Fault detection based on time series modeling and multivariate statistical process control

A. Sánchez-Fernández[a], F.J. Baldán[b], G.I. Sainz-Palmero[a], J.M. Benítez[b], M.J. Fuente[a,*]

[a] *Department of System Engineering and Automatic Control, EII, University of Valladolid, Valladolid, Spain*
[b] *Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain*

Abstract

Monitoring complex industrial plants is a very important task in order to ensure the management, reliability, safety and maintenance of the desired product quality. Early detection of abnormal events allows actions to prevent more serious consequences, improve the system's performance and reduce manufacturing costs. In this work, a new methodology for fault detection is introduced, based on time series models and statistical process control (MSPC). The proposal explicitly accounts for both dynamic and non-linearity properties of the system. A dynamic feature selection is carried out to interpret the dynamic relations by characterizing the auto- and cross-correlations for every variable. After that, a time-series based model framework is used to obtain and validate the best descriptive model of the plant (either linear o non-linear). Fault detection is based on finding anomalies in the temporal residual signals obtained from the models by univariate and multivariate statistical process control charts. Finally, the performance of the method is validated on two benchmarks, a wastewater treatment plant and the Tennessee Eastman Plant. A comparison with other classical methods clearly demonstrates the over performance and feasibility of the proposed monitoring scheme.

Keywords: Fault detection, Dynamic feature selection, Time-series modeling, Statistical process control charts

[*] Corresponding author, M.J. Fuente. Department of Systems Engineering and Automatic Control, EII, C/ Paseo del Cauce N.59, Post code: 47011, University of Valladolid, Valladolid, Spain, Tlf: +34 983 42 33 55

Email addresses: alvar.sanchez@uva.es (A. Sánchez-Fernández), fjbaldan@decsai.ugr.es (F.J. Baldán), gresai@eii.uva.es (G.I. Sainz-Palmero), j.m.benitez@decsai.ugr.es (J.M. Benítez), mjfuente@eii.uva.es (M.J. Fuente)

1. INTRODUCTION

The increasing complexity of modern industrial processes brings with it an increase in the importance of process monitoring to ensure plant safety and product quality [1]. Early detection of abnormal events allows actions to prevent more serious consequences, improve the system's behaviour and reduce maintenance and operation costs. The main objective of anomaly, or fault, detection is to identify any abnormal event indicating a distance from the process behaviour as compared to its normal behaviour. In other words, an anomaly occurs when the system deviates significantly from its normal situation during the on-line operation. The second goal is fault diagnosis (or isolation), which determines the root cause of the detected anomaly.

Process monitoring can generally be divided into three categories: model-based methods, knowledge-based methods, and data-based methods [2–4]. The first category is also called analytical redundancy (AR), in which an explicit model is used. Unusual events are detected by referencing the measured process behaviour against the model. However, as modern industrial processes become more and more complex it is difficult and time consuming to develop an accurate model that characterizes all the physical and chemical phenomena occurring in industrial processes. Fault detection using knowledge-based techniques is usually a heuristic process based on the available knowledge of the system's behaviour and the experience of expert plant operators. However, the creation of the process knowledge base is always a time consuming and difficult operation, requiring the long-term accumulation of expert knowledge and experience.

Furthermore, data-based anomaly-detection methods rely on the availability of historical data of the inspected system under normal operation mode. These methods have become more and more popular in recent years, especially in complex industrial processes, where models and knowledge are difficult to obtain in practice and where, because of the wide utilization of distributed control systems, large amounts of data have been collected. Such data contain the most process information and can be used for modelling and monitoring the process. In particular, multivariate statistical process control (MSPC), such as Principal Component Analysis (PCA), Partial Least Squares (PLS), etc. have been used to monitor different industrial processes [5, 6].

Industrial plants are normally non-linear and dynamic. To deal with the dynamic property in the processes, dynamic PCA utilizing an augmented matrix with time-lagged variables, which models the auto-correlation and cross-correlation among data samples, was proposed in [7] and used in different applications [8, 9]. Canonical variate analysis (CVA) has also been proposed for dynamic process monitoring [10], where both past data and future measurements are used to estimate the process state space model and build a fault detection scheme. However, there are some limitations for these techniques, as the high dimensionality of the augmented matrix, which cannot be applied to monitoring a large number of auto-correlated, cross-correlated and collinear variables.

Some process variables can affect other variables with a time-delay [11];

a time-delayed process variable might have a stronger relationship with other variables (delayed or not) than the non-delayed one. So, high cross-correlation among process variables with different time-delays is possible, and it is difficult to select both auto-correlation and cross-correlation together, i.e., dynamic features, for every process variable. However, it is expected that a proper selection will improve the monitoring scheme's behaviour.

To solve the problem of non-linearity, some extensions of non-linear PCA have been reported in the literature, such as [12], which develops a non-linear PCA based on autoassociative neural networks; while, more recently, [13] presents a similar idea, but using neural networks called invariant autoencoders (AE), equivalent to non-linear PCA, to extract a robust and non-linear representation of the process data. However, these methods usually require more computation and inevitably lead to the convergence to the local minimum during the network training. In addition, the number of principal components (PCs) must be specified in advance before training the neural networks.

A different idea is proposed in [14] and this is called Kernel PCA (KPCA), where non-linear data in the input space is transformed into linear data in a high dimensional feature space, through a non-linear mapping, a kernel trick, to calculate the principal components in the feature space [15]. However, the drawback of KPCA is that the computation time may increase with the number of samples, and the data pattern in the feature space is rather hard to interpret in the input space; so it is hard to identify the variables causing the fault. In addition, the fault detection of KPCA is very sensitive to the parameters of the kernel function used to implement it, especially for the radial basis function or the Gaussian kernel, the most commonly used one. Finally, there is no theoretical framework for specifying the optimal value for those parameters.

Motivated by the above considerations, the objective of this paper is to present a dynamic and non-linear fault detection and diagnosis methodology, that explicitly accounts for the dynamic relations in the process through dynamic feature selection, and for the non-linear relationship between the process variables through a time-series based model, capturing both the non-linear and the dynamic correlation in the process. Thus, the residuals, which are the difference between the process measurements and the output of the time-series model, can be monitored by conventional SPC charts, because the time-series model is used to remove the non-linear and the dynamic characteristics of the process. Therefore, the main objective of this paper is to combine the advantages of the SPC monitoring scheme and time-series modelling to enhance the performance of the monitoring scheme and widen its applicability in practice for complex and large-scale industrial processes.

The proposal works as follows: first, a dynamic feature selection method to characterize the dynamic relations of every process variable is carried out; then, a time series model is developed for every variable taking into account the features previously obtained; the residuals are then processed by different SPC charts. Every residual is processed by univariate statistics, the EWMA chart [16, 17], to detect and diagnose faults when one of the residuals exceeds its control limits. As this frequently happens, the systems have a very large number of variables, so it is better to process all the variables together to obtain a unique statistic to check whether there is a fault in the system or not. In this paper, the residuals are also processed by the well-known PCA algorithm, using the classical Hotelling's and SPE statistics to detect the faults.

The major contributions of the current work can be summarized as: (1) proposing a general methodology to monitor dynamic and non-linear complex process; (2) proposing a dynamic feature characterization for the dynamic processes; and (3) ensuring consistent monitoring performance compared to other fault detection methods driven by data.

The rest of this paper is organized as follows: Section 2 briefly reviews the theoretical concepts used related to the dynamic feature selection, the statistical process control and the time series modelling; Section 3 presents an explanation of the proposed approach; and Section 4 outlines the application of the proposed methodology to two complex systems: a waste water treatment plant and the Tennessee Eastman plant. Finally, Section 5 reviews the main points discussed in this work and concludes the study.

2. THEORY

In order to carry out this proposal, it is necessary to identify m different models, for a system with m sensors measuring m system variables, from the measurement data recorded by the sensors. In particular, for each system variable $x_i, i = 1, ..., m$, we are looking for a forecasting time-series model f_i based on a subset or a transformation of the other variables which are best able to predict this variable. At the same time, the aim is to find nominal system models with dynamic dependencies, i.e., where a variable might have influence on a target in several time steps in the future (but not immediately). Thus, it is aiming for a vectorized time-series model, where the prediction model f_i for each variable $x_i, i = 1, ..., m$ includes the necessary time-lags on a subset of other variables, which are the best to explain this variable. So, the data set containing the variable to be modelled is spanned to include lags along the m original non-delayed variables as inputs to the model.

$$x_i(t) = [x_1(t), ..., x_1(t - k_1), x_i(t - 1), ...x_i(t - k_i), ..., x_m(t), ...x_m(t - k_m)] \quad (1)$$

where $\{k_1, k_2, ..., k_m\} \subseteq \{0, ..., L\}$, thus allowing variables without lags to participate in the model definition. L denotes the maximal lags and will be set to a concrete value calculated with a dynamic feature selection. In this way, models are potentially obtained where no lags, different lags from the same variable, or different lags from different variables, may appear as inputs.

2.1. Dynamic feature selection

The interaction among different measured variables might be more appropriately represented on the basis of different time-delays. In order to model every variable, it is necessary to know which are the best relations for using. So a dynamic feature selection concerning auto- and cross-correlation with different time-delays is carried out. First, the matrix $\mathbf{X}$ (n samples $\times$ m variables), with the original variables in normal operation conditions, is augmented, taking for each observation its previous L observations and stacking the data matrix in the following manner:

$$\mathbf{X_a} = [\mathbf{X}_t|\mathbf{X}_{t-1}|\ldots|\mathbf{X}_{t-L}] \tag{2}$$

where the operator $|$ is the matrix concatenation operator, $\mathbf{X_a} \in \Re^{(n-L)\times(m(L+1))}$ is the augmented matrix, $\mathbf{X}_t$ is the data matrix $\mathbf{X}$ at the time instant t and $\mathbf{X}_{t-L}$ at the time instant $t-L$, that is, with a delay of L time samples.

To implement the dynamic feature selection, the relationship between two variables x_t^i and x_l^j at two different time instants is calculated as the absolute value of the correlation coefficient :

$$\Re_{\mathbf{i}}(\mathbf{x_t^i}, \mathbf{x_l^j}) = \left| x_t^{iT} x_l^j / \|x_t^i\| \|x_l^j\| \right| \tag{3}$$

where $i, j = 1, ..., m$ and $t, l = 1, ..., L$. This coefficient is a direct measure of the correlation between variables and, after the calculation of vector $\Re_{\mathbf{i}}$ for the i-th variable, the dynamic feature selection is carried out by selecting the variables in matrix $\mathbf{X_a}$ with high correlation values, i.e., $\Re_{\mathbf{i}} > \delta_i$, are selected as inputs to model the variable x_i, with δ_i being a cut off parameter for each variable that depends on the correlation present in the system.

Other method used for the dynamic feature selection is Dynamic Partial Least Squares (DPLS) [18]. It is a well known method to reduce the dimensionality of a system and it searches for a new feature set composed of linear combinations of the original ones. This method then adjusts a linear model using least squares over these new discovered features.

$$\mathbf{X} = \mathbf{TP^T} + \mathbf{E} \tag{4}$$

$$\mathbf{Y} = \mathbf{TQ^T} + \mathbf{F} \tag{5}$$

where $\mathbf{X}$ is the predictor matrix ($n \times m$), $\mathbf{Y}$ is the response matrix ($n \times p$), $\mathbf{T}$ is the first k terms of the latent variables or the score vectors, $\mathbf{P}$ and $\mathbf{Q}$, respectively, are the loading vectors of the data matrices $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{E}$ and $\mathbf{F}$ are the residual terms of PLS. In general, each score is extracted through deflating $\mathbf{X}$ and $\mathbf{Y}$ by the well known algorithm of the non-linear iterative partial least squares (NIPALS) until all variance in the data structure is explained [19]. These score vectors are calculated as:

$$\mathbf{T} = \mathbf{XR} \tag{6}$$

This matrix $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$ is very useful to show which variables in $\mathbf{X}$ are most related to the model response $\mathbf{Y}$. The way in which this method can perform the regression is:

$$\mathbf{Y} = \mathbf{XB_{PLS}} + \mathbf{F}, \quad \mathbf{B_{PLS}} = \mathbf{RQ^T} \tag{7}$$

So in this case, the augmented matrix $\mathbf{X_a}$ (eq. 2) is constructed and a PLS regression model is calculated, taking each of the system variables, $\mathbf{Y} = \mathbf{x_i}$, $i = 1, ...m$ as output variable and the matrix $\mathbf{X_a}$ as the predictor, without the corresponding i-th variable that is being modelled. The dynamic feature selection is carried out by selecting the variables in matrix $\mathbf{X_a}$ with high absolute value of regression coefficients, i.e., high values of the vector $\mathbf{B_{PLS}}$.

A third way for the dynamic feature selection to be done is by combining both methods, i.e., first to do a dynamic feature selection with the correlation coefficient between variables and, after that, refine the selection using the PLS method. So, the variables and their delays with bigger correlation coefficient are chosen first, and these variables are arranged in a data matrix $\mathbf{X_i}$, one for each variable in the system, $i = 1, ...m$. After that, a PLS regression model is obtained for the variable i-th, while the variables with bigger absolute values of the corresponding regression coefficients, $\mathbf{B_{PLSi}}$, are chosen as final input variables.

For the special case of ARIMA models, where the data of each variable are modelled using only the delayed information from each respective variable, the dynamic feature selection is done using the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) [20].

2.2. Time series modelling

Many methods have been developed to analyze and forecast time-series [21], making the choice of an appropriate method an important task. A brief summary of time-series models is presented:

Autoregressive Integrated Moving Average. The three univariate time-series models which have been widely applied are the autoregressive (AR), the moving average (MA) and the autoregressive and moving average models (ARMA), and when the system is not stationary, the ARIMA model ([20, 22]). When a time-series exhibits seasonality, a SARIMA model can be used to measure the seasonal effect or eliminate seasonality. In this time-series model (ARIMA), a variable is modelled through its own lags, and does not explicitly use the information contained in other related time-series. In industrial processes the variables are not independent and besides the autocorrelation of each variable, there exists high a cross-correlation among different process variables with different time-delays. To capture the relationships between them, time-series models with exogenous or input variables can be used, so the autoregressive moving average model with external inputs (ARMAX) and its variants (ARX, ARIMAX, SARIMAX,...) can be suitable.

Neural Networks. Neural networks are a useful non-linear modelling technique [23], especially the Multilayer Perceptions (MLP) neural networks, which are capable of approximating, to arbitrary accuracy, any continuous function as long as they contain enough hidden units [24]. Their attractive properties have led to the rise of several types of NNs and applications in the literature in different fields ([25–27]).

Support vector regression. Support vector machine (SVM) analysis is a popular machine learning tool for classification and regression, first identified by

[28]. It supposes a training dataset (x_i, y_i) $i = 1, ...N \subset \mathcal{X} \times \mathbb{R}$ where $\mathcal{X}$ denotes the space of input patterns. The goal is to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y_i for all training data and, at the same time, is as smooth as possible. This is known as ε- SV regression:

$$f(x) = \langle w, x \rangle + b \tag{8}$$

with $w \in \mathcal{X}$, $b \in \mathbb{R}$. When the problem is non-linear, an SVM can be used as a non-linear regression function by using the kernel trick to map the input feature space to a higher dimensional feature space, where a linear decision function is constructed. ([29–31])

2.3. Statistical process control techniques

The aim of Statistic Process control (SPC) is to monitor a process to detect abnormal behaviour. In order to detect faults, the statistical control charts (also referred to as monitoring charts) are used [32]. They present a value over time and some thresholds that must not be surpassed in normal operating conditions. So, these control charts are crucial in detecting whether a process is still working under normal operating conditions (usually termed in-control) or not (out-of-control). Within this framework, different control charts have been developed to monitor the system over time, including [33]:

- Shewhart control chart [34]: this represents the time evolution of the mean of a variable with upper and lower limits. If one of these thresholds is exceeded, a fault is detected. To avoid false alarms, it is usual to require a certain number of consecutive instants with abnormal values to activate the fault alarm.
- Cumulative Sum (CUSUM) charts [35, 36]: here the method represents the cumulative addition of deviations in every observation. It is able to detect small variations faster than Shewhart charts. This technique requires a value k of past observations to be set and these are used to calculate the cumulative sum at the current time.
- Exponentially Weighted Moving Average (EWMA) control chart [16]: this method filters the data. It computes a decision function for each observation $z_i(t)$ based on the current data and the past average values:

  $$z_i(t) = \lambda x_i(t) + (1 - \lambda) z_i(t-1) \tag{9}$$

  where x_i is the i-th value of the monitored variable at time t, and λ is the degree of weighting that determines the temporal memory of the EWMA decision function. With $0 \leqslant \lambda \leqslant 1$, lower values of λ give more influence to the past values, while higher values gives more importance to the current value. The control limits are calculated as:

  $$UCL, LCL = \mu_0 \pm L\sigma_0 \sqrt{\frac{\lambda}{2 - \lambda}} \tag{10}$$

where L is the width of the control limits (which determines the confidence limits, usually specified depending on the false alarm rate) and μ_0 and σ_0 are the mean and standard deviations of anomaly-free data.

These control charts are univariate SPC charts and their performance is based on the prior assumption that the process data are not correlated. Such an assumption may not be valid in industrial processes. Then two options can be considered. The first is to use a time-series modelling, i.e. to model the variables and use the residuals, resulting from this time-series modelling that are approximately uncorrelated, to monitor the process with the conventional SPC charts. The second option is to use a multivariate statistical process control tool, such as Principal Component Analysis (PCA).

2.3.1. PCA

Principal Component Analysis develops a linear transformation of the original data into a set of uncorrelated variables (principal components). The first principal component has the largest possible variance, the second has the second largest variance, and so on. Choosing the first principal components, it is possible to reduce dimensionality without losing too much information.

This method decomposes the covariance matrix $\mathbf{S}$ of the original data matrix $\mathbf{X}(n \times m)$ with data collected from the system in normal operation conditions, using Singular Value Decomposition:

$$\mathbf{S} = \mathbf{P}\mathbf{\Lambda_k}\mathbf{P^T} + \tilde{\mathbf{P}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{P}}^{\mathbf{T}} \tag{11}$$

where $\mathbf{\Lambda_k}$ contains, in its diagonal, the k most significant eigenvalues of $\mathbf{S}$, in decreasing order. Their associated eigenvectors are contained in $\mathbf{P}$. The residual eigenvectors and eigenvalues $(m-k)$ can be found in $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{\Lambda}}$, respectively.

Fault detection using PCA is performed with *Hotelling's statistic* (T^2) and the *Square Prediction Error (SPE)* statistic ([37–39]). The plant is in normal operation conditions, for an α significance level, if T^2 is under its threshold T^2_α:

$$T^2 = x^T\mathbf{D}x < T^2_\alpha \tag{12}$$

where $\mathbf{D} = \mathbf{P}\mathbf{\Lambda_k^{-1}}\mathbf{P^T}$.

The SPE statistic (or Q), for a new observation x, considers normal condition when Q is under the threshold Q_α:

$$Q = x^T\tilde{\mathbf{C}}x < Q_\alpha \tag{13}$$

where $\tilde{\mathbf{C}} = \tilde{\mathbf{P}}\tilde{\mathbf{P}}^{\mathbf{T}}$. The method to obtain the thresholds for T^2 and Q is explained in [40].

3. PROPOSED APPROACH

The detection of an abnormal situation in industrial processes can be accomplished using techniques based on data. Here, a time-series model was integrated with SPC charts to develop a methodology for monitoring these situations. The goal is to detect abnormal events, i.e., faults in a plant, as fast as possible, by solely analyzing data. This method consists of four steps: dynamic feature selection, time-series modelling, construction of the control chart limits and fault detection using SPC tools. The general methodology proposed is outlined in Figure 1 and in the following steps:

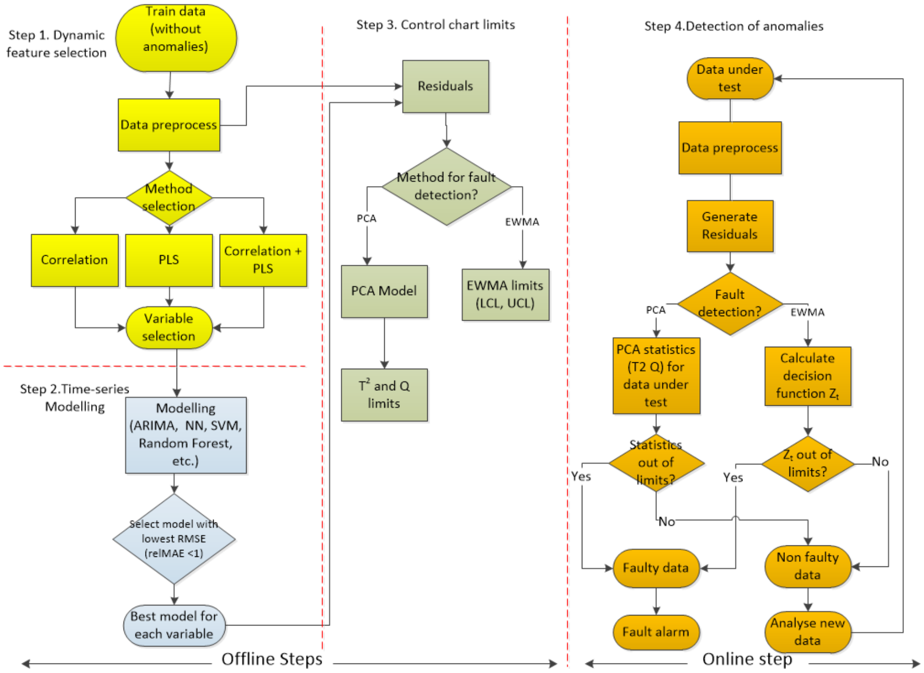
Step 1. Dynamic feature selection
Train data (without anomalies)
Data preprocess
Method selection
Correlation
PLS
Correlation + PLS
Variable selection
Step 2.Time-series Modelling
Modelling (ARIMA, NN, SVM, Random Forest, etc.)
Select model with lowest RMSE (relMAE <1)
Best model for each variable
Step 3. Control chart limits
Residuals
Method for fault detection?
PCA
EWMA
PCA Model
EWMA limits (LCL, UCL)
T^2 and Q limits
Step 4.Detection of anomalies
Data under test
Data preprocess
Generate Residuals
Fault detection?
PCA
EWMA
PCA statistics (T2 Q) for data under test
Calculate decision function Z_t
Statistics out of limits?
Z_t out of limits?
Yes
No
Yes
No
Faulty data
Non faulty data
Fault alarm
Analyse new data
Offline Steps
Online step

Figure 1: Flowchart of the proposed anomaly detection methodology

Step 1. Dynamic feature selection. The variables in an industrial plant are not usually independent. There are auto- and cross-correlations among them, i.e., the interaction between different variables might be more appropriately represented on the basis of different time-delays. So, here, a dynamic feature selection method concerning auto- and cross- correlation with different time-delays is presented. This step is composed of sub-steps as follows:

1. Collection of training data representing the normal operation conditions of the process.
2. Data pre-processing, which consists of data analysis, to eliminate outliers, to scale the data to the adequate range, and so on. The outliers are identified and modified using the Tukey method [41]
3. To select the most appropriate variables with their time delays to represent every variable of the system, i.e., to carry out the dynamic feature selection, using the different methods presented in section 2.1.

Step 2.-Time-series Modelling. The second step is to build a reference descriptive model using the data set defined in the first step representing normal situations. In this step, another data pre-processing, which consists of the analysis (trends, seasonality in the time series) and scaling of the data is first carried out. Then, the normal operation data are divided into two groups by n-cross validation, training and test data. Third, different models can be built for every variable of the system, i.e., the goal is to find the best time-series descriptive model for every system variable with the dynamic features selected in step 1 from training data. To automate the process of model building, grid search can be used for hyperparameter optimization. Some measures of the error committed in forecasting each test time series with each respective model are used to select the preference of these models. These error measures are:

- rMSE (root-mean-square error): the square root of the mean squared error (MSE)
- sMAPE (Symmetric mean absolute percentage error): the average value of the absolute value of the error in percentage.

  $$sMAPE = 100\%/n \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|F_t| + |A_t|)/2} \tag{14}$$

  where n is the number of samples, F_t is the forecast at time t and A_t is the current measure at that instant.
- relMAE (relative mean absolute error): divides the average value of the absolute error by the mean absolute error obtained by another model considered as a reference. In this article, this reference model is the “naive ”forecast, which supposes that the next observation will be the same as the current one.

  $$relMAE = \frac{MAE}{MAE_p} \tag{15}$$

  where MAE is the mean absolute error of the tested model and MAE_P is the same measure, but applied to the reference model.

The model with the lowest rMSE value will be selected, for every variable, provided that the relMAE is under 1. The other measures: sMAPE and relMAE will be used to know the performance of the selected model.

Step 3.-Computation of the control chart limits. In this step, the control limits of every chart used in this methodology need to be calculated. They are based on the residuals, calculated as the difference between the prediction for each variable using the respective time-series model chosen in step 2 and the actual measurement of this variable. Here, it is possible to use two different control charts:

1. The EWMA control chart. This chart works with the variables individually, i.e., a control chart is calculated for every residual, and in this case the upper control limit (UCL) and the lower control limit (LCL) of every control chart need be computed using eq. 10.
2. PCA. In this case all the residuals are taken into account together, and two charts are defined to detect anomalies. First a PCA model is calculated using the residuals in normal operation conditions, and the thresholds for the Hotelling's and Q statistics are calculated using the equations defined in section 2.3.1.

These three steps are calculated off-line.

Step 4.- Detection of anomalies. This step is calculated on-line and is composed of various sub-steps:

1. Collecting test data from the plant that may possibly contain abnormal situations. The aim now is to detect whether there are anomalies in this test data set.
2. Pre-process the data in the same way as with the training data. The dynamic feature selection performed in step 1 is carried out.
3. Generate the residuals, which are the difference between the measurements and the output of the time-series models constructed in step 2, for every variable.
4. Calculate the decision function $z_i(t)$ as eq. 9 for every residual for the EWMA control chart. Compute the statistics T^2 and Q with the equations 12 and 13 respectively, for the PCA control charts.
5. Detection for abnormal situations. Compare the defined statistics with their corresponding thresholds. If any of the control charts defined (the decision function $z_i(t)$, the T^2 or the Q statistics) exceed their respective thresholds, for a number of consecutive observations (Obs), an anomaly is declared. There must be a consecutive number of observations before declaring an alarm, this is done in order to avoid false alarms. In this case, a fault alarm is sent to the operator so that the appropriate corrective actions can be taken. If no anomaly is detected (the three control charts do not exceed their control limits), there is no anomaly and the monitoring process goes on.

To detect abnormal situations, the residuals can be used as an indicator. These residuals are close to zero when the behaviour of the monitored system is normal. However, when an abnormal situation or fault occurs, the residuals deviate significantly from zero, indicating the presence of a new situation that is distinguishable from the normal one.

An overview of this fault detection methodology is presented in Algorithm 1.

4. ILLUSTRATIVE EXAMPLES: SIMULATION CASE STUDIES

This section presents the results of applying the proposed methodology to two well-known benchmarks: the Tennessee Eastman Process and a Waste Wa-

Algorithm 1 *Fault detection*

```
for Non Faulty Data, off-line do
    Step 1. Dynamic feature selection
    Normalize data
    Do the dynamic feature selection for each variable using the three methods: correlation, PLS and a combination of both methods.
    Step 2. Time series-modeling
    Calculate different time-series models for each variable: ARIMA, ARMAX, NN, SVR, random Forrest, etc.
    Choose the best model for each variable, using the rMSE, and relMAE indexes
    Step 3. Computing the control chart limits
    Get predictions for each variable                    ▷ Using every chosen model
    Obtain residuals:
        $Residual^{NF}(t) = Observed(t) - Predicted(t)$
    Develop $PCA_{residuals}$                            ▷ PCA model with $Residual^{NF}(t)$
    Define the control limits UCL and LCL for the control charts
    Define the thresholds $T^2_\alpha$ and $Q_\alpha$ for the $T^2$ and $Q$ statistics.
end for

Step 4. Detection of anomalies
Analyze new data, on-line:
for t=1 to n do                                          ▷ For each instant
    Get and normalize a new observation: $x_n(t)$
    for i=1 to m do                                      ▷ For every variable m
        Do the dynamic feature selection defined in step 1
        $Prediction^i(t) : \hat{x}^i_n(t) = f(model^i, (t, t-1, t-2, ...))$
        $Residual^i(t) = x^i_n(t) - \hat{x}^i_n(t)$
    end for

    Fault detection based on EWMA chart:
    for i=1 to m do                                      ▷ For every variable m
        Compute the decision function $z^i(t) = \lambda Residual^i(t) + (1-\lambda) z^i(t-1)$
        if $z^i(t) > UCL$ then
            $Fault = TRUE$ in time $t$
        end if
    end for

    Fault detection based on PCA of residuals:
    Calculate $T^2(t)$ and $Q(t)$ for $Residual(t)$
    if $T^2(t) > T^2_\alpha$ or $Q(t) > Q_\alpha$ then
        Fault=TRUE in time $t$
    end if
end for
```

ter Treatment Plant (WWTP).

The performance of the proposed monitoring methodology is validated by the most common indexes for evaluating process monitoring performance: the missed detection rate (MDR), the false alarms rate (FAR), the fault detection delay and the number of detected faults ([5, 42–44]). The objective of a fault detection technique is for it to be robust to data independently of the training set, sensitive to all the possible faults of the process, and quick to detect the faults. The robustness of each statistic was determined by calculating the false alarm rate during normal operating conditions for the test set and comparing it against the level of significance upon which the threshold is based. The sensitivity of the fault detection techniques were quantified by calculating the missed detection rate and the promptness of the measures is based on the detection delays. Missed detection rate (MDR) denotes that faulty data are misclassified as faultless data, i.e., the MDR is the number of faulty data samples not exceeding the control limits over the total number of faulty data:

$$MDR = 100\frac{N_{F,N}}{N_F}\% \tag{16}$$

where $N_{F,N}$ is the number of faulty samples identified as normal and N_F is the number of faulty samples.

The FAR is the number of normal data samples classified as faulty data over the total number of faultless data and is defined as:

$$MDR = 100\frac{N_{N,F}}{N_N}\% \tag{17}$$

where $N_{N,F}$ is the number of normal samples identified as faults and N_N is the number of normal samples.

4.1. Case Study 1: Tennessee Eastman Process

The first case study is the Tennessee Eastman Process (TEP) (Fig. 2) [45]. This benchmark was created to provide a realistic industrial process for control and monitoring studies. It contains five major units: a reactor, a stripper, a condenser, a recycle compressor, and a separator. A detailed process description including the process variables and the specific plant wide closed-loop system can be found in [46].

The available data for this plant consist of 22 continuous process measurements, 11 manipulated variables and 19 sampled process measurements. So, 52 variables are available. The data was generated with a sampling interval of 3 min. There are two data sets available: the training set and the test set, each of which have 22 time-series of 500 and 960 observations, respectively. The first one is faultless data, and the other 21 series are data from different fault situations. The faults start at the first observation in the training data sets, and at the 160-th observation in the test data sets. Process faults are detailed in Table 1. Faults 3, 9 and 15 are hard to detect [47]. These data are generated by [46] and can be downloaded from http://web.mit.edu/braatzgroup/links.html.

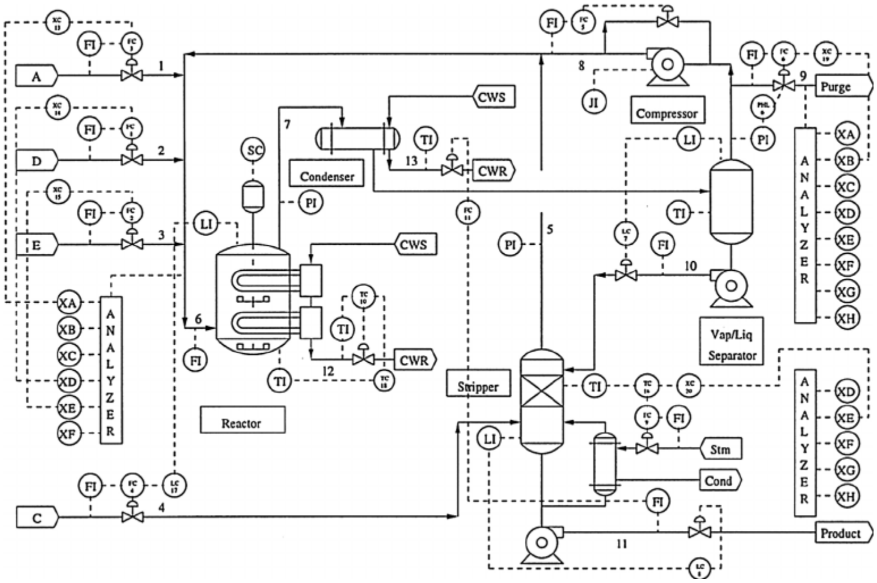

Figure 2: Tennessee Eastman Process

Table 1: TEP faults

Fault #	Description	Type
1	A/C feed ratio, B composition constant (Stream 4)	Step
2	B composition, A/C ratio constant (Stream 4)	Step
3	D feed (Stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (Stream 1)	Step
7	C header pressure loss-reduced availability (Stream 4)	Step
8	A, B and C compositions (Stream 4)	Random variation
9	D feed temperature (Stream 2)	Random variation
10	C feed temperature (Stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16	Unknown	-
17	Unknown	-
18	Unknown	-
19	Unknown	-
20	Unknown	-
21	Stream 4 valve	Sticking

4.1.1. Experimental methodology

The training normal data (no faults) $\mathbf{X} \in \Re^{500\times 52}$ are used for the off-line calculations of this proposal, which can be applied using the following steps (as presented in algorithm 1).

Step 1.- Dynamic feature selection. The first step is the dynamic feature selection, i.e., to discover which delayed variables must be used to model each variable; this is done using the three methods explained in section 2.1 for all the variables and their 15 first delays: correlation analysis, PLS feature extraction and a combination of both techniques. So it is necessary to construct the matrix X_a defined in Eq. 2, with $L = 15$. The 10 most correlated delayed variables are used for each variable for the next step: modelling, this is done to avoid over-fitting models, especially for the neural networks.

Step 2.- Building the time-series model. The faultless training data (500 observations) were used to model each variable with its corresponding dynamic features as inputs. First, these data were normalized according to the time-series model to be calculated. For example, the data for the neural networks must be normalized to range [0.2, 0.8] not the most common $[0, 1]$, because it is necessary to leave some room for bigger observations when new data are processed. For the ARIMA models, the data were normalized to have zero mean and differentiate the series if it was not stationary, etc. Then, different types of models were developed for every system variable. In this work, the developed time-series models were:

- ARIMA models, with the structure ARIMA(p,q,i), where the order of each part of the model AR (p), MA (q), and Integral part (i) is specified. In the experiments carried out, the variation of these parameters was: $p = [0, ..., 15]$, $q = [0, ..., 15]$ and $i = [0, 1, 2]$, with the additional condition that: $p + q \leq 20$.

- Neural network models. The NN used is always a Perceptron Multilayer network (MLP) with three layers, where the input layer has 10 inputs corresponding to the inputs selected in the dynamic feature selection of step 1, and the output layer has only one neuron. The number of neurons in the hidden layer is a parameter to be modified in each of the calculated models, and this parameter varies as follows: $neurons\ in\ hidden\ layer = [5, ..., 6 \times (2 \times number\ of\ inputs)]$.

- Support Vector Machine Regression (SVR) with a Gaussian Kernel: $k(u, v) = exp(-\gamma|u - v|^2)$. In this model it is necessary to adjust two parameters: C and γ, in the experiments carried out these parameters vary as follows: $C = [0.1,\ 1,\ 10, 100,\ 1000]$ and $\gamma = [0.01,\ 0.1,\ 1,\ 10,\ 100]$

- Support Vector Machine Regression (SVR) with a sigmoid kernel: $k(u, v) = tanh(\gamma u^T v + coef)$. In this model it is necessary to adjust two parameters: C and γ, in the experiments carried out these parameters vary as follows: $C = [0.1,\ 1,\ 10, 100,\ 1000]$ and $\gamma = [0.01,\ 0.1,\ 1,\ 10,\ 100]$

- Random forest models. Number of variables randomly sampled as candidates at each split : [200, 300, 400]. Number of trees to grow: [1500, 2000] this value should not be set at too small a number to ensure that every input variable gets predicted at least a few times.

The model with the lowest rMSE value using test data was selected. In this application, the selected time-series model for every variable and its structure can be seen in Table 2, where ANN are the neural networks with correlation coefficients variable selection and PLS ANN are the neural networks with PLS based variable selection. The final number of neurons in the hidden layer are shown in this Table. Note that some variables have an ARIMA model with parameters (0,0,0). These variables are white noise and they cannot be modelled, so in this case the model used to calculate the residuals is just the mean value of the variable in normal conditions.

Table 2: Selected time-series model for every variable

Variable	Model	Variable	Model
1	ANN: Hidden Layer:26	27	ARIMA: (0,0,0)
2	ANN: Hidden Layer: 13	28	ARIMA: (0,1,0)
3	ANN: Hidden Layer: 6	29	ARIMA: (2,2,0
4	ANN: Hidden Layer: 13	30	ARIMA: (1,1,0)
5	ANN: Hidden Layer: 13	31	ARIMA: (1,3,0)
6	ANN: Hidden Layer: 8	32	PLS ANN: Hidden Layer: 3
7	ANN: Hidden Layer: 23	33	ARIMA: (2,0,0)
8	ANN: Hidden Layer: 5	34	ARIMA: (1,1,0)
9	ANN: Hidden Layer: 5	35	ARIMA: (1,1,0)
10	ANN: Hidden Layer: 12	36	ARIMA: (1,0,0)
11	ANN: Hidden Layer: 5	37	ARIMA: (0,0,0)
12	ANN: Hidden Layer: 10	38	ARIMA:(1,2,0)
13	ANN: Hidden Layer: 13	39	ARIMA: (0,1,0)
14	ANN: Hidden Layer: 7	40	ARIMA: (0,0,0)
15	ANN: Hidden Layer: 5	41	ARIMA: (0,0,0)
16	PLS ANN: Hidden Layer: 3	42	ANN: Hidden Layer: 20
17	ANN: Hidden Layer: 24	43	ANN: Hidden Layer: 16
18	ARIMA: (2,2,1)	44	ANN: Hidden Layer: 7
19	PLS ANN:Hidden Layer: 8	45	ANN: Hidden Layer: 24
20	ARIMA: (2,2,0)	46	ANN: Hidden Layer: 13
21	ANN: Hidden Layer: 7	47	ANN: Hidden Layer: 11
22	ANN: Hidden Layer: 15	48	ANN: Hidden Layer: 11
23	ARIMA: (0,1,0)	49	ANN: Hidden Layer: 9
24	ARIMA: (1,1,0)	50	ARIMA: (1,2,1)
25	ARIMA:(3,1,0)	51	ANN: Hidden Layer: 21
26	PLS ANN: Hidden Layer: 4	52	ANN: Hidden Layer: 15

Step 3.- Computing the control chart limits. Now a forecast is obtained for every variable using the respective model and then compared with its actual measured value, resulting in a residual. Based on these residuals, the control limits for both control charts have to be calculated.

1. *Residuals EWMA control chart (EWMAres).* The thresholds can be ob-

tained using the equation 10, and were adjusted experimentally by doing some tests with different values of λ and getting the minimum number of consecutive anomalous observations (Obs) for each value to activate the fault alarm, using the training faultless data. The selected values after some experiments were $\lambda = 0.7$ and $Obs = 6$. This combination gives a good result in terms of false alarms rate, missed detection rate and minimum detection time as can be seen in the results.

2. *Fault detection based on PCA of residuals (PCAres).* Here, it is first necessary to create a PCA model using the calculated residuals. This PCA is developed with 85% of variance in the selected principal components (this is a trade-off between maximizing information contained in principal components and minimizing the number of them). Now, it is necessary to calculate the thresholds for the statistics T^2 and Q, taking into account the fact that it is necessary to consider a consecutive number of observations exceeding the threshold (Obs) in order to detect a fault. The thresholds of these statistics were calculated theoretically as explained in [37] and, after that, these limits were tuned experimentally for an imposed significance level (ISL or α) of 1%. This value is the expected percentage of alarms for the system under normal operation conditions. The value of $Obs = 3$ is calculated experimentally to ensure 0% alarms in the Q and T^2 statistics in the training set.

4.1.2. Detection Results and discussion

Here, step 4 of the methodology is carried out, i.e., the on-line detection of anomalies. The fault detection performance of this proposal has been compared with various data-driven multivariate statistical methods applied to the TE plant: PCA, dynamic PCA (DPCA), Canonical Variate Analysis (CVA) (using T_s^2, T_r^2, and Q statistics) [42], and Kernel PCA (KPCA) [48]. The comparison is made in terms of performance indexes defined at the beginning of this section: number of detected faults (bigger is better), false alarms rate, missed alarms rate (lower is better) for each fault and detection delay (also lower is better).

The number of faults detected for every method is presented in Table 3. The better performance of the proposed methods is clear, because, while the other methods can detect between 16 to 18 faults, *EWMAres* method can detect 20 out of 21 faults and *PCAres* with the Q statistical is able to detect all the faults.

Table 3: Faults detected by each method. TEP

	PCA T^2	**PCA** Q	**DPCA** T^2	**DPCA** Q	**CVA** T_s^2	**CVA** T_r^2	**CVA** Q	**KPCA** T^2	**KPCA** Q	**EWMAres**	**PCAres** T^2	**PCAres** Q
Faults detected	16	18	17	18	18	18	16	18	18	20	17	**21**

To check the robustness of the methods, the False Alarms Rate (FAR), also called Type I error, is calculated using the training data set (500 observations) for normal operation conditions ([5, 42–44]). The thresholds in all the methods have been defined to obtain a false alarm rate in normal operation conditions similar to the ISL, i.e., in terms of 1%. After this, the Type I error is checked by testing the second normal data set of 960 samples. The results for the methods are presented in Table 4. It can be observed that the minimum false alarms rate for the testing data (0.39%) is obtained by the *PCAres* method with the T^2 statistic proposed in this paper, whereas the DPCA method with the Q statistic achieves the highest false alarm rate (28.1%), a very high value.

All the statistics in the CVA method also have a very high false alarm rate, in the range (8% to 12%). From the aspect of engineering practice, it is important to trigger a fault alarm after detecting 3 to 6 abnormal samples consecutively. Therefore, the Type I error (0.39% and 1.06% of *PCAres* with the T^2 and Q statistic respectively) is acceptable, since it is calculated by counting single samples. Note that, to avoid this false alarm rate when the system is working on-line, a number of consecutive samples has to exceed the threshold to detect an abnormal situation. In this paper, three is considered to be the minimum number of consecutive anomalous samples to detect a fault. So, in all the situations considered in the paper (normal and the 21 faults), the real false alarm rate is 0%. Based on this analysis, it can be concluded that the proposed methods are applicable to monitoring dynamic processes.

Table 4: False Alarms Rate (FAR) in %. TEP

Method and Statistic	**Training data**	**Test data**
PCA T^2	0.2	1.4
PCA Q	0.4	1.6
DPCA T^2	0.2	0.6
DPCA Q	0.4	28.1
CVA T_s^2	1.3	8.3
CVA T_r^2	0	12.6
CVA Q	0.9	8.7
KPCA T^2		1.5
KPCA Q		2
EWMAres	0.94	1.6
PCAres T^2	0.3	0.39
PCAres Q	1.04	1.06

Table 5 shows, in percentages, the Missed Detection Rate (MDR), which is calculated by dividing the number of faulty observations identified as non faulty by the total number of faulty observations, for each method and fault. The proposed methods (*EWMAres*, *PCAres* with statistic Q) outperform the other methods PCA, DPCA, KPCA and CVA, with T_s^2 and Q statistics for most fault cases. CVA with the statistical T_r^2 also gives good MDR results in many faults, but not very far from the *EWMAres* and *PCAres* with Q, except for faults 5, 10 and 19, which are very different. However, in the cases of faults 8, 11, 13 and 18, the detection performance is greatly enhanced by the proposed monitoring techniques, and also in faults 3, 9 and 15, which are very difficult to detect. Nevertheless, as the authors state in paper [42], they have changed the thresholds to perform this MDR comparison for the PCA, DPCA and CVA methods with respect to the ones used to calculate the FAR so as to get better results, while in KPCA and the methods proposed in this paper, the thresholds to implement MDR and FAR are the same.

Table 5: Missed Detection Rate (MDR), in %. TEP

Fault	**PCA**	**PCA**	**DPCA**	**DPCA**	**CVA**	**CVA**	**CVA**	**KPCA**	**KPCA**	**EWMAres**	**PCAres**	**PCAres**
	T^2	Q	T^2	Q	T_s^2	T_r^2	Q	T^2	Q		T^2	Q
1	0.8	0.3	0.6	0.5	0.1	**0**	0.3	**0**	**0**	**0**	0.625	0.125
2	2	1.4	1.9	1.5	1.1	1	2.6	2	2	**0**	2	1.5
3	99.8	99.1	99.1	99	98.1	98.6	98.5	96	92	**89.1**	99.875	93.8
4	95.6	3.8	93.9	**0**	68.8	**0**	97.5	91	**0**	0.25	81.5	6.625
5	77.5	74.6	75.8	74.8	**0**	**0**	**0**	75	73	38	76.125	65.3
6	1.1	**0**	1.3	**0**	**0**	**0**	**0**	1	**0**	**0**	**0**	**0**
7	8.5	**0**	15.9	**0**	38.6	**0**	48.6	**0**	**0**	**0**	0.75	**0**
8	3.4	2.4	2.8	2.5	2.1	1.6	48.6	3	4	**0**	2.75	1.6
9	99.4	98.1	99.5	99.4	98.6	99.3	99.3	96	96	**88.4**	99.75	93.8
10	66.6	65.9	58	66.5	16.6	**9.9**	59.9	57	49	28.9	66.625	43.3
11	79.4	35.6	80.1	19.3	51.5	19.5	66.9	76	19	**7.0**	42.75	27.375
12	2.9	2.5	1	2.4	**0**	**0**	2.1	3	2	**0**	1.5	0.5
13	6	4.5	4.9	4.9	4.7	4	5.5	6	5	**0.25**	6.5	4.0
14	15.8	**0**	6.1	**0**	**0**	**0**	12.2	21	**0**	**0**	**0**	**0**
15	98.8	97.3	96.4	97.6	92.8	90.3	97.9	95	93	**85.5**	99.5	90.9
16	83.4	75.5	78.3	70.8	16.6	**8.4**	42.9	70	48	15.9	64.375	33.75
17	25.9	10.8	24	5.3	10.4	**2.4**	13.8	26	5	2.6	15.25	2.625
18	11.3	10.1	11.1	10	9.4	9.2	10.2	10	10	**0**	11.5	9.0
19	99.6	87.3	99.3	73.5	84.9	**1.9**	92.3	97	51	56.6	98.75	41.8
20	70.1	55	64.4	49	24.8	**8.7**	35.4	59	45	16.1	43.625	20.1
21	73.6	57	64.4	55.8	44	**34.2**	54.7	65	47	51.5	69	62.1

Table 6 contains the detection delay for every method and every fault, showing the great improvement in reducing the detection delay (number of samples) of the *PCAres* method with the statistic Q in comparison to the PCA, DPCA and CVA methods, while eliminating false alarms. The detection delay of the proposed method is in line with the KPCA method, as they are equivalent or very similar in 10 of the 21 faults considered. However, our method detects faults 3,9, and 15, while the KPCA does not. In addition, the *PCAres* with the Q statistic is better in 7 of the remaining faults, while the KPCA is best in the other 4 faults. At the same time, the proposed method with the Q statistic gives the best detection time for faults 4, 5, 6, 7, and 14, where the faults were accurately detected at the occurring time sample. On the other hand, the *EWMAres* method achieves results better than or similar to those of the PCA, DPCA and CVA with the Q statistic.

The comparisons with the four indexes (number of faults detected, FRA, MDR and detection delay) obviously demonstrate the effectiveness and superiority of the proposed methodology, especially for the *PCAres* with the Q statistic.

4.2. Case Study 2: Waste Water Treatment Plant

Once it had been proved that the proposed methodology can detect all the faults with a short detection delay and a reduced number of missed and false alarms, it was tested on a very complex biological system: a Waste Water Treatment Plant (WWTP) (Figure 3) to detect different faults with different fault magnitudes. This benchmark is called BSM2 (Benchmark Simulation Model

Table 6: Detection delays (in samples). TEP

Fault	**PCA**	**PCA**	**DPCA**	**DPCA**	**CVA**	**CVA**	**CVA**	**KPCA**	**EWMAres**	**PCAres**	**PCAres**
	T^2	Q	T^2	Q	T_s^2	T_r^2	Q			T^2	Q
1	7	3	6	5	2	3	2	5	2	5	**1**
2	17	12	16	13	13	15	25	**10**	13	16	12
3									86		**50**
4		3	151	2	462	1		3	1	**0**	**0**
5	16	1	2	2	1	1	**0**	1	1	10	**0**
6	10	1	11	1	1	1	**0**	1	1	**0**	**0**
7	1	1	1	1	1	1	**0**	1	1	**0**	**0**
8	23	20	23	21	20	20	21	25	22	22	**15**
9											**233**
10	96	49	101	50	25	23	44	**20**	42	108	38
11	304	11	195	**7**	292	11	27	23	12	10	13
12	22	8	3	8	2	2	**0**	3	3	7	2
13	49	37	45	40	42	39	43	41	36	47	**35**
14	4	1	6	1	2	1	1	1	2	**0**	**0**
15		740			677			**9**	674		654
16	312	197	199	196	14	**9**	11	**9**	12	33	12
17	29	25	28	24	27	20	23	**19**	20	27	**19**
18	93	84	93	84	83	79	84	**74**	80	94	79
19				82		11			280		**10**
20	87	87	87	84	82	66	72	**59**	71	86	75
21	563	285	522	286	273	511	302	**252**	475	563	510

No. 2) and was developed by the Working Groups of COST Action 682 and 624, and the IWA Task Group ([49, 50]).

These plants are installations whose function is to process waste waters and make them able to be used for other purposes or to discharge them into the environment. This process increases the water quality. The model of the plant is implemented in Simulink (Matlab). To generate faults, some parts of it were changed. The default plant control system was running throughout the simulation.

The variables measured in this model are 16 state variables such as flow, slowly biodegradable substrate, oxygen, nitrates, etc. [50], at each measurement point. There are 20 measurement points, as can be seen in Fig. 3, so 320 measurements can be obtained. However, in a real WWTP, it is not possible to get these measurements instantly, so the method used here is applied over a set of 7 variables, which are easier to obtain in a real plant, and are obtained by combining some of the model variables (see Table 7). The model simulation runs for 609 days and, for this paper, the measurements were recorded every 8 hours. The faults considered are: an O_2 sensor fault, an alkalinity variation and various problems with flow, such as leaks and pipe jams, with different fault sizes. Finally, 16 faulty test data sets were made available to test the proposal (see Table 8).

4.2.1. Experimental methodology

The experimental methodology is similar to that in the previous case study: a normal training data (no faults) set $\mathbf{X} \in \Re^{1827 \times 140}$ is used for the off-line calculations of this proposal, which can be applied using the following steps (as was presented in algorithm 1).

Step 1.- Dynamic feature selection. The first step was dynamic feature se-

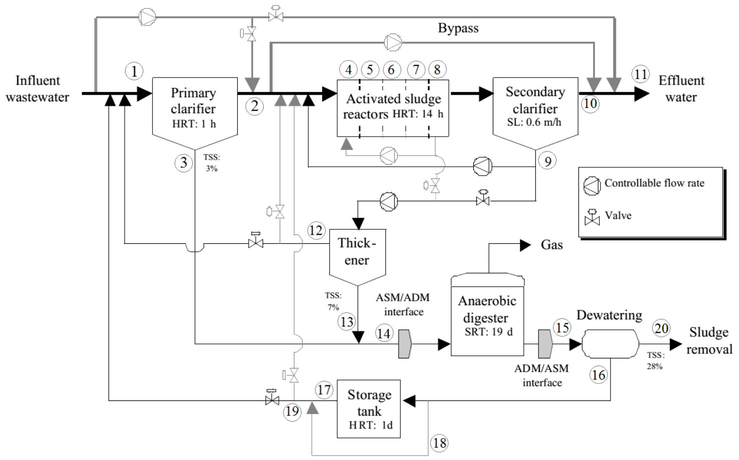

Figure 3: BSM2 plant

Table 7: Variables in BSM2 plant

Used variables	**Variables in the model**
COD (Chemical Oxygen Demand)	S_i, S_s, X_i, X_s
O_2	S_O
Alkalinity	S_{alk}
Nitrogen	S_{NO}, S_{NH}, S_{ND}
Solids suspended	TSS (Total suspended solids)
Flow	Flow rate
Temperature	Temperature

lection, i.e., to discover which delayed variables must be used to model each variable; this is done using the three methods explained in section 2.1 for all the variables and their 15 first delays. For every variable, the 10 most correlated delayed variables are used for the next step: modelling. The dynamic feature selection for the ARIMA model is done using the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), as has been explained.

Step 2.- Building the time-series model. The second step with this workbench was to develop the time-series models. The faultless training data (1827 observations) was used to model each variable with its corresponding dynamic features as inputs. First, these data were normalized according to the time-series model to be calculated, as explained in the TE case study. Then, different types of models were developed for every system variable, in this case these models were:

Table 8: Faults in BSM2 plant

Fault #	**Description**
1 to 5	O_2 sensor failure, from -50 % to +70 %
6 to 8	Influent alkalinity change, from -50 % to +40 %
9 to 11	Reactor 1 alkalinity change, from -30 % to +20 %
12 and 13	Change in lower flow of primary Dec, from -50 % to -30%
14	Change in storage tank output flow, - 50 %
15 and 16	Qr and Qw flows change, from -50% to -25% of Q_r

- ARIMA models, with structure ARIMA(p,q,i), where the order of each part AR (p), MA (q), and Integral part (i) is specified. In the experiments carried out, the variations of these parameters are: $p = [0, ..., 15]$, $q = [0, ..., 15]$ and $i = [0, 1, 2]$, with the additional condition that: $p + q \leq 20$.

- Neural network models. The NN used is always a Perceptron Multilayer network (MLP) with three layers, where the input layer has 10 inputs corresponding to the inputs selected in the dynamic feature selection of step 1 and the output layer has only one neuron. The number of neurons in the hidden layer is a parameter to be modified in each of the calculated models, and this parameter varies as follows: $neurons\ in\ hidden\ layer = [5, ..., 6 \times (2 \times number\ of\ inputs)]$.

The model with the lowest rMSE value using test data in normal operation conditions is selected. The selected time-series model for some of the variables and its structure can be seen in Table 9, where ANN are the neural networks with correlation coefficients variable selection and PLS ANN are the neural networks with PLS based variable selection, and the final number of neurons in the hidden layer are shown in the Table. In this case study, as the wastewater treatment plant is a very non-linear process, the best time-series model for most of the variables are non-linear, specially a neural network model; all the variables modelled with an ARIMA model are the ones presented in Table 9.

Step 3.- Computing the control chart limits. Now a residual is calculated as the difference between the prediction for every variable using the respective model and its measured value. Based on these residuals, the control limits for both control charts have to be calculated.

1. *Residuals EWMA control chart (EWMAres).* These thresholds are adjusted as in the previous benchmark using the faultless training data. The thresholds were adjusted experimentally to have a minimum number of consecutive observations outside the thresholds with different values of λ. The values chosen were $Obs = 5$ and $\lambda = 0.9$, which achieve the lowest possible delay with no false alarms and all the faults detected.
2. *Fault detection based on PCA of residuals (PCAres).* A PCA model using the residuals was built. This PCA is performed with a number of principal components that achieve 85% of the variance (as before, this value

Table 9: Selected time-series models for some variables in the WWTP

Variable	Model	Variable	Model
1	PLS ANN: Hidden Layer: 26	28	ARIMA: (2,2,1)
2	ANN: Hidden Layer: 21	35	ARIMA: (4,2,1)
3	PLS ANN: Hidden Layer: 26	42	ARIMA: (4,2,1)
4	ANN: Hidden Layer: 26	56	ARIMA:(5,1,1)
5	PLS ANN: Hidden Layer: 26	77	ARIMA: (2,1,1)
6	ANN: Hidden Layer: 26	84	ARIMA (4,2,1)
7	PLS ANN: Hidden Layer: 21	98	ARIMA: (2,2,1)
8	ANN: Hidden Layer: 26	105	ARIMA: (2,2,1)
9	PLS ANN: Hidden Layer: 21	126	ARIMA: (2,2,1)
10	PLS ANN: Hidden Layer: 26	133	ARIMA: (2,2,1)
11	PLS ANN: Hidden Layer: 21	140	ARIMA: (2,2,1)

maximizes information while minimizing the dimensionality). The thresholds of the T^2 and Q statistics are calculated theoretically and they are then tuned experimentally for an imposed significance level (ISL) of 5% in normal operation conditions. Also, as in the example above, in order to avoid false alarms when the system is working on-line, it is necessary to overpass these limits in $Obs = 5$ consecutive samples to detect the fault. This number of consecutive observations avoids the appearance of false alarms and also gives the smallest delay and the highest fault detection rate.

4.2.2. Detection results and discussion

In this sub-section, step 4 of the methodology is carried out, i.e., the on-line detection of anomalies. In this case study, this proposal will be compared with PCA with 70% of variance explained by its selected principal components and its T^2 and Q thresholds adjusted theoretically with $ISL = 5\%$. This study is in terms of the performance indexes defined above: number of detected faults (bigger is better), false alarms rate, missed alarms rate (lower is better) for each fault and detection delay (also lower is better).

In this example for all the methods, PCA and the proposed methods in this paper, the thresholds are adjusted to obtain an $ISL = 5\%$ in normal operation conditions and the threshold has to exceed some consecutive number of samples so there will be zero false alarms in working conditions. The false alarm rate for the test data is in Table 10, where it is possible to see that the best result is for the ($EWMAres$) method with (0.32%) and also for the ($PCAres$) with the Q statistics.

Table 11 shows the Missed Detection Rate (MDR), in percentages. Here, once more the $EWMAres$ method gives the best results, as it is able to detect more faulty observations than other methods, with big differences. Faults 6 to 16 are very easy to detect and all the methods works well, except the PCA with the T^2 statistic, which is the worst. For faults 1 to 5, which are very difficult

Table 10: False Alarms Rate (FAR). BSM2

Method and Statistic	**est data**
PCA T^2	2.6
PCA Q	4.8
EWMAres	0.32
PCAres T^2	3.2
PCAres Q	2

to detect, the *PCAres* method with the Q statistical is the second best method after the *EWMAres* one.

Table 11: Missed Detection Rate (MDR), in %. BSM2

Fault	**PCA** T^2	**PCA** Q	**EWMAres**	**PCAres** T^2	**PCAres** Q
1	96.41	10.53	**0.12**	69.83	6.93
2	96.66	42.11	**3.98**	87.42	27.86
3	97.18	87.55	**27.47**	96.92	82.16
4	97.82	71.50	**25.03**	96.79	86.01
5	91.78	0.26	**0.13**	44.16	0.26
6	27.47	**0.26**	**0.26**	**0.26**	**0.26**
7	86.65	**0.26**	**0.26**	**0.26**	**0.26**
8	26.32	**0.26**	**0.26**	**0.26**	**0.26**
9	2.05	**0.39**	**0.39**	**0.39**	**0.39**
10	0.39	0.39	**0.26**	0.39	**0.26**
11	27.47	**0.26**	**0.26**	0.39	**0.26**
12	**0.26**	**0.26**	**0.26**	**0.26**	**0.26**
13	**0.26**	**0.26**	**0.26**	**0.26**	**0.26**
14	**0.26**	**0.26**	**0.26**	**0.26**	**0.26**
15	**0.26**	**0.26**	**0.26**	**0.26**	**0.26**
16	14.89	**0.26**	**0.26**	**0.26**	**0.26**

The detection delay obtained by each method is shown in Table 12. It is clear that the best method in this case is the *PCAres* with Q statistics, because it gives the smallest delay for all the faults except for Fault 2. Also, all the methods are able to detect all the faults, except the *PCAres* with the statistical T^2, which cannot detect faults 3 and 4, two faults very difficult to detect. In general, the detection delay is bigger for the T^2 statistic than the Q statistic. So, with the best results in 15 out of 16 faults, it is clear that the *PCAres* method (with Q statistic) is the best option in the fault detecting task. The *EWMAres* method is also very good for this example, as it has the lowest false alarm rate, the lowest missed detection rate and a good detection time.

Table 12: Detection delays (in samples). BSM2

Fault	**PCA**	**PCA**	**EWMAres**	**PCAres**	**PCAres**
	T^2	Q		T^2	Q
1	258	772	23	253	**12**
2	258	725	**184**	253	193
3	258	211	223	nd	**194**
4	259	278	223	nd	**208**
5	259	7	3	**2**	**2**
6	726	7	3	**2**	**2**
7	541	7	3	**2**	**2**
8	775	7	3	**2**	**2**
9	506	8	4	**3**	**3**
10	8	8	3	3	**2**
11	775	7	3	3	**2**
12	7	7	3	**2**	**2**
13	7	7	3	**2**	**2**
14	7	7	3	**2**	**2**
15	7	7	3	**2**	**2**
16	692	7	3	**2**	**2**

nd=not detected

5. CONCLUSIONS

This paper presents a dynamic and non-linear fault detection and diagnosis methodology. The proposed methodology explicitly accounts for the dynamic relations in the process data through dynamic feature selection, and for the non-linear relationship between the variables of the process through a time-series model available to capture both the non-linear, if it exists, and the dynamic correlation in the process data. After that the residuals, which are the difference between the process measurements and the output of the model, are monitored using conventional SPC charts, such as the EWMA control chart if the residuals are evaluated individually, or a Multivariate Statistical Process Control (MSPC) chart when the residuals are processed all together; in this case they are evaluated with the PCA algorithm and the classical Hotelling's and SPE statistics.

This methodology was applied to two plants: the Tennessee Eastman Process and a Waste Water Treatment Plant, and compared with other fault detection methods. The method based on individual residuals with the univariate control chart (EWMA) gives a good performance, but it is not better than the other methods of the comparison, specially for the TE plant, but it does give a very good result with the WWTP. However, the method based on the PCA of residuals is the best in the comparative, giving the highest number of faults detected and the lowest detection time (delay). The fault alarms rate (FAR) is also better than the other existing methods for the test data, while the missed detection rate (MDR) outperforms the other methods and is equivalent to the best one (CVA with the T_r statistic). So, comparisons with the four indexes (number of faults detected, FRA, MDR and detection delay) obviously demonstrate the effectiveness and superiority of the proposed methodology, specially for the *PCAres* with the Q statistic. Finally, these examples demonstrate that the proposed methodology efficiently detects faults for non-linear and dynamic processes.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Economy, Industry and Competitiveness and the European Regional Development Fund (FEDER) through the Projects: DPI2015-67341-C2-2-R, TIN2013-47210-P, TIN2016-81113-R and to the Junta de Andalucía through the project: P12-TIC-2958

6. References

[1] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, Industrial & Engineering Chemistry Research 52 (2013) 3543–3562.

[2] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, A review of process fault detection and diagnosis. Part I: Quantitative model-based methods, Computers & Chemical Engineering 27 (2003) 291–311.

[3] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies, Computers & Chemical Engineering 27 (2003) 313–326.

[4] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, K. Yin, A review of process fault detection and diagnosis. Part III: Process history based methods, Computers & Chemical Engineering 27 (2003) 327–346.

[5] A. Bakdi, A. Kouadri, A new adaptive PCA based thresholding scheme for fault detection in complex systems, Chemometrics and Intelligent Laboratory Systems 162 (Supplement C) (2017) 83 – 93.

[6] S. Qin, Survey on data-driven industrial process monitoring and diagnosis, Annual Reviews in Control 36 (2012) 220–234.

[7] W. Ku, R. Storer, C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems 30 (1995) 179–196.

[8] T. Villegas, M. J. Fuente, G. I. Sainz-Palmero, Fault diagnosis in a wastewater treatment plant using dynamic independent component analysis, in: 18th Mediterranean Conference on Control & Automation (MED), IEEE, 2010, pp. 874–879.

[9] J. Huang, X. Yan, Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and bayesian inference, Chemometrics and Intelligent Laboratory Systems 148 (2015) 115–127.

[10] A. Simoglou, E. Martin, A. Morris, Statisical performance monitoring of dynamic multivariate processes using state space modeling, Computers & Chemical Engineering, 26(6) (2002) 909–920.

[11] H. Kaneko, K. Funatsu, A new process variable and dynamics selection method based on a genetic algorithm-based wavelength selection method, AIChE Journal 58(6) (2012) 1829–1840.

[12] M. Kramer, Autoassociative neural networks, Computers & Chemical Engineering 16 (1992) 313–328.

[13] W. Yan, P. Guo, L. Gong, Z. Li, Nonlinear and robust statistical process monitoring based on variant autoencoders, Chemometrics and Intelligent Laboratory Systems 158 (2016) 31–40.

[14] B. Schokopf, A. Smola, K. R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.

[15] J. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghemb, I. Lee, Nonlinear process monitoring using kernel principal component analysis, Chemical Engineering Science 59 (2004) 223–234.

[16] J. Hunter, The exponentially weighted moving average, Journal of Quality Technology 18(4) (1986) 203–210.

[17] D. Garcia-Alvarez, M. J. Fuente, G. I. Sainz-Palmero, Design of residuals in a model-based fault detection and isolation system using statistical process control techniques, in: ETFA2011, 2011, pp. 1–7.

[18] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer New York, 2013.

[19] S. Wold, M. Sjostrom, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometrics and Intelligent Laboratory Systems 58 (2001) 109–130.

[20] R. H. Shumway, D. S. Stoffer, Time Series Analysis and its Applications, Springer-Verlag Berlin, 2006.

[21] J. D. Gooijer, R. J. Hyndman, 25 years of time series forecasting, International Journal of Forecasting 22 (2006) 443–473.

[22] G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day San Francisco, 1976.

[23] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer-Verlag New York, 2013.

[24] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2(5) (1989) 359–366.

[25] G. I. Sainz-Palmero, M. J. Fuente, P. Vega, Recurrent neuro-fuzzy modelling of a wastewater treatment plant, European Journal of Control 10 (2004) 83–95.

[26] M. Khashei, M. Bijari, An artificial neural network (p, d,q) model for timeseries forecasting, Expert Systems with Applications 37 (2010) 479–489.

[27] E. Heidari, M. A. Sobati, S. Movahedirad, Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (MLP-ANN), Chemometrics and Intelligent Laboratory Systems 55 (2016) 73–85.

[28] V. Vapnik, The Nature of Statistical Learning Theory, Springer New york, 1995.

[29] K. W. Lau, Q. H. Wu, Local prediction of non-linear time series using support vector regression, Pattern Recognition 41 (2008) 1539–1547.

[30] E. E. Elattar, J. Goulermas, Q. Wu, Electric load forecasting based on locally weighted support vector regression, IEEE Transactions on Systems, Man and Cybernetics. 40(4) (2010) 438–447.

[31] S. Saludes-Rodil, M. J. Fuente, Fault tolerance in the framework of support vector machines based model predictive control, Engineering Applications of Artificial Intelligence 23 (2010) 1127–1139.

[32] A. Ferrer, Statistical control of measures and processes, Comprehensive Chemometrics - Chemical and Biochemical Data Analysis 1 (2009) 97–126.

[33] D. Garcia-Alvarez, G. I. Sainz-Palmero, M. J. Fuente, P. Vega, Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant, in: IFAC Proceedings Volumes, Vol. 42, 2009, pp. 952–957, 7th IFAC Symposium on Advanced Control of Chemical Processes.

[34] W. Shewhart, Application of statistical methods to manufacturing problems, Journal of the Franklin Institute 226 (2) (1938) 163 – 186.

[35] R. Woodward, P. Goldsmith, Cumulative sum techniques, Mathematical and Statistical Techniques for Industry, Oliver & Boyd Edinburgh, 1964.

[36] E. S. Page, Continuous inspection schemes, Biometrika 41 (1-2) (1954) 100–115.

[37] I. Jolliffe, Principal Component Analysis, Springer Verlag New York, 2002.

[38] T. Kourti, J. MacGregor, Multivariate SPC methods for process and product monitoring, Journal of Quality Technology 28 (1996) 409–428.

[39] C. F. Alcala, S. J. Qin, Reconstruction-based contribution for process monitoring, Automatica 45 (2009) 1593–1600.

[40] J. E. Jackson, A User's Guide to Principal Components, Wiley New York, 1991.

[41] M. Frigge, D. C. Hoaglin, B. Iglewicz, Some Implementations of the Boxplot, The American Statistician 43(1) (1989) 50–54.

[42] E. L. Russell, L. H. Chiang, R. D. Braatz, Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems 51(1) (2000) 81–93.

[43] K. Detroja, R. Gudi, S. Patwardhan, Plant-wide detection and diagnosis using correspondence analysis, Control Engineering Practice 15 (2007) 1468–1483.

[44] C. Tong, T. Lan, X. Shi, Fault detection and diagnosis of dynamic processes using weighted dynamic decentralized PCA approach, Chemometrics and Intelligent Laboratory Systems 161 (Supplement C) (2017) 34–42.

[45] J. J. Downs, E. F. Vogel, A plant-wide industrial process control problem, Computers & Chemical Engineering 17 (1993) 245–255.

[46] L. Chiang, E. Russell, R. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag London, 2000.

[47] B. Jiang, D. Huang, X. Zhu, F. Yang, R. D. Braatz, Canonical variate analysis-based contributions for fault identification, Journal of Process Control 26 (2015) 17–25.

[48] Y. Zhang, Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM), Industrial & Engineering Chemistry Research 47 (18) (2008) 6961–6971.

[49] J. Alex, L. Benedetti, J. Copp, K. Gernaey, U. Jeppsson, I. Nopens, M. Pons, C. Rosen, J. Steyer, P. Vanrolleghem, Benchmark Simulation Model no. 2 (BSM2), Tech. rep., IWA Taskgroup on Benchmarking of Control Systems for WWTPs. Department of Industrial Electrical Engineering and Automation, Lund University, Lund, Sweden, (2008).

[50] I. Nopens, L. Benedetti, U. Jeppsson, M.-N. Pons, J. Alex, J. B. Copp, K. V. Gernaey, C. Rosen, J.-P. Steyer, P. A. Vanrolleghem, Benchmark simulation model no 2: finalisation of plant layout and default control strategy, Water Science & Technology 62 (2010) 1967–1974.