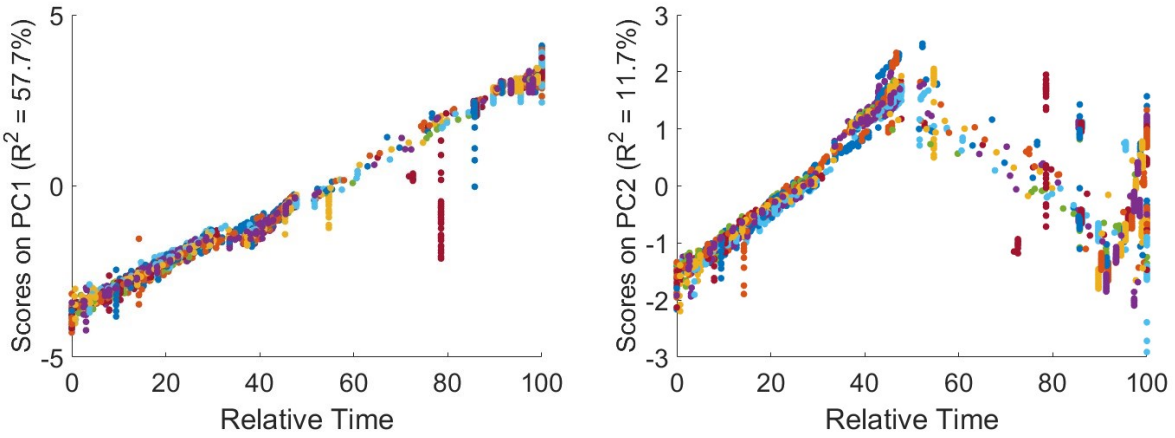


Figure 4.52 *Dataset 5: Relative time of the calibration batches of (a) PC1 and (b) PC2.*

The scores evolve more evenly with the relative time on the first principal component. Indeed, after a relative time equal to 50, the scores on the second principal component do not show a uniform pattern. This is different with respect to what has been seen with the other datasets. The reason may rely in the process itself, indeed, as described in §2.5 the process is divided into different steps. As mentioned by Chamaco et al. (2008) a process made of different steps may be modelled by applying a different model to each processing step. Indeed, by using a variable-wise unfolding PCA a constant correlation is imposed, and the batch dynamics are suppressed.

Once the relative time has been evaluated, the control limit can be evaluated from the distribution of the distances of the means of the batches from the common trajectory. A limit is evaluated from the inverse of a normal distribution with a 95% confidence level for each valid cell. The control limits are shown in Figure 4.53.

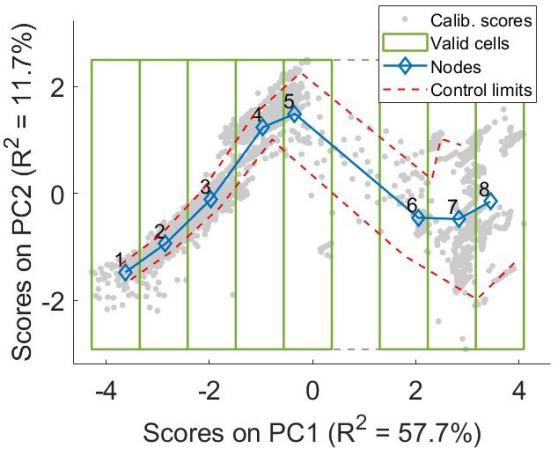


Figure 4.53 *Dataset 5: reconstructed common trajectory and control limits around it for the calibration batches*

The control limits are able to fairly represent the variability in the first 5 valid cells. As expected, in the last 3 valid cells the control limits are wider in order to account for the increased

variability. The percentage of means of batches inside the control limit is 17.5%. The amount of points out of the control limit is higher than the expected 5%, however it is still a reasonable amount considering that not all the cells have a normal distribution of the distances of the means of the batches from the common trajectory.

The control chart on the *SPE* is built by considering the empirical distribution of the *SPE* in each valid cell with a confidence level of 95%. The built control chart is shown in Figure 4.54.

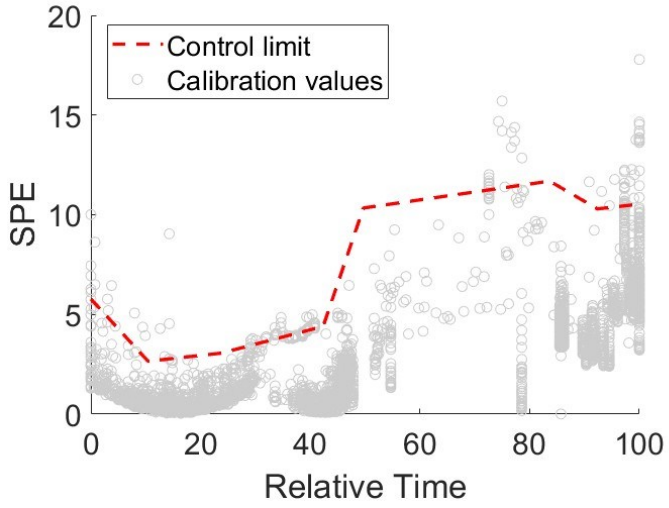


Figure 4.54 *Dataset 5: SPE control chart with the SPE of the calibration scores.*

The *SPE* shows an evolution similar to the scores. Indeed, until $r_t = 50$ the statistic is dense and compact, after half of the relative time the points shows a greater variability.

Despite the evolution of the *SPE*, the percentage of points out of the control limit is 5.1% which is coherent with the chosen confidence level.

In order to perform process monitoring, there is the need to evaluate C_D^{max} and C_{SPE}^{max} from the calibration set. These values are evaluated in accordance to what explained in §3.8. The number of consecutive points out of the control limits for all the batches in each chart is shown in Figure 4.55.

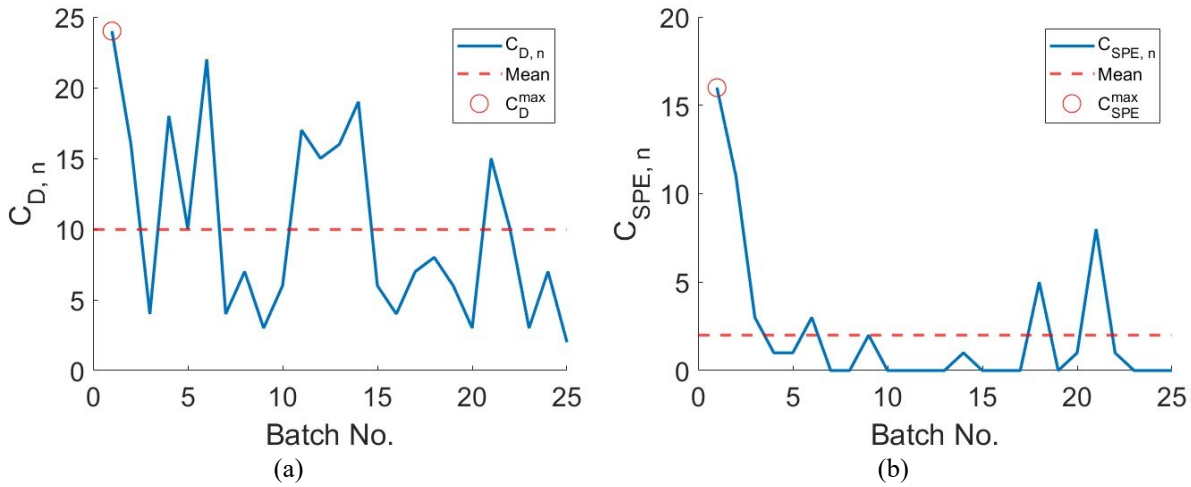


Figure 4.55 *Dataset 5: calibration set. Maximum number of consecutive points out of (a) the control limits around the common trajectory and (b) the control limit of the SPE.*

In both control charts, the batch where the maximum of consecutive point out of the control limit was found is the first. Indeed $C_D^{max} = 24$, while $C_{SPE}^{max} = 16$.

4.5.2 Dataset 5: monitoring using the assumption-free model

After the model has been calibrated, it is possible to assess its monitoring performances using the validation set. It contains 41 batches, 3 of which are NOC, the remaining ones are faulty. Each sample is autoscaled and projected into the score plot, and its relative time is estimated along with the *SPE*. This procedure is repeated for all the samples in a batch run and for all the batches in the dataset. The results of the monitoring are shown in Figure 4.56.

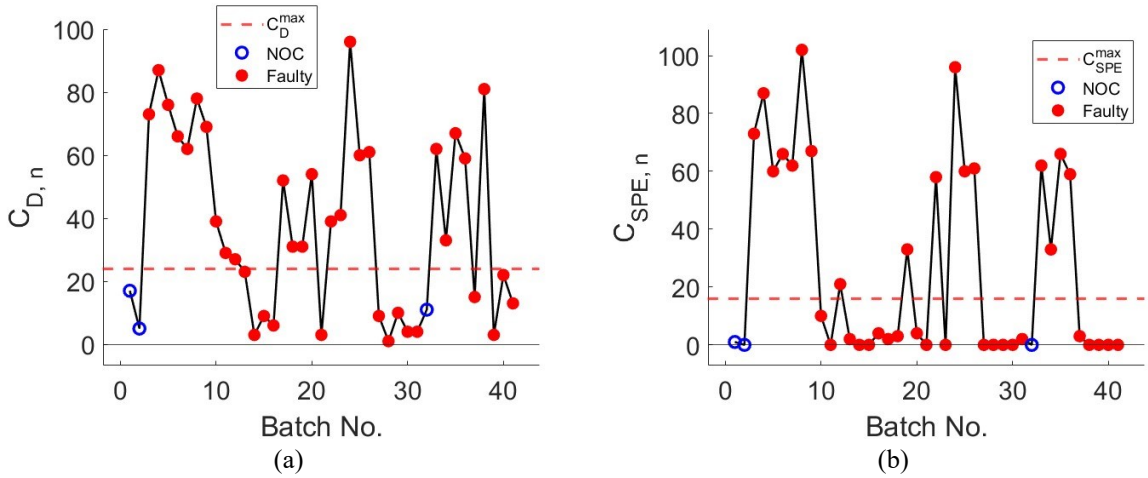


Figure 4.56 *Dataset 5: monitoring of the validation set. Maximum number of consecutive points outside the control limit on (a) the score plot and on (b) the SPE The truly faulty batches are indicated in red.*

All the NOC batches have been correctly identified by the model. However, some faulty batches have been misclassified. The faulty batches that were considered normal did not trigger an alarm on any control chart. No batches triggered an alarm only when considering the *SPE* control chart.

Validation batch No. 10 was classified as faulty due to the control limits around the common trajectory. Figure 4.57 shows the control charts for this batch.

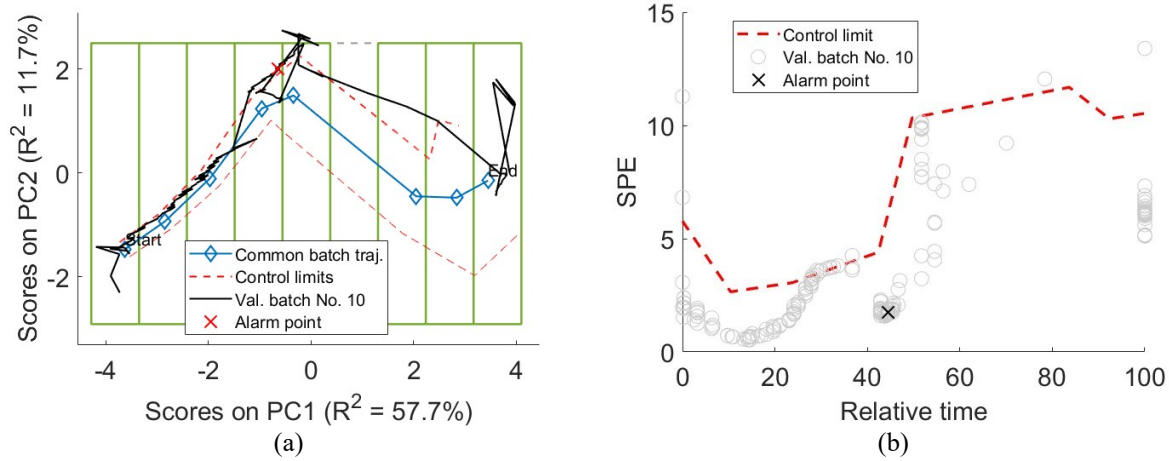


Figure 4.57 *Dataset 5: Validation batch No. 10. (a) Control chart on the score plot and (b) control chart on the SPE.*

Indeed, the batch does not show any unusual behaviour from the *SPE* control chart. However, the scores went out from the control limits between the third and the fourth node for a number of points greater than C_D^{max}. The sample that triggered an alarm was the 105th (174 samples in total). It is not known when the disturbance that caused the departure from normality occurred, however, the fault is identified halfway through the process.

Validation batch No. 15 triggered an alarm on both control charts, the results of the monitoring performed on this batch are shown in Figure 4.58.

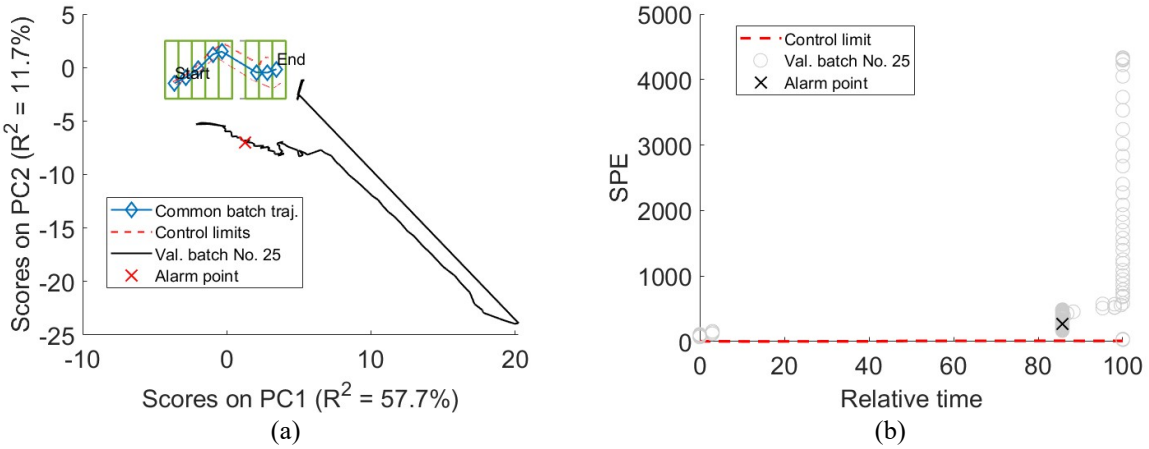


Figure 4.58 *Dataset 5: Validation batch No. 25. (a) Control chart on the score plot and (b) control chart on the SPE.*

It is immediately noticeable how in both control charts the batch exhibits an abnormal behaviour. Indeed, from the control chart on the score plot the batch is not even inside the grid limits and shows a completely different path with respect to the NOC batches. Moreover, from the control chart on the *SPE* the batch is outside the control limit from the beginning, showing a different relation between variables which become grater towards the end of the process. The sample that triggered the alarm was the 21st and the alarm has been triggered by the *SPE* control chart.

Validation batch No. 15 is a faulty batch which has not been identified by the model. The control charts are shown in Figure 4.59.

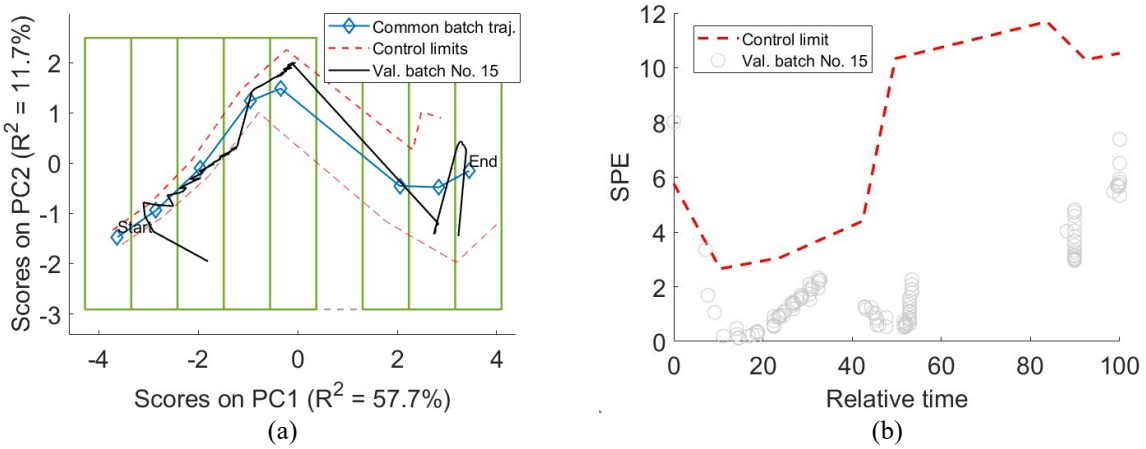


Figure 4.59 *Dataset 5: Validation batch No. 15. (a) Control chart on the score plot and (b) control chart on the SPE.*

Indeed, the batch does not show any difference with respect to the NOC batches. As a matter of fact, except for few points at the beginning of the processing, the scores are always inside the control limits around the common trajectory. Moreover, the *SPE* are all inside the control limits. From the analysis of the scores and of the SPE, it was not possible to detect any fault for this batch.
After all the batches in the validation set have been monitored, the performance indicators are evaluated (Table 4.24)

Table 4.24 *Dataset 5: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	63.2	% of batches
FPR	0	% of batches
ARL	41	Samples

All the NOC batches have been correctly identified, therefore the $FPR = 0\%$. About two thirds of the faulty batches have been classified as such, indeed the evaluated $TPR = 63.2\%$. In order to evaluate the *ARL*, all the faulty batches have been considered as they were upset from the beginning of the process, therefore its value is 41 samples.

4.5.3 Dataset 5: monitoring using a standard MPCA method

The data are not aligned, therefore, the alignment is required before unfolding the matrix in the variable direction. To perform the alignment both DTW according to Kassidas et al. (1998) and to Ramaker et al. (2003) are carried out. To perform the final alignment, the geometric mean of the weights of both DTW is considered as suggested by Gonzalez-Martinez et al. (2011). Once the data are aligned the matrix is batch-wise unfolded and autoscaled. A PCA model is built on the unfolded matrix. Sartori (2023), chose 5 PCs to build the PCA model. This led to 63.2% of total variance captured.
The scores obtained by the PCA are plotted and the score plot with the 95% confidence limit is shown in Figure 4.60.

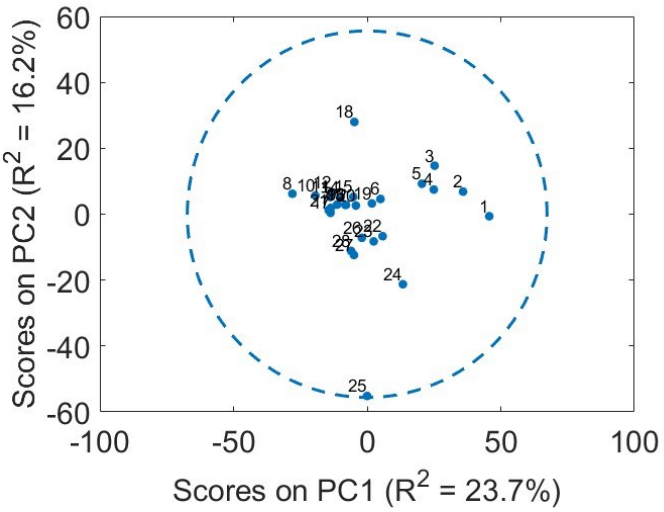


Figure 4.60 *Dataset 5: score plot obtained for a standard MPCA model. Each point represents a batch, the dotted line is the 95% confidence interval on the multivariate distribution of the scores.*

From the score plot of the first 2 PCs no batches are confidence limit. However, batch No.25 is close to the limit.

The control chart on the *SPE* is built according to §1.2.1 using a 95% confidence level. The result is shown in Figure 4.61.

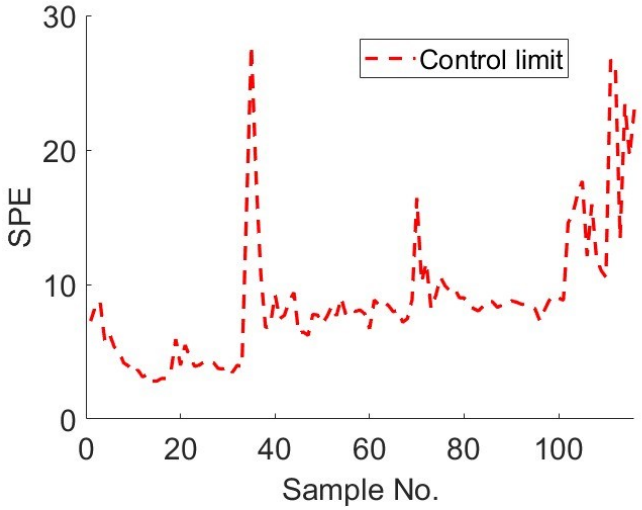


Figure 4.61 *Dataset 5: SPE control chart built on a batch-wise unfolding MPCA obtained using (1.13).*

The percentage of points out of the control limit is 5.5%. This value is coherent with the choice done on the confidence level.

In order to perform process monitoring Sartori (2023) set the maximum number of consecutive points out of the control limit on T^2 equal to 5, while the corresponding value on the *SPE* was set to 21. The monitoring was carried out on all the batches of the validation set and the results are reported in Table 4.25.

Table 4.25 *Dataset 5: performance indicators of the monitoring scheme.*

Performance indicators	Value	Units
TPR	60.5	% of batches
FPR	33.3	% of batches
ARL	94	Samples

Not all the faulty batches have been identified by the model, indeed $TPR = 60.5\%$. Meaning that only 2 faulty batches out of 3 trigger an alarm. On the other hand, when dealing with NOC batches the $FPR = 33.3\%$, which indicates that one third of the normal batches is classified as faulty. $ARL = 94$ samples, therefore a fault is identified on average close to the end of the batch run considering that the average number of samples in the calibration set is 129.

4.5.4 Dataset 5: comparison of the results

The monitoring has been carried out on Dataset 5 with both the methods considered in this study. The monitoring performances are reported in Table 4.26.

Table 4.26 *Dataset 5: comparison of the performance indicators of the monitoring for both methods used*

Performance indicator	Method	Value	Units
TPR	Assumption-free model	63.2	% of batches
	Standard model	60.5	
FPR	Assumption-free model	0	% of batches
	Standard model	33.3	
ARL	Assumption-free model	41	Samples
	Standard model	94	

From the results of both methods, it can be noticed how they have the same capability of correctly classify a normal batch (even though the assumption-free modelling performed slightly better). Moreover, the capability of the assumption-free modelling of dealing with faulty batches is superior with respect to the one of a monitoring performed using a batch-wise unfolding PCA. Indeed, for the assumption-free modelling $FPR = 0\%$ while for the batch-wise unfolding PCA $FPR = 33.3\%$. In terms of detection speed, the assumption-free modelling outperforms the batch-wise unfolding PCA, indeed the first one, can detect a fault in half of the time that is required by the second.

From the results obtained, it can be noticed that the assumption-free model has better performances. Indeed, not only has a *FPR* which is 0% compared to the 33.3% of the batch-wise unfolding PCA, but also has an *ARL* significantly lower than the other method. Therefore, for this dataset the assumption-free monitoring outperformed the batch-wise unfolding PCA.

Conclusions

The objective of this thesis was to further improve what had been already carried out during previous studies (Fracassetto, 2022; Sartori, 2023) on the implementation of the assumption-free modelling for batch process monitoring proposed by Westad et al. (2015).

In this study, detailed guidelines on how to design and use an assumption-free monitoring model for batch process monitoring have been given. Each step of the calibration of the assumption-free model has been analysed in detail in order to assist the implementation of such methodology. Moreover, the assumptions made in previous studies on this topic have been verified. Particularly, the assumption of a normal distribution of the distances of the means of the batches in the valid cells from the common trajectory was confirmed. This verification was carried out by creating a criterion for assigning a sign to the distances by closing the common trajectory by joining the first and the last node. After the signs have been given, an Anderson-Darling test (Anderson and Darling, 1952) is performed on the population of each cell in order to assess whether or not the distribution is normal. The outcome is that the percentage of cells with a normal distribution depends on the dataset. However, distances are normally or close-to-normally distributed in many cells, and the choice of evaluating the control limit from the inverse of a normal distribution is justified.

Furthermore, by analysing the distribution of the residuals in a valid cell, a new methodology has been developed for estimating the control limit of the squared prediction error statistics. This methodology involves the evaluation of a non-parametric distribution of the *SPE*, and a 95% confidence limit is selected from the obtained cumulative density function.

Additionally, an in-depth description on how to calibrate the alarms on both control charts has been given. Indeed, the number of points needed to trigger an alarm for each control chart is calculated by considering the maximum number of consecutive points out of the control limit among the calibration batches.

In order to evaluate the monitoring performances of the monitoring model, five different datasets, coming both from simulated and industrial process, have been used. The datasets contained both aligned and non-aligned data in order to judge the capability of the model in dealing with both types of data. The datasets of the latter type are of particular importance for two main reasons. Firstly, because batches coming from the same process do not have the same duration and therefore real data are often not aligned. Secondly, because alignment techniques are known to generate artifacts when some batches are significantly shorter than the reference batch; furthermore they are also computationally demanding.

In order to assess the capability of an assumption-free model in monitoring a batch process, the results are compared to the ones obtained by Sartori (2023) after applying a standard multi-way

principal component analysis model to the same datasets of this study. The monitoring performances are compared in terms of detection strength and detection speed. The former judges the capability of the model in correctly classifying a batch, while the latter evaluates how fast a fault is detected after its occurrence. The performances indicators have been evaluated for all the datasets and using both methods.

The result of the study was that the performances of the two methods are comparable for the aligned datasets (Dataset 1 and Dataset 2). Indeed, both methodologies are able to correctly identify all the faulty batches. Dealing with normal batches, in Dataset 2 the same result has been obtained using the two methods. However, using Dataset 1, the assumption-free model performed slightly better than the standard MPCA model. The detection speed is equivalent in both methods when considering these aligned datasets. Therefore, no evident benefits arise in the use of either of the two methods when dealing with this type of data.

From the obtained results with non-aligned data (Dataset 3, Dataset 4 and Dataset 5), the retrieved conclusions are substantially different. The assumption-free model outperformed the other method for all datasets. In all the non-aligned datasets, the assumption-free model resulted in greater or equal true positive rate, and therefore in a higher sensibility on faulty batches. Moreover, the false positive rate of the assumption-free modelling is always 0%, showing a perfect capability of identifying normal operating conditions batches. Finally, in terms of detection speed, with the exception of Dataset 3 where values are comparable, the assumption-free model detects a fault much faster than the monitoring performed after batch-wise unfolding the data matrix.

Therefore, the assumption-free model performs better with respect to the standard method used in this study. Moreover, it is easier to calibrate and to implement when the data are not aligned. As a matter of fact, an assumption-free model can immediately use a measurement taken in an industrial plant without the need for any kind of alignment.

However, there are some limitations which arise when dealing with a process that involves different processing steps. This behaviour has been appreciated in Dataset 4 and Dataset 5 where the scores are not uniform in the score plot and therefore building the common trajectory may result difficult. This problem could be solved by increasing the number of PCs and by adjusting the grid search algorithm and all the steps of the calibration of the assumption-free modelling to accommodate for the additional dimensions included. This possibility has also been highlighted by Westad et al. (2015). Adding further dimensions may allow to capture more variability leading to better monitoring performances. Indeed, the included additional variability may avoid the situation encountered in Dataset 4, where the *SPE* identified only one of the two types of faults present in the dataset.

In conclusion, the assumption-free modelling strategy proved to be a more reliable, sound and easier-to-calibrate method with respect to the standard one, it being understood that further improvements are still possible.

Nomenclature

a	=	a principal component
A	=	number of principal components
ARL	=	average run length
C_D^{max}	=	highest number of consecutive points out of the confidence limit of the common trajectory among the calibration batches
$C_{D,n}$	=	maximum number of consecutive points out of the control limit around the common trajectory of batch n
$\mathbf{C}_D$	=	vector containing all the $C_{D,n}$
C_{SPE}^{max}	=	highest number of consecutive points out of the confidence limit of the *SPE* among the calibration batches
$C_{SPE,n}$	=	maximum number of consecutive points out of the control limit on the *SPE* of batch n
$\mathbf{C}_{SPE}$	=	vector containing all the $C_{SPE,n}$
$d_{n,u,\perp}$	=	distance between $t_{a,u,\perp}$ and $\bar{t}_{a,n,u}$
$d_{n,u}$	=	distance between $\bar{t}_{a,u}$ and $\bar{t}_{a,n,u}$
$d_u^{c.i.}$	=	distance of the control limit in the valid cell u from the common trajectory
$\mathbf{e_n}$	=	row of the residual matrix
$\mathbf{e}_{n,k}$	=	residuals of the n batch from sample 1 to sample k
$\mathbf{e}_{new}$	=	residuals of the new observation
$\mathbf{E}$	=	residual matrix
$\mathbf{E}_k$	=	residual matrix from sample 1 to sample k
$\mathbf{E_u}$	=	matrix of the residuals of the scores contained in valid cell u
$F_{A,N-1,\alpha}$	=	fisher distribution
FN	=	false negative
FP	=	false positive
FPR	=	false positive rate
i	=	number of points of the common trajectory before the point of the common trajectory considered
k	=	sample
K	=	number of samples
$l_{PC1,w}$	=	dimension of the cell on PC1 at iteration w
$l_{\mathrm{PC2},w}$	=	dimension of the cell on PC2 at iteration w
L	=	number of samples not included into the model

M	=	total number of score in valid cell u
m_{PC1}	=	lower limit of the grid on PC1
M_{PC1}	=	higher limit of the grid on PC1
m_{PC2}	=	lower limit of the grid on PC2
M_{PC1}	=	higher limit of the grid on PC2
n_{int}	=	points into which the common trajectory is divided
n_{valid}^{max}	=	highest number of valid cells
$n_{PC1,w}$	=	number of cells in the direction of PC1 at iteration w
$n_{PC2,w}$	=	number of cells in the direction of PC2 at iteration w
n_{PC1}^{max}	=	maximum number of cells in the direction of PC1
n_{PC2}^{max}	=	maximum number of cells in the direction of PC2
N	=	number of batches
$\mathbf{p}_{\mathrm{a}}$	=	loading. eigenvector associated to the eigenvalue λ_a
$\mathbf{P}$	=	loading matrix
$\mathbf{P}_k^{\mathbf{T}}$	=	loading matrix truncated to sample k
r_t	=	relative time
$r_{t,i}$	=	relative time of the i point of the common trajectory
$RMSECV$	=	root mean square error of cross-validation
$RMSECV_a$	=	root mean square error of cross-validation related to the a principal component
$\mathbf{s}_{k,u}$	=	vector of the sampling number of batch k in valid cell u
$s_{k,u}^{max}$	=	maximum sampling number of batch k in valid cell u
$\bar{s}_u$	=	mean of the maximum sampling number in valid cell u
$\mathbf{s}_u^{max}$	=	vector containing $s_{k,u}^{max}$ of the batches present in valid cell u
SPE	=	square prediction error
SPE_{lim}	=	statistical limit on the SPE
$SPE_{lim,k}$	=	statistical limit on the SPE of time instant k
SPE_u^{lim}	=	statistical limit on the SPE of time instant of the valid cell u
SPE_n	=	SPE of the n observation
$SPE_{n,k}$	=	SPE of the n batch from sample 1 to sample k
SPE_{new}	=	SPE of the new observation
$\mathbf{SPE}_u$	=	vector of the SPE of the scores contained in valid cell u
$t_{a,m}$	=	score on the a PC in valid cell u
$t_{a,m,n}$	=	score on the a PC in valid cell u of batch n
$\bar{t}_{a,n,u}$	=	mean of the score on the a PC in the valid cell u of batch n
$\bar{t}_{a,u}$	=	mean of the score on the a PC in the valid cell u
$t_{a,u,\perp}$	=	projection of a $\bar{t}_{a,n,u}$ onto the common trajectory
$\mathbf{t}_{\mathrm{n}}$	=	row of the score matrix

$\mathbf{t}_{new}$	=	score of the new observation
$\mathbf{t}_{n,k}$	=	score of the batch *n* from sample 1 to sample *k*
$\mathbf{t}_{PC1}$	=	scores on PC1
$\mathbf{t}_{PC2}$	=	scores on PC2
$\mathbf{T}$	=	score matrix
T^2	=	hotelling T^2statistic
T_n^2	=	hotelling T^2statistic of the *n* sample
$T_{n,k}^2$	=	hotelling T^2statistic of the *n* batch of the sample *k*
T_{lim}^2	=	statistical limit on the hotelling T^2statistic
TN	=	true negative
TP	=	true positive
TPR	=	true positive rate
u	=	valid cell
V	=	number of variables
w	=	iteration of the grid search algorithm
$\mathbf{X}$	=	2D matrix containing the data
$\widehat{\mathbf{X}}$	=	reconstructed matrix of the original data
$\boldsymbol{X_{3D}}$	=	3D array containing the data
$\widehat{\mathbf{X}}_k$	=	reconstructed value of the original variables from sample 1 to sample *k*
$\mathbf{X}_k$	=	data matrix from sample 1 to sample *k*
$\mathbf{x}_{n,k}$	=	vector of the data of batch *n* from sample 1 to sample *k*
$\mathbf{x}_{new}$	=	new observation measured
$\widehat{\boldsymbol{x}}_{new}$	=	reconstructed value of $\boldsymbol{x}_{new}$
$\hat{y}_l$	=	predictions for samples that are not included in model formulation
y_l	=	real samples not included in model formulation
Z_α	=	standard normal distribution

Greek letters

α	=	confidence level
β	=	fraction of batches that need to be present in a cell in order to consider it valid
γ	=	fraction of the total scores that need to be present in all valid cells in order to consider the grid valid
λ_a	=	eigenvector of the *a PC*
$\boldsymbol{\Lambda^{-1}}$	=	inverse of the diagonal matrix of the eigenvalues
μ	=	mean

μ_k	=	mean at sample k
μ_u	=	mean in valid cell u
σ^2	=	standard deviation
${\sigma_k}^2$	=	standard deviation at sample k
σ_u	=	standard deviation in valid cell u

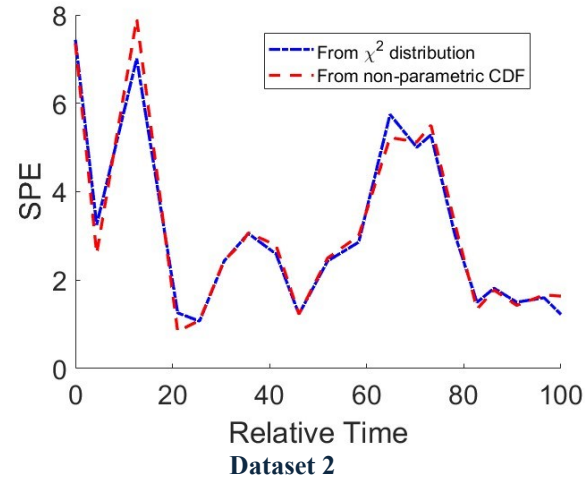
Acronyms

BWU	=	batch-wise unfolding
CDF	=	cumulative density function
DTW	=	dynamic time warping
MPCA	=	multi-way principal component analysis
NOC	=	normal operation condition
PC	=	principal component
PCA	=	principal component analysis
RGTW	=	relaxed greedy time warping
SPM	=	statistical process monitoring
VWU	=	variable-wise unfolding

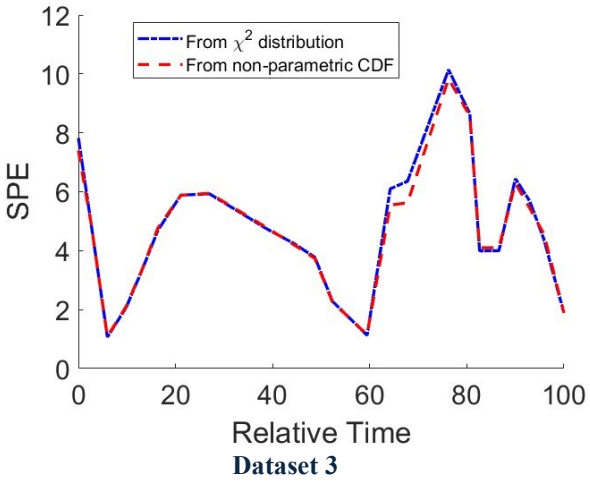
Appendix

In this appendix the differences between the control charts on the *SPE* obtained from the non-parametric distribution and the χ^2 distribution are shown. The first approach is described in §3.7.2, while the second one is described in §3.7.

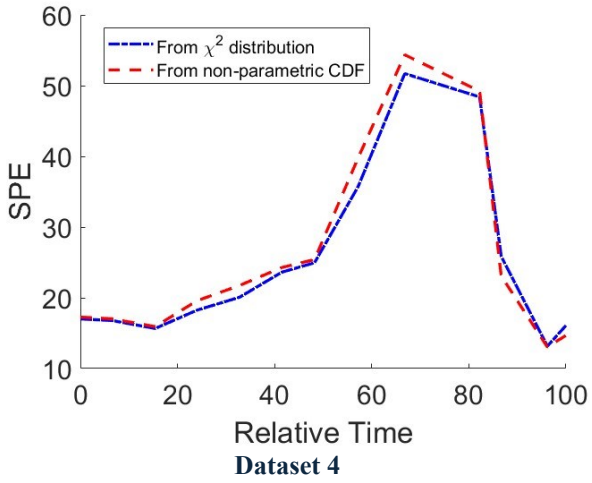
A.1. Comparison between control charts



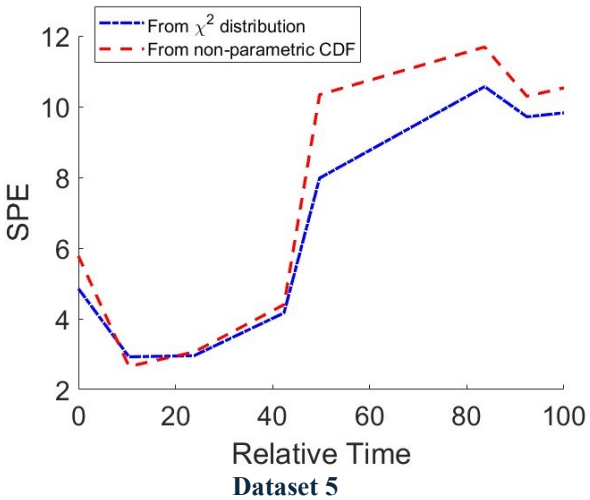
Dataset 2



Dataset 3



Dataset 4



Dataset 5

References

1. Anderson, T. W., Darling, D. A., 1954. A test of goodness of fit. *J. Am. Stat. Assoc.* **49**, 765-769.
2. *Aspen ProMV Getting Started Guide* (2017). Aspen Technology. p.31.
3. Birol, G., Ündey, C., Çinar, A., (2002). A modular simulation package for fed-batch fermentation: Penicillin production. *Comput. Chem. Eng*. **26**, 1553–1565.
4. 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. *Makromol. Chem. Suppl.*, **10**, 105–128.
5. Camacho, J., Picó, J., & Ferrer, A. (2008). Bilinear modelling of batch processes. Part I: Theoretical discussion. *J. Chemometr.*, **22**, 299–308.
6. 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? *Anal. Chim. Acta*, **642**, 59–68.
7. Fracassetto, A. (2022). Batch process monitoring using an assumption-free modeling methodology. *Tesi di Laurea Magistrale in Ingegneria Chimica e dei Processi Industriali*, DII, Università di Padova.
8. García-Muñoz, S., Kourti, T., MacGregor, J. F., Mateos, A. G., & Murphy, G. (2003). Troubleshooting of an industrial batch process using multivariate methods. *Ind. Eng. Chem. Res.*, **42**, 3592–3601.
9. González-Martínez, J.M., Ferrer, A., Westerhuis, J.A., (2011). Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping. *Chemom. Intell. Lab. Sys.*, **105**, 195-206.
10. González-Martínez, J. M., Camacho, J., & Ferrer, A. (2018). MVBatch: A matlab toolbox for batch process modeling and monitoring. *Chemom. Intell. Lab. Sys.*, **183**, 122–133.
11. Kassidas, A., MacGregor, J.F., Taylor, P.A., (1998). Synchronization of batch trajectories using dynamic time warping. *AIChE J.,* **44**, 864–875.
12. Kosanovich, K. A., Dahl, K. S., & Piovoso, M. J. (1996). Improved process understanding using multiway principal component analysis. *Ind. Eng. Chem.*, **35**, 138–146.
13. Kourti T. (2003). Multivariate dynamic data modelling for analysis and statistical process control of batch processes, start-ups and grade transitions. *J. Chemom.*, **17**, 93-109.
14. Lei, F., Rotboll, M., & Jorgensen, S. B. (2001). A biochemically structured model for Saccharomyces cerevisiae. *J. Biotechnol.*, **88**, 205–221.
15. Nomikos, P., & MacGregor, J. F. (1994). Monitoring batch processes using multiway principal component analysis. *AIChE J.*, **40**, 1361–1375.

16. Nomikos, P., & Macgregor, J. F. (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics*, **37(1)**, 41–59.
17. Ramaker, H. J., van Sprang, E. N. M., Westerhuis, J. A., Smilde, A. K. (2003), Dynamic time warping of spectroscopic BATCH data. *Anal. Chim. Acta*, **498**, 133–153.
18. Rato, T. J., Rendall, R., Gomes, V., Chin, S.T., Chiang, L.H., Saraiva, P.M., Reis, M.S., (2016). A Systematic Methodology for Comparing Batch Process Monitoring Methods: Part I-Assessing Detection Strength. *Ind. Eng. Chem. Res*. **55**, 5342–5358.
19. Sartori, F., Facco, P., Zuecco, F., Bezzo, F., Barolo, M. (2023). Optimal indicator-variable approach for trajectory synchronization in uneven-length multiphase batch processes. *Ind. Eng. Chem. Res.*, **62**, 18511-18525.
20. Sartori, F. (2023). Effective implementation of Industry 4.0 approaches for process monitoring in the batch manufacturing of specialty chemicals. *Ph.D. Thesis*, Università di Padova.
21. Sun, W., Meng, Y., Palazoglu, A., Zhao, J., Zhang, H., Zhang, J., (2011). A method for multiphase batch process monitoring based on auto phase identification. *J. Process Contr.* **21**, 627–638.
22. Westad, F., Gidskehaug, L., Swarbrick, B., & Flåten, G. R. (2015). Assumption free modeling and monitoring of batch processes. *Chemom. Intell. Lab. Sys.*, **149**, 66–72.
23. Wise B., Gallagher N. B. (1996). The process chemometrics approach to process monitoring and fault detection. *J. Process Contr.*, **6**, 329-348.
24. https://github.com/jogonmar/MVBatch/releases (Latest access 29/07/2024)