Pseudo-sample based contribution plots: innovative tools for fault diagnosis in kernel-based batch process monitoring

Raffaele Vitale[a,*], Onno E. de Noord[b], Alberto Ferrer[a]

[a] *Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain*

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

Abstract

This article explores the potential of kernel-based methods for fault diagnosis in batch process monitoring by combining Kernel-Principal Component Analysis and three common techniques which permit to analyze batch data by means of bilinear models: variable-wise unfolding, batch-wise unfolding and landmark feature extraction. Gower's idea of pseudo-sample projection is exploited to develop novel tools, the pseudo-sample based contribution plots, for diagnostic purposes. The results show that, when the datasets under study are affected by severe non-linearities, the proposed approach performs better than classical ones.

Keywords: kernel-based techniques, batch process monitoring, pseudo-sample projection, contribution plots, fault detection, fault diagnosis

1. Introduction

Nowadays, most of the manufacturing industries in the world perform batch processes in their plants. To guarantee and preserve high quality of the final products and to minimize the number of off-specification process runs, batch monitoring schemes are designed so that faults and failures might be quickly, easily and efficiently recognized and their possible root causes might be correctly identified. These two phases of process monitoring are also known as fault detection and fault diagnosis, respectively. After that, specific countermeasures can be adopted for restoring the possible faulty situation to normal operating conditions (NOC).

The aforementioned monitoring schemes are usually constructed resorting to particular data-driven approaches [1–3]. In fact, during a batch run, j=1, 2, ..., J process variables are commonly measured at t=1, 2, ..., T time points. Data collected for i=1, 2, ..., I batches are then arranged in a three-way array (I×J×T). Although many methods exist for directly analyzing data with this structure, the most widely used technique to extract exploitable information from them consists in

[*] Corresponding author:
Telephone number: +34684099819
Email address: rvitale86@gmail.com (Raffaele Vitale)

unfolding this three-way array into a matrix and then fitting a bilinear model by means of Principal Component Analysis (PCA) or Partial Least Squares regression (PLS) [1]. In general, this unfolding procedure is executed by variable-wise unfolding (VWU), batch-wise unfolding (BWU) or landmark feature extraction (LFE). The first separates the single sub-matrices associated to the evolution of each batch and rearranges them preserving the variable direction leading to a new array with dimensions $IT \times J$. This approach does not take into account the dynamics of the process under study and has been found to be valid only when its correlation structure is more or less constant along the batch run [1]. The second unfolds the original three-way structure so that all the sub-matrices, each one related to the evolution of a single variable in all the batches, are organized side by side (the final two-way array has dimensions $I \times JT$). It then permits to model the linear dynamics of the analyzed process and capture changing relationships among process variables. By the third, F landmark features of the evolution of each batch are defined and organized in a new matrix with dimensions $I \times F$. A good survey of these techniques can be found in [4]. However, when resorting to PCA or PLS for dealing with non-linear process datasets (defined as those characterised by measured variables that are linked by non-linear relationships), their analysis and interpretation may be seriously jeopardized, since both assume their underlying structure is linear [5]. In the last decades, many novel methods have been proposed to handle these situations such as non-linear PLS [6–8] or artificial neural networks [9]. Nevertheless, such approaches often require the optimization of many adjustable parameters and may show overfitting and local minima. A good alternative is represented by the so-called kernel-based techniques [10], which also comprehend support vector machines [11] and have already been broadly used in chemistry [12, 13], biology [14], informatics [15, 16], continuous process monitoring [17, 18] and on-/off-specification batch run discrimination [19]. The first aim of this article is to explore their potential to design off-line monitoring schemes for batch processes, whose complex nature usually results in strong non-linearities, difficult to model if the physical and/or chemical phenomena generating them are not completely known.

Unfortunately, these techniques suffer from a severe drawback: the information about the weights or the loadings of the original variables is lost. This makes classical diagnostic tools, like contribution plots [20], unfeasible. Many possibilities to recover this information exist, but authors commonly abstain from resorting to them for three reasons: i) their implementation and the interpretation of the final results is not straightforward; ii) most of them do not permit to graphically visualize the relation between original variables and final models; iii) when they are used for continuous process monitoring, their application is always based on the comparison of the possible detected faults with a database of previously observed ones, which is rarely available when dealing with real industrial data [13, 21–24]. Recently, the principles of non-linear bi-plots and so-called pseudo-sample projection, described by Gower and Hardings in 1988 [25], has been extended and used to overcome these limitations. However, only specific applications aimed at finding the original variables with the highest discriminant power among various classes of objects and, then, solving classification or discrimination (supervised) problems have been reported in the scientific literature [19, 26–28].

On the other hand, when a batch process monitoring scheme is designed, data resulting from runs which evolved under NOC are used to construct a so-called *in-control* model. Incoming batch data are then projected onto its space for the evaluation of the quality of the new analyzed process runs.

This approach has a clear unsupervised nature. For this reason and since the fault diagnosis plays a fundamental role in every process monitoring scheme, the principle of pseudo-sample projection will be first adapted to the case in which a kernel-based batch monitoring scheme is built. Then, it will be exploited to construct novel tools for diagnostic purposes, the so-called pseudo-sample based contribution plots.

2. Materials and methods

2.1. Datasets

Three different datasets will be analysed. The first is a simulated one containing 30 different batches during which 10 variables are measured at 25 sampling times: as such batches are characterized by differences in the variance of some variables, but not in their mean values, they will be used to mimic two different scenarios, in which increases and decreases in the process variable variability have to be detected and diagnosed, respectively (see Section 1 of the Supporting Material for the details about the data simulation).

The second one relates to a real 4-stage batch chemical process, described in [29], and consists of 22 process variables measured at 1-minute intervals for 36 batches of a single product grade. These variables mainly include pressures, temperatures and flow rates. Normal batch lengths are between 398 and 460 minutes and, therefore, before modeling the data, variable trajectory synchronization was performed by applying Dynamic Time Warping [30] stage-wise to guarantee all the process runs had the same evolution pace.

The third dataset includes eight landmark features extracted from the variable trajectories of 49 batches (26 NOC and 23 off-specification) of a pharmaceutical batch drying process, originally described in [31].

2.2. Kernel-based techniques: basic principles

The framework of all the kernel-based data analysis methods is common and based on the so-called *kernel transformation*, whose mathematical formulation is given by:

$$K(\mathbf{x}_a, \mathbf{x}_{a'}) =< \phi(\mathbf{x}_a), \phi(\mathbf{x}_{a'}) > \tag{1}$$

where $\mathbf{x}_a^{\mathrm{T}}$ and $\mathbf{x}_{a'}^{\mathrm{T}}$ are two row vectors of the original data matrix to which a specific mapping function ϕ is applied, while < and > denote the inner product. If one applies this transformation to every possible couple of vectors constituting a generic array, $\mathbf{X}$, with dimensions $N \times M$, it will be converted into a squared symmetric $N \times N$ kernel matrix, $\mathbf{K}$, whose elements constitute dissimilarity or distance measurements between two different observations. This will project the original data onto a new space, the so-called feature space, permitting to describe in a linear way possible non-linear relationships. Details about the mathematical principles of the kernel transformation and the properties of the feature space can be found elsewhere [5]. Once the kernel matrix has been obtained, a classical bilinear technique is applied to it: in case one uses PCA or PLS, a Kernel-PCA (K-PCA) or a Kernel-PLS (K-PLS) model is generated, respectively.

When dealing with kernel-based techniques, it is not needed to know the mapping function *a priori*: $\mathbf{K}$ can be obtained by applying many generic kernel functions. For the purposes of this article, only four will be taken into account: the linear, the second- and third-order polynomial and the Gaussian. Their mathematical formulations are listed in Table 1 together with the indication of their possible adjustable parameters.

Table 1: Kernel functions referred to in this article and list of their adjustable parameters

Kernel type	Kernel function	Adjustable parameters
Linear	$\mathbf{x}_a^{\mathrm{T}}\mathbf{x}_{a'}$	None
2nd-order polynomial	$(\mathbf{x}_a^{\mathrm{T}}\mathbf{x}_{a'})^2$	None
3rd-order polynomial	$(\mathbf{x}_a^{\mathrm{T}}\mathbf{x}_{a'})^3$	None
Gaussian	$\exp(-\frac{\lVert\mathbf{x}_a-\mathbf{x}_{a'}\rVert^2}{2\sigma})$	σ

2.3. Pseudo-samples and pseudo-sample projection

A pseudo-sample corresponds to a particular observation, which carries all the weight in one single variable. For example, the vector $[0, 0, \ldots, 1, 0, \ldots, 0]$ represents one of the possible pseudo-samples associated to the m-th variable of a generic dataset. By projecting an observation like this onto the latent structure of a classical 1-component PCA model, the score for this new sample is calculated by Equation 2:

$$t_{new} = [0, 0, \ldots, 1, 0, \ldots, 0]\mathbf{p} = p_m \tag{2}$$

This score is equal to the m-th value of the loadings vector $\mathbf{p}$, and, thus, gives information about the contribution of the variable x_m to the model. Creating a pseudo-sample matrix, $\mathbf{V}_m$, which contains in its m-th column values ranging from the minimum to the maximum of the variable x_m and 0 in all the other entries, as:

$$\mathbf{V}_m = \begin{bmatrix} 0, & 0, & \ldots, & \min(x_m), & 0, & \ldots, & 0 \\ & & & \cdots & & & \\ & & & \cdots & & & \\ 0, & 0, & \ldots, & \max(x_m), & 0, & \ldots, & 0 \end{bmatrix} \tag{3}$$

and projecting it onto the latent space, a trajectory is obtained according to the following equation:

$$\mathbf{V}_m\mathbf{p} = \begin{bmatrix} \min(x_m)p_m \\ \cdots \\ \cdots \\ \max(x_m)p_m \end{bmatrix} \tag{4}$$

In a higher-dimensional model space, the matrix resulting from the previous operation would define the geometrical locus of all the points lying along the direction determined by the origin of the latent space and the point, whose coordinates represent the weights of x_m on the A calculated components. Nevertheless, as proven in Appendix A, it is possible to resort to the pseudo-sample principle also when dealing with kernel-based methods.

In [19, 26, 27], the analysis of the pseudo-sample trajectories, obtained by projecting one pseudo-sample matrix per original variable under study, permitted to recover the information related to their importance or discriminant power after the application of kernel-based methods for classification and discrimination purposes. In particular, these trajectories were compared with the maximum separation direction between two classes of objects. When a monitoring scheme is built, no directions of separation are defined due to the unsupervised nature of the aforementioned approach. For this reason, the pseudo-sample projection strategy has to be slightly modified in order to be applied for fault diagnosis in such cases.

2.4. Adaptation of the pseudo-sample projection strategy to batch process monitoring

The whole procedure for building kernel-based batch monitoring schemes and recovering the information about the contribution of the original variables, enabling this way the fault diagnosis, comprises two phases (*Model building* and *Model exploitation*).
Phase I - Model building: i) Preprocess and unfold into the training matrix $\mathbf{X}$ ($N \times M$) the three-way dataset containing the observations resulting from the batches which evolved under NOC[i]; ii) Convert $\mathbf{X}$ into a kernel matrix $\mathbf{K}$ ($N \times N$) by using a specific kernel function; iii) Double-center $\mathbf{K}$ as follows:

$$\mathbf{K}_c = \mathbf{K} - \overline{\mathbf{K}}_m - \overline{\mathbf{K}}_n + \overline{\mathbf{K}}_{n,m} \tag{5}$$

where $\overline{\mathbf{K}}_m$, $\overline{\mathbf{K}}_n$ and $\overline{\mathbf{K}}_{n,m}$ are squared matrices ($N \times N$) containing the column means, the row means and the overall mean of $\mathbf{K}$, respectively; iv) Fit a PCA model on $\mathbf{K}_c$:

$$\mathbf{K}_c = \mathbf{T}^{\mathbf{K}}\mathbf{P}^{\mathbf{K}^{\mathrm{T}}} + \mathbf{E}^{\mathbf{K}} \tag{6}$$

where $\mathbf{T}^{\mathbf{K}}$ ($N \times A$), $\mathbf{P}^{\mathbf{K}}$ ($N \times A$) and $\mathbf{E}^{\mathbf{K}}$ ($N \times N$) are the score, loading and residual matrices of the fitted K-PCA model. v) Calculate the D-statistic and the Squared Prediction Error (SPE) for every observation of $\mathbf{K}$ as follows:

$$D_n^{\mathbf{K}} = \mathbf{t}_n^{\mathbf{K}^{\mathrm{T}}}\mathbf{S}^{\mathbf{K}^{-1}}\mathbf{t}_n^{\mathbf{K}} \tag{7}$$

$$\mathrm{SPE}_n^{\mathbf{K}} = \sum_{n=1}^{N} e_{n,n}^{\mathbf{K}^2} \tag{8}$$

and their corresponding control limits as in [32] and [33], respectively. $\mathbf{t}_n^{\mathbf{K}^{\mathrm{T}}}$ is the n-th row of $\mathbf{T}^{\mathbf{K}}$, $\mathbf{S}^{\mathbf{K}}$ is the K-PCA scores covariance matrix ($A \times A$)[ii], while $e_{n,n}^{\mathbf{K}}$ represents the $n \times n$-th element of $\mathbf{E}^{\mathbf{K}}$; vi) Provided that all the training process runs are *in-control*, proceed with phase II. Otherwise, remove outliers and return to step ii).

[i] As the pharmaceutical batch drying process training and test data were provided beforehand as two-way arrays obtained by LFE, the unfolding step was not required in this specific case and they were straight preprocessed in this form by standard auto-scaling.

[ii] When VWU is applied for the analysis of batch data, one observation of the final unfolded matrices represents a single time instant of a specific process run. In this article, in such cases, the calculation of its D-statistic value was executed using its predicted score, centered by the mean of the training ones corresponding to the same sampling point, and the respective instantaneous score covariance matrix.

Phase II - Model exploitation i) Preprocess and unfold into the matrix $\mathbf{X}_{Test}$ ($N' \times M$) the three-way dataset including the test process runs, as for the training set[i]; ii) Convert $\mathbf{X}_{Test}$ into a kernel matrix $\mathbf{K}_{Test}$ ($N' \times N$) by using the same kernel function as for $\mathbf{X}$. By this operation, the dissimilarity/distance values between the N' test and the N training observations are calculated, which allows $\mathbf{K}_{Test}$, after appropriate preprocessing, to be projected onto the K-PCA model space defined by the loadings matrix $\mathbf{P}^{\mathbf{K}}$; iii) Double-center $\mathbf{K}_{Test}$ as follows:

$$\mathbf{K}_{c,Test} = \mathbf{K}_{Test} - \overline{\mathbf{K}}_{m,Test} - \overline{\mathbf{K}}_{n',Test} + \overline{\mathbf{K}}_{n,m,Test} \tag{9}$$

where $\overline{\mathbf{K}}_{m,Test}$ and $\overline{\mathbf{K}}_{n,m,Test}$ are $N' \times N$ matrices containing the column means and the overall mean of $\mathbf{K}$, respectively. Notice that, unlike Equation 5, $\overline{\mathbf{K}}_{n}$ is substituted by the term $\overline{\mathbf{K}}_{n',Test}$ ($N' \times N$), containing the row means of $\mathbf{K}_{Test}$; iv) Project $\mathbf{K}_{c,Test}$ onto the *in-control* K-PCA model space as:

$$\mathbf{T}^{\mathbf{K}}_{Test} = \mathbf{K}_{c,Test}\mathbf{P}^{\mathbf{K}} \tag{10}$$

where $\mathbf{T}^{\mathbf{K}}_{Test}$ ($N' \times A$) is the predicted score matrix associated to $\mathbf{K}_{Test}$. Then, calculate the residual test matrix, $\mathbf{E}^{\mathbf{K}}_{Test}$ ($N' \times N$), as:

$$\mathbf{E}^{\mathbf{K}}_{Test} = \mathbf{K}_{c,Test} - \mathbf{T}^{\mathbf{K}}_{Test}\mathbf{P}^{\mathbf{K}^{\mathrm{T}}} \tag{11}$$

v) Calculate the D-statistic and the Squared Prediction Error (SPE) for every observation of $\mathbf{K}_{Test}$ as follows:

$$D^{\mathbf{K}}_{n',Test} = \mathbf{t}^{\mathbf{K}^{\mathrm{T}}}_{n',Test}\mathbf{S}^{\mathbf{K}^{-1}}\mathbf{t}^{\mathbf{K}}_{n',Test} \tag{12}$$

$$\mathrm{SPE}^{\mathbf{K}}_{n',Test} = \sum_{n=1}^{N} e^{\mathbf{K}^{2}}_{n',n,Test} \tag{13}$$

where $\mathbf{t}^{\mathbf{K}^{\mathrm{T}}}_{n',Test}$ is the n'-th row of $\mathbf{T}^{\mathbf{K}}_{Test}$[ii], while $e^{\mathbf{K}}_{n',n,Test}$ represents the n'×n-th element of $\mathbf{E}^{\mathbf{K}}_{Test}$; vi) If a faulty batch is detected, create a $V \times M$ pseudo-sample matrix, $\mathbf{V}_{m,NOC}$, for each one of the M original variables as in Equation 3 using its minimum and its maximum values in the whole training matrix, $\mathbf{X}$; vii) Build another set of $\mathbf{V}_{m,Fault}$ pseudo-sample matrices ($V' \times M$) using, in this case, the minimum and the maximum values of the M original variables in $\mathbf{X}_{Test}$ relative to the faulty batch[iii]; viii) Apply to each couple of pseudo-sample matrices the same kernel transformation as for the training data in order to obtain a pair of pseudo-sample kernel matrices, $\mathbf{K}_{\mathbf{V}_{m,NOC}}$ ($V \times N$) and $\mathbf{K}_{\mathbf{V}_{m,Fault}}$ ($V' \times N$). By this operation, the dissimilarity/distance values between the N training observations and both the V pseudo-samples in $\mathbf{V}_{m,NOC}$ and the V' pseudo-samples in $\mathbf{V}_{m,Fault}$ are calculated, which allows $\mathbf{K}_{\mathbf{V}_{m,NOC}}$ and $\mathbf{K}_{\mathbf{V}_{m,Fault}}$, after appropriate preprocessing, to be projected onto the K-PCA model space defined by the loadings matrix $\mathbf{P}^{\mathbf{K}}$; ix) Double-center every $\mathbf{K}_{\mathbf{V}_{m,NOC}}$ and every $\mathbf{K}_{\mathbf{V}_{m,Fault}}$ so that:

$$\mathbf{K}_{\mathbf{V}_{m,NOC},c} = \mathbf{K}_{\mathbf{V}_{m,NOC}} - \overline{\mathbf{K}}_{m,\mathbf{V}_{m,NOC}} - \overline{\mathbf{K}}_{v,\mathbf{V}_{m,NOC}} + \overline{\mathbf{K}}_{n,m,\mathbf{V}_{m,NOC}} \tag{14}$$

$$\mathbf{K}_{\mathbf{V}_{m,Fault},c} = \mathbf{K}_{\mathbf{V}_{m,Fault}} - \overline{\mathbf{K}}_{m,\mathbf{V}_{m,Fault}} - \overline{\mathbf{K}}_{v',\mathbf{V}_{m,Fault}} + \overline{\mathbf{K}}_{n,m,\mathbf{V}_{m,Fault}} \tag{15}$$

[iii] Special circumstances in which this operation needs to be slightly adjusted will be further discussed in the following sections.

where $\overline{\mathbf{K}}_{m,\mathbf{v}_{m,NOC}}$ $(V \times N)$ and $\overline{\mathbf{K}}_{m,\mathbf{v}_{m,Fault}}$ $(V' \times N)$ carry the column means of $\mathbf{K}$, $\overline{\mathbf{K}}_{n,m,\mathbf{v}_{m,NOC}}$ $(V \times N)$ and $\overline{\mathbf{K}}_{n,m,\mathbf{v}_{m,Fault}}$ $(V' \times N)$ include the overall mean of $\mathbf{K}$, while the v-th row of $\overline{\mathbf{K}}_{v,\mathbf{v}_{m,NOC}}$ $(V \times N)$ and the v'-th row of $\overline{\mathbf{K}}_{v',\mathbf{v}_{m,Fault}}$ $(V' \times N)$ contain the mean of the v-th row of $\mathbf{K}_{\mathbf{V}_{m,NOC}}$ and the mean of the v'-th row of $\mathbf{K}_{\mathbf{V}_{m,Fault}}$, respectively; x) Project the two groups of double-centered pseudo-sample kernel matrices onto the K-PCA model space, as follows:

$$\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,NOC}} = \mathbf{K}_{\mathbf{V}_{m,NOC},c}\mathbf{P}^{\mathbf{K}} \tag{16}$$

$$\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,Fault}} = \mathbf{K}_{\mathbf{V}_{m,Fault},c}\mathbf{P}^{\mathbf{K}} \tag{17}$$

where $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,NOC}}$ $(V \times A)$ and $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,Fault}}$ $(V' \times A)$ represent the pseudo-sample K-PCA predicted score matrices associated to $\mathbf{K}_{\mathbf{V}_{m,NOC}}$ and $\mathbf{K}_{\mathbf{V}_{m,Fault}}$, respectively. Two different pseudo-sample trajectories will be then constructed per each original variable, which can be considered as representations in the latent space of the K-PCA model of its real variability range in the NOC batches and in the process run detected as faulty, respectively; xi) Calculate for every couple of pseudo-sample trajectories related to the same m-th variable the so-called *Discriminant Distance* (DD_m) as:

$$DD_m = \sqrt{\sum_{a=1}^{A} \left(\frac{\widetilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a} - \widetilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}}{s_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}}} \right)^2} \tag{18}$$

which will be used as an index of the m-th variable contribution to the fault. $\widetilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a}$ and $\widetilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}$ are the median values of the column vectors of $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,Fault}}$ and $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,NOC}}$, respectively, related to the a-th component of the model. $s_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}}$ is the standard deviation of the column vector of $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,NOC}}$ related to the a-th component of the model. This term will permit to distinguish specific situations in which the absolute difference between the two median values, $\widetilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a}$ and $\widetilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}$, might not be statistically significant considering the variability range of the pseudo-sample predicted scores associated to the NOC observations; xii) Represent the DD_m values in a bar plot for all the M original variables, obtaining a pseudo-sample based contribution plot (named *Discriminant Distance* or *DD plot* from now on).

3. Results and discussion

The three datasets, described in Section 2.1, will be used in order to assess and compare the performance of classical bilinear techniques to that of kernel-based methods coupled to pseudo-sample projection in batch process monitoring. VWU and BWU will be tested on the simulated and the real chemical data. They will be combined with the two most common preprocessing approaches applied in process chemometrics: Variable Centering and Scaling (VCS) and Trajectory Centering and Scaling (TCS). This will permit to evaluate how VCS and TCS affect the quality of the final monitoring schemes. VCS mean-centers and scales to unit variance each j-th process variable. TCS consists of mean-centering and scaling to unit variance each j-th process variable at each t-th sampling point, allowing their variation around their average trajectories to be subsequently modelled. Specifically, the comparison will be carried out considering: i) Variable Centering and

Scaling-VWU (VCS-VWU), ii) Trajectory Centering and Scaling-VWU (TCS-VWU) and Trajectory Centering and Scaling-BWU (TCS-BWU).

On the other hand, the third dataset will permit to evaluate how bilinear and kernel methods perform when LFE is applied to a batch dataset.

In any considered case-study, the number of principal components (PCs) calculated for building the PCA- and K-PCA-based monitoring schemes was determined by cross-validation[iv]. The cross-validatory approach was extended in order to take into account also a certain set of possible kernel transformations of the original training data. The final combination of number of PCs and kernel function was selected as that guaranteeing the best compromise between degree of non-linearity of the model and reconstruction error. The authors are aware that such criterion might not be the most suitable in a monitoring context [34]. Nevertheless, in all the applications under study, it permitted to obtain very satisfying outcomes in terms of fault detection and diagnosis accuracy.

3.1. Simulated dataset - Variability increase detection case-study

The first approach tested on the simulated dataset was VCS-VWU. After preprocessing and unfolding the three-way arrays, 10 NOC process runs were randomly selected for building the PCA and K-PCA *in-control* models. Regarding the latter, a second-order polynomial function was chosen for the kernel transformation.

The comparison of the two monitoring schemes was then carried out after having adjusted the control limits of the resulting SPE and D-statistic control charts so that the Overall Type I (OTI) risk value, the false alarm rate, was approximately equal to the corresponding imposed significance level (ISL) α (5% and 1% in this specific case)[v]. The expression for the OTI risk value is:

$$OTI = 100\frac{nf}{I_{NOC}T} \tag{19}$$

where nf denotes the number of sampling points detected as faulty, I_{NOC} represents the number of considered NOC batches and T is the total number of sampling points per batch. The control limit adjustment was assessed after the projection of the 5 NOC batches, left out of the training set, onto the model space. As highlighted in Table 2, the OTI values are quite similar for both considered monitoring schemes, which indicates that the control limits were correctly adjusted.

The fault detection power of the different control charts resulting from classical PCA and K-PCA was evaluated according to the Overall Type II ($OTII$) risk value, calculated as:

$$OTII = 100\frac{nnf}{I_{faulty}T} \tag{20}$$

[iv] In case VWU was resorted to for the analysis of the batch data, at each iteration, the bunch of observations associated to a specific batch were removed from the training set. On the other hand, one single row was iteratively left out when the unfolding step was performed by BWU or LFE.

[v] Notice that the significance level imposed for the control limits of the SPE and D-statistic charts may vary from case to case. This is due to the fact that their adjustment is assessed by using an external test set, which may have different size depending on the original data array under study. The same applies for the confidence limits of every pseudo-sample based contribution plot displayed from now on, calculated by a jackknife procedure.

Table 2: Overall Type I (OTI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

	PCs	R^2	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	2	0.705	6.4%	1.6%	4.8%	1.6%
K-PCA (second-order polynomial)	2	0.743	6.4%	0.8%	4.0%	0.8%

where nnf represents the number of non-signaled faulty sampling points and I_{faulty} is the number of faulty batches, projected onto the *in-control* model. A good performance would lead to a $OTII$ value close to 0. The results, obtained using the 15 faulty test runs, are reported in Table 3.

Table 3: Overall Type II ($OTII$) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	79.5%	89.1%	0.3%	5.6%
K-PCA (second-order polynomial)	0.3%	1.1%	0.0%	0.0%

The K-PCA-based SPE control chart shows $OTII$ values approximately equal to zero, much lower than the ones resulting from the classical PCA model. On the other hand, the D-statistic control charts seem to be characterized by a similar good accuracy in terms of fault detection power. As an example, the SPE and the D-statistic control charts obtained by both classical PCA and K-PCA for the first faulty batch contained in the test set are represented in Figure 1.

Once a specific fault has been detected by one or both of the considered control charts, it is fundamental to diagnose it and verify which variables contribute the most to the *out-of-control* signal. In general, this is done by the so-called contribution plots [20]. In Figure 2, the classical SPE and D-statistic contribution plots, related to the first sampling point of the first faulty test batch found to be beyond the 95% control limit in the corresponding PCA-based control charts (time instant #10 and #1, respectively), are displayed. The SPE contribution plot for time instant #10 (Figure 2a) points out an issue, which affects variables x_8, x_9 and x_{10}. Unfortunately, the conclusions one would reach looking at it would be completely wrong. In fact, these three variables do not show any difference in their evolution between NOC and faulty batches, as highlighted in Section 1 of the Supporting Material. Therefore, the fault diagnosis would be completely mistaken. On the other hand, in the D-statistic contribution plot (Figure 2b), variables x_8, x_9 and x_{10} are correctly identified as having a low contribution to the fault at time instant #1. However, its interpretation is not straightforward: all the variables from x_1 to x_7 were simulated so that their variance in the faulty batches was approximately twice the variance in the NOC process runs. Thus, such large differences in their contributions were not expected and are not coherent with the nature of the simulated dataset. Furthermore, the SPE and D-statistic contribution plots related to all the

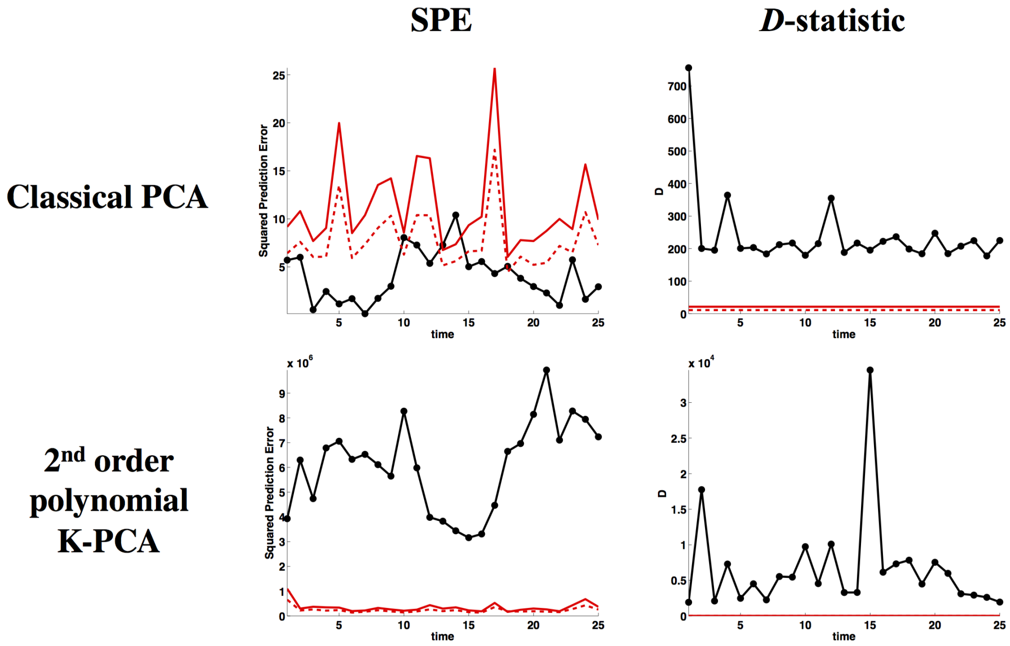


Figure 1: SPE and D-statistic control charts obtained by both classical PCA and K-PCA for the first faulty test batch. The dotted and the solid red lines represent the 95% and the 99% control limits, respectively

other sampling times were found to have consistent profiles as the ones displayed in Figure 2 (not shown).

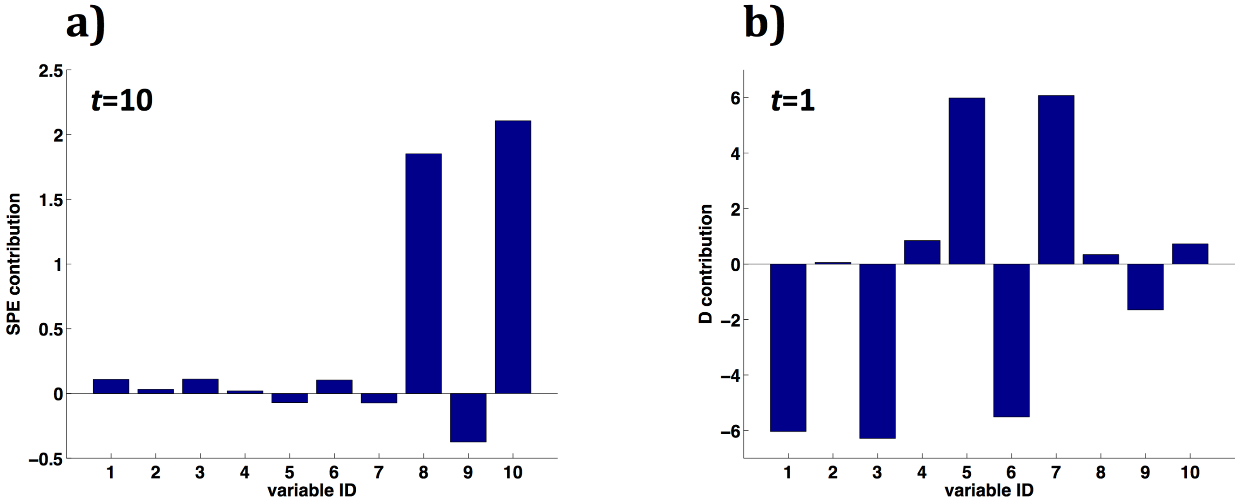


Figure 2: Classical PCA: a) SPE and b) D-statistic contribution plots related to first sampling point of the first faulty test batch found to be beyond the 95% control limit in the corresponding PCA-based control charts (time instant #10 and #1, respectively)

Regarding the kernel-based approach, the fault diagnosis was enabled by resorting to the strategy described in Section 2.4. Figure 3 shows the *DD* plot, related to the first faulty test batch. Notice that this is an overall contribution plot, taking into account the whole evolution of each variable in both the faulty and the NOC process runs under study and not focusing on an individual value it assumes at a certain time instant. However, the proposed pseudo-sample approach can be easily adapted to diagnose faults, occurring during specific time intervals of the batch under study. For instance, in this case, instantaneous pseudo-sample based contribution plots might have also been constructed[vi], but, it was not necessary because, for this particular process run, both SPE and *D*-statistic were found beyond the 99% control limits at all the sampling points (see Figure 1). However, all the possible instantaneous *DD* plots showed very similar outcomes as that in Figure 3 (not shown).

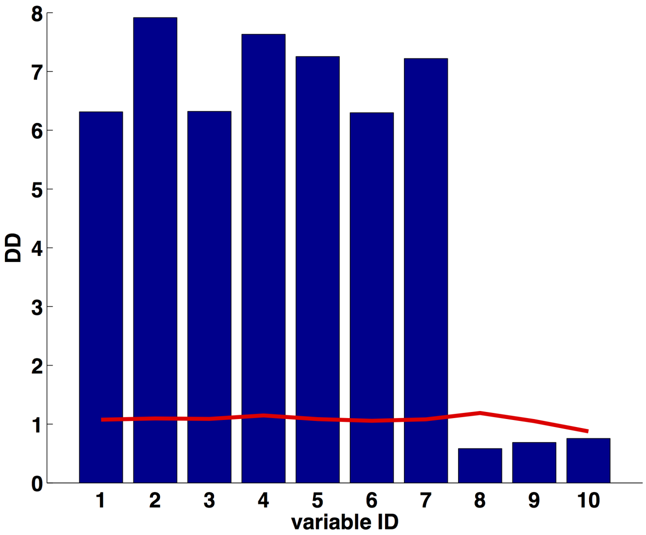


Figure 3: K-PCA: Overall *DD* plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure

For the sake of clarity, the original pseudo-sample trajectories are represented in Figure 4.

In order to ease the interpretation of Figure 3 and quickly identify the process variables that are different in the faulty batch compared to the NOC runs, 90% confidence limits were calculated for the different *DD* values using the jackknife procedure, proposed in [35]. As expected, variables x_8, x_9 and x_{10} have no statistically significant contributions to the fault (values below the respective confidence limits). Furthermore, in accordance with the way the data were simulated, similar contributions were found for variables from x_1 to x_7.

Exploiting the features of the pseudo-sample trajectories represented in Figure 4, a modification

[vi] Specifically by i) computing per each of the M considered variables one single pseudo-sample using the value it assumes when a specific *out-of-control* signal is detected; ii) transforming it by applying the same kernel function as for the training data; iii) double-centering the resulting pseudo-sample kernel vector and projecting it onto the K-PCA model space, as detailed in Section 2.4; iv) substituting the resulting score value to the median value $\tilde{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a}$ in Equation 18.

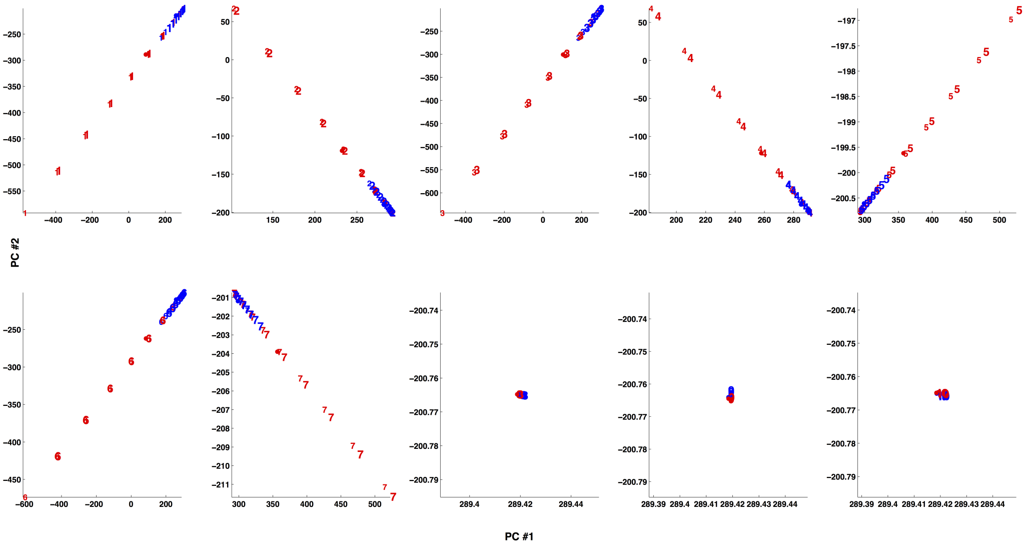


Figure 4: K-PCA: Original pseudo-sample trajectories obtained by the procedure described in Section 2.4. The scores represented as blue numbers are calculated starting from the minimum and the maximum values each preprocessed variable assumes in all the training NOC batches. The scores represented as red numbers are calculated starting from the minimum and the maximum values each preprocessed variable assumes in the considered *out-of-control* test batch. Their graphical comparison permits to evaluate whether the original variables assume values either inside or outside their *in-control* variability range, during the occurrence of a fault

of the original *DD* plot can also be developed to selectively focus on changes in the variance of the process variables under study, as in this specific case. This variant is named *Pseudo-sample Trajectory Variance Ratio plot* (*VR* plot) and the contribution of each variable to the fault is evaluated according to the VR_m index, calculated as:

$$VR_m = \sum_{a=1}^{A} \frac{max(\{s^2_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}}\},\{s^2_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a}}\})}{min(\{s^2_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}}\},\{s^2_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a}}\})} \qquad (21)$$

where $s^2_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,NOC},a}}$ and $s^2_{\mathbf{t}^{\mathbf{K}}_{\mathbf{V}_{m,Fault},a}}$ correspond to the variance of the column vectors of $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,NOC}}$ and $\mathbf{T}^{\mathbf{K}}_{\mathbf{V}_{m,Fault}}$, respectively, related to the a-th component of the model. In Figure 5 the overall *VR* plot related to the first faulty test batch is displayed.

Also in this case, a completely correct fault diagnosis was obtained.

The rationale behind the definition of the *VR* index is associated to the frequent need in process monitoring to distinguish whether an *out-of-control* signal is generated by shifts in the average level of specific measured variables or by changes in their variability. In fact, although the *DD* plot has proved to be an effective tool for detecting the latter deviations in case they represent the only differences between *in-control* and faulty batches and a second-order polynomial function is found to be the optimal for the data under study, in more complex scenarios, when a different kernel transformation is performed, coupling it with the *VR* plot may constitute a feasible and valid

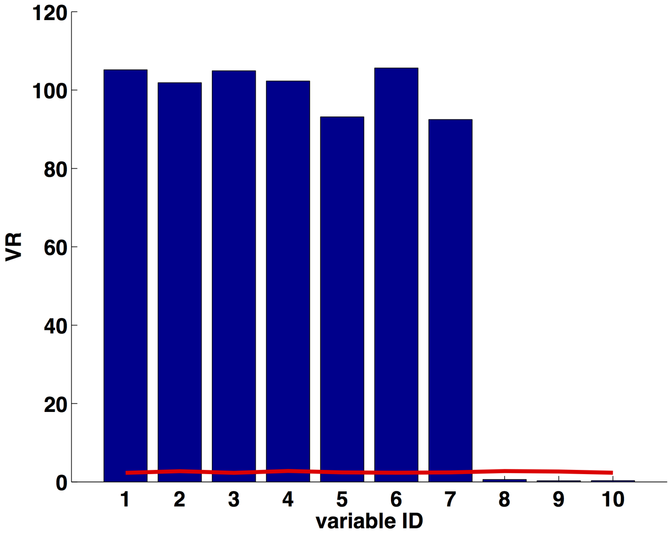


Figure 5: K-PCA: Overall VR plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the VR values of every variable by a jackknife-based procedure

option for unveiling both types of variation.

The same modeling strategy was followed when TCS-VWU and TCS-BWU were used to preprocess and unfold the original simulated three-way arrays. The results are detailed in Section 2 of the Supporting Material, which also includes the classical contribution and the DD and VR plots used for the diagnosis of a particular fault in the various circumstances.

Regarding TCS-VWU, a similar good accuracy was found for both the PCA- and the K-PCA-based SPE and D-statistic control charts. Nevertheless, the interpretation of the classical SPE and D-statistic contribution plots is not obvious, which drastically jeopardizes the fault diagnosis. On the contrary, both the DD and the VR plots permit to correctly identify the process variables showing a different evolution with respect to an *in-control* situation.

Concerning TCS-BWU, as each row of the final unfolded data arrays contains all the information registered for one batch, the comparison was carried out assessing the Type I (TI) and Type II (TII) risk values. They stand for the percentage of NOC process runs detected as abnormal and the percentage of faulty ones detected as NOC, respectively. Owing to the limited number of analyzed batches, in this case, the control limits of the SPE and D-statistic charts were adjusted so that the TI risk value was equal to an α of 20%. Here, no significant differences in terms of fault detection power were found in either the SPE or the D-statistic control charts obtained by classical PCA and K-PCA (with a Gaussian data transformation and a σ parameter value of 89.1). Also the fault diagnosis was correctly addressed by both the studied approaches. Here, three aspects have to be carefully taken into account:

- since every row of the final training and test matrices contains information associated to the entire evolution of a particular process run, a single SPE and a single D-statistic value is calculated for each one of them. For the same reason, the procedure to construct the DD plots employed for diagnostic purposes has to be redefined as specified before. Notice that this point also applies when LFE is resorted to for building batch monitoring schemes;

- applying BWU to the original three-way data structures permits to compare the considered faulty process run with the training NOC ones time instant-to-time instant and, then, compute a contribution or a *DD* value per each sampling point of the evolution of every variable;
- as in this case the pseudo-sample based strategy takes into account the variability range of every variable at a specific time instant of the evolution of all the training batches and the single value that the same variable assumes at the same time instant in a specific run detected as faulty, the *VR* indices cannot be calculated and the respective pseudo-sample based contribution plot cannot be constructed.

The similarity between the performance of classical PCA and K-PCA results from coupling TCS and BWU. In fact, their combination permits to remove the strongly non-linear dynamic trend of the evolution of the considered variables from the data and, at the same time, model their variation about their average trajectories at the single sampling points, reducing the process analysis to a stationary problem [4]. This is also proven by the fact that a Gaussian function with a relatively high σ parameter was selected after a previous step of cross-validation as the optimal one for dealing with such data (N.B. the higher the σ parameter, the more linear the kernel transformation).

3.2. Simulated dataset - Variability decrease detection case-study

A similar comparison was carried out inverting the training and the test sets of the previous simulated case-study in order to verify how classical PCA and K-PCA perform when a decrease in the variability of the process variables has to be detected and diagnosed. In particular, the *in-control* models were built on 10 out of the 15 batches characterized by a higher variance of the measured variables (randomly chosen), the adjustment of the resulting control chart limits was assessed after the projection of the 5 batches, left out of the training set, and the fault detection and diagnosis power of the final monitoring schemes was evaluated as described before using the remaining 15 process runs. The results are displayed in Section 3 of the Supporting Material.

VCS-VWU did not lead to satisfactory performance with either PCA or K-PCA. This is a consequence of the fact that mean-centering and scaling the complete variable trajectories of the test batches (showing a lower standard deviation) by the larger standard deviation of those of the training runs modified their profiles so that they showed very slight fluctuations around zero. For this reason, the scores associated with every sampling time of the evolution of a new batch were concentrated around the origin of the latent space, which, combined with the large variability of the training process runs, made the detection of any fault impossible.

After TCS-VWU, the *D*-statistic control charts resulting from both the second-order polynomial K-PCA and classical PCA models were found to be characterized by a similar fault detection accuracy. Nevertheless, only the *DD* and *VR* plots permitted to correctly identify the process variables showing a decrease in their variability in the test batches, while the classical SPE and *D*-statistic contribution plots were not able to recognize them and suffer from lack of interpretability.

Finally, the two monitoring schemes constructed after TCS-BWU were compared. In this case, the *D*-statistic control charts resulting from a Gaussian K-PCA (σ=19.3) and a classical PCA model proved to have a high accuracy in terms of fault detection power and both the classical contribution and the *DD* plots enabled a correct fault diagnosis.

A summary of the outcomes obtained in each of the described simulated case-study is displayed in Table 4.

Table 4: Summary of the results obtained in the different simulated case-studies. The table shows which of the classical PCA- or K-PCA-based control charts and contribution plots were able to detect and diagnose the considered faults, respectively.

		Variability increase		Variability decrease	
		Fault detection	Fault diagnosis	Fault detection	Fault diagnosis
VCS-VWU	PCA	*D*	NO	NO	NO
	K-PCA	*D*, SPE	*DD*, *VR*	NO	NO
TCS-VWU	PCA	*D*, SPE	NO	*D*	NO
	K-PCA	*D*, SPE	*DD*, *VR*	*D*	*DD*, *VR*
TCS-BWU	PCA	*D*, SPE	*D*	*D*	*D*
	K-PCA	*D*, SPE	*DD*	*D*	*DD*

3.3. Chemical process dataset

In order to build a first K-PCA monitoring scheme on the chemical process dataset, VCS-VWU was applied. A training set, containing observations associated with 20 NOC batches, a first test set, including 11 NOC process runs, and a second test set, consisting of 5 batches manufactured with a deviating steam supply were obtained by unfolding the original three-way arrays. For increasing the quality of the final monitoring scheme, a multi-stage modeling strategy was resorted to: the training set was further divided into 4 sub-matrices, each one related to a different process phase, on which, 4 cross-validated K-PCA models were built. As highlighted in Table 5, except for stage #1, a Gaussian kernel function was selected for the original data transformation.

Table 5: Parameters of the K-PCA *in-control* models built for every process stage

	Stage #1	Stage #2	Stage #3	Stage #4
Function Type	3-rd order polynomial	Gaussian	Gaussian	Gaussian
CV-σ	-	23.8	15.0	12.8
PCs	3	2	2	2
R^2	0.852	0.850	0.798	0.775

After performing K-PCA on the training data and assessed the adjustment of the limits of the resulting control charts using the 11 NOC batches of the first test set, the observations related to the faulty process runs were transformed in the same way as for the corresponding training set and projected onto the corresponding model space. As an example, the case of the fifth faulty test batch is reported. Figure 6 shows its respective SPE and the D-statistic control charts.

In every process stage, the SPE values are always beyond the 95% control limit and, at most of the time samples, beyond the 99% control limit, which evidently highlights an issue affecting the evolution of this process run all over its duration. To investigate the nature of this problem and pinpoint which variables evolve differently with respect to an *in-control* situation, the *DD* plots were exploited also in this case. In Figure 7, those related to every process stage and associated to the fifth faulty test batch are displayed. They clearly point out an issue which affects variable x_{22} during all the evolution of the batch. This exactly corresponds to the steam pressure, which is known to be the main cause generating the deviating behavior of all the process runs included in the second test set. Furthermore, regarding stage #3, a further problem is highlighted. In this case, the variable pinpointed as having the highest contribution to the fault is x_4.

At a first glance, the fault diagnosis might be considered correct, since the main problem associated to the steam pressure is properly identified. Nevertheless, by inspecting more carefully the original variable trajectories, it was possible to realize that the contribution of variable x_1 in stage #2 was much lower than expected (see Figure 8).

This variable contribution masking effect is due to the combination between VCS-VWU and the pseudo-sample projection strategy adopted for fault diagnosis in this article. In fact, the reader should remind that the pseudo-sample matrices are filled in with values ranging from the minimum to the maximum of a certain column of the original preprocessed data related to the *in-control* process runs and to the batch detected as faulty. When VCS-VWU is applied, each of these columns contains the complete evolution of a single variable, which preserves its non-linear dynamic trend. If the minimum and the maximum of the variable under study in the two cases roughly correspond, the resulting pseudo-sample trajectories would cover approximately the same variability range (see the animation contained in the Supporting Material). This would generate a really small difference between the medians of the pseudo-sample score distributions and, thus, a really small contribution for this variable, even if its evolution in the faulty batch is different compared to the *in-control* process runs. That is exactly what happens with variable x_1 in stage #2 for the fifth batch of the second test set, as highlighted in Figure 8.

Similar results were obtained in terms of fault detection accuracy for TCS-VWU, while the resulting *DD* plots enabled a slightly clearer, but not completely satisfying fault diagnosis (not shown). The cross-validated values of the σ parameters of the kernel functions were found to be all between 50 and 75 for the different process stages, which means a more linear kernel transformation is needed for adequately modeling this dataset if TCS-VWU is used.

A simple way to definitely solve the problem of the inconsistency of the variable contributions to the fault of the previous *DD* plots is to switch to TCS-BWU. Here, a single K-PCA model was built on the whole training set, because no significant differences were detected when a multi-stage strategy was applied. A Gaussian function was selected to transform the unfolded data. The optimized σ parameter was fixed at a value of approximately 700, which lead to an approximately linear kernel transformation. This corroborates what was stated before about the coupling of TCS and BWU.

Figure 9 indicates that the fifth faulty test batch is correctly identified as an outlier by the SPE control chart.

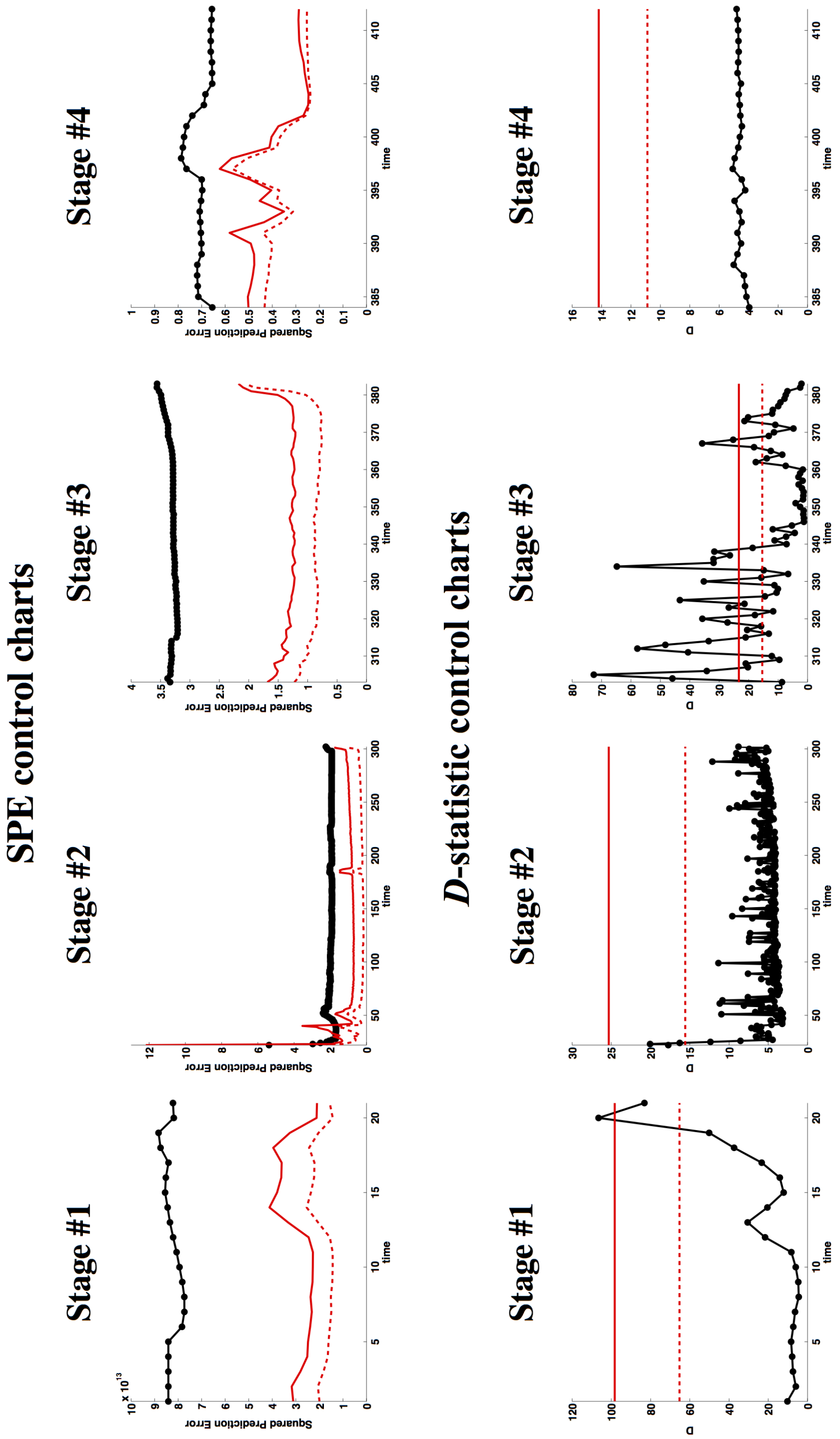


Figure 6: K-PCA: SPE and D-statistic control charts related to every stage of the evolution of the fifth faulty batch contained in the second test set (VCS-VWU). The dotted and the solid red lines represent the 95% and the 99% control limits, respectively

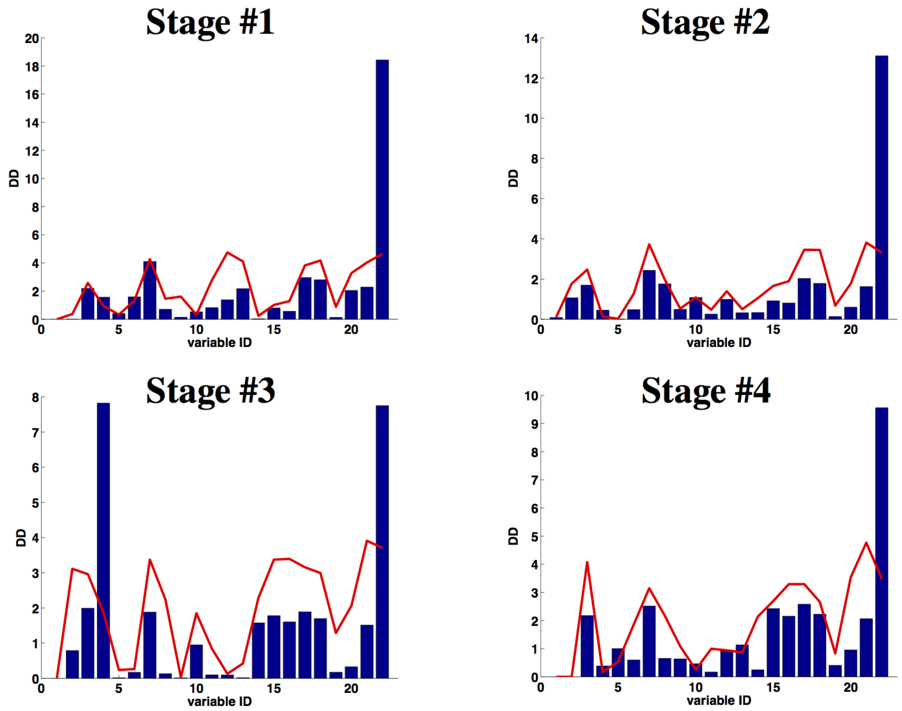


Figure 7: K-PCA: Overall stage-*DD* plots related to the fifth faulty test batch (VCS-VWU). The solid red lines represent the 95% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure

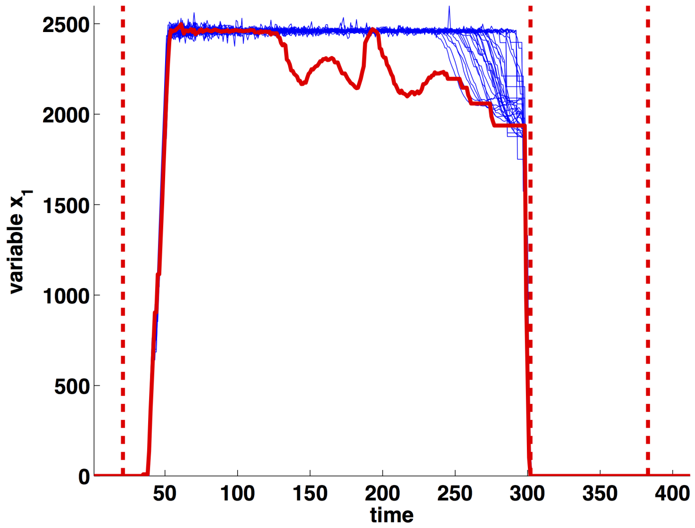


Figure 8: Temporal evolution of the variable x_1 of the chemical process dataset in the NOC batches (blue thin lines) and in the fifth faulty process run contained in the second test set (red thick line). The vertical dotted red lines mark the end point of every process stage

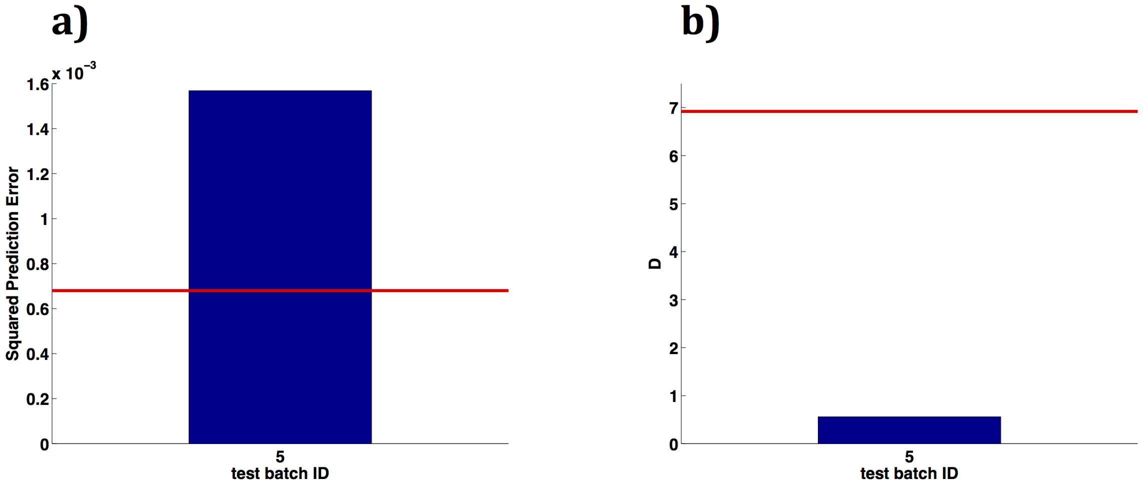


Figure 9: K-PCA: a) SPE and b) *D*-statistic control charts related to the fifth faulty batch contained in the second test set (TCS-BWU). The solid red lines represent their respective 90.9% control limit

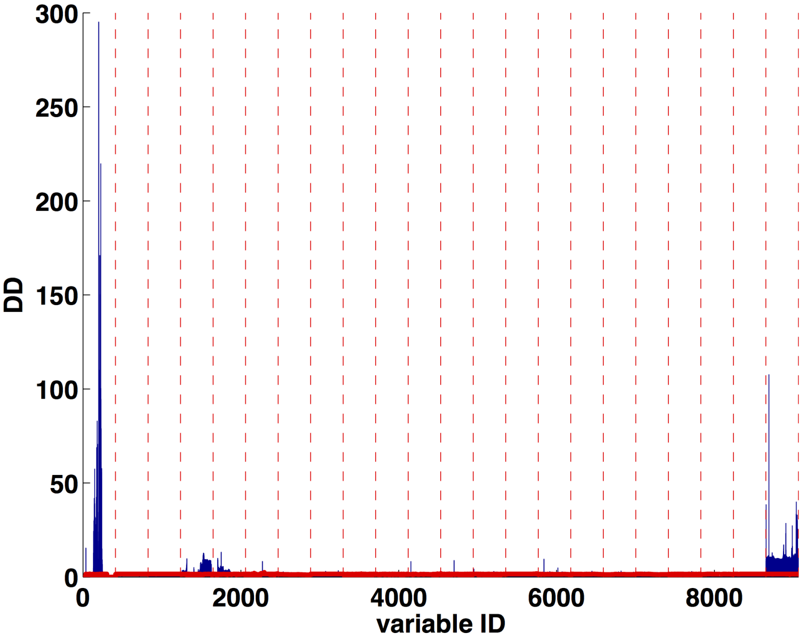


Figure 10: K-PCA: Overall *DD* plot related to the fifth faulty test batch (TCS-BWU). The solid red line represents the 95% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure. The vertical dotted red lines mark the separation between the *DD* values at the different sampling times of two consecutive process variables

In this case, a more accurate fault diagnosis is enabled, as proven by the *DD* plot related to the fifth batch of the second test set, displayed in Figure 10 where also the contribution of variable x_1 is correctly highlighted. Furthermore, it seems to be much higher than the other ones, in contrast with what was observed when VWU was applied for the analysis of these data. The temporal evolution of the original variables, found to have a high contribution to the fault (x_1, x_4, x_5 and x_{22}), in the training NOC batches and the faulty test process run under study is represented in Section 4 of the Supporting Material.

The analysis of the chemical process dataset by means of classical PCA also resulted in outcomes, which were coherent with those obtained when K-PCA was resorted to (not shown). This may be associated to the fact that not so strong non-linear relationships were present in the original data. However, the quality of the final monitoring schemes was found to be strictly dependent on the number of PCs, which could represent an additional issue to be solved for a correct fault detection and diagnosis.

3.4. Pharmaceutical batch drying process

The pharmaceutical batch drying process dataset [31] represents a specific case-study, in which LFE is used for monitoring batch processes. Here, the *in-control* models were built on 17 process runs, which evolved under NOC, while the remaining 9 were used to assess the adjustment of the limits of the resulting SPE and D-statistic control charts according to the TI risk values, as done when TCS-BWU was applied previously in this article. Owing to the low number of analyzed batches, the imposed significance level α was set at a value of 11.1. The fault detection power of the control charts constructed by both PCA and K-PCA was evaluated in terms of TII values, calculated after the projection of the 23 faulty test runs onto the respective model space. The results are listed in Table 6 and 7.

Table 6: Type I (TI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

	PCs	R^2	$\text{SPE}_{\text{ISL}=11.1\%}$	$D_{\text{ISL}=11.1\%}$
Classical PCA	2	0.790	11.1%	11.1%
K-PCA (third-order polynomial)	2	0.830	11.1%	11.1%

Table 7: Type II (TII) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

	$\text{SPE}_{\text{ISL}=11.1\%}$	$D_{\text{ISL}=11.1\%}$
Classical PCA	30.4%	69.2%
K-PCA (third-order polynomial)	17.4%	65.2%

In this case, the use of a third-order polynomial kernel transformation clearly improved the accuracy of the SPE control chart, leading to a higher number of test batches correctly detected as faulty (TII=17.4%, α=11.1%) than when resorting to classical PCA (TII=30.4%, α=11.1%), while similar TII values were found for the two different D-statistic control charts. In order to evaluate the fault diagnosis ability of both the monitoring schemes, the classical SPE contribution plot and the DD plot (constructed, for the reason previously mentioned, according to the adapted procedure described in Section 3.1) related to the first faulty test batch are displayed in Figure 11. The D-statistic contribution plot is not shown, as this specific process run did not exhibit a value for this index beyond the control limit.

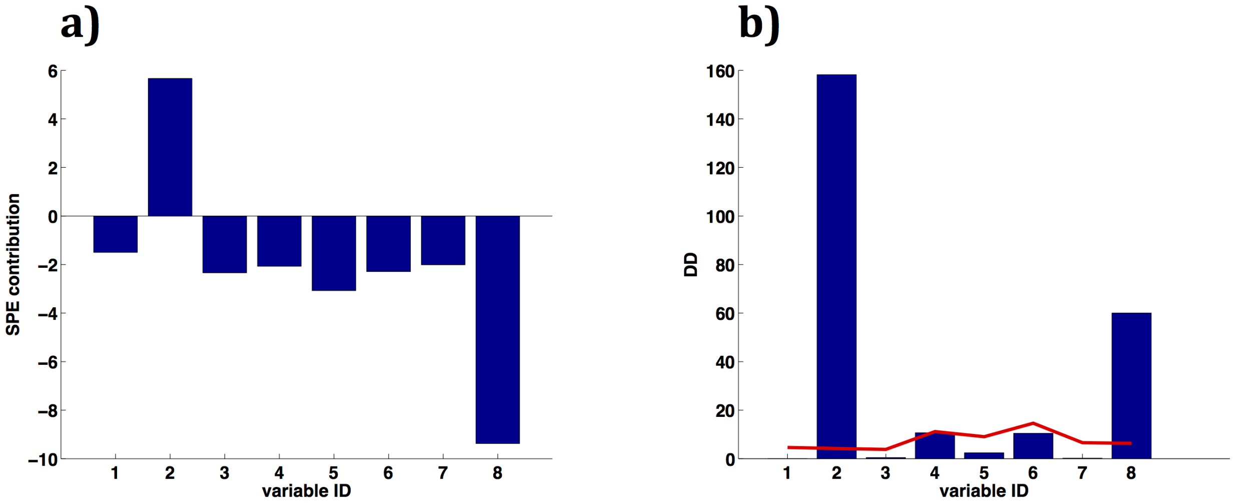


Figure 11: a) SPE contribution plot (classical PCA) and b) DD plot (K-PCA) related to first faulty test batch. The solid red line represents the 94.1% confidence limit calculated for the DD values of every variable by a jackknife-based procedure

The fault diagnosis was correctly addresses by both the approaches under study. The variables pinpointed as having the highest contribution to the fault were x_2 (dryer temperature) and x_8 (batch duration). In the original article, these were found to be the most critical parameters affecting the quality of the final product, together with the level of the solvent collector tank (variable x_1), which showed a normal behavior in this particular batch [31].

4. Conclusions

In this article a novel approach for fault detection and diagnosis, based on the combination of K-PCA and pseudo-sample projection, was proposed to deal with complex batch process datasets, usually difficult to model if affected by strong non-linearities and in case the physico-chemical phenomena generating them are not completely known. In particular, an innovative tool, the so-called pseudo-sample based contribution plot (in the form of both DD and VR plot) was developed to overcome the main drawback of the kernel-based methods, that is, their hard interpretability (which does not permit to evaluate the importance of the original variables in the final models). Two interesting points arose from the results of the data analysis:

- when strongly non-linear data structures had to be analyzed, non-linear kernel-based techniques showed clear advantages in comparison to classical bilinear approaches. In fact, in most of the simulated case-studies, even if both the K-PCA- and classical PCA-based monitoring schemes were found to be characterized by a similar fault detection power, in terms of fault diagnosis, classical contribution plots suffered from lack of interpretability and did not lead to identify the process variables evolving differently with respect to an *in-control* situation. On the other hand, the pseudo-sample based contribution plots always resulted in a completely correct diagnosis of the detected faults;

- even if no severe non-linear relationships affected the original analyzed data, it was still possible to resort to kernel-based methods and pseudo-sample projection, obtaining very similar results to classical PCA. In fact, in all the circumstances in which classical PCA performed well, the cross-validation procedure, by which the best kernel transformation and its adjustable parameters were chosen, selected an approximately linear function (in general Gaussian with a relatively high value of σ), guaranteeing an equally correct fault detection and diagnosis.

In addition, the outcomes of the analysis of the chemical process dataset highlighted that one of the two proposed pseudo-sample based contribution plots, the *DD* plot, enables a more correct and accurate fault diagnosis if it is used in combination with TCS-BWU and a single *DD* index is calculated for every sampling time of the evolution of every *out-of-control* batch.
However, the pseudo-sample based approach proposed in this article shows a particular limitation, represented by the fact that the information about the process variables affected by a fault is recovered in a univariate way. New research aimed at trying to unveil possible interactions among process variables is now in progress, which can be useful to further strengthen the methodology.

5. Acknowledgments

This research work was partially supported by the Spanish Ministry of Economy and Competitiveness under the project DPI2011- 28112-C04-02 and Shell Global Solutions International B.V. (Amsterdam, The Netherlands).

6. References

[1] J. Camacho, J. Picó, A. Ferrer, Bilinear modelling of batch processes. Part I: theoretical discussion, J. Chemometr. 22 (2008) 299–308.

[2] J. Camacho, J. Picó, A. Ferrer, Bilinear modelling of batch processes. Part II: a comparison of PLS soft-sensors, J. Chemometr. 22 (2008) 533–547.

[3] J. González-Martínez, J. Camacho, A. Ferrer, Bilinear modelling of batch processes. Part III: parameter stability, J. Chemometr. 28 (2014) 10–27.

[4] S. Wold, N. Kettaneh-Wold, J. MacGregor, K. Dunn, Comprehensive Chemometrics, 1st Edition, Elsevier B.V., Oxford, UK, 2009, Ch. 2.10, pp. 163–197.

[5] D. Cao, Y. Liang, Q. Xu, Q. Hu, L. Zhang, G. Fu, Exploring nonlinear relationships in chemical data using kernel-based methods, Chemometr. Intell. Lab. 107 (2011) 106–115.

[6] B. Walczak, D. Massart, The radial basis functions-partial least squares approach as a flexible non-linear regression technique, Anal. Chim. Acta 331 (1996) 177–185.

[7] B. Walczak, D. Massart, Application of radial basis functions-partial least squares to non-linear pattern recognition problems: diagnosis of process faults, Anal. Chim. Acta 331 (1996) 187–193.

[8] F. Yacoub, J. MacGregor, Product optimization and control in the latent variable space of nonlinear PLS models, Chemometr. Intell. Lab. 70 (2004) 63–74.

[9] J. Gasteiger, J. Zupan, Neural networks in chemistry, Angew. Chem. Int. Ed. Engl. 32 (1993) 503–527.

[10] B. Schölkopf, A. Smola, Learning with Kernels, 1st Edition, MIT Press, Cambridge, UK, 2002.

[11] H. Li, Y. Liang, Q. Xu, Support vector machines and its applications in chemistry, Chemometr. Intell. Lab. 95 (2009) 188–198.

[12] P. Williams, Influence of water on prediction of composition and quality factors: the aquaphotomics of low moisture agricultural materials, J. Near Infrared Spectroscop. 17 (2009) 315–328.

[13] C. Tan, M. Li, Mutual information-induced interval selection combined with kernel partial least squares for near-infrared spectral calibration, Spectrochim. Acta Pt. A-Mol. Biomol. Spectrosc. 71 (2008) 1266–1273.

[14] M. Embrechts, S. Ekins, Classification of metabolites with kernel-partial least squares (K-PLS), Drug Metab. Dispos. 35 (2007) 325–327.

[15] J. Arenas-Garcia, G. Camps-Valls, Efficient kernel orthonormalized PLS for remote sensing applications, IEEE Trans. Geosci. Remote Sens. 46 (2008) 2872–2881.

[16] V. Struc, N. Pavesic, Gabor-based kernel partial-least-squares discrimination features for face recognition, Informatica 20 (2009) 115–138.

[17] R. Sun, F. Tsung, A kernel-distance-based multivariate control chart using support vector methods, Int. J. Prod. Res. 41 (2003) 2975–2989.

[18] J. Lee, C. Yoo, S. Choi, P. Vanrolleghem, I. Lee, Nonlinear process monitoring using kernel principal component analysis, Chem Eng. Sci. 59 (2004) 223–234.

[19] R. Vitale, O. de Noord, A. Ferrer, A kernel-based approach for fault diagnosis in batch processes, J. Chemometr. 28 (2014) 697–707.

[20] T. Kourti, J. MacGregor, Multivariate SPC methods for process and product monitoring, J. Qual. Technol. 28 (1996) 409–428.

[21] K. Bennett, M. Embrechts, Advances in Learning Theory: Methods, Models and Applications, 1st Edition, IOS Press, Amsterdam, The Netherlands, 2003.

[22] R. Kewley, M. Embrechts, C. Breneman, Data strip mining for the virtual design of pharmaceuticals with neural networks, IEEE Trans. Neural Netw. 11 (2000) 668–679.

[23] B. Üstün, W. Melssen, L. Buydens, Visualization and interpretation of support vector regression models, Anal. Chim. Acta 595 (2007) 299–309.

[24] C. Alcala, S. Qin, Reconstruction-based contribution for process monitoring with kernel principal component analysis, Ind. Eng. Chem. Res. 49 (2010) 7849–7857.

[25] J. Gower, S. Hardings, Nonlinear biplots, Biometrika 75 (1988) 445–455.

[26] P. Krooshof, B. Üstün, G. Postma, L. Buydens, Visualisation and recovery of the (bio)chemical interesting variables in data analysis with support vector machine classification, Anal. Chem. 82 (2010) 7000–7007.

[27] G. Postma, P. Krooshof, L. Buydens, Opening the kernel of kernel partial least squares and support vector machines, Anal. Chim. Acta 705 (2011) 123–134.

[28] A. Smolinska, L. Blanchet, L. Coulier, K. Ampt, T. Luider, R. Hintzen, S. Wijmega, L. Buydens, Interpretation and visualization of nonlinear data fusion in kernel space: study on metabolomic characterization of progression of multiple sclerosis, PLoS One 7 (2012) e38163.

[29] O. de Noord, Improvements to multivariate data analysis and monitoring of batch processes by multilevel methods, J. Chemometr. 26 (2012) 340–344.

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

[31] S. García-Muñoz, T. Kourti, J. MacGregor, A. Mateos, G. Murphy, Troubleshooting of an industrial batch process using multivariate methods, Ind. Eng. Chem. Res. 42 (2003) 3592–3601.

[32] N. Tracy, J. Young, R. Mason, Multivariate control charts for individual observations, J. Qual. Technol. 24 (1992) 88–95.

[33] G. Box, Some theorems on quadratic forms applied in the study of analysis of variance problems: effect of inequality of variance in one-way classification, Ann. Math. Stat. 25 (1954) 290–302.

[34] J. Camacho, A. Ferrer, Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical

aspects, J. Chemometr. 26 (2012) 361–373.

[35] J. Westerhuis, S. Gurden, A. Smilde, Generalized contribution plots in multivariate statistical process monitoring, Chemometr. Intell. Lab. 51 (2000) 95–114.

Appendix A.

Let $\mathbf{X}_{(N\times M)}$ be a generic data matrix. The centered matrix, $\tilde{\mathbf{X}}$, is defined as:

$$\tilde{\mathbf{X}} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}})\mathbf{X} \tag{A.1}$$

The linear kernel of $\tilde{\mathbf{X}}$ corresponds to the inner product matrix, $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathrm{T}}$, which is proven to be equivalent to the double-centered Euclidean distance matrix, $\mathbf{B}$ [19]:

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{D}\mathbf{H}^{\mathrm{T}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathrm{T}} \tag{A.2}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ and $\mathbf{D}$ is the Euclidean distance matrix, obtained from the original measured variables.
Performing a Singular Value Decomposition (SVD) on $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathrm{T}}$ and extracting only the first eigenvector yields:

$$\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{v}_1 = \lambda_1\mathbf{v}_1 \tag{A.3}$$

$\mathbf{v}_1$ is also equal to the first normalized scores vector obtained by executing SVD on $\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}$:

$$\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \tag{A.4}$$

$$\tilde{\mathbf{X}}\mathbf{u}_1 = \mathbf{t}_1 \tag{A.5}$$

$$\mathbf{v}_1 = \frac{\mathbf{t}_1}{||\mathbf{t}_1||} = \frac{\mathbf{t}_1}{\sqrt{\lambda_1}} = \frac{\tilde{\mathbf{X}}\mathbf{u}_1}{\sqrt{\lambda_1}} \tag{A.6}$$

where:

$$||\mathbf{t}_1|| = \sqrt{\mathbf{t}_1^{\mathrm{T}}\mathbf{t}_1} = \sqrt{\mathbf{u}_1\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\mathbf{u}_1} = \sqrt{\lambda_1\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1} = \sqrt{\lambda_1} \tag{A.7}$$

The projection of the kernel vector calculated by multiplying the pseudo-sample $\mathbf{s}^{\mathrm{T}} = [0, 0, \ldots, 1, 0, \ldots, 0]_{(1\times M)}$ by $\tilde{\mathbf{X}}^{\mathrm{T}}$ onto the space defined by $\mathbf{v}_1$ is proven to be equal to:

$$\mathbf{s}^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{v}_1 = \mathbf{s}^{\mathrm{T}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\frac{\mathbf{u}_1}{\sqrt{\lambda_1}} = \mathbf{s}^{\mathrm{T}}\frac{\lambda_1\mathbf{u}_1}{\sqrt{\lambda_1}} = \sqrt{\lambda_1}\mathbf{s}^{\mathrm{T}}\mathbf{u}_1 = \sqrt{\lambda_1}u_{1,m} \tag{A.8}$$

and thus contains information about the weight of the m-th variable of the original data matrix. This demonstrates that pseudo-sample projection can be used as a tool for recovering the information related to the weights of the M original measured variables on a particular principal component of a PCA model built on the simplest possible kernel matrix (i.e. the double-centered Euclidean distance matrix). However, as stated in [25], the same property is valid for all those kernel transformations generating sets of distance, which may be embedded in a Euclidean space. Since the polynomial kernel is calculated as function of the linear one (which is equal to the double-centered Euclidean distance matrix, **B**) and the Gaussian as function of the Euclidean distance, their Euclidean nature is verified.

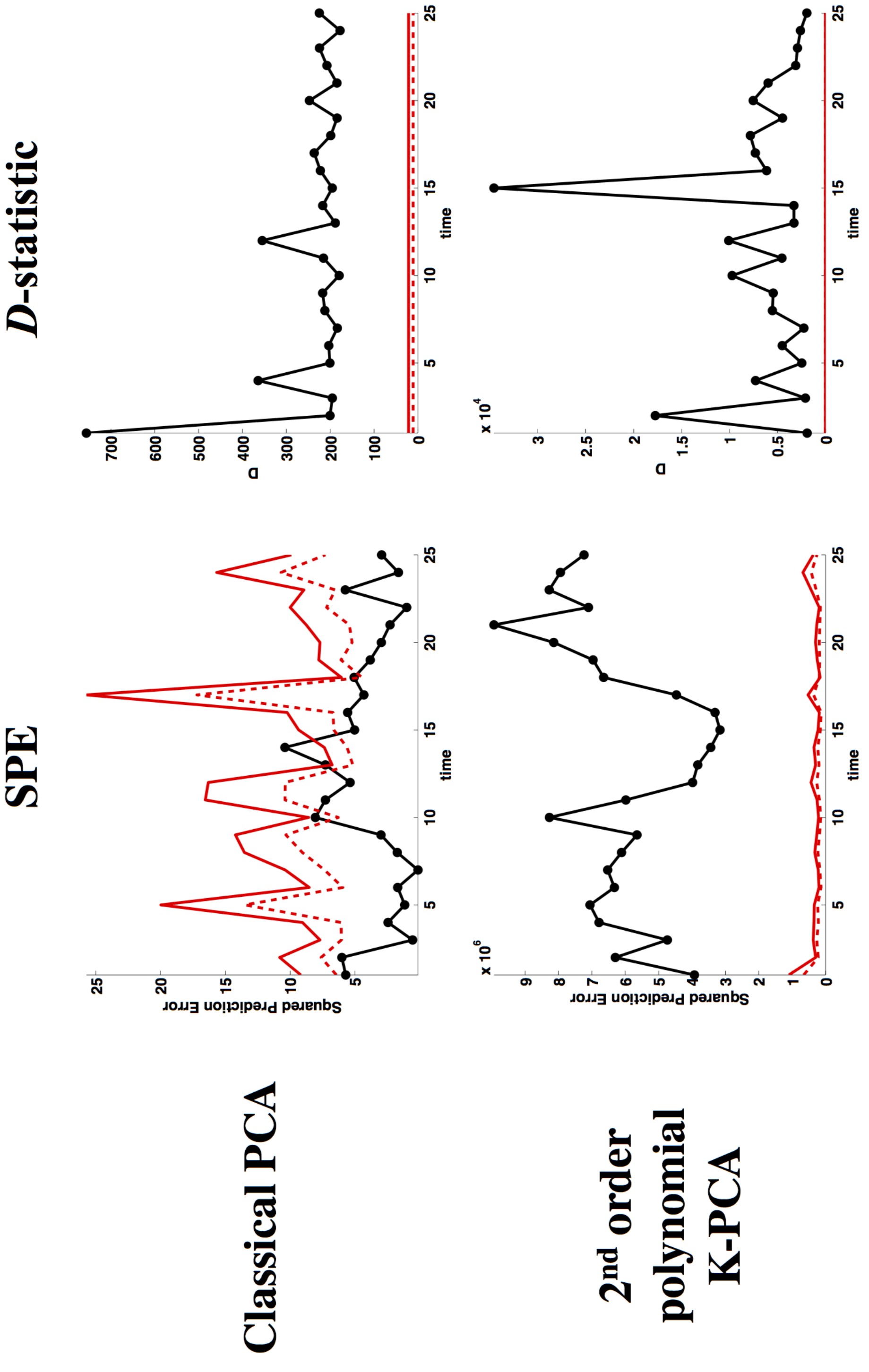

SPE
D-statistic
Classical PCA
2nd order polynomial K-PCA
Squared Prediction Error
D
time
x 10^6
x 10^4
0
5
10
15
20
25
100
200
300
400
500
600
700
0.5
1
1.5
2
2.5
3
1
2
3
4
5
6
7
8
9

Figure 1

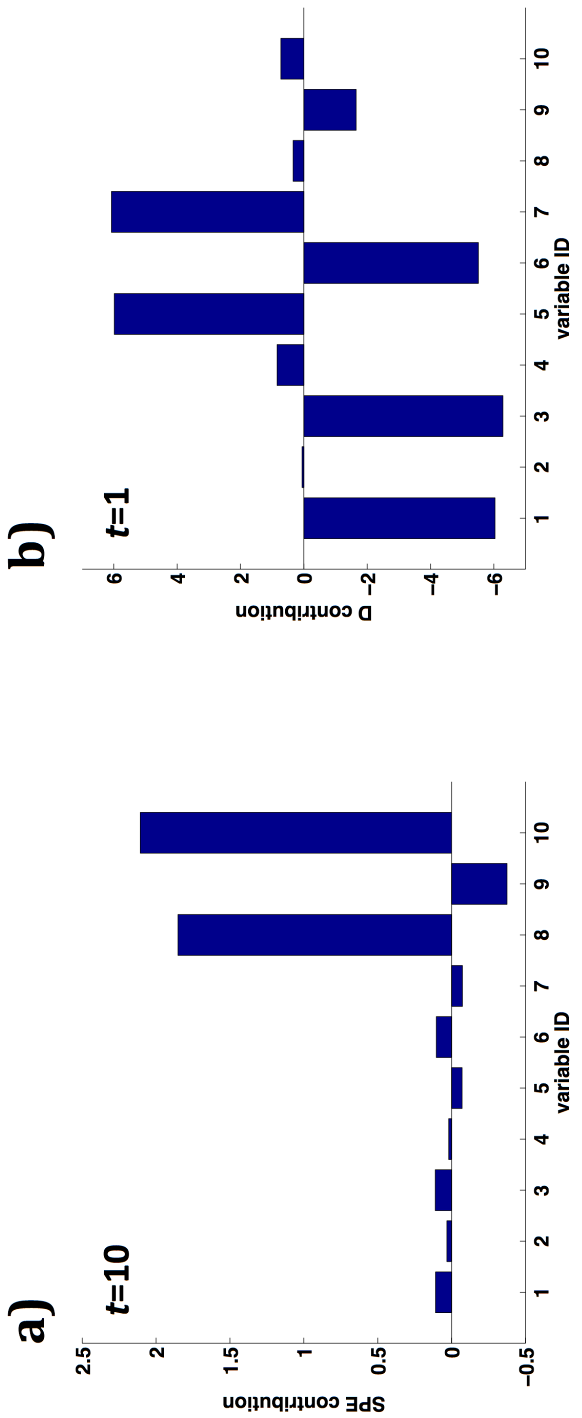

a)
t=10
SPE contribution
2.5
2
1.5
1
0.5
0
−0.5
variable ID
1
2
3
4
5
6
7
8
9
10
b)
t=1
D contribution
6
4
2
0
−2
−4
−6
variable ID
1
2
3
4
5
6
7
8
9
10

Figure 2

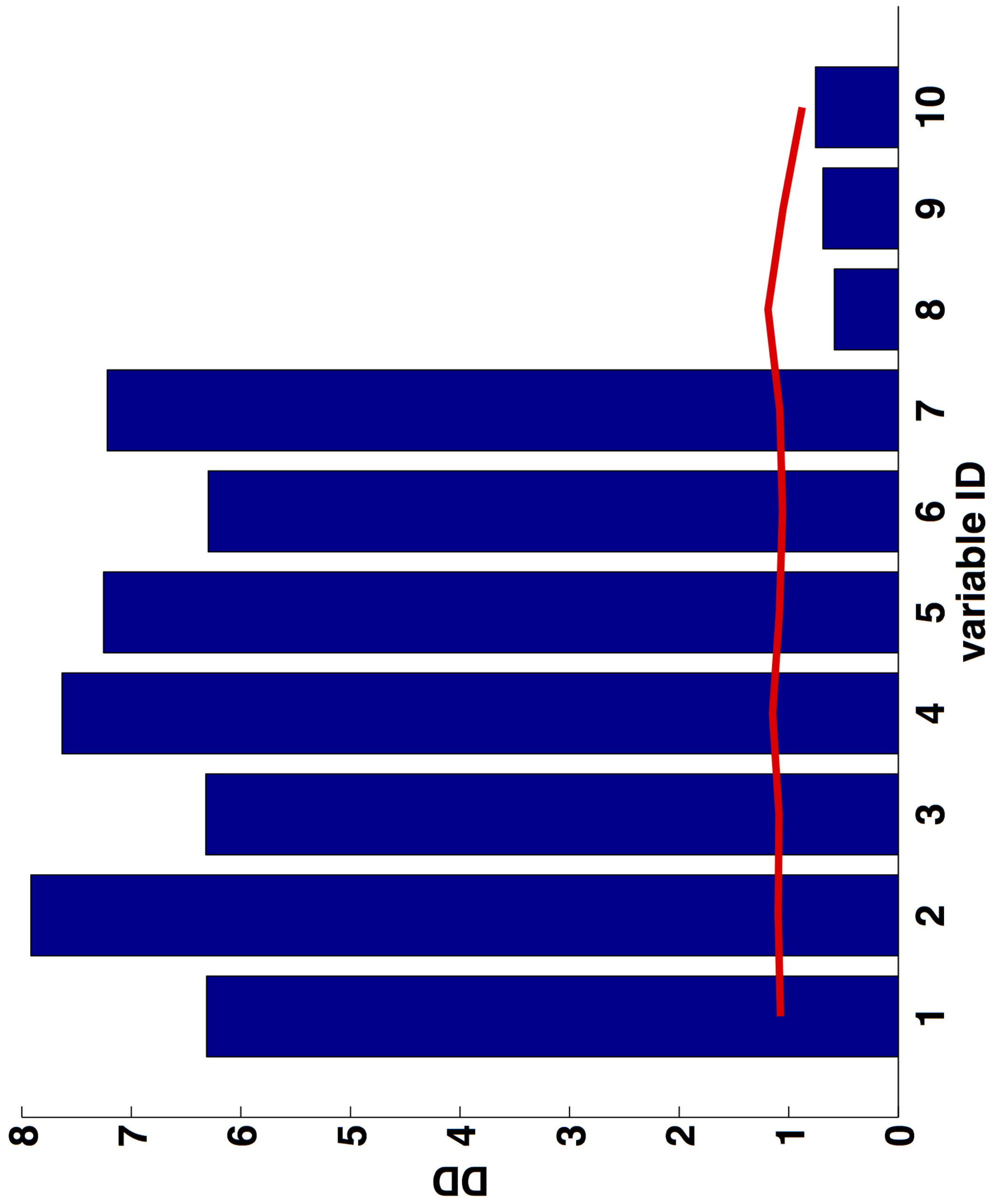


Figure 3

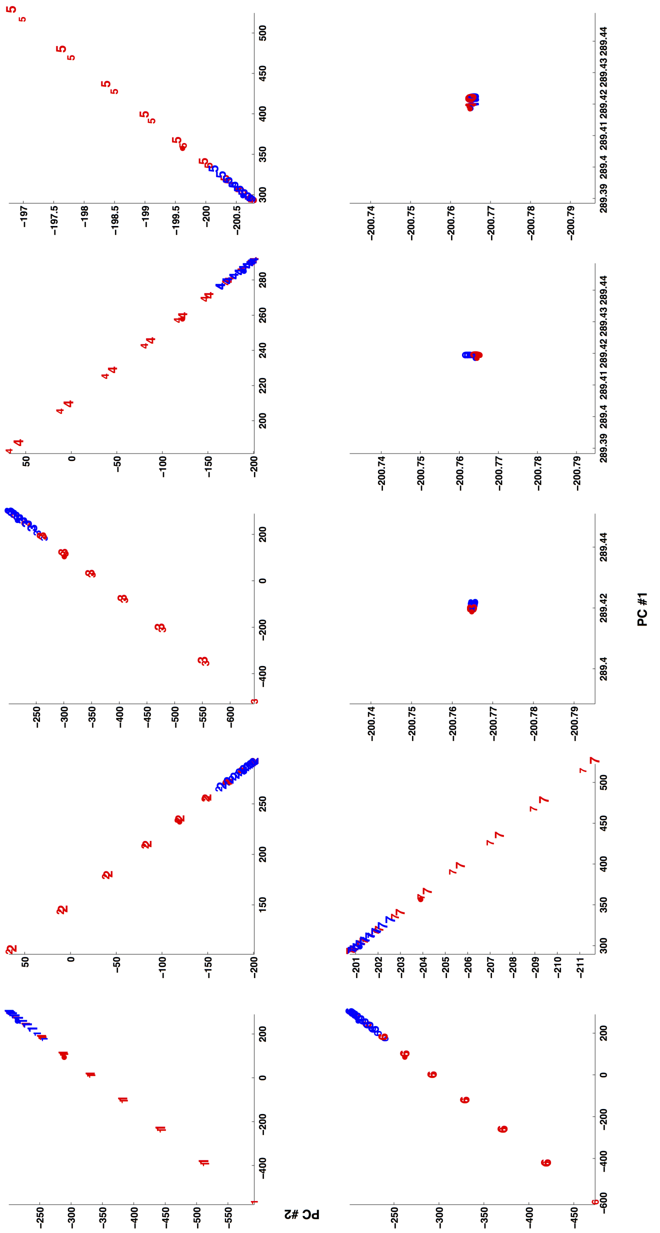


Figure 4

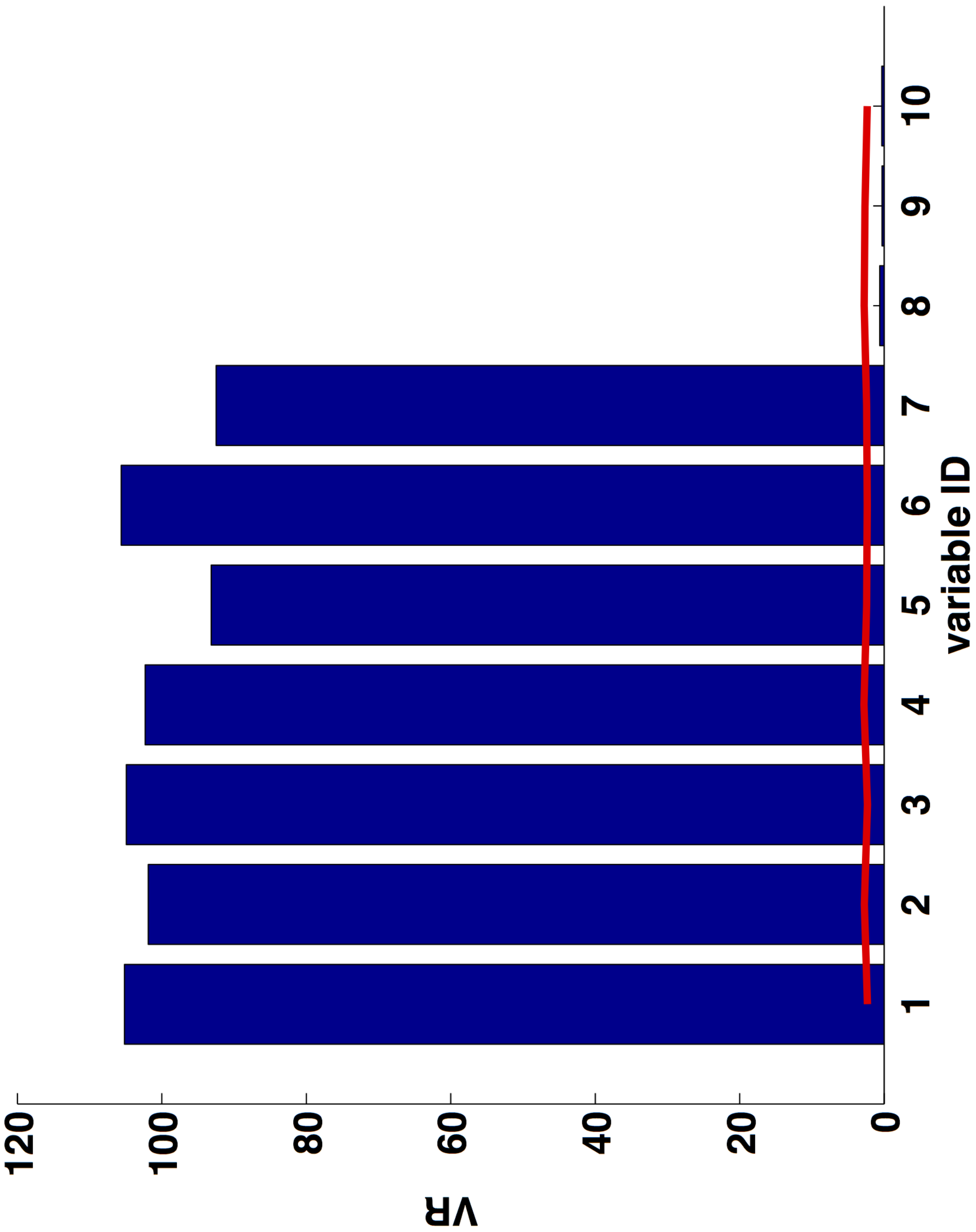


Figure 5

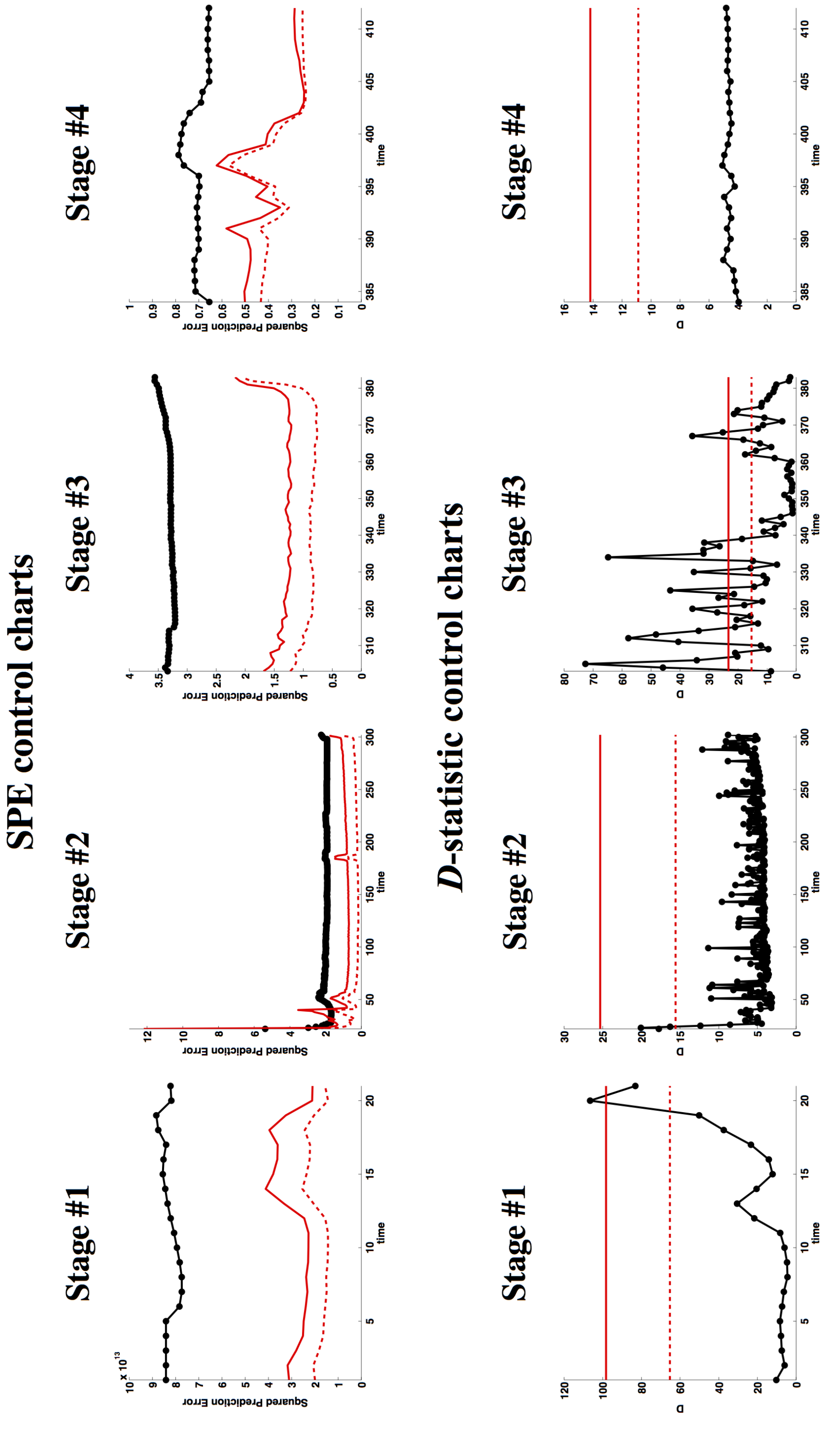

SPE control charts
Stage #1
Stage #2
Stage #3
Stage #4
Squared Prediction Error
time
x 10^13
D-statistic control charts
D
0
5
10
15
20
25
30
40
50
60
70
80
100
120
250
300
310
320
330
340
350
360
370
380
385
390
395
400
405
410

Figure 6

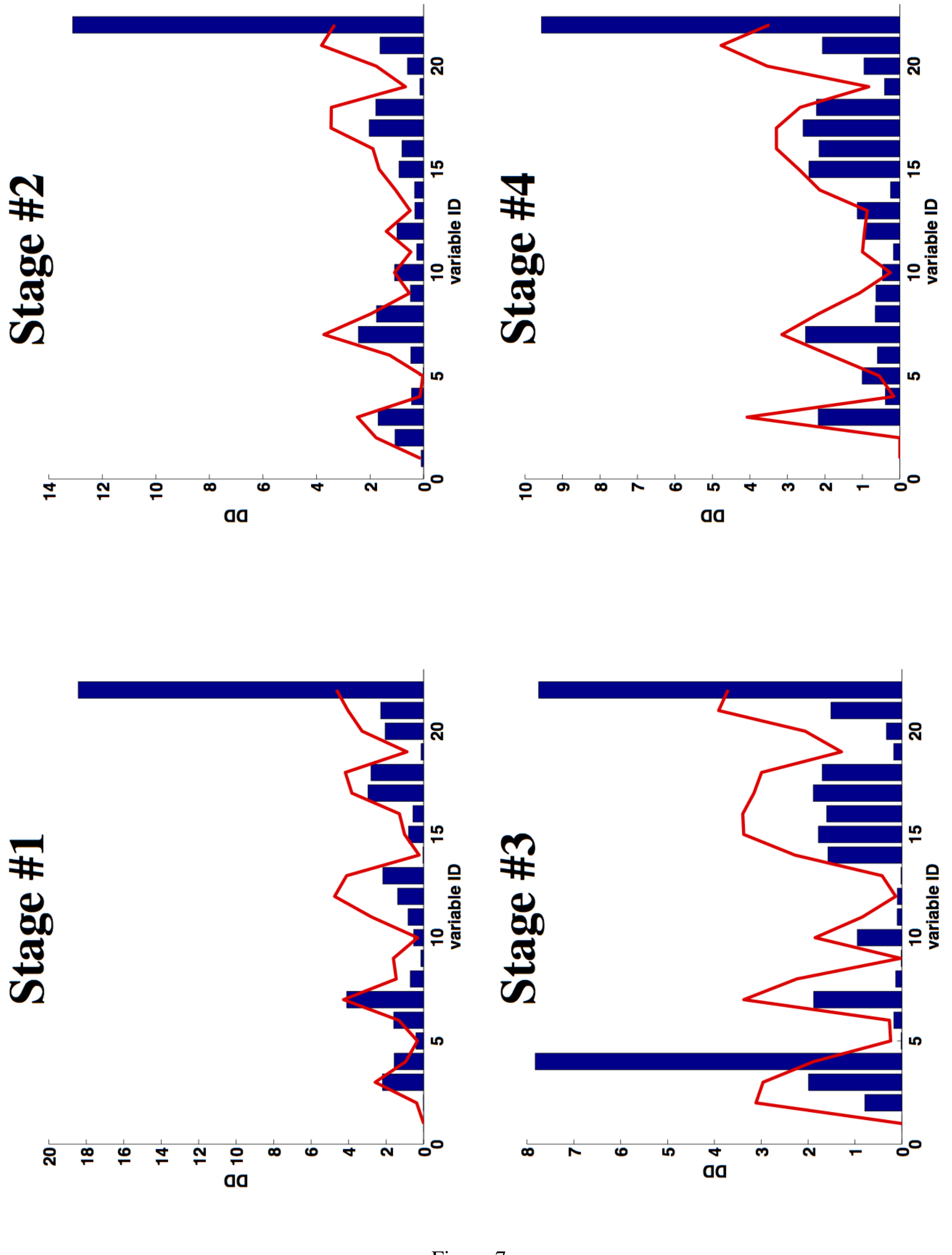

Stage #1
DD
0
2
4
6
8
10
12
14
16
18
20
variable ID
0
5
10
15
20
Stage #2
DD
0
2
4
6
8
10
12
14
variable ID
0
5
10
15
20
Stage #3
DD
0
1
2
3
4
5
6
7
8
variable ID
0
5
10
15
20
Stage #4
DD
0
1
2
3
4
5
6
7
8
9
10
variable ID
0
5
10
15
20

Figure 7

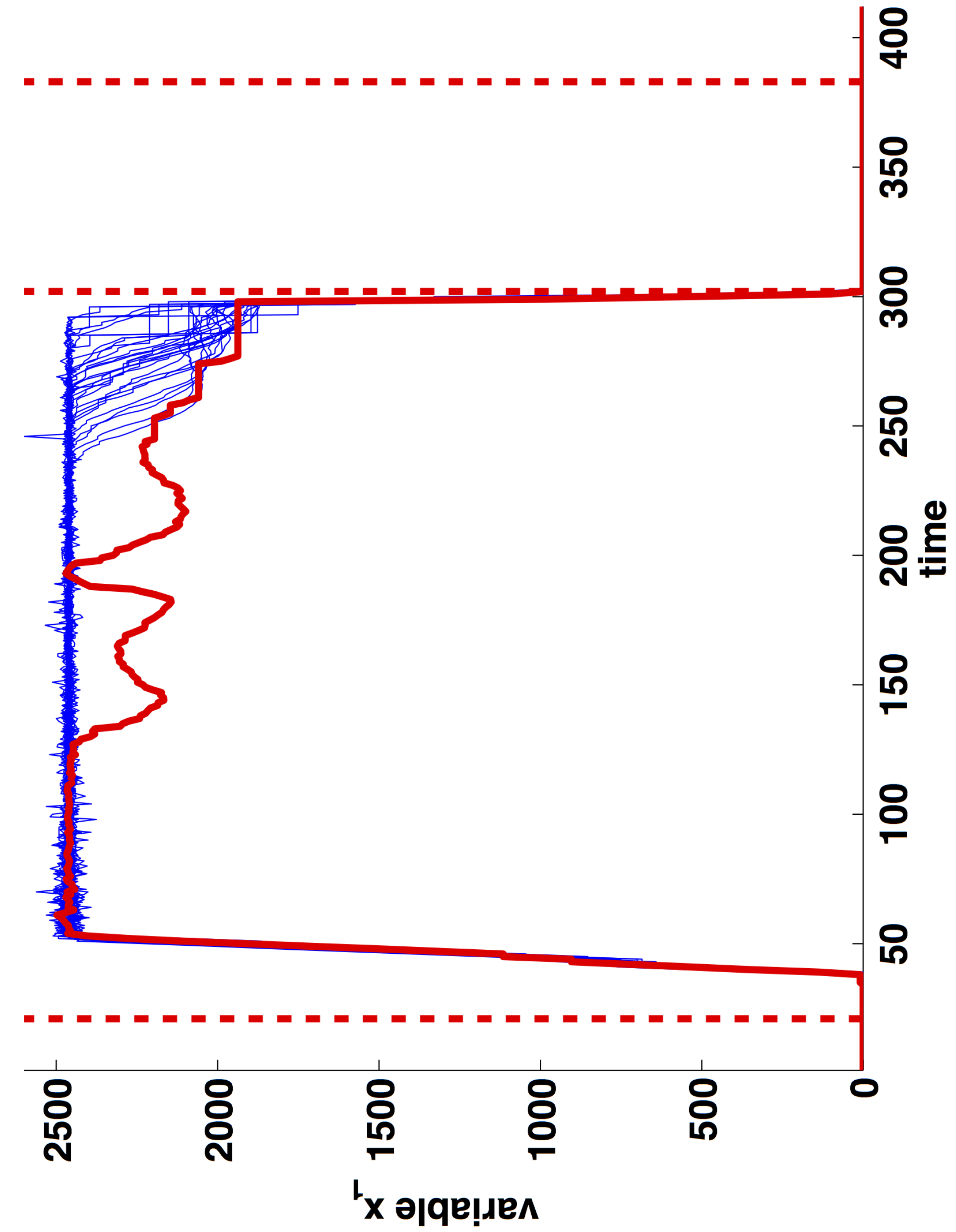

variable x_1
2500
2000
1500
1000
500
0
50
100
150
200
250
300
350
400
time

Figure 8

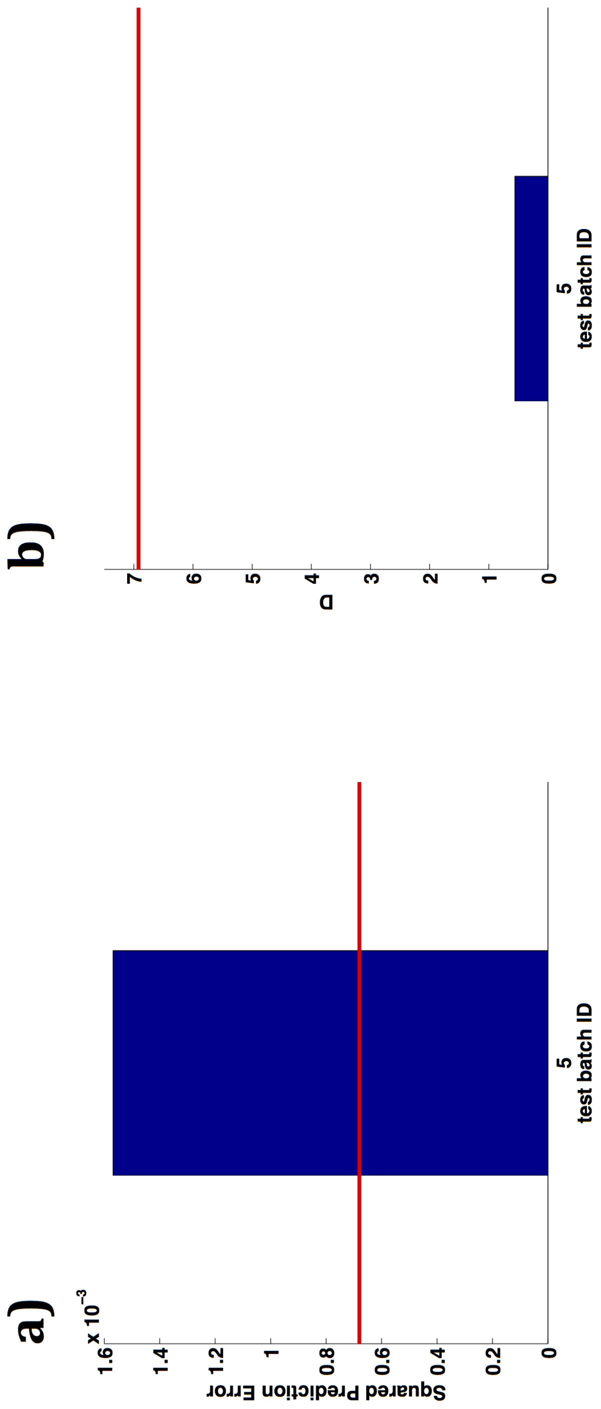

a)
Squared Prediction Error
x 10^−3
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
5
test batch ID
b)
D
7
6
5
4
3
2
1
0
5
test batch ID

Figure 9

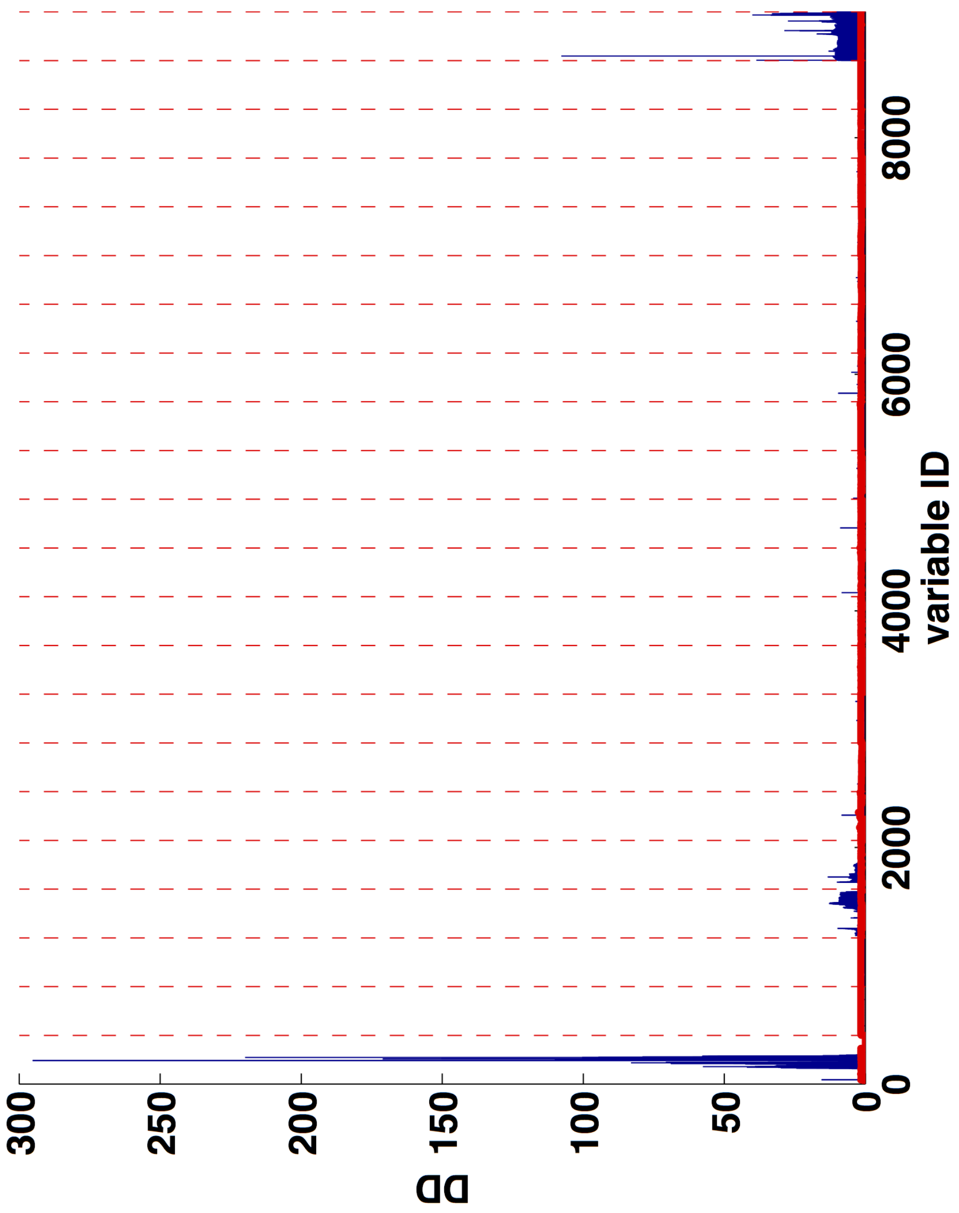


Figure 10

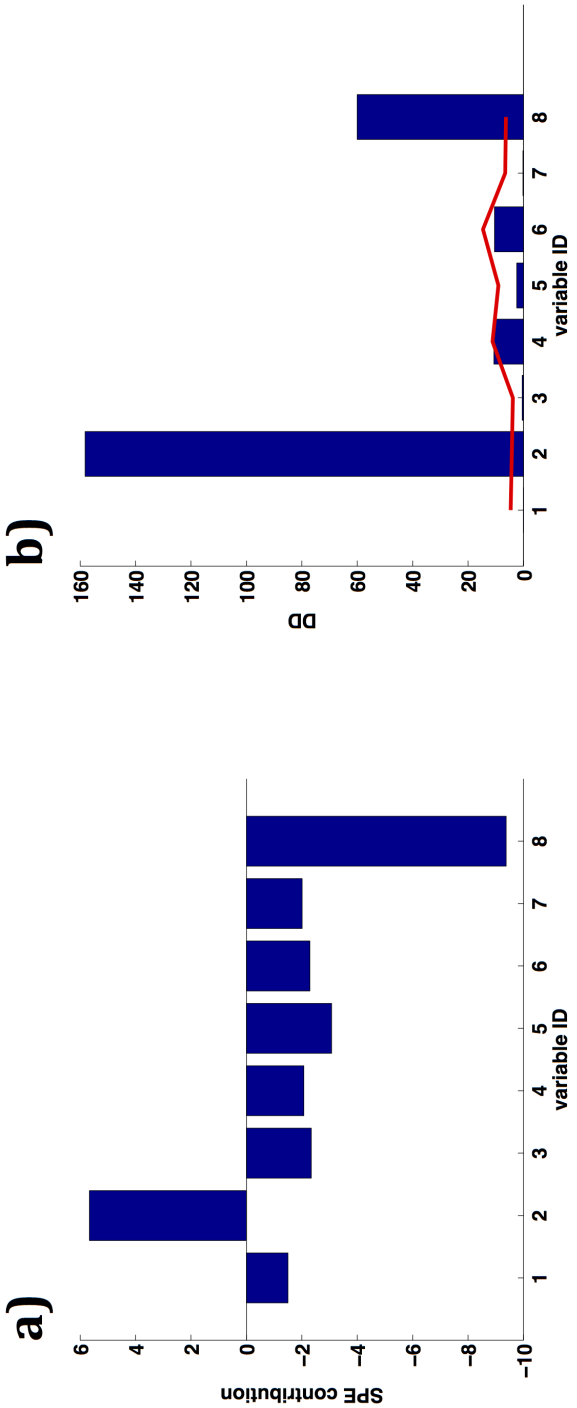

a)
SPE contribution
6
4
2
0
−2
−4
−6
−8
−10
1
2
3
4
5
6
7
8
variable ID
b)
DD
160
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
variable ID

Figure 11

Figure captions

- Figure 1: SPE and D-statistic control charts obtained by both classical PCA and K-PCA for the first faulty test batch. The dotted and the solid red lines represent the 95% and the 99% control limits, respectively

- Figure 2: Classical PCA: a) SPE and b) D-statistic contribution plots related to first sampling point of the first faulty test batch found to be beyond the 95% control limit in the corresponding PCA-based control charts (time instant #10 and #1, respectively)

- Figure 3: K-PCA: Overall *DD* plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure

- Figure 4: K-PCA: Original pseudo-sample trajectories obtained by the procedure described in Section 2.4. The scores represented as blue numbers are calculated starting from the minimum and the maximum values each variable assumes in all the training NOC batches. The scores represented as red numbers are calculated starting from the minimum and the maximum values each variable assumes in the considered *out-of-control* test batch. Their graphical comparison permits to evaluate whether the original variables assume values either inside or outside their *in-control* variability range, during the occurrence of a fault

- Figure 5: K-PCA: Overall *VR* plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the *VR* values of every variable by a jackknife-based procedure

- Figure 6: K-PCA: SPE and D-statistic control charts related to every stage of the evolution of the fifth faulty batch contained in the second test set (VCS-VWU). The dotted and the solid red lines represent the 95% and the 99% control limits, respectively

- Figure 7: K-PCA: Overall stage-*DD* plots related to the fifth faulty test batch (VCS-VWU). The solid red lines represent the 95% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure

- Figure 8: Temporal evolution of the variable x_1 of the chemical process dataset in the NOC batches (blue thin lines) and in the fifth faulty process run contained in the second test set (red thick line). The vertical dotted red lines mark the end point of every process stage

- Figure 9: K-PCA: a) SPE and b) *D*-statistic control charts related to the fifth faulty batch contained in the second test set (TCS-BWU). The solid red lines represent their respective 90.9% control limit

- Figure 10: K-PCA: Overall *DD* plot related to the fifth faulty test batch (TCS-BWU). The solid red line represents the 95% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure. The vertical dotted red lines mark the separation between the *DD* values at the different sampling times of two consecutive process variables

- Figure 11: a) SPE contribution plot (classical PCA) and b) *DD* plot (K-PCA) related to first faulty test batch. The solid red line represents the 94.1% confidence limit calculated for the *DD* values of every variable by a jackknife-based procedure

Table 1

Kernel type	Kernel function	Adjustable parameters
Linear	$\mathbf{x}_a^{\mathrm{T}}\mathbf{x}_{a'}$	None
2nd-order polynomial	$(\mathbf{x}_a^{\mathrm{T}}\mathbf{x}_{a'})^2$	None
3rd-order polynomial	$(\mathbf{x}_a^{\mathrm{T}}\mathbf{x}_{a'})^3$	None
Gaussian	$\exp(-\frac{\|\mathbf{x}_a-\mathbf{x}_{a'}\|^2}{2\sigma})$	σ

Table 2

	PCs	R^2	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	2	0.705	6.4%	1.6%	4.8%	1.6%
K-PCA (second-order polynomial)	2	0.743	6.4%	0.8%	4.0%	0.8%

Table 3

	$\mathrm{SPE}_{\mathrm{ISL}=5\%}$	$\mathrm{SPE}_{\mathrm{ISL}=1\%}$	$D_{\mathrm{ISL}=5\%}$	$D_{\mathrm{ISL}=1\%}$
Classical PCA	79.5%	89.1%	0.3%	5.6%
K-PCA (second-order polynomial)	0.3%	1.1%	0.0%	0.0%

Table 4

		Variability increase		Variability decrease	
		Fault detection	Fault diagnosis	Fault detection	Fault diagnosis
VCS-VWU	PCA	*D*	NO	NO	NO
	K-PCA	*D*, SPE	*DD*, *VR*	NO	NO
TCS-VWU	PCA	*D*, SPE	NO	*D*	NO
	K-PCA	*D*, SPE	*DD*, *VR*	*D*	*DD*, *VR*
TCS-BWU	PCA	*D*, SPE	*D*	*D*	*D*
	K-PCA	*D*, SPE	*DD*	*D*	*DD*

Table 5

	Stage #1	Stage #2	Stage #3	Stage #4
Function Type	3-rd order polynomial	Gaussian	Gaussian	Gaussian
CV-σ	-	23.8	15.0	12.8
PCs	3	2	2	2
R^2	0.852	0.850	0.798	0.775

Table 6

	PCs	R^2	$\text{SPE}_{\text{ISL}=11.1\%}$	$D_{\text{ISL}=11.1\%}$
Classical PCA	2	0.790	11.1%	11.1%
K-PCA (third-order polynomial)	2	0.830	11.1%	11.1%

Table 7

	$\mathrm{SPE}_{\mathrm{ISL}=11.1\%}$	$D_{\mathrm{ISL}=11.1\%}$
Classical PCA	30.4%	69.2%
K-PCA (third-order polynomial)	17.4%	65.2%

Table captions

- Table 1: Kernel functions referred to in this article and list of their adjustable parameters

- Table 2: Overall Type I (*OTI*) risk values for the SPE and the *D*-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

- Table 3: Overall Type II (*OTII*) risk values for the SPE and the *D*-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

- Table 4: Summary of the results obtained in the different simulated case-studies. The table shows which of the classical PCA- or K-PCA-based control charts and contribution plots were able to detect and diagnose the considered faults, respectively

- Table 5: Parameters of the K-PCA *in-control* models built for every process stage

- Table 6: Type I (*TI*) risk values for the SPE and the *D*-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

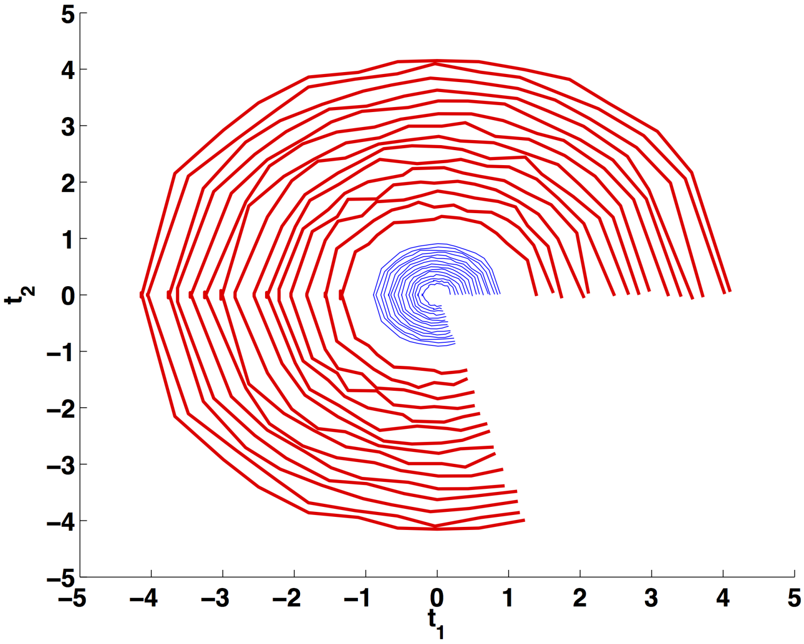
- Table 7: Type II (*TII*) risk values for the SPE and the *D*-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

Pseudo-sample based contribution plots: innovative tools for fault diagnosis in kernel-based batch process monitoring

Supporting Material

Section 1 - Data simulation

The data simulation was carried out generating a 750x2 set of scores, which define 2 classes of 15 different trimmed circular profiles of 25 observations each, as shown in SM.1.

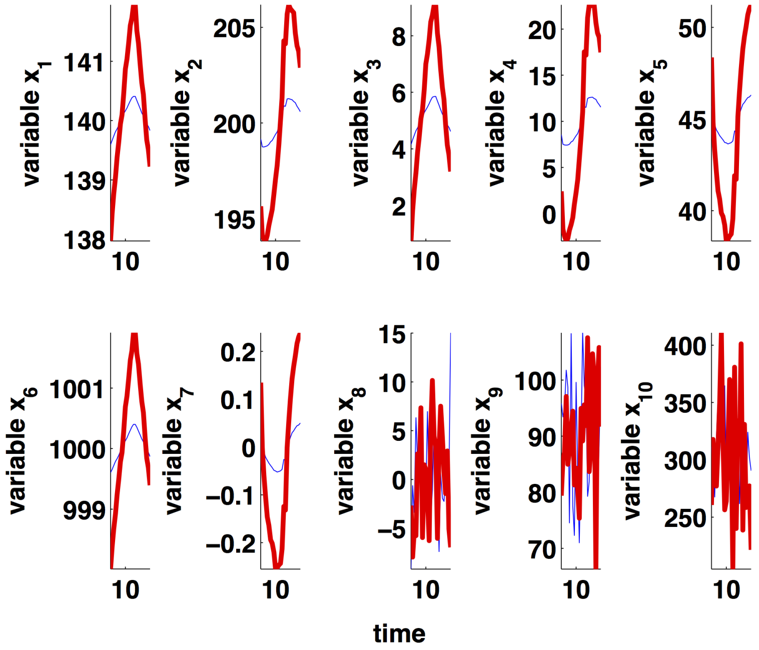


SM. 1: On- (blue thin line) and off-specification (red thick line) simulated batch scores trajectories

Every trimmed circular profile represents a proper batch score trajectory, which may describe its progression in the latent variable space of a 2-

component PCA model built on a VWU data array. Multiplying this set of scores by a 2x7 transposed matrix of loadings (obtained performing PCA on real process data), a 750x7 dataset was constructed, which contains the trajectories of 7 variables in 30 different runs, constituted by 25 time samples each. Three noisy variables were also included for verifying the effectiveness of the pseudo-sample-based approach for diagnostic purposes.

As shown in SM. 2, the data simulation generated off-specification batches characterized by an increase in the variance of variables x_1-x_7, but not in their mean value with respect to the NOC ones.



SM. 2: Temporal evolution of the variables of the simulated dataset for a NOC (blue thin line) and a faulty (red thick line) batch

Section 2 - Variability increase detection case study

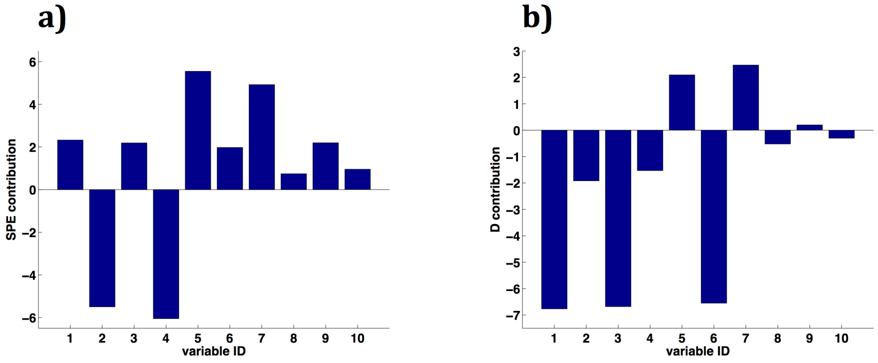
TCS-VWU

Table SM. 1: Overall Type I (OTI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

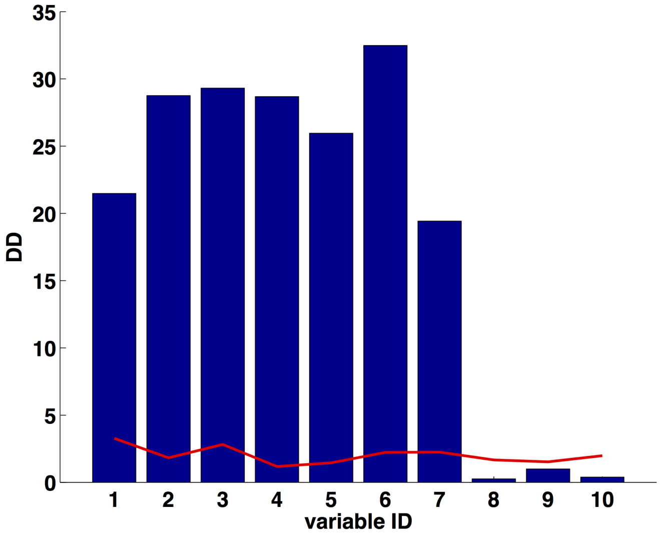
	PCs	R^2	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	2	0.631	6.4%	1.6%	4.8%	1.6%
K-PCA (second-order polynomial)	2	0.729	6.4%	1.6%	5.6%	1.6%

Table SM. 2: Overall Type II ($OTII$) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

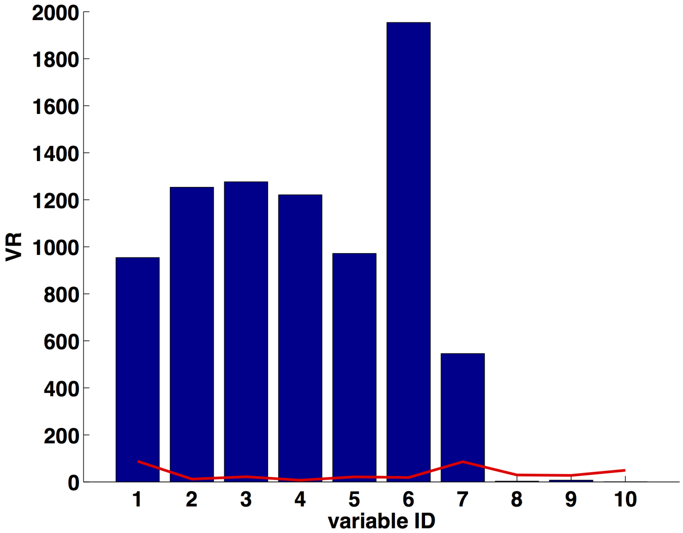
	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	6.9%	13.6%	0.0%	0.0%
K-PCA (second-order polynomial)	0.0%	0.0%	0.0%	0.0%



SM. 3: Classical PCA: a) SPE and b) D-statistic contribution plots related to the first sampling point of the first faulty test batch. The displayed profiles are consistent with those observed for all the other *out-of-control* signals



SM. 4: K-PCA: Overall DD plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the DD values of every variable by a jackknife-based procedure



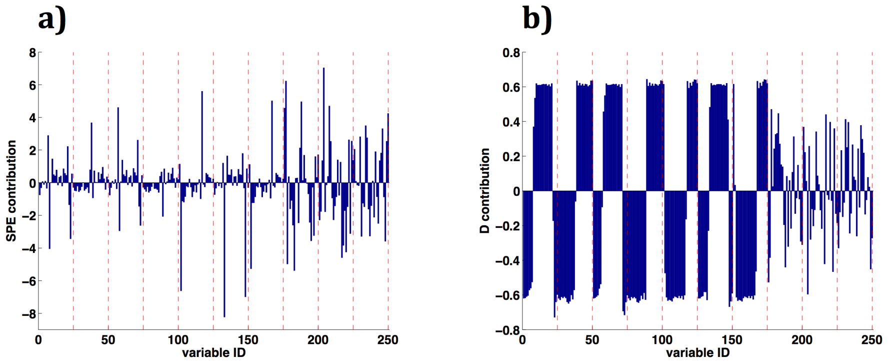
SM. 5: K-PCA: Overall VR plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the VR values of every variable by a jackknife-based procedure

Table SM. 3: Type I (TI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

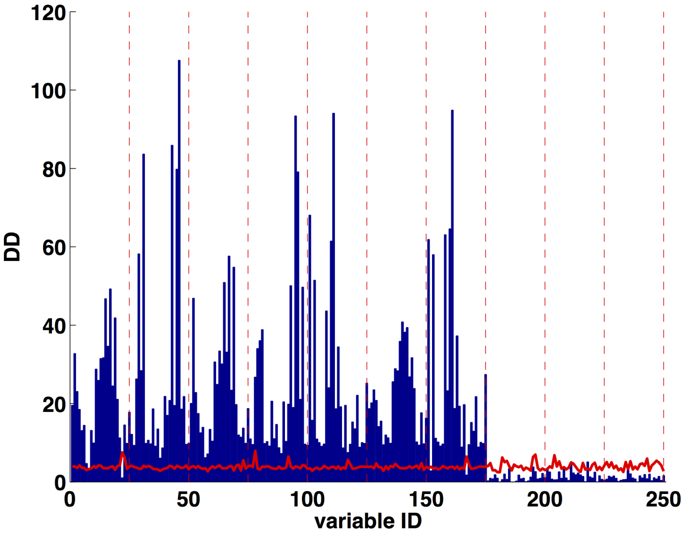
	PCs	R^2	$\mathrm{SPE}_{\mathrm{ISL}=20\%}$	$D_{\mathrm{ISL}=20\%}$
Classical PCA	2	0.746	20.0%	20.0%
K-PCA (Gaussian, σ=89.1)	2	0.976	20.0%	20.0%

Table SM. 4: Type II (TII) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

	$\mathrm{SPE}_{\mathrm{ISL}=20\%}$	$D_{\mathrm{ISL}=20\%}$
Classical PCA	0.0%	0.0%
K-PCA (Gaussian, σ=89.1)	0.0%	0.0%



SM. 6: Classical PCA: a) SPE and b) D-statistic contribution plots related to the first faulty test batch. The vertical dotted red lines mark the separation between the contributions at the different sampling times of two consecutive process variables



SM. 7: K-PCA: Overall DD plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the DD values of every variable at the different time instants by a jackknife-based procedure. The vertical dotted red lines mark the separation between the DD values at the different sampling times of two consecutive process variables

Section 3 - Variability decrease detection case study

VCS-VWU

Table SM. 5: Overall Type I (OTI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

	PCs	R^2	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	2	0.703	4.8%	1.6%	4.8%	1.6%
K-PCA (second-order polynomial)	2	0.697	4.8%	2.4%	6.4%	2.4%

Table SM. 6: Overall Type II ($OTII$) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

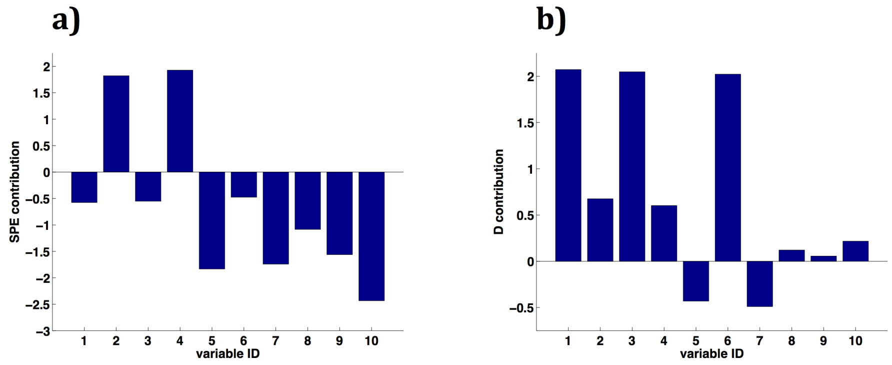
	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	96.5%	98.9%	51.7%	84.0%
K-PCA (second-order polynomial)	100%	100%	90.1%	100%

Table SM. 7: Overall Type I (OTI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

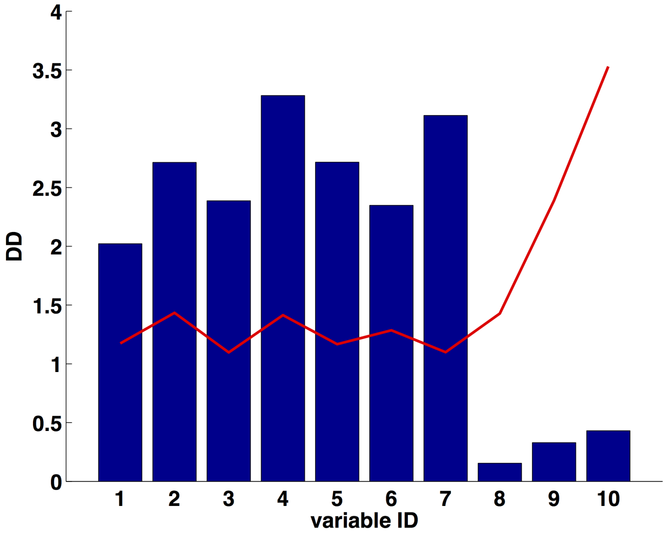
	PCs	R^2	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	2	0.638	4.8%	1.6%	4.8%	3.2%
K-PCA (second-order polynomial)	3	0.812	5.6%	2.4%	4.8%	1.6%

Table SM. 8: Overall Type II ($OTII$) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

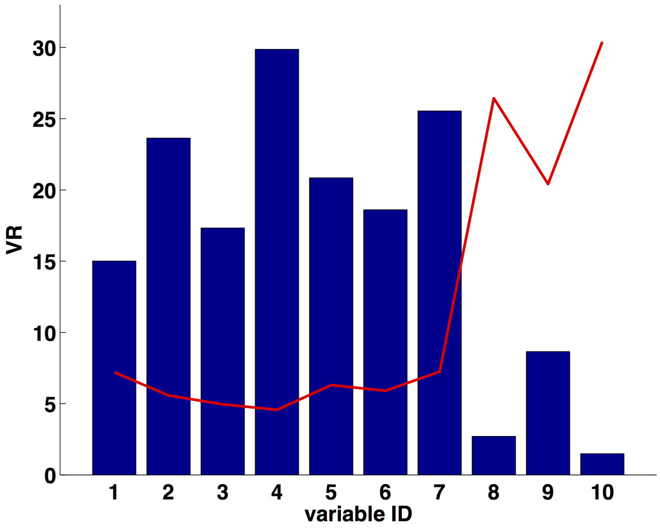
	$\text{SPE}_{\text{ISL}=5\%}$	$\text{SPE}_{\text{ISL}=1\%}$	$D_{\text{ISL}=5\%}$	$D_{\text{ISL}=1\%}$
Classical PCA	85.9%	91.7%	0.7%	24.5%
K-PCA (second-order polynomial)	56.3%	76.5%	0.0%	0.0%



SM. 8: Classical PCA: a) SPE and b) D-statistic contribution plots related to the first sampling point of the tenth faulty test batch. The displayed profiles are consistent with those observed for all the other *out-of-control* signals



SM. 9: K-PCA: Overall DD plot related to the tenth faulty test batch. The solid red line represents the 90% confidence limit calculated for the DD values of every variable by a jackknife-based procedure



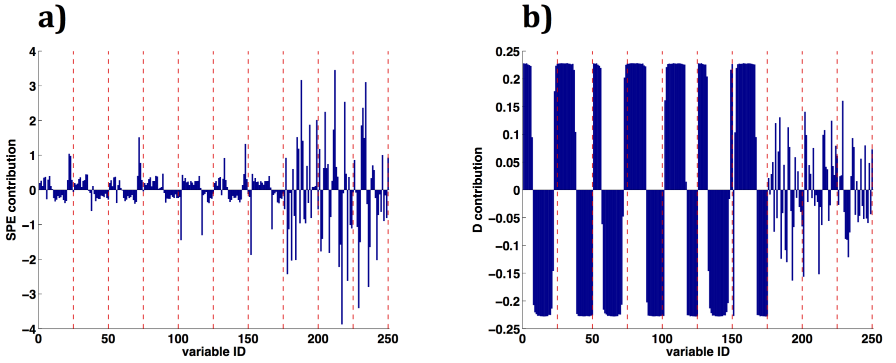
SM. 10: K-PCA: Overall VR plot related to the tenth faulty test batch. The solid red line represents the 90% confidence limit calculated for the VR values of every variable by a jackknife-based procedure

Table SM. 9: Type I (TI) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes. The table lists also the number of principal components (PCs) and the goodness of fit (R^2) of the two different models

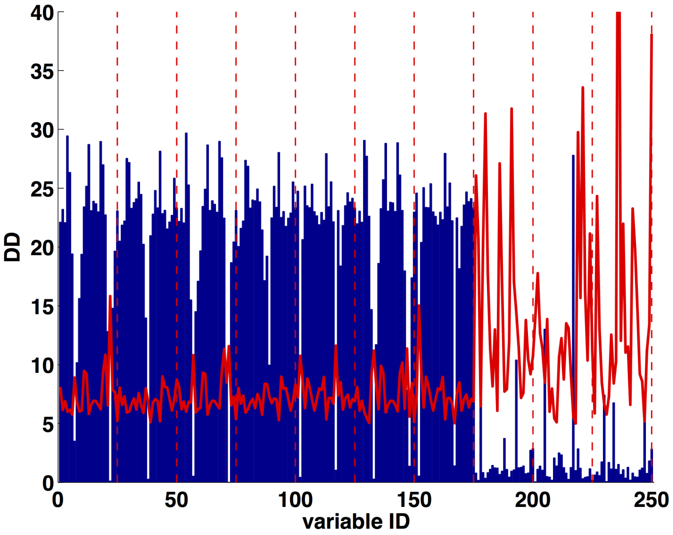
	PCs	R^2	$\text{SPE}_{\text{ISL}=20\%}$	$D_{\text{ISL}=20\%}$
Classical PCA	2	0.737	20.0%	20.0%
K-PCA (Gaussian, σ=19.3)	3	0.943	20.0%	20.0%

Table SM. 10: Type II (TII) risk values for the SPE and the D-statistic control charts resulting from both the classical PCA- and the K-PCA-based monitoring schemes

	$\text{SPE}_{\text{ISL}=20\%}$	$D_{\text{ISL}=20\%}$
Classical PCA	60.0%	0.0%
K-PCA (Gaussian, σ=19.3)	33.3%	0.0%

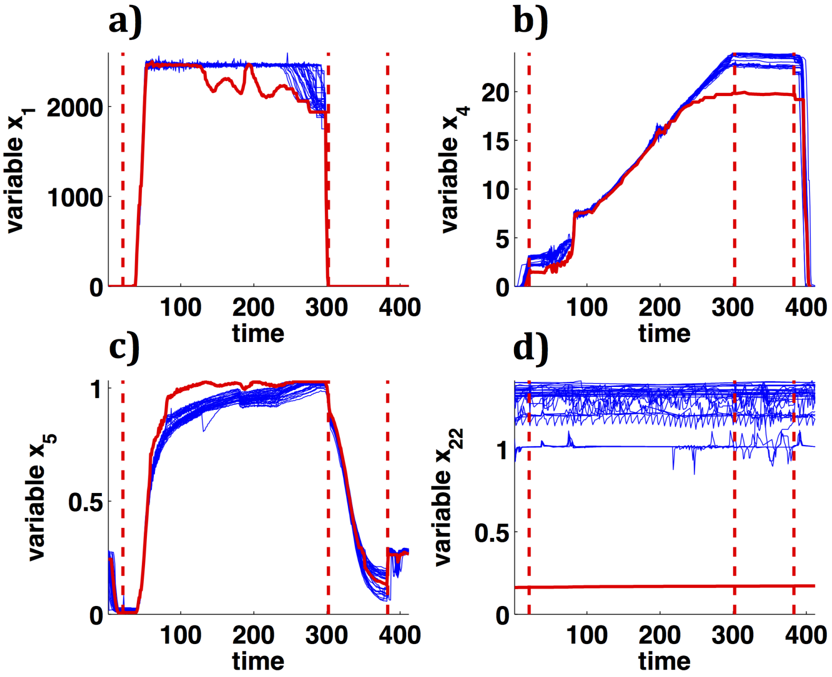


SM. 11: Classical PCA: a) SPE and b) D-statistic contribution plots related to the first faulty test batch. The vertical dotted red lines mark the separation between the contributions at the different sampling times of two consecutive process variables



SM. 12: K-PCA: Overall DD plot related to the first faulty test batch. The solid red line represents the 90% confidence limit calculated for the DD values of every variable at the different time instants by a jackknife-based procedure. The vertical dotted red lines mark the separation between the DD values at the different sampling times of two consecutive process variables

Section 4 - Further details on the chemical process dataset under study



SM. 13: Temporal evolution of the variables of the chemical process dataset, identified as having a high contribution to the fault, in the NOC batches (blue thin lines) and in the fifth faulty test process run (red thick line). The vertical dotted red lines mark the end point of every process stage. a) variable x_1; b) variable x_4; c) variable x_5; d) variable x_{22}