A Systematic Comparison of PCA-based Statistical Process Monitoring Methods for High-dimensional, Time-dependent Processes

Tiago Rato,[†] Eric Schmitt,[*,‡] Bart De Ketelaere,[¶] Mia Hubert,[‡] and Marco Reis[†]

CIEPQPF - Department of Chemical Engineering, University of Coimbra, Department of Mathematics, KU Leuven, and MeBioS - Department of Biosystems, KU Leuven

E-mail: eric.schmitt@wis.kuleuven.be

Abstract

High-dimensional and time-dependent data pose significant challenges to Statistical Process Monitoring (SPM). Most of the high-dimensional methodologies to cope with these challenges rely on some form of Principal Component Analysis (PCA) model, usually classified as non-adaptive and adaptive. Non-adaptive methods include the static PCA approach and Dynamic PCA for data with autocorrelation. Methods, such as Dynamic PCA with Decorrelated Residuals, extend Dynamic PCA to further reduce the effects of autocorrelation and cross-correlation on the monitoring statistics. Recursive PCA and Moving Window PCA, developed for non-stationary data, are adaptive. These fundamental methods will be systematically compared on high-dimensional, time-dependent processes (including the Tennessee Eastman benchmark process) to provide practitioners with guidelines for appropriate monitoring strategies

[*]To whom correspondence should be addressed
[†]CIEPQPF - Department of Chemical Engineering, University of Coimbra
[‡]Department of Mathematics, KU Leuven
[¶]MeBioS - Department of Biosystems, KU Leuven

and a sense of how they can be expected to perform. The selection of parameter values for the different methods is also discussed. Finally, the relevant challenges of modeling time-dependent data are discussed, and areas of possible further research are highlighted.

Keywords. Control charts, time-dependent data, high-dimensional data, Principal Component Analysis.

Introduction

Quality control charts are a widely used tool, developed in the field of Statistical Process Monitoring (SPM) to identify when a system is deviating from typical behavior. Some general reviews on this topic are provided by Woodall and Montgomery [1] and Qin [2]. A common scenario encountered in SPM is that of processes with a large number of variables and time-dependence in the form of autocorrelation and non-stationarity. These characteristics challenge many well known multivariate SPM methodologies, such as the Hotelling's T^2,[3] MEWMA [4] and MCUSUM,[5] leading to the widespread use of methods based on latent structures. One of the most important of these methods is Principal Component Analysis (PCA),[6,7] which is capable of performing monitoring tasks on high-dimensional processes, and accounting for time-dependence after some modifications of the base procedure. Applications come primarily from industry, where the goal is early fault detection in chemical or mechanical processes. Examples include catalytic cracking in petroleum refining,[8] furnace deterioration monitoring,[9] solid epoxy resin product quality,[10] and machine vibration monitoring.[11] A number of works in the literature provide an overview of PCA-based process monitoring or compare it to other methods,[8,9,11–13] but to our knowledge none provide a broad, cross-method coverage of the behavior of even the most basic methods when applied to time-dependent processes. Given the prevalence of precisely this type of data in fields such as industry, information technology, precision agriculture, health care and economy, this paper sets out to illustrate and compare the detection performance of fundamental meth-

ods on a collection of simple, but informative, high-dimensional, time-dependent processes. Working with such processes allows us to provide precise insights into the drivers of detection performance. We focus on simple, fundamental methods because these are the most likely to be used in practice, and their performance is indicative of that of extensions developed to obtain, for example, greater interpretability. A broad comparison of this sort allows us to examine the relative merits of these methods on a common basis, whereas currently one is obliged to assess them based on results from a heterogeneous collection of processes in different articles. Moreover, as some of the results obtained in this comparison study contradict expectations, revealing surprising monitoring behavior, having a clear understanding of the basic characteristics of each method is important for both practitioners and researchers developing extensions for them.

SPM aims to detect deviations from typical process behavior during two distinct phases of process measurement, called Phase I, and Phase II. Phase I monitoring is the practice of retrospectively evaluating whether a previously completed process was statistically in-control. On the other hand, Phase II monitoring is the practice of determining whether new observations from the process are in-control as they are obtained. During both phases, time dependence in the form of autocorrelation and/or non-stationarity can be present. Autocorrelation arises when the in-control measurements within one time series are not serially independent, while non-stationarity arises when the parameters governing a process, such as the mean or covariance, change over time. In this work, only the problem of Phase II monitoring will be addressed.

We assume that we observe a large number, p, of time series $\boldsymbol{x}_j(t_i)$, $(1 \leqslant j \leqslant p)$, typically corresponding to variables in the process, during a calibration period $t_1, t_2, \ldots, t_T$ that collectively constitute a high-dimensional data set. As time continues, more measurements become available. When the data are not time-dependent, control charts based on Principal Component Analysis (PCA) have been successfully applied in high-dimensional settings. These methods train a model on an existing data matrix $\boldsymbol{X}_{T,p}$, that is represen-

tative of typical process behavior. The j-th column contains the j-th time series $\boldsymbol{x}_j(t_i)$ for $1 \leqslant i \leqslant T$. The number of rows of $\boldsymbol{X}_{T,p}$ refers to the number of calibration observations, and p is the number of measured variables. Such methods compare a new observation at time t, $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \ldots, \boldsymbol{x}_p(t)]'$, to the data in $\boldsymbol{X}_{T,p}$, and evaluate whether it is typical. This is called static PCA because the trained model contains no dynamic components. Only the current measurement is used in the process evaluation at each time, t. Moreover, the base model remains unchanged as new observations are obtained. Therefore, no attempt is made to model relationships between observations at different time points (autocorrelation), and it will not adjust as underlying parameter values change (non-stationarity).

Three classes of approaches have been proposed to extend PCA methods to cope with time-dependent data. These are Dynamic PCA (DPCA), Recursive PCA (RPCA), and Moving Window PCA (MWPCA). DPCA was developed to handle autocorrelation, whereas RPCA and MWPCA are intended to deal with non-stationary data. The rest of the paper is organized as follows. In Section 2, static PCA and the associated control limits are briefly reviewed. Then, DPCA, RPCA, MWPCA and extensions for these methods are discussed. Section 3 details the simulation scenarios used to compare the methods, and reports on the results obtained. In Section 4, we discuss the results arising from this comparison study.

Review of PCA-based methods for SPM

Static PCA

Principal component analysis defines a linear relationship between the original variables of a data set, mapping them to a set of uncorrelated variables. In general, static PCA assumes that an $(n \times p)$ data matrix $\boldsymbol{X}_{n,p} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]'$ is observed. Let $\mathbf{1}_n = [1, 1, \ldots, 1]'$ be of length n. Then the sample mean can be calculated as $\bar{\boldsymbol{x}} = \frac{1}{n}\boldsymbol{X}'_{n,p}\mathbf{1}_n$ and the sample covariance matrix as $\boldsymbol{S} = \frac{1}{n-1}(\boldsymbol{X}_{n,p} - \mathbf{1}_n\bar{\boldsymbol{x}}')'(\boldsymbol{X}_{n,p} - \mathbf{1}_n\bar{\boldsymbol{x}}')$. Each p-dimensional vector $\boldsymbol{x}$ is transformed into a score vector $\boldsymbol{y} = \boldsymbol{P}'(\boldsymbol{x} - \bar{\boldsymbol{x}})$, where $\boldsymbol{P}$ is the $p \times p$ loading matrix,

containing columnwise the eigenvectors of $\boldsymbol{S}$. More precisely, $\boldsymbol{S}$ can be decomposed as $\boldsymbol{S} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}'$. Here, $\boldsymbol{\Lambda} = \text{diag}(\lambda_1,\ \lambda_2, \ldots,\ \lambda_p)$ contains the eigenvalues of $\boldsymbol{S}$ in descending order. Throughout this paper, before modeling and monitoring is performed in our methods, the data will be preprocessed using autoscaling, so that $\boldsymbol{S}$ will essentially be a correlation matrix.

In many cases, due to redundancy between the variables, using $k < p$ of the components still fits the data well. The k-dimensional scores are $\boldsymbol{y}_k = \boldsymbol{P}_k'(\boldsymbol{x} - \bar{\boldsymbol{x}})$, where $\boldsymbol{P}_k$ contains only the first k columns of $\boldsymbol{P}$. To select the number of components to retain in the PCA model, one can resort to several methods (see e.g. Valle et al.[14] and Jolliffe[15]). In this study we use the Cumulative Percentage of Variance (CPV) to select k, due to its simplicity and requirement of recursively estimate the number of retained components in the case of adaptive methods. The CPV measures the amount of variation captured by the first k latent variables:

$$\text{CPV}(k) = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j} 100\%,$$

and k is selected such that the CPV is greater than a given threshold.

PCA control charts are based on the Hotelling's T^2 statistic and the Q-statistic (a.k.a. *Squared Prediction Error, SPE*). For any p-dimensional vector $\boldsymbol{x}$, the Hotelling's T^2 is

$$T^2 = (\boldsymbol{x} - \bar{\boldsymbol{x}})'\boldsymbol{P}_k\boldsymbol{\Lambda}_k^{-1}\boldsymbol{P}_k'(\boldsymbol{x} - \bar{\boldsymbol{x}}) = \boldsymbol{y}_k'\boldsymbol{\Lambda}_k^{-1}\boldsymbol{y}_k$$

where $\boldsymbol{\Lambda}_k = \text{diag}(\lambda_1,\ \lambda_2, \ldots,\ \lambda_k)$ is the diagonal matrix consisting of the k largest eigenvalues of $\boldsymbol{S}$. The Q-statistic is defined as

$$Q = (\boldsymbol{x} - \bar{\boldsymbol{x}})'(\boldsymbol{I} - \boldsymbol{P}_k\boldsymbol{P}_k')(\boldsymbol{x} - \bar{\boldsymbol{x}}) = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$$

with $\hat{\boldsymbol{x}} = \boldsymbol{P_k}\boldsymbol{P}_k'(\boldsymbol{x} - \bar{\boldsymbol{x}})$. The Hotelling's T^2 is the squared Mahalanobis distance of $\boldsymbol{x}$ in the PCA model subspace, and the Q-statistic is the quadratic orthogonal distance to the

PCA subspace. If the number of observations is large, then assuming temporal independence and multivariate normality of the scores, the $100(1-\alpha)\%$ control limit for Hotelling's T^2 is approximately the $(1-\alpha)$ percentile of the χ^2 distribution with k degrees of freedom; thus $T^2_\alpha \approx \chi^2_k(\alpha)$.

The most commonly used approximation for the control limit of the Q-statistic is given by Jackson and Mudholkar[7]. This approximation typically performs well, but strongly relies on the assumption that the $k+1, \ldots, p$ eigenvalues are small. This assumption may be violated in the presence of high autocorrelation (especially when applying an adaptive method, such as RPCA or MWPCA), if faulty observations enter the calculation of the updated covariance matrix. If this occurs, the limits can become uninformative. To circumvent this issue we resort to the general result of Box[16], which shows that the Q-statistic is approximately distributed as a scaled χ^2-distribution with h degrees of freedom, denoted as $g\chi^2_h$. Provided that all the eigenvalues of $\boldsymbol{S}$ are available, the parameters are given by:

$$\theta_i = \sum_{j=k+1}^{p} \lambda_j^i \text{ for } i = 1, 2;\ g = \frac{\theta_2}{\theta_1};\ \text{and } h = \frac{\theta_1^2}{\theta_2}.$$

The control limit for the Q-statistic, Q_α, is then taken as the $(1-\alpha)$ quantile of the $g\chi^2_h$ distribution. We compare the difference in performance between the two limit derivations in Figure 1 for RPCA on an AR(1) process. Here, we see that after the introduction of a fault at $t = 500$ the Box[16] approximation continues to produce realistic limits, whereas the Jackson and Mudholkar[7] limit drops to an unrealistically low value as a result of faulty observations contaminating the covariance matrix.

Dynamic PCA

Dynamic PCA was first proposed in Ku et al.[17] as a way to extend static PCA to auto-correlated, multivariate systems. Ku et al.[17] state that in addition to the current observed variables, the respective lagged values up to a proper order, l, can also be included as PCA

model inputs. DPCA applies PCA to an augmented data set, $\tilde{\boldsymbol{X}}(l)$, constructed of lagged replicates of the original variables:

$$\tilde{\boldsymbol{X}}(l) = [\boldsymbol{X}(t), \boldsymbol{X}(t-1), \ldots, \boldsymbol{X}(t-l)]\,.$$

Here $\boldsymbol{X}(t-j)$ denotes the data matrix $\boldsymbol{X}$ shifted j times into the past (i.e., with j lags). By estimating the linear relationships for the dimensionality reduction, this method also implicitly estimates the autoregressive structure of the data. For instance, an AR(1) process will be modeled if lagged values up to order one are included in the model input, i.e., $\tilde{\boldsymbol{X}}(1) = [\boldsymbol{X}(t), \boldsymbol{X}(t-1)]$. Furthermore, a DPCA model with a sufficient number of lags is also capable of modeling an invertible moving average process, since such a process can be approximated by an autoregressive model.

In order to specify the number of lags, Ku et al.[17] provide an algorithm which adds an order to the lag structure, evaluates whether this brings any new linear relationship to the model and keeps it if it does. The authors indicate that typically only a few lags are retained, but as we encounter later on, the number indicated by this method is often too low to convincingly model the process. More recently, Rato and Reis[18] detail an approach for selecting the number of lags by variable, allowing for a more refined model of the process being monitored, which we will use. Treasure et al.[19] propose an extension of DPCA based on subspace identification. Conceptually and in terms of implementation this is a more complex approach, and the authors state that fault detection is comparable to DPCA. However, it has the advantage that one may restrict interpretation to the subspace representation of the process, which is likely to be more comprehensible than a DPCA model. Under conventional implementation of DPCA, the lagged vector is composed by the last l consecutive observations, including the ones signaled as faulty. Therefore, no attempt is made to accommodate the presence of faults in the lagged vector, which may lead to the wrong classification of subsequent observations. Using a missing data method to replace faulty observations is a

potential solution, but this is not explored in this paper.

Dynamic PCA with Decorrelated Residuals

Introducing lagged variables in DPCA allows the description of the autocorrelation present in the data. However, the T^2 and Q statistics can still exhibit autocorrelation. In the case where enough lags are selected, the Q-statistic should indeed have no serial correlation. Yet, even in this case, there is autocorrelation in the scores, and subsequently in the T^2-statistic. To overcome this issue, Rato and Reis[20] proposed a combination of DPCA with a missing data estimation technique (Nelson et al.[21], Arteaga and Ferrer[22]) in order to obtain better time-decorrelated statistics that they call DPCA-DR. In this method, a DPCA model is constructed as in the previous section, and from it we obtain the usual scores $\boldsymbol{y}_k = \boldsymbol{P}_k'(\boldsymbol{x} - \bar{\boldsymbol{x}})$. An additional vector of estimated scores $\hat{\boldsymbol{y}}_k$ is computed by assuming that the current observation vector $\boldsymbol{x}(t)$ is missing. This is a one-step-ahead prediction of the scores based on the implicit AR model estimated by DPCA. Moreover, the application of this methodology gives an estimate of the scores that best agree with the last l known measurements. Given these scores, the following Hotelling's T^2 statistic is defined:

$$T^2_{prev} = (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)' \boldsymbol{S}^{-1}_{prev} (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k),$$

where $\boldsymbol{S}_{prev}$ is the sample covariance matrix of the difference between the observed and estimated scores, $(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)$, that monitors the DPCA reference subspace. Control limits for this statistic can be determined empirically to obtain the desired false detection properties. Likewise, a monitoring statistic for the residual subspace, which replaces the Q-statistic, is defined as:

$$T^2_{res} = \boldsymbol{r}' \boldsymbol{S}^{-1}_{res} \boldsymbol{r} = (\boldsymbol{x} - \boldsymbol{P}_k \hat{\boldsymbol{y}}_k)' \boldsymbol{S}^{-1}_{res} (\boldsymbol{x} - \boldsymbol{P}_k \hat{\boldsymbol{y}}_k) \tag{1}$$

where $\boldsymbol{S}_{res}$ is the sample covariance matrix of the residuals in the reconstructed data, obtained with the estimated scores $(\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{P}_k \hat{\boldsymbol{y}}_k)$. Limits for this statistic are also determined

empirically. Xie et al.[23] introduce another PCA-based monitoring approach based on subspace identification that achieves decorrelation results similar to DPCA-DR. It is a more complex modelling approach but has the advantage of producing a final model that allows for greater interpretability.

Recursive PCA

If the stationarity assumption of non-adaptive PCA models, such as those above, is violated, then model parameter estimates obtained during the calibration phase may not be appropriate for future monitoring. RPCA with a forgetting factor[24] (henceforth RPCA), and MWPCA[25] have been proposed to monitor non-stationary processes. RPCA uses the idea of incorporating new observations and exponentially downweighting old ones to calculate the mean and covariance matrix used in PCA. A new observation is evaluated when it is obtained. If the T^2 or Q statistics exceed their limits because the observation is a fault or an outlier, then the model is not updated. However, when the observation is in-control, it is desirable to update the estimated mean and covariance/correlation from the previous period.

More precisely, assume that the mean and covariance of all observations up to time t have been estimated by $\bar{\boldsymbol{x}}_t$, and $\boldsymbol{S}_t$. Then at time $t+1$ the T^2 and Q-statistic are evaluated for the new observation $\boldsymbol{x}_{t+1} = \boldsymbol{x}(t+1) = [\boldsymbol{x}_1(t+1), \ldots, \boldsymbol{x}_p(t+1)]'$. If both values do not exceed their cut-off value, the data matrix $\boldsymbol{X}_{t,p}$ is augmented with observation $\boldsymbol{x}_{t+1}$ as $\boldsymbol{X}_{t+1,p} = [\boldsymbol{X}'_{t,p}\ \boldsymbol{x}_{t+1}]'$. Next, the model parameters are updated by means of a forgetting factor $0 \leqslant \eta \leqslant 1$. Denoting n_t as the total number of observations measured at time t, the updated mean is defined as:

$$\bar{\boldsymbol{x}}_{t+1} = (1 - \frac{n_t}{n_t + 1}\eta)\boldsymbol{x}_{t+1} + \frac{n_t}{n_t + 1}\eta\ \bar{\boldsymbol{x}}_t, \tag{2}$$

and the updated covariance matrix is defined as:

$$\boldsymbol{S}_{t+1} = (1 - \frac{n_t}{n_t + 1}\eta)(\boldsymbol{x}_{t+1} - \bar{\boldsymbol{x}}_{t+1})(\boldsymbol{x}_{t+1} - \bar{\boldsymbol{x}}_{t+1})' + \frac{n_t}{n_t + 1}\eta \boldsymbol{S}_t. \tag{3}$$

This is equivalent to computing a weighted mean and covariance of $\boldsymbol{X}_{t+1}$ where older values are downweighted exponentially. Using a forgetting factor $\eta < 1$ allows RPCA to automatically give lower weight to older observations. As $\eta \to 1$, the model forgets older observations more slowly. The eigenvectors of $\boldsymbol{S}_{t+1}$ are then used to obtain a loading matrix $\boldsymbol{P}_{t+1}$. Once the new value of k is determined and the new eigenvalues calculated, the control limits of the T^2 and Q-statistics can be updated according to the formulas described earlier.

Moving Window PCA

MWPCA updates at each time point while restricting the observations used in the estimations to those which fall within a specified *window* of time. With each new observation, this window excludes the oldest observation and includes the observation from the previous time period. Thus, for window size H, the data matrix at time t is $\boldsymbol{X}_t = [\boldsymbol{x}_{t-H+1}, \boldsymbol{x}_{t-H+2}, \ldots, \boldsymbol{x}_t]'$, and at time $t+1$ it is $\boldsymbol{X}_{t+1} = [\boldsymbol{x}_{t-H+2}, \boldsymbol{x}_{t-H+3}, \ldots, \boldsymbol{x}_{t+1}]'$. The updated $\bar{\boldsymbol{x}}_{t+1}$ and $\boldsymbol{S}_{t+1}$ can then be calculated using the observations in the new window. While completely recalculating the parameters for each new window is straightforward, and intuitively appealing, methods have been developed to improve on computational speed (see for example Wang et al.[25] and Jyh-Cheng[26]). As was the case for RPCA, the model is not updated when an observation is determined to be out-of-control. A good introduction to MWPCA can be found in Kruger and Xie[9].

Simulation Studies

In this section we evaluate the performance of the PCA-based methods on a variety of pure time-dependent processes. This was intentionally done because it allows for complete control

of the data generation, and eliminates confounding behavior that can arise in simulations that attempt to model complex real world processes. We find that even though we are restricting ourselves to this limited set of scenarios, the study yields useful and surprising insights. The process settings and fault scenarios are varied to provide a comprehensive overview of how they affect performance for each process scenario. The intention is to provide an overview of potential situations the practitioner might encounter, and to illustrate the type of monitoring performance that can reasonably be expected from the methods we consider.

We simulate faults where they appear in complex systems: in the inner latent components (the scores, $\boldsymbol{y}_t$) and measurement sensors (the data, $\boldsymbol{x}_t$). Sensor faults are self-explanatory. Conceptually, the scores represent latent structures of the process (fundamental structures in the process that cannot be directly observed). Thus, while sensor faults may indicate an issue with a particular sensor or element of the process, score faults indicate systematic faults. Five types of time-dependent processes are considered: an autoregressive (AR) process, a moving average (MA) process, an autoregressive process with a unit root (ARI), an integrated moving average (IMA) process, and a process that is non-stationary in the loadings structure (NSS). Following convention (e.g. Burnham et al.[27], Choi et al.[28]) we generate data at the subspace level so that we can explicitly control the features monitored by the PCA models, though we do not make use of this knowledge when monitoring. Doing so makes it possible to compute objective performance metrics, as we know the true underlying behavior of the system under Normal Operating Conditions (NOC) and faulty conditions. To obtain each observation at time t we began by generating five scores, $\boldsymbol{y}_t$, according to the equation of the desired process. For all process types, we introduce variation onto the process dynamics through $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_5, 0.01\boldsymbol{I}_5)$, where $\boldsymbol{I}_5$ is the 5×5 identity matrix. These are then transformed into a 100-dimensional data set of measurements computed as

$$\boldsymbol{x}_t = \boldsymbol{P}_0 \boldsymbol{y}_t + \boldsymbol{e}_t, \tag{4}$$

where $\boldsymbol{P}_0$ is a 100×5 matrix with orthogonal columns randomly generated once and kept constant for all simulation runs over all processes. The decision to simulate 100-dimensional data was motivated, in part, by a lack of other studies considering high-dimensional processes, even though PCA is often proposed for precisely that scenario. The $\boldsymbol{e}_t$ are 100×1 vectors of white noise errors, distributed as $\mathcal{N}(\boldsymbol{0}_{100}, 0.000025\boldsymbol{I}_{100})$, that simulate measurement noise, as is done, for instance, in (Ku et al.[17] and Lakshminarayanan et al.[29]). The $\boldsymbol{e}_t$ can be seen as the error at the sensor level, and are set to a small value here under the assumption that sensors are typically reliable. For all methods and simulations, a CPV of 95% is used. Since the statistics are not guaranteed to be i.i.d. in the dynamic context, an alternative to the analytical expressions for the limits and the chosen α level is necessary. The control limits of the non-adaptive methods are determined based on a validation data set with 5000 NOC observations. A different approach was necessary to ensure that the adaptive methods also achieved the desired False Detection Rate (FDR) on NOC data since the control limits are not constant. To do so, we search for values of α for the T^2 and Q (call these α_{T^2} and α_Q), which, when used as input in the analytical expressions for the control limits of these statistics, result in an FDR_{T^2} and an FDR_Q whose sum equals the desired total FDR for the model. To accomplish this, we impose that FDR_{T^2}=FDR_Q, and assume that T^2 and Q are independent, which is plausible for these simulations. Note that the adjusted values of α_{T^2} and α_Q do not correspond to the statistical significance of the monitoring statistics, since the theoretical expressions for the control limits are not valid under the conditions simulated in this study. Further, they do not need to be equal to each other. These limits are set such that the false detection rate of each method was 1% (i.e., the combined use of scores and residual statistics through a logical gate OR gives an overall false detection of about 1%). In Appendix A, we provide a comparison of the control limits obtained using this and the conventional approach for selecting the α values.

The number of lags for the dynamic PCA methods was selected using the method of Rato and Reis[18]. To select the values of η and H, we considered a range of possible values and

their corresponding α values, and selected one giving good monitoring results, in terms of false detection rate, on a validation NOC data set. Although the selection of forgetting parameters for adaptive methods is critical to their proper implementation, this topic is not well covered in the literature.

Faults are introduced to the process on either the first score in $\boldsymbol{y}_t$, or the first measurement variable (sensors) in $\boldsymbol{x}_t$, by simple addition of a step deviation with magnitude defined as d times the standard deviation of the first element of $\boldsymbol{\varepsilon}_t$ for score faults, and $\boldsymbol{e}_t$ for sensor faults.

This approach of varying d gives a sense of how difficult it is to detect a fault for a given process type (how large the fault would needs to be), and provides a basis for comparison between methods and across process types. We expect that the T^2 and T^2_{prev} statistics will detect the score faults, and the Q and T^2_{res} statistics will detect the measurement faults, because they monitor distance of the observations on the model subspace and from it, respectively. Each process type, fault type, and value of d constitutes a scenario. Each faulty data set contains 1000 observations (the first 500 under NOC and the remaining 500 under the effect of a fault). One hundred faulty data sets are generated for each scenario to assess the stability of the results. We considered increasing values of d to illustrate how the methods behave as the faults become more clear. Negative deviations were also investigated, and yielded symmetrical results. An overview of the parametrization used for each of the methods on the simulation scenarios we consider are given in Tables 1 and 2.

In the above tables, LV stands for the number of latent variables. Parameter values are left blank in the ARI(1,1) and IMA(1,1) cases for the PCA and DPCA models because in practice they were found to be unsuitable for these processes and no simulation results are presented. In the case of RPCA and MWPCA ranges and averages are given for the ARI and IMA cases, where the number of components changes as the process evolves.

We are particularly interested in the detection rates (DR: not to be confused with the decorrelated residuals of DPCA-DR) of the methods. In order to compare them visually, we

will plot the average DR (i.e., the ratio between the number of alarms over the number of faulty observations) of each of the methods as a function of the deviation size for the 500 time points after the fault is introduced. Additionally, whiskers are plotted around the lines indicating the 25% and 75% quantiles. These detection rates correspond to the True Positive Rate (TPR), and one minus these detection rates gives the False Negative Rate (FNR) (the rate at which faults are incorrectly classified as in-control). Similarly, the False Positive Rate (FPR) can be inferred from the zero deviation case. To assess the monitoring statistics behavior over time, we also present plots with the percentage of runs that correctly give an out-of-control signal at each faulty time period for the largest magnitude deviations applied to the scores and measurements of each process we studied. The x-axis is time on a log scale to highlight the detection power of the methods at the earliest time points, where it is most relevant. Therefore, these plots are meant to give a sense of how quickly on average each of the methods detects the faults. Monitoring results prior to the introduction of the fault are not shown since these conform on average to the false detection rate of 1% that we selected.

AR(1) process

The AR process is investigated because of its natural relevance for studying the properties of DPCA and DPCA-DR. Furthermore, this is a particularly relevant process type because the high sampling rate of many contemporary sensors inherently introduces autocorrelation into the data. The AR(1) process is defined as (Box et al.[30]):

$$\boldsymbol{y}_t = \phi \boldsymbol{y}_{t-1} + \boldsymbol{\varepsilon}_t, \tag{5}$$

where $\boldsymbol{y}_t$ are the serial observations of the underlying latent model ($\boldsymbol{y}_t$ in Equation (4)) and ϕ is the AR coefficient. In simulating the AR(1) process, five scores with a relatively high AR coefficient ϕ equal to 0.90 were used to generate the data according to Equation (5). Although negative autocorrelation is a possibility, we do not consider it since the rapid

sampling rate of modern processes means that positive autocorrelation is far more common.

Figure 2 displays the obtained fault detection rates for the 100 replications. It shows that static PCA and DPCA are both capable of detecting the simulated score faults at approximately the same level of accuracy. The Q-statistic corresponding to static PCA does not exhibit autocorrelation and for DPCA only small levels of autocorrelation are observable. This situation means that both models are successful in describing the system. Still, the Hotelling's T^2 is highly autocorrelated, which undermines the detection of deviations at the scores level, leading to weak detection of faults on the score subspace. As a consequence, large values of d are necessary before good detection results are observed. On the other hand, DPCA-DR produces monitoring statistics without significant autocorrelation but is only able to detect faults much larger than the ones presented here (for instance, a fault of magnitude 40 standard deviations has, on average, a detection rate of 0.63). This is a direct result of the DPCA-DR estimation step and subsequent differencing between the observed and estimated scores (see Equation (1)). Thus, when a fault is introduced, the estimated scores by missing data do not comply with the full data scores, causing the T^2_{prev}-statistic to signal an alarm. However, after this initial detection, the subsequent scores fitted by DPCA-DR begin to resemble faults, since the previous, faulty observation is added as a lag and ultimately T^2_{prev} returns to in-control status, as seen in the left plot of Figure 3, which illustrates the rate of fault detection for the time period after the fault is introduced. A similar, smaller, adaptation is observed for DPCA as well, but it still produces a detection rate on par with PCA. On the other hand, DPCA-DR is far more capable of detecting sensor faults than other methods, while PCA and DPCA do not show a large increase in detection after the fault is introduced for the deviation values displayed in the plot (they can fully detect larger faults not covered by that range).

There is an observable difference between the behavior of RPCA and MWPCA and the non-adaptive methods. Considering the left plot of Figure 2, we see that the adaptive methods do not detect score faults as accurately as PCA or DPCA, though between RPCA

and MWPCA there is no notable difference for this type of fault. The right plot of Figure 2 shows RPCA and MWPCA producing equally weak sensor fault detection results. Consulting both plots in Figure 3, it is clear that the reason for the weaker detection of score faults is because RPCA and MWPCA adapt to the fault after initially detecting it. This may not be an issue since detection in the moments after a fault occurs is of primary interest. The adaptation of the methods to the faults occurs because low values of the faulty observations are still within the old control limits, but these are relatively high in the context of the distribution of the NOC period statistics and therefore pull the control limits higher, allowing yet more faulty observations to enter the updating calculations.

Interestingly, these results illustrate the robustness of static PCA in detecting faults in systems with simple dynamics, even when the monitoring statistics exhibit autocorrelation. One would expect DPCA to outperform it by a considerable margin, but in practice the difference is small. Therefore, PCA is regarded as the most suitable monitoring scheme for detecting score faults in this case scenario. This is in fact in line with Russell et al.[12], who provided a comparison of DPCA against PCA on simulated data from the Tennessee Eastman process in which they find that the two methods give similar results. DPCA-DR offers the best detection of sensor faults.

MA(1) process

Like AR processes, MA processes are fundamental time-dependent processes. The MA(1) process is defined as (Box et al.[30]):

$$\boldsymbol{y}_t = \boldsymbol{\varepsilon}_t - \varphi \boldsymbol{\varepsilon}_{t-1}. \tag{6}$$

where φ is the MA coefficient.

If an MA process is invertible, it can be equivalently described as an AR process with an infinite order. Moreover, as at some point the equivalent AR coefficients may become close

to zero, DPCA methods should be able to model them under these conditions. The φ in Equation (6) is set to 0.90 for all of the scores. Note that none of the procedures covered in this study are specifically designed for modeling MA processes.

Figure 4 reveals that PCA is incapable of detecting score faults. This may come as a surprise since this MA(1) can be reformulated as an AR process with weaker autocorrelation than 0.9, and we have observed in the AR(1) case that PCA is capable of providing good monitoring. However, to monitor the MA(1) process, far more lags may be necessary even though the autocorrelation is weaker, and since PCA has no lags, it cannot model the process well. Indeed, we find that when applying DPCA, more lags are needed to model the MA(1) process than the AR(1), highlighting their importance in modeling this process. The score fault detection power of DPCA is the highest of the methods in this scenario. However, autocorrelation on the monitoring statistics is still present. The performance of PCA and DPCA in the MA(1) case is related with their relative capability to properly model the process. For faults in the scores, no observable change occurs in their T^2-statistics, which indicates that both models are unable to extract the correct scores structure. Therefore, faults introduced at the scores level are not explained by the DPCA model, but ultimately translated into deviations on its Q-statistic. For the sensor faults, PCA is again more accurate than DPCA. DPCA-DR displays the strongest sensor faults, and decent detection for score faults. This greater performance in score detection happens because the DPCA-DR model has the same deficiency as DPCA in what regards their computation, i.e., the scores remain consistent with the behavior of the system before the fault. However, the missing data estimation stage intrinsic to the method is able to follow the fault, due to faults in the lags, leading to different estimates of scores. As a result, an out-of-control state is observed since the difference between these two values is assessed and captured by both the T^2_{prev} and T^2_{res} statistics. We note that this is a different result from the AR(1) case, where both approaches of DPCA-DR to compute the scores eventually became consistent and so fault detection diminishes. Figure 5 shows that both DPCA and DPCA-DR exhibit some delay

before they completely detect the introduced score faults, and we see a similar delay for DPCA in the case of the sensor faults. This is because at each time period a new lag with a fault in it is included. The initial time periods following the fault still have a significant number of normal observations in the vector of lagged observations. However, once enough lags contain faulty observations, the scores become sufficiently distant from the estimated subspace that they can be detected using the residual statistics.

Just like PCA, the adaptive methods are not capable of determining the scores correctly and are therefore unable to detect faults in them. Both show roughly the same ability to detect sensor faults, having decent detection rates for this scenario, with a slight advantage for RPCA. As the right plot of Figure 5 shows, RPCA and MWPCA with the selected forgetting factor and window size do not adapt to these simulated faults when the magnitude of the fault is high. However, we observed that different parameters lead to substantially different results, which highlights the need for their proper selection.

Considering these results, for this process we recommend DPCA for monitoring for score faults, and PCA or DPCA-DR for monitoring for sensor faults. A general comment on the monitoring results for the MA(1) process is that smaller deviations can be detected than in the AR(1) case. This is because the process variation is lower, making faults in the MA(1) relatively more obvious than in the AR(1) case. We shall see that subsequent processes with different process variation obey this relationship between process variation and ease of fault detection as well.

ARI(1,1) process

DPCA and DPCA-DR are designed for monitoring autocorrelated data. A unit root is added to the autocorrelated series to evaluate how well they perform when non-stationarity is present, and conversely to explore the performance of RPCA and MWPCA when non-

stationary data is autocorrelated. The ARI(1,1) process is defined as (Box et al.[30]):

$$\boldsymbol{y}_t = \boldsymbol{y}_{t-1} + \phi(\boldsymbol{y}_{t-1} - \boldsymbol{y}_{t-2}) + \boldsymbol{\varepsilon}_t. \tag{7}$$

The latent variables of an ARI(1,1) process are generated with ϕ equal to 0.90 for all scores. Due to the time-dependent characteristics of this system, parameter determination during the modeling stage, such as selection of the number of latent variables and lags, heavily influences the performance of the final monitoring statistics. Even when these parameters are optimally selected, most of the monitoring statistics follow a non-stationary pattern; namely the statistics of PCA and DPCA. This makes PCA and DPCA extremely unreliable since they produce a large number of false alarms when the system is under NOC. Moreover, there is no visible change in these monitoring statistics when a fault occurs. Therefore, PCA and DPCA models, with a single constant control limit are not suitable for this process. Consequently, PCA and DPCA are excluded from the main comparison. Since the process variation of the ARI(1,1) process is so high relative to the variation of the error terms, it happens that a CPV of 95% does not always include all five important components. Thus DPCA-DR, RPCA and MWPCA are all observed to take between four and five components.

As shown in Figure 6, the estimated DPCA-DR model tends to perform better under these conditions. Even though DPCA-DR was not specifically designed for ARI processes, by simple manipulation of Equation (7) an AR like structure can be obtained. More precisely, current observations can be defined as a linear combination of their lagged versions. This is the underlying assumption of the DPCA model used by the DPCA-DR, which fits appropriate loadings to explain the data. However, due to the higher process complexity, a significantly large number of lags is required to accurately describe it. The monitoring statistics have low autocorrelation and are generally beneath their respective control limits for NOC. Figure 7 reveals that DPCA-DR does initially detect score faults, however when all lags contain faulty observations, the detection capacity of this method becomes very low. It is also observed

that the monitoring performance is highly irregular and dependent of the amount of faulty observations in the lagged variables. When the process is subjected to step deviations on the sensor measurements, DPCA-DR is the only method effectively capable of detecting them.

The left plot of Figure 6 shows that, as expected for a non-stationary process, RPCA and MWPCA both detect score faults quite well, and the average behavior of the methods is essentially the same. We see though, that both methods do display high variability, meaning that on some runs the methods do much better or worse than usual at detecting the fault. This result is a symptom of the non-stationarity of the data: the fault may occur during either periods of high or low volatility, and may go with or against the direction of the non-stationarity. In the event that the fault moves counter to the direction of the process non-stationarity it may be masked by the movement of the process, but if the fault moves with the non-stationarity it may stand out more because it is amplified by the process. This effect is most visible for $d = 400$, which lies between smaller fault magnitudes that are rarely detected and $d = 600$, which is nearly always detected. Two reasons explain the high magnitudes of the d values. First, process variation dominates the variation of error terms, which are based on a reference i.i.d. distribution. Secondly, the even though RPCA and MWPCA are adaptive, they still cannot completely account for the non-stationarity in the data. As a consequence, the control limits are set high enough to attain the desired FDR on NOC data, but at a cost to detection power. In the right plot, we see that the adaptive methods do not display the same aptitude for detecting sensor faults, especially when compared to DPCA-DR. Figure 7 shows that the adaptive methods slowly adapt to score faults on some runs. In the case of sensor faults, only some of the simulation runs result in the process going out of control since the d value for which results are displayed show a transition point between d values where almost no detection occurs and full detection occurs. The runs for which faults were detected were determined to be completely out of control after the introduction of the fault, while the remaining runs remained in control.

Based on the results, we recommend RPCA and MWPCA for detecting ARI(1,1) score

faults and DCPA-DR for detecting measurement deviations.

IMA(1,1) process

Next, a multivariate IMA(1,1) process is considered. In the univariate context, this is the process type for which the popular EWMA gives the optimal one-step-ahead prediction. This process is defined as (Box et al.[30]):

$$\boldsymbol{y}_t = \boldsymbol{y}_{t-1} + \boldsymbol{\varepsilon}_t - \varphi\boldsymbol{\varepsilon}_{t-1}. \tag{8}$$

The IMA(1,1) process data was generated according to Equation (8) with φ equal to 0.90.

As in the discussion of the ARI(1,1) process, the non-stationarity of the process has a greater impact on the final monitoring statistics behavior than the autocorrelation. This is particularly noticeable in the T^2-statistics, which exhibit different mean values across replications, because the process means of the simulated series differ from each other due to the non-stationarity. Therefore, as for the ARI(1,1) process, it is not possible to determine a reliable control limit for T^2_{PCA} and T^2_{DPCA} based on historical data with a different mean and variance level, even though the process structure remains the same. Thus, these methods are not considered for this process type. In Figure 8, we see that DPCA-DR can detect sensor faults, with the advantage of both T^2_{prev} and T^2_{res} being serially decorrelated and capable of coping with the process dynamics. This result is achieved because the IMA(1,1) is interpreted in the DPCA step of DPCA-DR as an MA(1) process in much the same way as for the ARI(1,1) process. This leads to correct estimation of the latent variables even though the base model remains unchanged throughout the simulation.

RPCA and MWPCA detect score faults well. Under our parametrization, MWPCA detects for score faults somewhat better than RPCA. Neither method is competitive with DPCA-DR for sensor faults. High variability in score fault detection performance, similar to the detection performance noted for the ARI(1,1) process, is observed. This is again due

to the non-stationarity of the process.

The results in Figure 9 match what we observe with the fault detection curves; namely that the adaptive methods perform well on the score faults and DPCA-DR does well on the sensor fault. Given the above results, we recommend RPCA or MWPCA for monitoring the scores and DPCA-DR for the measurements.

NSS process

Next, we consider a process that is non-stationary in the loadings structure (NSS) as opposed to the ARI and IMA processes, which introduce non-stationarity at the score level. The performance of the methods is also studied when another type of non-stationarity is present to see if they still perform as they did in the previous two scenarios. The process we consider is non-stationary locally, but exhibits a periodic fluctuation that can be considered stationary on the larger scale. The NSS process expressing the described behavior introduces non-stationarity in the form of rotations on the base latent variables hyperplane, $\boldsymbol{P}_0$ (see Equation (4)). By application of such rotations, the latent variables hyperplane experiences a periodic fluctuation over all its axes (see Appendix B for details). In this case we set the amplitude to $15\pi/180$, which corresponds to a $\pm 15^\circ$ rotation on the base hyperplane, and the frequency to $1/1000$ (i.e., a full rotation is obtained at every 1000 observations). Since we are using 5000 observations to train the models, this corresponds to five complete periods. Each score was generated from a normal distribution with zero mean and variance 0.01 in order to make it comparable with the other processes. As for the previous cases study, faults were introduced in one of the scores or one of the measurements. The detection rates for these faults are depicted in Figure 10.

Since PCA and DPCA-DR are based on static models, they can only accurately describe the process over some local regions resembling the average behavior of the process even though the process exhibits periodic rotation. Therefore, whenever the fluctuation causes the data to depart from this average behavior, the monitoring statistics increase. This

happens in a cyclic way and is visible in the left plot of Figure 11, and to a lesser extent in the right plot. This situation also causes a generally lower detection rate for PCA and DPCA-DR because the control limits are inflated and thus mask the faults when they occur in periods closer to the reference model. Still, DPCA-DR shows a competitive performance, which resembles PCA in score faults and DPCA in sensor faults. As for DPCA, the periodic effect is mitigated, allowing it to have a high performance in both type of fault.

Since they are adaptive methods, the RPCA and MWPCA models should be able to adjust as the base hyperplane rotates. In order to achieve this performance, a forgetting factor or window size related to the rotation frequency should be used. For our NSS process, this requires the use of small parameter values. Figure 10 shows that these methods have weak fault detection as a consequence of their high adaptation, and therefore they are only detecting faults with large deviations. As shown in the right plot of Figure 11, both weakly detect the introduced score faults at their inception, but later adapt to them. The right plot of Figure 11 shows the adaptice methods obtaining a low detection rate.

Based on the results, we recommend the use of DPCA for detecting both types of faults.

The Tennessee Eastman process

The simulated processes we have considered thus far have the advantage of being simple and transparent, in the sense that the dynamics of the system and the changes caused by the introduction of faults are readily intelligible. However, these processes are not as complex as those encountered in most real applications. A simulation of the Tennessee Eastman (TE) chemical production process, introduced by Downs and Vogel[31], provides a more realistic testing environment. We will use the data sets employed by Russell et al.[12] (available at http://web.mit.edu/braatzgroup), of a controlled version of the Tennessee Eastman process. Twenty-one fault scenarios are considered, each corresponding to a data set containing 960 observations collected at a sample interval of 3 minutes, with the fault introduced after 8

hours. All the manipulated and measurement variables, except the agitation speed of the reactor's stirrer (which is always constant), are used for monitoring, giving a total of 52 variables. Manipulated variables are controlled input variables. Measurement variables are direct measurements of the process. Table 3, summarizes the 21 fault scenarios we will apply the methods to.

Table 4 shows the parameter settings of the models we used to monitor the TE process. The high number of retained components is due to the threshold of 95% used by the CPV criterion. Good detection results can also be obtained with lower thresholds though the evaluation of the methods does not yield a different interpretation.

Table 5 shows the fault detection rates of the monitoring statistics of each method across the fault scenarios. Since the TE process is autocorrelated, but does not exhibit significant non-stationarity, we expect that DPCA and DPCA-DR will be the methods best suited to identifying the faults. In practice, we find that this to be largely the case, with DPCA-DR delivering the highest fault detection across most of the scenarios. If we consider the detection rates of RPCA and MWPCA, we find that these are often much lower than those of the other methods due to adaptation to the fault. Despite the range of methods considered, we also find that none reliably detect faults 3, 9 or 15, which is consistent with the findings of Russell et al.[12].

Discussion and Conclusions

As expected, DPCA and DPCA-DR typically handle autocorrelation well, and RPCA and MWPCA do the same for some kinds of non-stationarity. Interestingly, our results show that static PCA is effective enough when the process exhibits simple dynamics, as in the AR(1) and NSS cases, and when the faults occur in the scores. Nonetheless, for processes with complex dynamics and time-dependent features, neither PCA nor DPCA can cope with the natural fluctuations of the data. DPCA-DR displays a markedly superior capability for

detecting sensor measurements faults in all cases, while score faults with low magnitude pass undetected except in the MA(1) and NSS cases. This characteristic is a consequence of DPCA-DR's ability to estimate the current scores using missing data imputation techniques, as it will be explained later. RPCA and MWPCA do detect score faults, but eventually adapt to them. The expectation had been that the adaptability of these methods would lead to a more accurate local description of the process and higher quality monitoring statistics as a consequence. Therefore, it was revealing to find while performing well for the ARI(1,1) and IMA(1,1) processes, the adaptive methods were less well suited to the NSS process than the non-adaptive methods. Indeed, the non-adaptive methods perform well because they cover the whole period of the process in the calibration phase. On the other hand, the adaptive models 'lag behind' slightly in the sense that the limits and parameters pertain to slightly older local realizations of the process, causing the weaker performance.

In Table 6 we summarize our simulation results, identifying poor performance (–), acceptable performance (∘), good performance (+) and the best performing method ($+^*$). In cases where two methods produce essentially identical results, and have the highest detection rates, both are classified as the best. These assessments are purely qualitative, and are meant as guidelines to compliment the quantitative results presented in the simulations section.

Since we have fixed the standard deviations of $\boldsymbol{\varepsilon}_t$, and $\boldsymbol{e}_t$, the difficulty of detecting score and sensor faults is comparable across processes. In Figure 12, we plot the log of lowest magnitude fault for which good detection is achieved by the best performing method against the proportional contribution of score and sensor errors to the total variation. In the left plot, the x-axis is $trace(cov(\boldsymbol{\varepsilon}_t))/trace(cov(\boldsymbol{y}_t)) \times 100$, and in the right plot it is $trace(cov(\boldsymbol{e}_t))/trace(cov(\boldsymbol{x}_t)) \times 100$, where $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ are obtained from a reference data set for each process. We see that this relationship is non-linear, even after applying the log, with the difficulty of detecting faults typically increasing the more dominant the process variation is. This is logical given that the process variance does not grow linearly between processes,

and when the process variation is large relative to the fault magnitude it tends to mask small faults. Still MA(1) faults are detected a bit earlier than we might otherwise expect because of the behavior of DPCA described in the section on the MA(1) simulation.

Additional process scenarios should yield further insights. While this may be the case, the results of this investigation highlight the complexity of control chart modeling of time-dependent processes; even for the fundamental cases we covered. Even when model parameters are well chosen, methods designed for a process type may be outperformed by more basic methods, as happens in the NSS process. Furthermore, the results also point to the importance of understanding synergies between the monitoring methods, such as we observe in the ARI(1,1) and IMA(1,1) processes. RPCA and MWPCA detect faults in the scores while DPCA-DR is more suitable for the measurements. For this reason, future research into methods more suited for a specific process may be of interest, even if they were not originally designed for that type of process.

In the simulations we performed, we only considered step faults. However, ramp faults are another common type of upsets. We also considered these and find that the results are largely what we would expect. Non-adaptive methods that can detect step faults of a given magnitude will also detect a ramp fault if it eventually reaches that magnitude. The speed of detection will be lower than for a step fault, and will depend on the slope of the ramp fault. All things being equal, to detect a ramp fault at the same speed as a step fault, they require that the deviation of the ramp fault be larger. Predictably, adaptive methods adapt to ramp faults unless the slope of the fault is very large. Thus, we recommend adaptive methods for step faults or for detecting ramp faults that are close to step faults in their slope. Otherwise, the adaptive methods will adjust to them without signaling a fault. Results for ramp fault versions of the simulations we performed are available in the Supplementary Materials. Performance characteristics for this type of fault would benefit from further investigation.

In analyzing the fault diagnosis performance of the methods on the TE data, we found

that DPCA-DR was generally the most effective method, followed distantly by PCA and DPCA. The poor performance of the adaptive methods indicates that selecting this sort of method should perhaps be avoided unless non-stationarity is of real concern. If non-stationarity is minimal or not expected, it is better to rely on methods which explicitly assume stationarity.

Further research should focus on a rigorous approach for setting control limits when the monitored series is time-dependent. With correct definitions of the control limits, more complex comparative analyses, such as the Average Run Length, would be possible. We note that for some of the processes studied, even the simplest metrics, such as the detection rates, are highly variable despite considering many simulated realizations. This phenomenon is mostly a result of the different behavior of the time-dependent data at different realizations, since the simulated deviations do not always have the same impact on the observed data. For all of the scenarios, we found it necessary to adjust the control limits, and α values to achieve the desired false detection rate. This introduces additional complexity for model specification and furthermore, we saw that good modeling performance on NOC data may not be enough to guarantee good fault detection properties.

In general, most of the methods display a constant fault detection rate following the introduction of the fault. This is especially true of the non-adaptive methods. As expected, the adaptive methods often report lower detection rates as the time from the fault grows because they adjust to the faulty scenario. More surprisingly, our results reveal that the detection rates of the dynamic methods can increase (as in the MA(1) process) or decrease (as in the IMA(1,1) process) once the lagged observations consist entirely of faulty observations.

An interesting finding of this study was the capability of DPCA-DR to deal with non-stationary processes. Since the studied processes (namely, AR(1), MA(1), ARI(1,1) and IMA(1,1)) can be expressed as a function of past observations, a regression model with a sufficient amount of past observations is able to explain the process dynamics properly. This modeling is, in fact, conducted by the underlying DPCA model of DPCA-DR. The missing

data estimation step also makes it possible to perform a one-step-ahead prediction of the observations and scores that, given information from past observations, best agree with the latent variables subspace. The estimated values will follow the same NOC non-stationary pattern in which the model was trained, and therefore deviations from that structure are reflected on their residuals and captured by the respective monitoring statistics. Surprisingly, these good modeling properties also lead to low detection capabilities when changes occur at the scores level. In this case, only the first faulty observations are signaled as an alarm, since they do not comply with the past observed behavior. As these faulty observations are then used to estimate the subsequent observations, the deviation is no longer observed and no detection is done. A similar complication occurs with DPCA when an atypical observation is obtained. While this atypical observation will be detected correctly, in the next time period, it will be added as the first lag of the next observation. This may lead to the wrong classification of the new observation since the previously atypical observation also has an impact on the scores computation at the current time. Investigation into possible solutions could improve the performance of these methods. Although we used a more accurate method to select the number of lags for DPCA and DPCA-DR, as a sensitivity analysis we also considered the lag selection method of Ku et al.[17]. In most of the processes studied, this method selected no lag for DPCA, making it equivalent to PCA, and revealing the need for the more refined approach.

The selection of η and H remains an unsettled problem. Choi et al.[28] have detailed an approach for adaptive values of η in RPCA, but this does not explain how to initialize this adaptive model and a widely accepted approach for that basic requirement still does not exist. Furthermore, the control limits of RPCA and MWPCA are problematic. We currently use fitted values of α in the classic control limit formulas for the monitoring statistics, but these α values do not directly correspond to intuitive values. Further research might investigate the use of control limits that are based on alternative distributional approximations, or a robust updating approach that prevents faulty observations from contaminating the

estimated covariance matrix.

We make the models used to simulate the processes discussed in this paper available to interested parties upon request.

Acknowledgments

Tiago J. Rato acknowledges the Portuguese Foundation for Science and Technology for his PhD grant (grant SFRH/BD/65794/2009). Marco S. Reis also acknowledges financial support through project PTDC/EQU-ESI/108374/2008 co-financed by the Portuguese FCT and European Unions FEDER through Eixo I do Programa Operacional Factores de Competitividade (POFC) of QREN (with ref. FCOMP-01-0124-FEDER-010397).

Part of the research was funded by the Flanders Agency for Innovation by Science and Technology (IWT).

Mia Hubert acknowledges the financial support of KU Leuven grant GOA/12/14 and of the IAP Research Network P7/06 of the Belgian Science Policy.

Bart De Ketelaere acknowledges the Flanders Agency for Innovation by Science and Technology (IWT) through the SBO POM project, and the Industrial Research Fund of the KU Leuven.

Mr. Rato and Mr. Schmitt contributed approximately equally to this work, with Mr. Rato managing the non-adaptive methods, and Mr. Schmitt, the adaptive ones. Prof. Reis, Dr. De Ketelaere, and Prof. Hubert provided insights and guidance.

References

(1) Woodall, W.; Montgomery, D. *Journal of Quality Technology* **2014**, *46*, 78–94.

(2) Qin, S. *Journal of Chemometrics* **2003**, *17*, 480–502.

(3) Hotelling, H. *The Annals of Mathematical Statistics* **1931**, *2*, 360–378.

(4) Lowry, C. A.; Woodall, W. H.; Champ, C. W.; Rigdon, S. E. *Technometrics* **1992**, *34*, pp. 46–53.

(5) Crosier, R. B. *Technometrics* **1988**, *30*, 291–303.

(6) Jackson, J. E. *Technometrics* **1959**, *1*, 359–377.

(7) Jackson, J. E.; Mudholkar, G. S. *Technometrics* **1979**, *21*, 341–349.

(8) Kourti, T. *International Journal of Adaptive Control and Signal Processing* **2005**, *19*, 213–246.

(9) Kruger, U.; Xie, L. *Advances in statistical monitoring of complex multivariate processes: with applications in industrial process control*; John Wiley and Sons Ltd., 2012.

(10) Chiang, L. H.; Colegrove, L. F. *Chemometrics and Intelligent Laboratory Systems* **2007**, *88*, 143 – 153.

(11) De Ketelaere, B.; Hubert, M.; Schmitt, E.

(12) Russell, E.; Chiang, L.; Braatz, R. *Chemometrics and Intelligent Laboratory Systems* **2000**, *51*, 81–93.

(13) Ferrer, A. *Quality Engineering* **2007**, *19*, 311–325.

(14) Valle, S.; Li, W.; Qin, S. J. *Industrial & Engineering Chemistry Research* **1999**, *38*, 4389–4401.

(15) Jolliffe, I. *Principal Component Analysis*, 2nd ed.; Springer: New York, 2002.

(16) Box, G. E. P. *The Annals of Mathematical Statistics* **1954**, *25*, pp. 290–302.

(17) Ku, W.; Storer, R. H.; Georgakis, C. *Chemometrics and Intelligent Laboratory Systems* **1995**, *30*, 179–196.

(18) Rato, T. J.; Reis, M. S. *Chemometrics and Intelligent Laboratory Systems* **2013**, *125*, 74–86.

(19) Treasure, R. J.; Kruger, U.; Cooper, J. E. *Journal of Process Control* **2004**, *14*, 279 – 292.

(20) Rato, T. J.; Reis, M. S. *Chemometrics and Intelligent Laboratory Systems* **2013**, *125*, 101–108.

(21) Nelson, P. R.; Taylor, P. A.; MacGregor, J. F. *Chemometrics and Intelligent Laboratory Systems* **1996**, *35*, 45–65.

(22) Arteaga, F.; Ferrer, A. *Journal of Chemometrics* **2002**, *16*, 408–418.

(23) Xie, L.; Kruger, U.; Lieftucht, D.; Littler, T.; Chen, Q.; Wang, S.-Q. *Industrial & Engineering Chemistry Research* **2006**, *45*, 1659–1676.

(24) Li, W.; Yue, H. H.; Valle-Cervantes, S.; Qin, S. J. *Journal of Process Control* **2000**, *10*, 471–486.

(25) Wang, X.; Kruger, U.; Irwin, G. W. *Industrial & Engineering Chemistry Research* **2005**, *44*, 5691–5702.

(26) Jyh-Cheng, J. *Journal of the Taiwan Institute of Chemical Engineer* **2010**, *44*, 475–481.

(27) Burnham, A. J.; MacGregor, J. F.; Viveros, R. *Chemometrics and Intelligent Laboratory Systems* **1999**, *48*, 167 – 180.

(28) Choi, S.; Martin, E.; Morris, A.; Lee, I.-B. *Ind. Eng. Chem. Res* **2006**, *45*, 3108–3118.

(29) Lakshminarayanan, S.; Shah, S. L.; Nandakumar, K. *AIChE Journal* **1997**, *43*, 2307–2322.

(30) Box, G. E. P.; Jenkins, G. M.; Reinsel, G. C. *Time Series Analysis: Forecasting and Control*, 3rd ed.; Prentice-Hall: New Jersey, 1994.

(31) Downs, J.; Vogel, E. *Computers and Chemical Engineering* **1993**, *17*, 245–255.

Appendix A: Comparison of control limits obtained from conventional and tuned α values

Throughout our simulations, we select values of α_{T^2} and α_Q so that FDR_{T^2}=FDR_Q=0.005 and the global FDR=0.01. Conventionally, one would expect to achieve an FDR of 1% by simply setting $\alpha_{T^2} = \alpha_Q = 0.005$. We will illustrate in the following example that this conventional approach leads to an unacceptably high FDR on non-stationary data. We fit RPCA and MWPCA models to an IMA(1,1) process using the forgetting factors given in Table 2, and implement two monitoring schemes with different control limits. In Table 7, we summarize the performance of the adaptive methods using conventional and tuned limits. Conventional limits lead to undesirably high detection rates, but the tuned limits give close to 1% false detection. When we examine the α values selected by our algorithm, we see that in this case those of for the limits for the Q-statistics are significantly smaller than convention would dictate. Figures 13 and 14 present the control charts corresponding to these results. We notice that the control charts with tuned limits have consistent monitoring statistics, while the Q-statistic charts based on conventional limits detect too many faults.

Appendix B: Rotation scheme for the NSS process

Here we provide in greater detail the scheme we use to rotate the hyperplane defining the data in the NSS process simulations. The NSS process introduces non-stationarity in the form of rotations on the base latent variables hyperplane, $\boldsymbol{P}_0$. These rotations are achieved by premultiplication of $\boldsymbol{P}_0$ over time by a rotation matrix, $\boldsymbol{R}(\boldsymbol{\theta}) = \boldsymbol{R}_1(\theta_1) \cdot \boldsymbol{R}_2(\theta_2) \cdot \ldots \cdot$

$\boldsymbol{R}_{p-1}(\theta_{p-1})$, defined by the vector of angles $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_{p-1}]'$, where,

$$\boldsymbol{R}_1(\theta_1) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & \cdots & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad \boldsymbol{R}_2(\theta_2) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 & \cdots & 0 \\ 0 & \sin\theta_2 & \cos\theta_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \text{etc}$$

For this case study, θ_i was generated as:

$$\theta_i = A_i \sin(2\pi f_i t) \tag{9}$$

where A_i is the amplitude, t is the sampling time and f_i is the frequency.

In this case, the rotation is performed by setting A_i as $15\pi/180$ and f_i as $1/1000$ for all $p-1$ rotation angles (see Equation (9)). This induces a periodic rotation of $\pm 15^\circ$ in the base hyperplane.

Supporting Information Available

Illustrations of process types

Presented in these supplementary materials are plots of NOC realizations of the first variable of each of the processes we considered, and the first score. This corresponds to the measurement and latent variable levels of the process. The intention is to provide a qualitative impression of how each of these processes behave, to improve the interpretability of the results in the simulation study.

Simulations with ramp faults

In this section we redo our earlier simulations, but introduce ramp faults instead of step faults. These results are similar to the step fault results, with the main difference being that RPCA and MWPCA have weaker fault detection performance because they adapt to the ramp faults as they occur. The performance of the non-adaptive methods on ramp faults and step faults differs in that larger magnitude faults are needed to detect the ramp faults earlier. Ramp faults are generated with a slope such that they take 500 observations to reach the full fault magnitude. Therefore, the larger the final magnitude, the higher the slope of the fault, and the more it resembles a step fault.

AR

In Figures 20 and 21 we show the detection rates and speed of detection for ramp faults on the AR(1) process. The ranking of the methods in terms of score and sensor fault detection is the same as in the step fault scenario. However, we note that the faults must now have much higher magnitudes to be detected during the entire fault period. This is because at the beginning of the fault the deviation is not yet fully developed.

MA

In Figures 22 and 23 we show the detection rates and speed of detection for ramp faults on the MA(1) process. The ranking of the methods by score fault detection is the same as in the step fault case.

ARI

In Figures 24 and 25 we show the detection rates and speed of detection for ramp faults on the ARI(1,1) process. The ranking of score and sensor fault detection accuracy is the same as in the step case, but here we see that RPCA and MWPCA have far weaker fault detection

capacities since they adapt to the ramp faults.

IMA

In Figures 26 and 27 we show the detection rates and speed of detection for ramp faults on the IMA(1,1) process. The ranking of the methods in terms of fault detection is largely consistent with what we saw for the step fault scenario, except that the score fault detection curves of DPCA-DR and RPCA cross. For the largest score fault magnitude considered, though, RPCA still outperforms DPCA-DR.

NSS

In Figures 28 and 29 we show the detection rates and speed of detection for ramp faults on the NSS process. As before DPCA, is the best method for detecting score faults, and DPCA-DR is best for detecting sensor faults. Static PCA delivers good performance as well, while the adaptive methods fair poorly.

Images and Tables

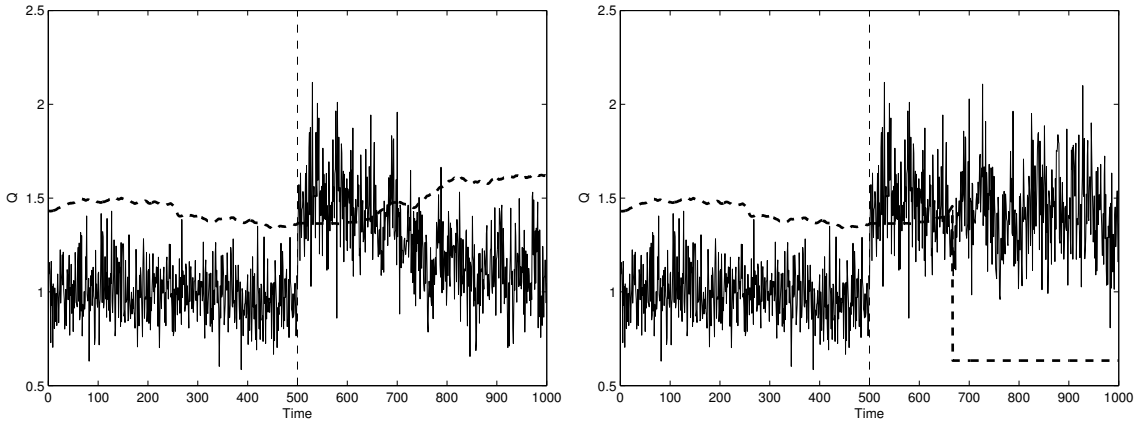

Figure 1: RPCA Q-statistic control charts for the Box (left) and Jackson and Mudholkar (right) limits on AR(1) data with a score fault at $t = 500$.

Table 1: Parameter settings for non-adaptive methods.

	PCA	DPCA		DPCA-DR	
Process	LV	LV	Lags	LV	Lags
AR(1)	5	9	0–2	9	0–2
MA(1)	5	46	10	46	10
ARI(1,1)	- -	- -	- -	4	50
IMA(1,1)	- -	- -	- -	37	10
NSS	14	144	10	144	10

Table 2: Parameter settings for adaptive methods.

	RPCA		MWPCA	
Process	LV	η	LV	H
AR(1)	5	0.9987	5	700
MA(1)	5	0.9999	5	800
ARI(1,1)	4-5 (4)	0.995	4-5 (4)	530
IMA(1,1)	5	0.999	5	800
NSS	7-14 (11)	0.93	5-12 (7)	102

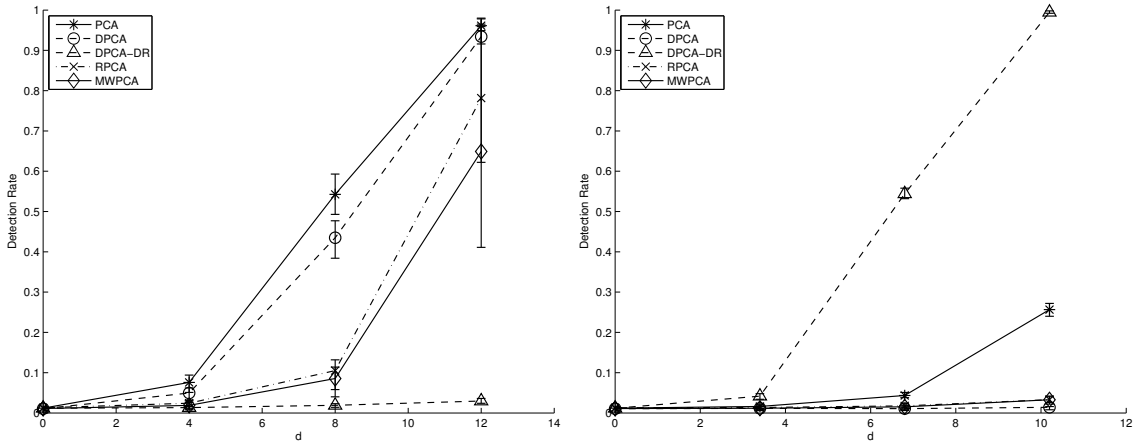

Figure 2: Fault detection rate curves for step deviations on one of the scores (left) and sensor measurements (right) of the AR(1) process. The fault magnitude is defined as d times the standard deviation.

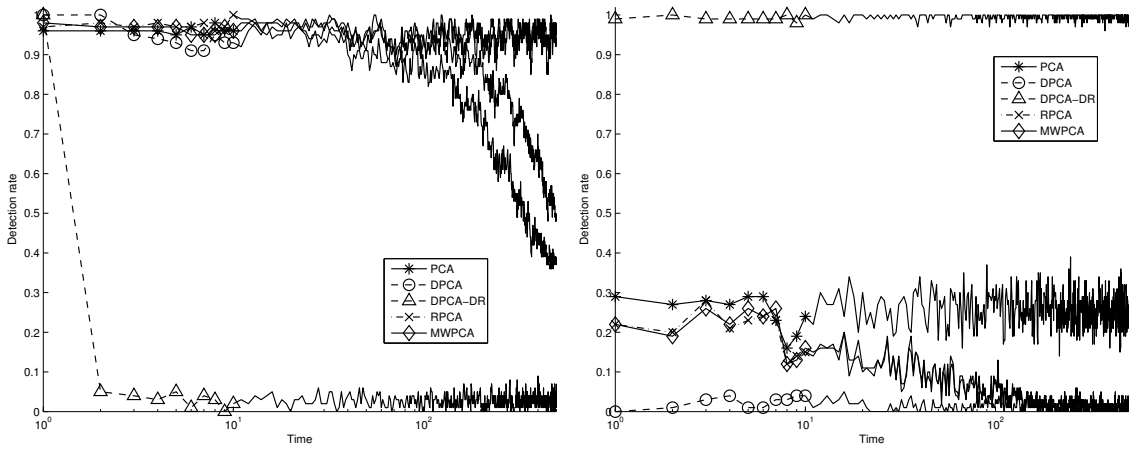

Figure 3: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor faults for the AR(1) process at each faulty time period averaged over all runs.

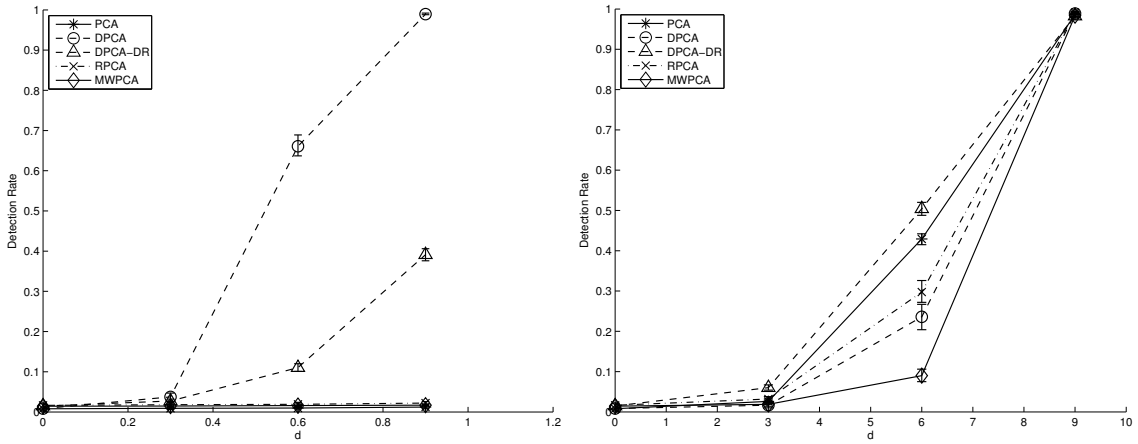

Figure 4: Fault detection rate curves for step deviations on one of the scores (left) and sensor measurements (right) of the MA(1) process. The fault magnitude is defined as d times the standard deviation.

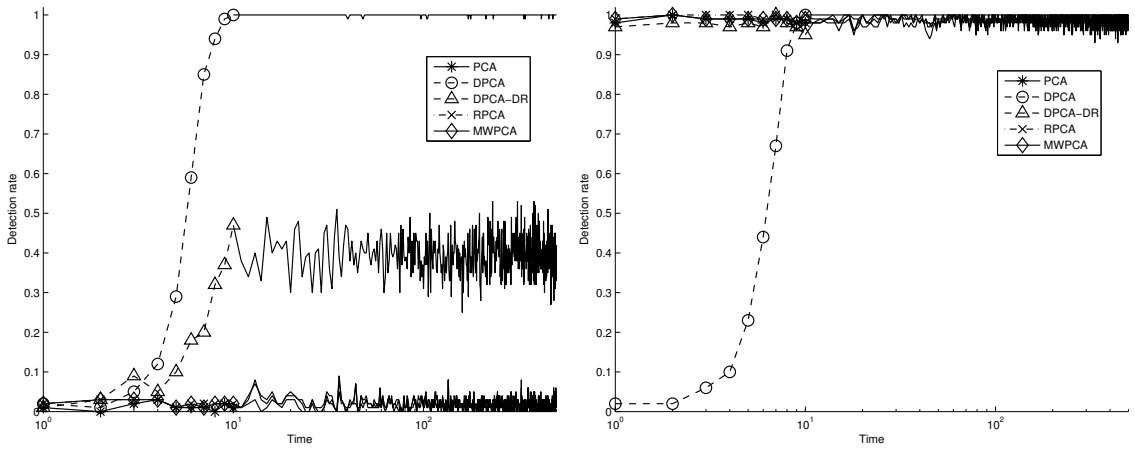

Figure 5: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor faults for the MA(1) process at each faulty time period averaged over all runs.

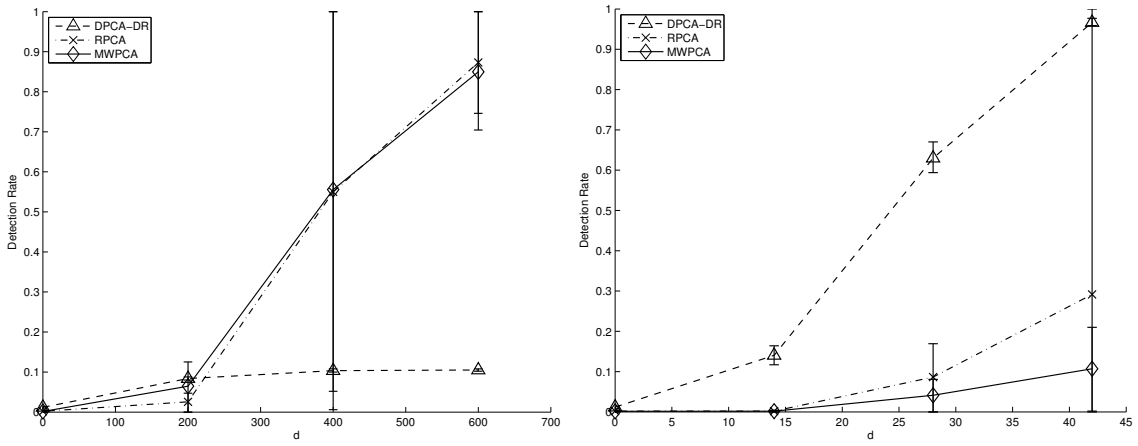

Figure 6: Fault detection rate curves for step deviations on one of the scores (left) and sensor measurements (right) of the ARI(1,1) process. The fault magnitude is defined as d times the standard deviation.

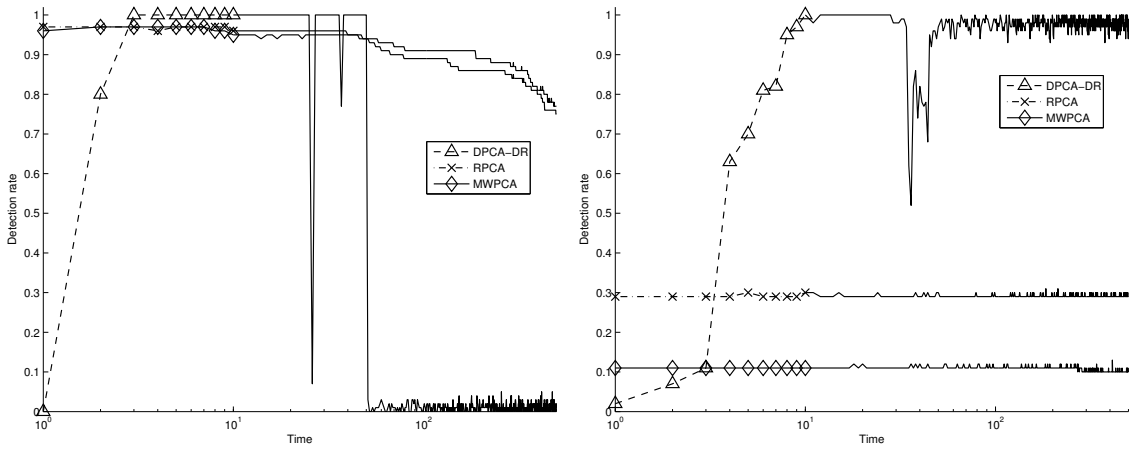

Figure 7: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor faults for the ARI(1,1) process at each faulty time period averaged over all runs.

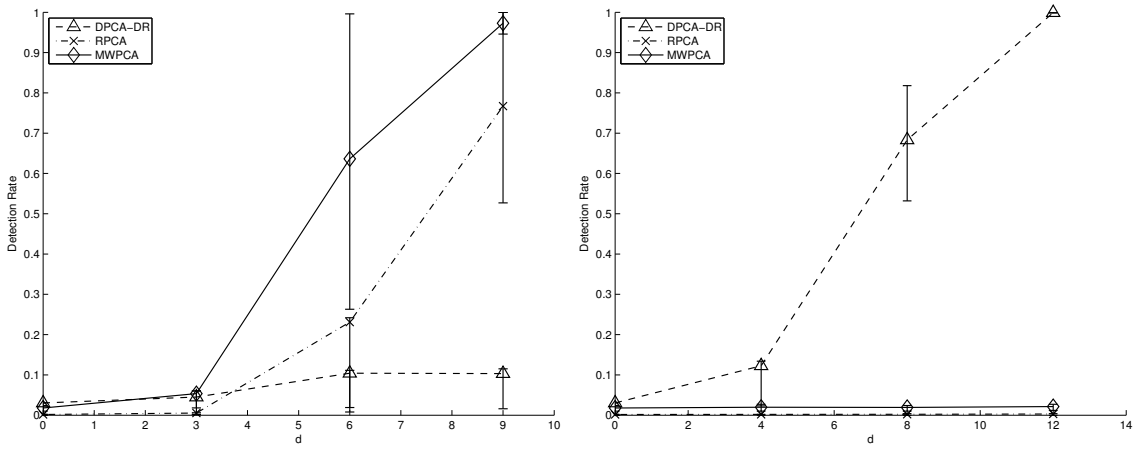

Figure 8: Fault detection curves for step deviations on one of the scores (left) and sensor measurements (right) of the IMA(1,1) process. The fault magnitude is defined as d times the standard deviation.

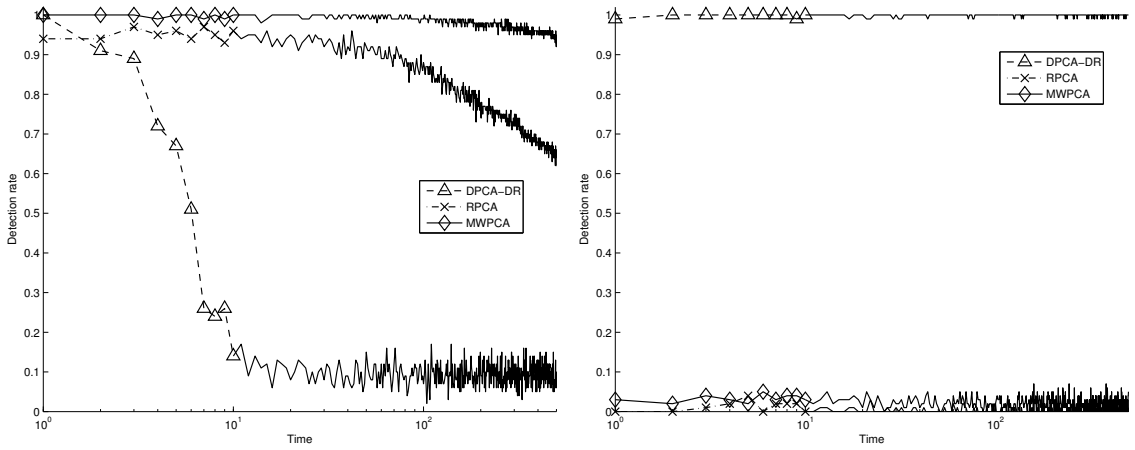

Figure 9: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor faults for the IMA(1,1) process at each faulty time period averaged over all runs.

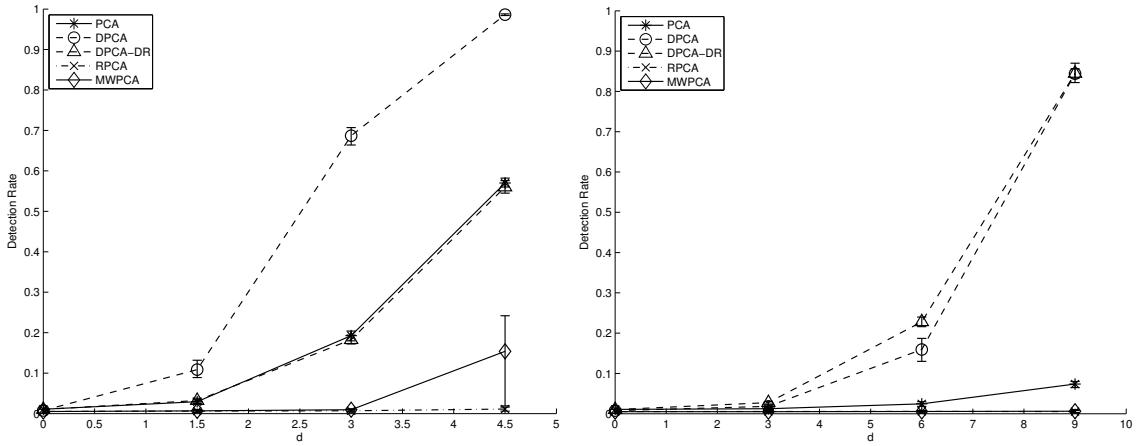

Figure 10: Fault detection rate curves for step deviations on one of the scores (left) and sensor measurements (right) of the NSS process. The fault magnitude is defined as d times the standard deviation.

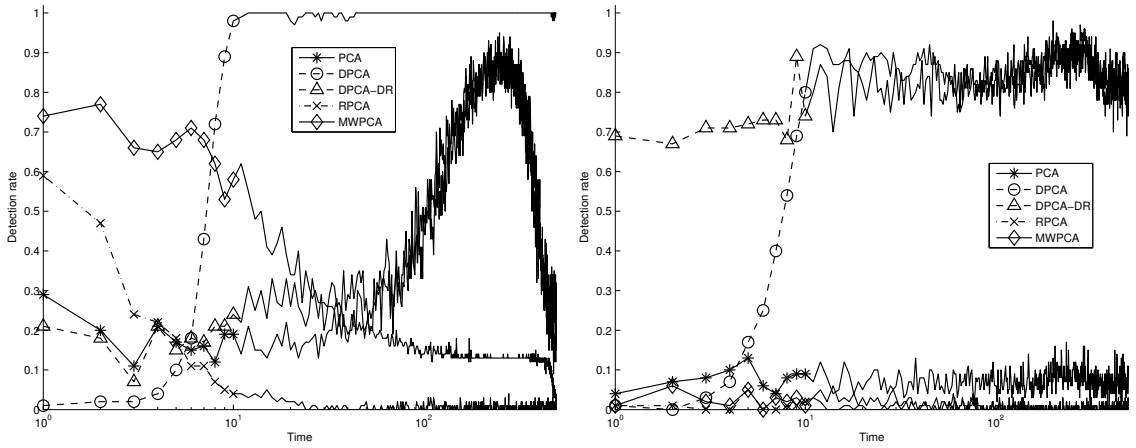

Figure 11: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor faults for the NSS process at each faulty time period averaged over all runs.

Table 3: Tennessee Eastman fault types

Fault	Description	Type
IDV(1)	A/C feed ratio, B composition constant (Stream 4)	Step
IDV(2)	B composition, A/C ratio constant (Stream 4)	Step
IDV(3)	D feed temperature (Stream 2)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss (Stream 1)	Step
IDV(7)	C header pressure loss reduced availability (Steam 4)	Step
IDV(8)	A, B, C feed composition (Stream 4)	Random variation
IDV(9)	D feed temperature (Stream 2)	Random variation
IDV(10)	C feed temperature (Stream 4)	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)	Unknown	
IDV(17)	Unknown	
IDV(18)	Unknown	
IDV(19)	Unknown	
IDV(20)	Unknown	
IDV(21)	The valve for Stream 4 was fixed at the state position	Constant position

Table 4: Parameter settings used to monitor the TE process.

	PCA	DPCA		DPCA-DR		RPCA		MWPCA	
Process	LV	LV	Lags	LV	Lags	LV	η	LV	H
TE	37	195	1–18	195	1–18	35	0.9983	32	163

Table 5: Tennessee Eastman fault detection results. The method giving the best performance is in bold.

Fault	PCA		DPCA		DPCA-DR		RPCA		MWPCA	
	T^2	Q	T^2	Q	T^2_{prev}	T^2_{res}	T^2	Q	T^2	Q
1	0.991	**0.996**	0.989	0.994	0.995	0.994	0.991	0.991	0.994	**0.996**
2	**0.983**	0.976	0.981	0.979	0.976	0.970	**0.983**	0.968	0.981	0.969
3	0.002	**0.010**	0.001	0.007	0.009	0.004	0.001	0.002	0.006	0.004
4	0.105	0.995	0.878	**0.999**	0.993	**0.999**	0.144	0.911	0.006	0.006
5	0.218	0.238	0.236	0.341	**0.999**	**0.999**	0.221	0.099	0.233	0.215
6	0.989	**0.999**	0.984	**0.999**	**0.999**	**0.999**	0.993	**0.999**	0.995	**0.999**
7	**0.999**	0.727	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	0.700	**0.999**	0.940
8	0.969	0.925	0.968	0.974	**0.975**	0.970	0.971	0.844	0.971	0.968
9	0.001	0.000	0.000	0.002	**0.006**	0.002	0.002	0.000	0.001	0.001
10	0.211	0.346	0.147	0.381	0.876	**0.880**	0.004	0.004	0.016	0.011
11	0.358	0.493	0.662	**0.919**	0.703	0.775	0.077	0.065	0.019	0.016
12	0.979	0.903	0.995	0.996	**0.998**	0.995	0.980	0.865	0.983	0.973
13	0.941	0.948	0.940	0.950	0.956	0.951	0.943	0.940	0.976	**0.998**
14	0.984	0.843	0.998	0.999	0.602	0.995	0.981	0.874	**0.999**	**0.999**
15	0.001	0.007	0.000	0.005	**0.039**	0.021	0.000	0.002	0.010	0.002
16	0.062	0.326	0.055	0.306	**0.891**	0.874	0.000	0.005	0.001	0.004
17	0.753	0.936	0.864	0.970	**0.973**	0.971	0.754	0.891	0.197	0.218
18	0.889	**0.898**	0.888	0.900	**0.898**	**0.898**	0.891	0.895	0.890	0.888
19	0.010	0.030	0.069	0.270	**0.421**	0.397	0.004	0.000	0.007	0.005
20	0.189	0.469	0.255	0.597	**0.819**	0.797	0.102	0.094	0.074	0.060
21	0.332	0.449	0.377	0.422	**0.472**	0.315	0.000	0.000	0.015	0.011

Table 6: Summary of detection performance.

	PCA		DPCA		DPCA-DR		RPCA		MWPCA	
Type	Score	Sensor	Score	Sensor	Score	Sensor	Score	Sensor	Score	Sensor
AR	$+^*$	$\circ$	$+^*$	$-$	$-$	$+^*$	$+$	$-$	$+$	$-$
MA	$-$	$+$	$+^*$	$+$	$\circ$	$+^*$	$-$	$\circ$	$-$	$\circ$
ARI	- -	- -	- -	- -	$-$	$+^*$	$+^*$	$-$	$+^*$	$-$
IMA	- -	- -	- -	- -	$-$	$+^*$	$+$	$-$	$+^*$	$-$
NSS	$\circ$	$-$	$+^*$	$+^*$	$\circ$	$+^*$	$-$	$-$	$-$	$-$

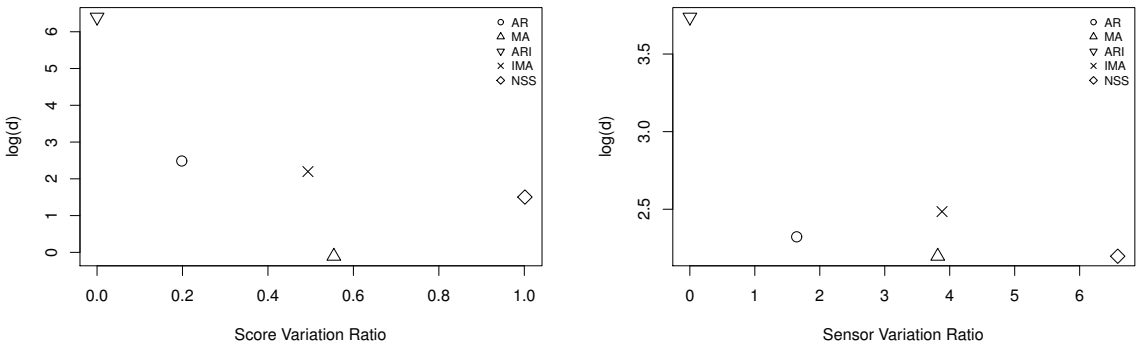

Figure 12: Minimum $\log(d)$ for which good fault detection is achieved against relative contribution of score and sensor error to total score and measurement variation.

Table 7: Detection performance of PCA control charts on the ARI process.

Method	FF_{opt}	Limits	α_{T^2}	α_Q	DR_{T^2}	DR_Q	DR_g
RPCA	0.999	conventional	0.005	0.005	0.7%	8.4%	9%
RPCA	0.999	tuned	0.004	1.67×10^{-6}	0.8%	0%	0.8%
MWPCA	800	conventional	0.005	0.005	0.6%	7.5%	8.1%
MWPCA	800	tuned	0.002	9.62×10^{-7}	0.2%	0%	0.2%

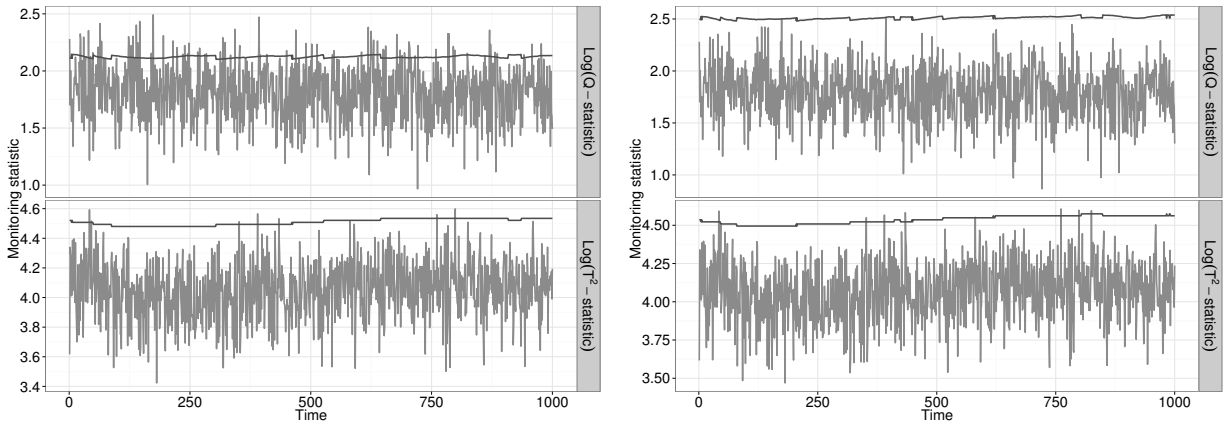

Figure 13: RPCA control charts based on conventional (left) and tuned (right) α values applied to an IMA(1,1) process.

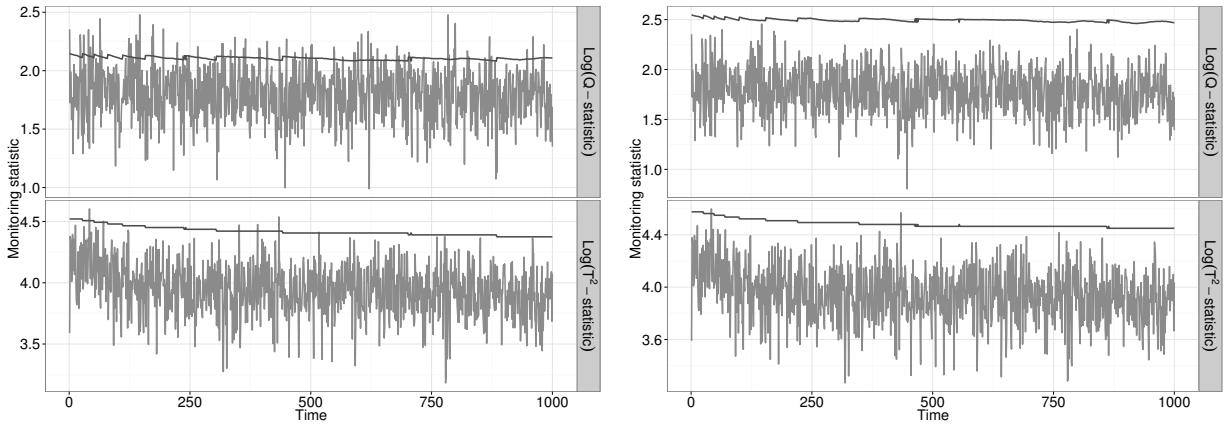

Figure 14: MWPCA control charts based on conventional (left) and tuned (right) α values applied to an IMA(1,1) process.

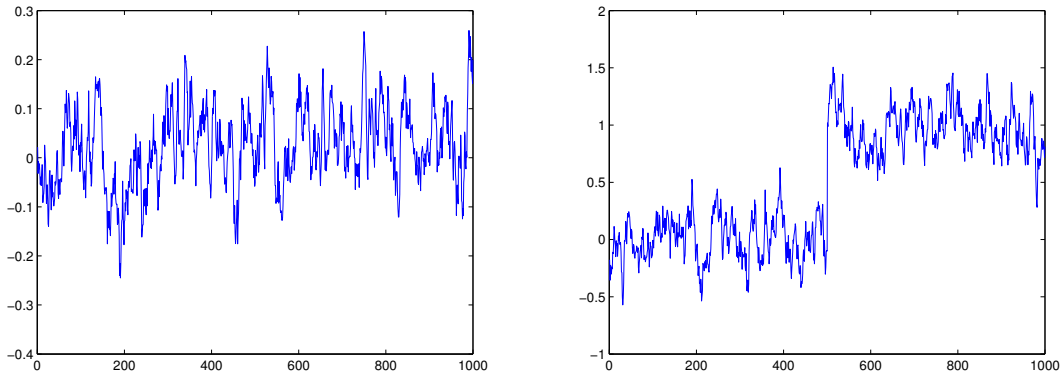

Figure 15: The first variable (left) and the first score (right) for an AR(1) process.

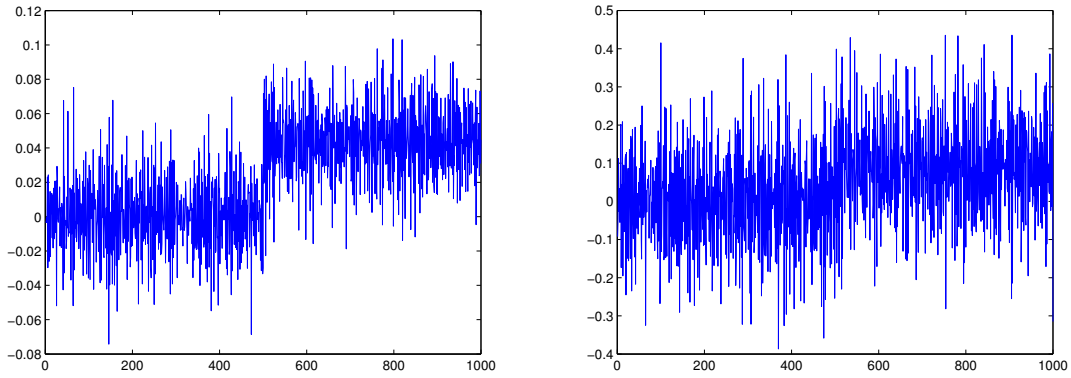

Figure 16: The first variable (left) and the first score (right) for an MA(1) process.

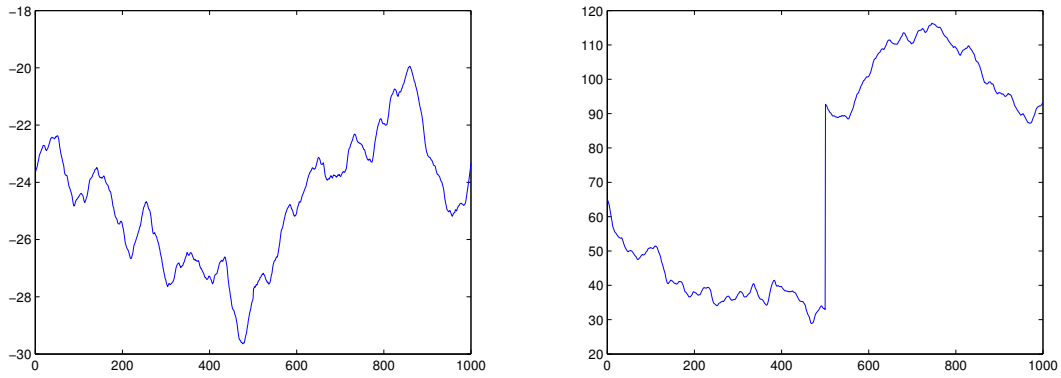

Figure 17: The first variable (left) and the first score (right) for an ARI(1,1) process.

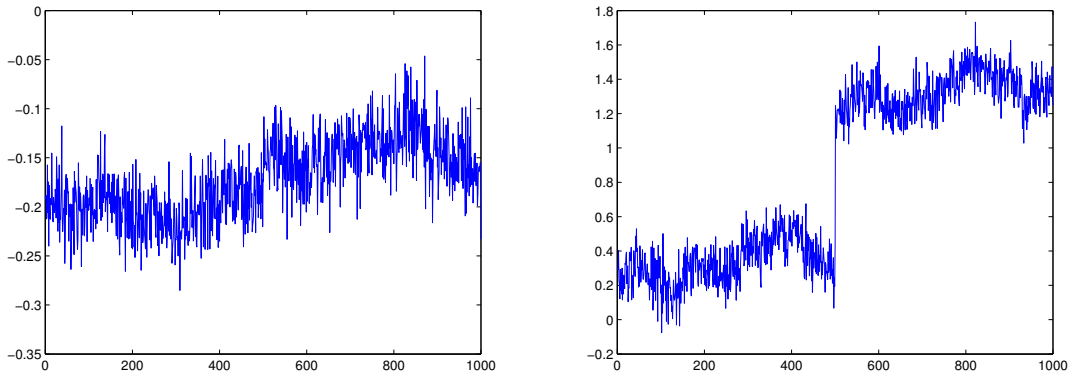

Figure 18: The first variable (left) and the first score (right) for an IMA(1,1) process.

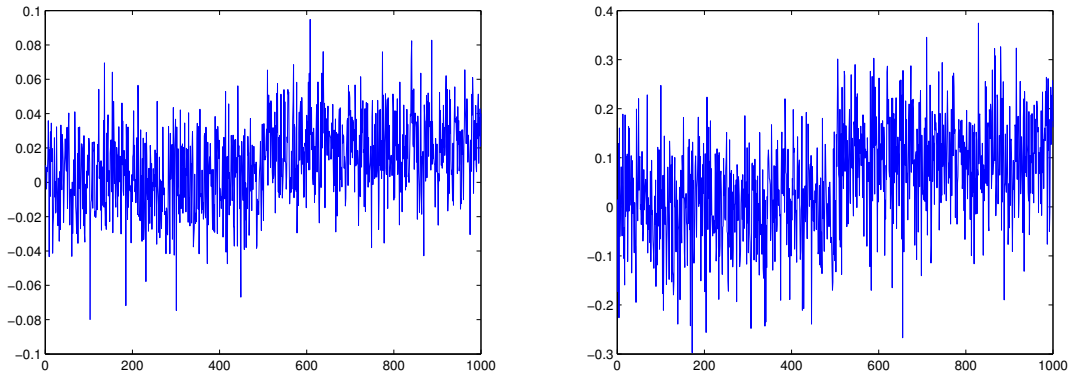

Figure 19: The first variable (left) and the first score (right) for an NSS process.

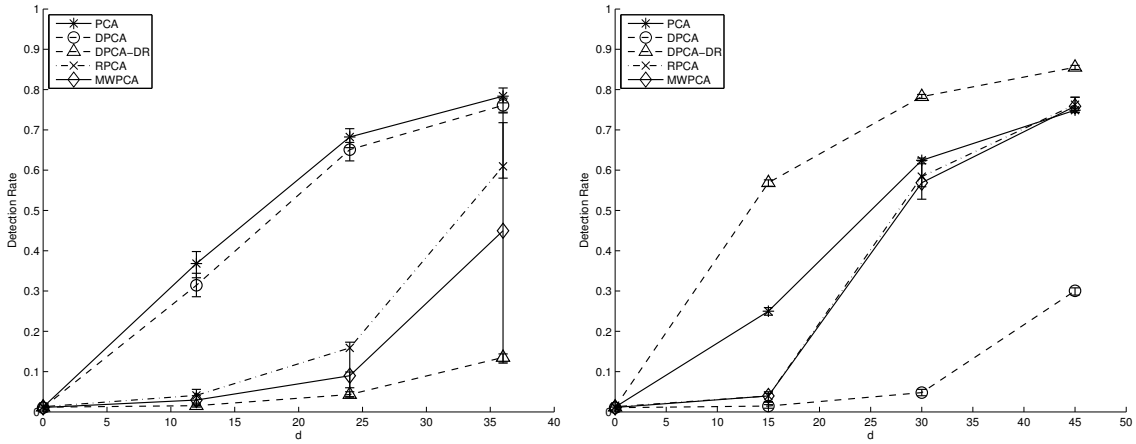

Figure 20: Fault detection rate curves for ramp deviations on one of the scores (left) and sensor measurements (right) of the AR(1) process. The fault magnitude is defined as d times the standard deviation.

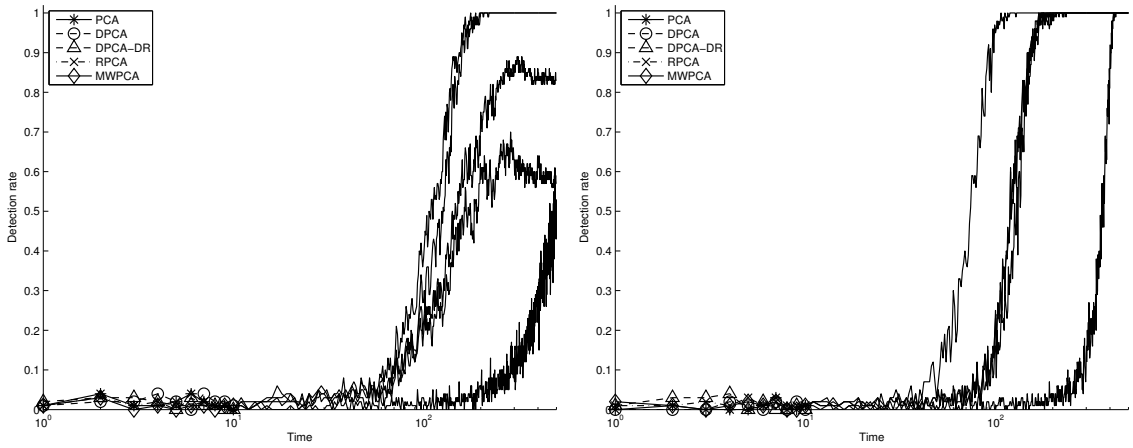

Figure 21: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor ramp faults for the AR(1) process at each faulty time period averaged over all runs.

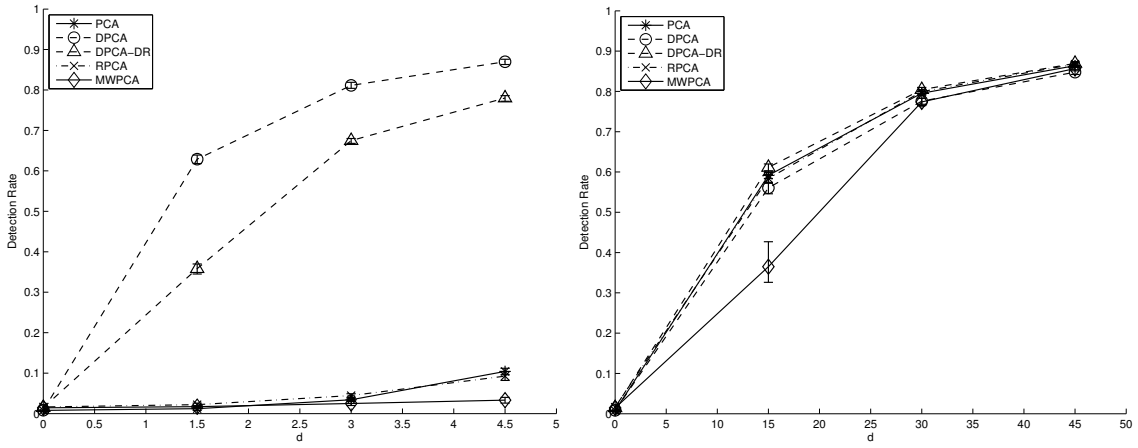

Figure 22: Fault detection rate curves for ramp deviations on one of the scores (left) and sensor measurements (right) of the MA(1) process. The fault magnitude is defined as d times the standard deviation.

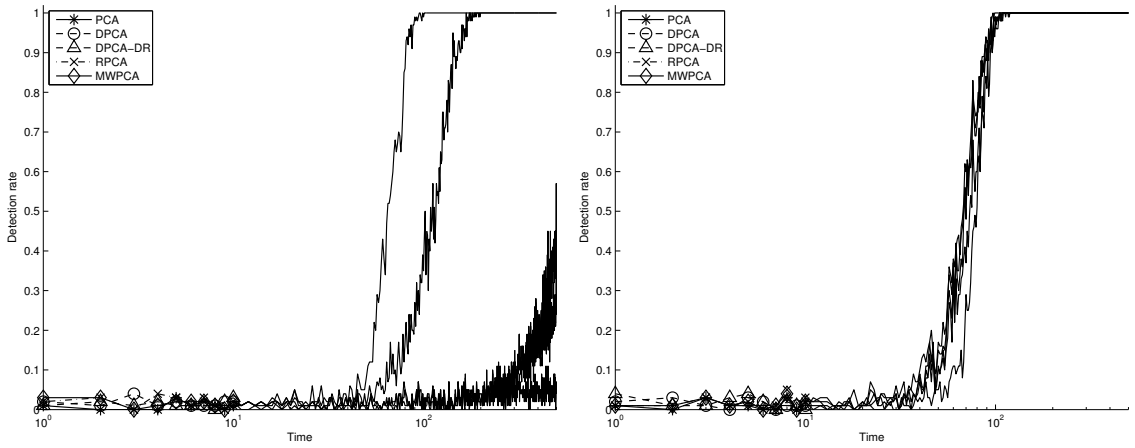

Figure 23: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor ramp faults for the MA(1) process at each faulty time period averaged over all runs.

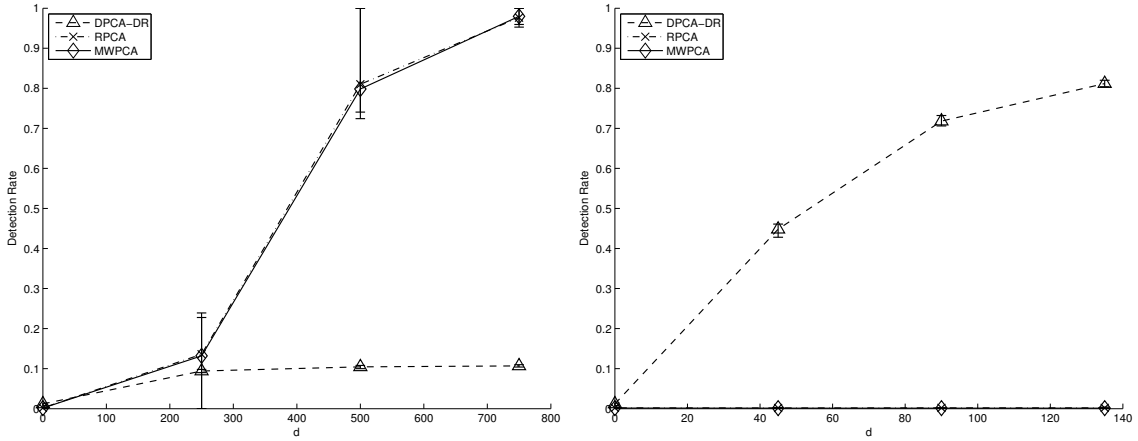

Figure 24: Fault detection rate curves for ramp deviations on one of the scores (left) and sensor measurements (right) of the ARI(1,1) process. The fault magnitude is defined as d times the standard deviation.

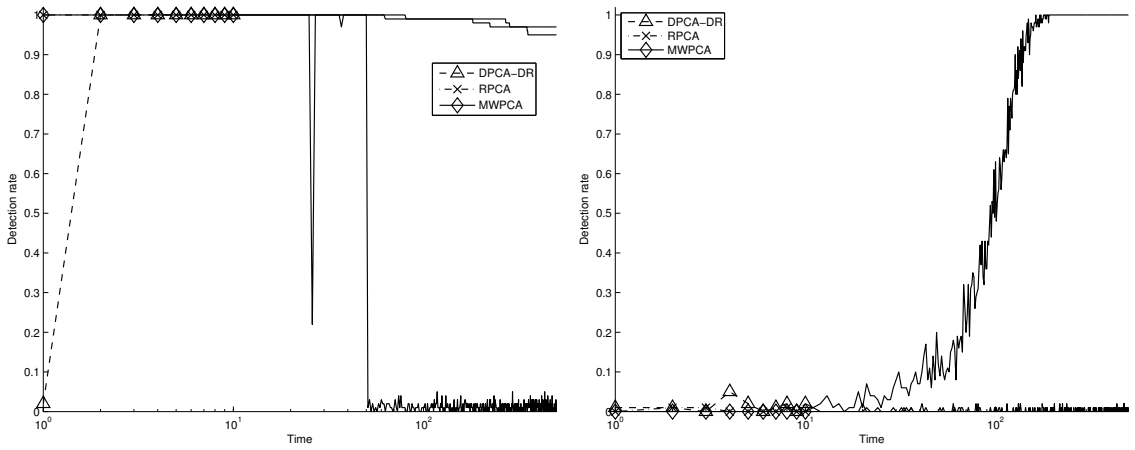

Figure 25: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor ramp faults for the ARI(1,1) process at each faulty time period averaged over all runs.

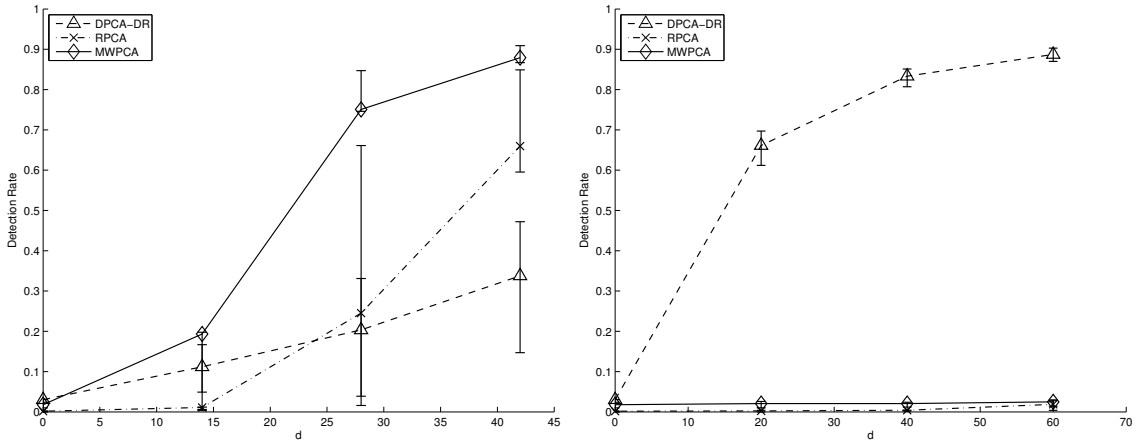

Figure 26: Fault detection curves for ramp deviations on one of the scores (left) and sensor measurements (right) of the IMA(1,1) process. The fault magnitude is defined as d times the standard deviation.

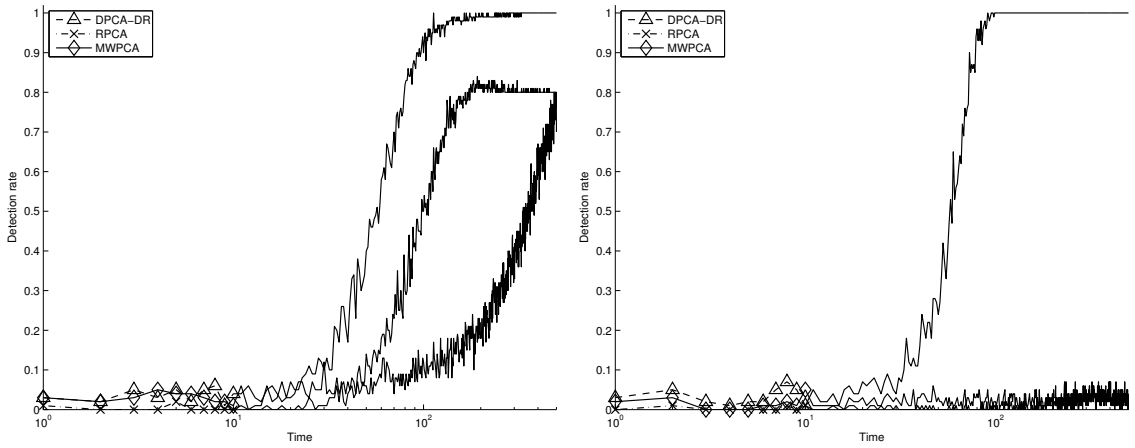

Figure 27: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor ramp faults for the IMA(1,1) process at each faulty time period averaged over all runs.

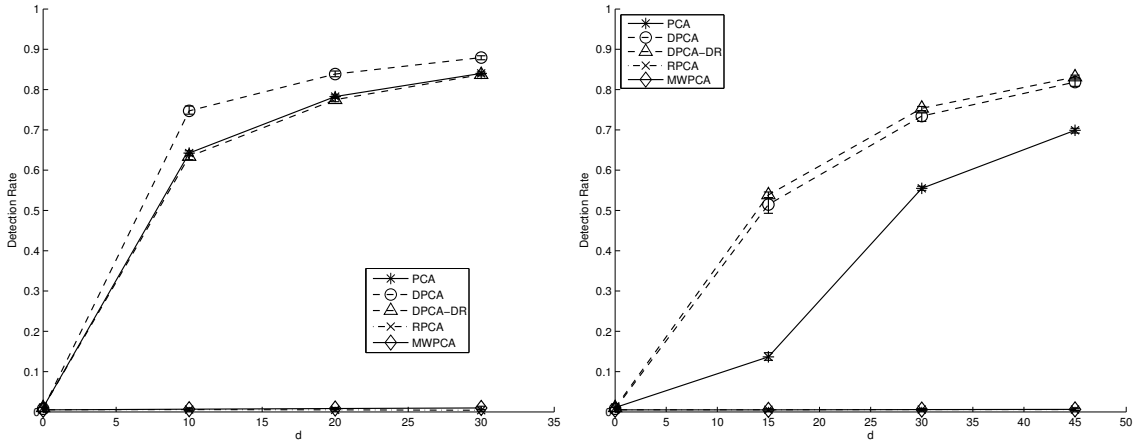

Figure 28: Fault detection rate curves for ramp deviations on one of the scores (left) and sensor measurements (right) of the NSS process. The fault magnitude is defined as d times the standard deviation.

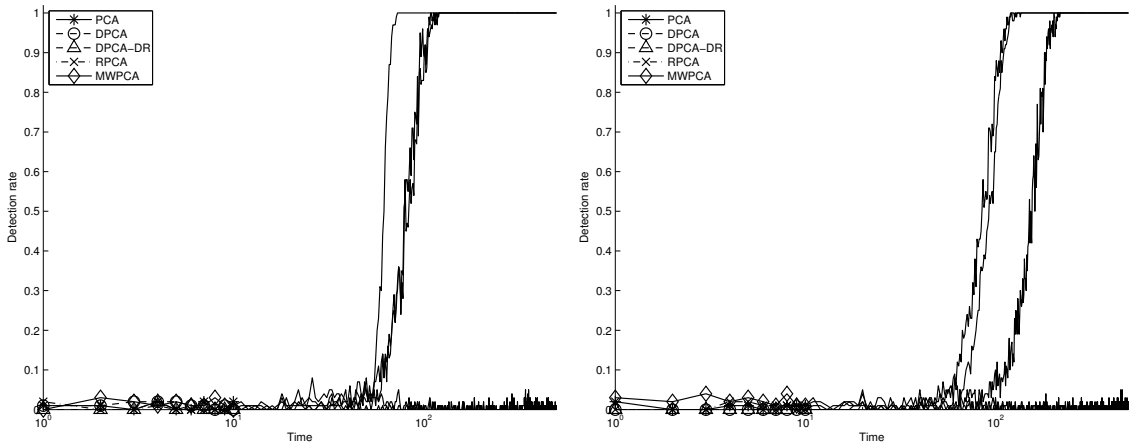

Figure 29: Score (left) and sensor (right) fault detection rate on the maximum simulated score and sensor ramp faults for the NSS process at each faulty time period averaged over all runs.

This material is available free of charge via the Internet at `http://pubs.acs.org/`.