**Control Engineering Practice, Volume 42, September 2015, Pages 74–88**
**doi:10.1016/j.conengprac.2015.04.012**

# Statistical Process Monitoring of a Multiphase Flow Facility

C. Ruiz-Cárcel[a], Y. Cao[a], D. Mba[a,b], L.Lao[a], R. T. Samuel[a]

*a School of Engineering, Cranfield University, Building 52 School of Engineering, MK430AL (UK)*
*b School of Engineering, London South Bank University, 103 Borough Road, London, SE1 0AA (UK)*
*Corresponding atuthor contact: c.ruizcarcel@cranfield.ac.uk*

**Abstract**

Industrial needs are evolving fast towards more flexible manufacture schemes. As a consequence, it is often required to adapt the plant production to the demand, which can be volatile depending on the application. This is why it is important to develop tools that can monitor the condition of the process working under varying operational conditions. Canonical Variate Analysis (CVA) is a multivariate data driven methodology which has been demonstrated to be superior to other methods, particularly under dynamically changing operational conditions. These comparative studies normally use computer simulated data in benchmark case studies such as the Tennessee Eastman Process Plant [1].
The aim of this work is to provide a benchmark case to demonstrate the ability of different monitoring techniques to detect and diagnose artificially seeded faults in an industrial scale multiphase flow experimental rig. The changing operational conditions, the size and complexity of the test rig make this case study an ideal candidate for a benchmark case that provides a test bed for the evaluation of novel multivariate process monitoring techniques performance using real experimental data. In this paper, the capabilities of CVA to detect and diagnose faults in a real system working under changing operating conditions are assessed and compared with other methodologies. The results obtained demonstrate that CVA can be effectively applied for the detection and diagnosis of faults in real complex systems, and reinforce the idea that the performance of CVA is superior to other algorithms.

*Keywords:* fault detection; diagnosis; multivariate; canonical; experimental; process

## Nomenclature

| | | | |
|---|---|---|---|
| $\sum_{ff}$ | Future covariance matrix | $T_k^2$ | Hotelling $T^2$ statistic at time $k$ |
| $\sum_{fp}$ | Cross covariance matrix | $T^2_{UCL}$ | Upper control limit for the $T^2$ statistic |
| $\sum_{pp}$ | Past covariance matrix | $\boldsymbol{u_i}$ | $i^{th}$ column in matrix $\boldsymbol{U}$ |
| $c_{i,j}$ | Contribution of the $j^{th}$ variable to the $i^{th}$ canonical variate | $\boldsymbol{U}$ | Rotation matrix |
| $C_j$ | Total contribution of the $j^{th}$ variable | $\boldsymbol{v_i}$ | $i^{th}$ column in matrix $\boldsymbol{V}$ |
| $\boldsymbol{D}$ | Matrix of canonical correlations | $\boldsymbol{V}$ | Rotation matrix |
| $e$ | Euler constant | $\boldsymbol{V_r}$ | Reduced rotation matrix |
| $\boldsymbol{E}$ | Residual matrix | $x$ | Random variable |
| $f$ | Number of future samples considered | $\boldsymbol{y}$ | Observation vector |
| $g$ | Argument of Kernel function $K$ | $\boldsymbol{Y_f}$ | Future matrix |
| $\boldsymbol{H}$ | Hankel matrix | $\boldsymbol{y_{f,k}}$ | Future vector at time $k$ |
| $h$ | Bandwidth for the estimation of $p(x)$ | $\hat{\boldsymbol{y}}_{\boldsymbol{f,k}}$ | Normalized future vector at time $k$ |
| $\boldsymbol{I}$ | Identity matrix | $\bar{\boldsymbol{y}}_{\boldsymbol{f,k}}$ | Sample mean of $y_{f,k}$ |
| $\boldsymbol{J}$ | Transformation matrix | $\hat{\boldsymbol{y}}_{\boldsymbol{j}}$ | $j^{th}$ elemnt of normalised vector $\hat{y}$ |
| $k$ | Arbitrary time point | $\boldsymbol{y_k}$ | Observation vector at time $k$ |
| $K$ | Kernel function | $\boldsymbol{Y_p}$ | Past matrix |
| $\boldsymbol{L}$ | Transformation matrix | $\boldsymbol{y_{p,k}}$ | Past vector at time $k$ |
| $M$ | Number of columns in past and future matrices | $\hat{\boldsymbol{y}}_{\boldsymbol{p,k}}$ | Normalized past vector at time $k$ |
| $m$ | Number of sensors | $\bar{\boldsymbol{y}}_{\boldsymbol{p,k}}$ | Sample mean of $y_{p,k}$ |
| $N$ | Number of observations in the data set | $z_i$ | $j^{th}$ elemnt of canonical variate vector $\boldsymbol{z}$ |
| $p$ | Number of past samples considered | $\boldsymbol{z_k}$ | Canonical variates vector at time $k$ |
| $p(x)$ | Probability density function of random variable $x$ | $\boldsymbol{Z}$ | Matrix of canonical variates |
| $Q_k$ | $Q$ statistic (squared prediction error) at time $k$ | $\alpha$ | Significance level |
| $Q_{UCL}$ | Upper control limit for the $Q$ statistic | $\gamma_i$ | Canonical correlations in matrix $\boldsymbol{D}$ |
| $r$ | Reduced dimension selected | $\boldsymbol{\varepsilon_k}$ | Residual vector at time $k$ |
| $s$ | Arbitrary value | | |

## 1. Introduction

Data driven methods are widely used for fault detection and diagnosis applications in real industrial systems. In particular, multivariate monitoring techniques such as the Principal Component Analysis (PCA) [2] or the Partial Least Squares (PLS) can take into account the correlation between the different variables measured in the process, and they show advantages against the traditional univariate methods [3]. However, there is a need for more effective techniques that can deal with problems like changing operational conditions or nonlinear systems [3-5] . Ku et al. [6] proposed the use of lagged variables to take into account time correlation to extend PCA to dynamic system monitoring (DPCA). Similarly, a dynamic version of PLS was proposed by Komulainen [7]. Despite of their success, DPCA and DPLS have been reported not to be as efficient as other state-space based methodologies such as Canonical Variate Analysis (CVA) when applied to systems working under variable loading conditions, principally due to the representation of the system dynamics [8; 9]. Other popular approaches typically used for condition monitoring are neural networks, machine learning or fault-tree analysis [10; 11].

The literature gives examples of successful application of different improved algorithms based on well-known techniques such as PCA [12-16] , PLS [17; 18], and CVA [8; 17] for fault detection and diagnosis using computer simulated data. Particularly, the Tennessee Eastman process simulator [1] has been widely used for the assessment and comparison of the performance of different algorithms due to its realistic level of complexity and the challenges attached to the fact that it is a highly non-linear system. The popularity of this particular benchmark case is demonstrated by the high number of researchers who have used it in the last years to prove the validity of a large variety of approaches based on PCA [19-22], PLS [23; 24], CVA [25; 26], Fisher Discriminant Analysis (FDA) [27], Latent Subspace Projection (LSP) [28], Sensitive Fault Analysis (SFA) [29] or Bayesian Networks [30; 31] among others.

Canonical Variate Analysis is a data driven methodology which maximizes the correlation between two sets of variables [10]. In the recent years, different researchers have applied CVA for fault detection and diagnosis using computer simulated data [4; 8; 9; 32], data acquired in small test rigs [33] or particular parts of a system [34]. Despite this success, there are no examples of application of CVA in real and complex systems working under varying operational conditions. This underlines the authors' motivation for generating a common benchmark case study which can be used for the assessment of novel process monitoring techniques using real data available in the public domain for the assessment of new monitoring techniques.

In the light of these challenges, this paper provides a significant contribution in the following aspects:

- Generate a benchmark case study for the development of new monitoring techniques and its implementation in real industrial facilities [35].
- Study the capabilities of CVA as a method capable of detecting faults in real systems working under variable operating conditions
- Compare the performance of CVA against other traditional monitoring algorithms such as PCA, PLS and their dynamic approaches
- Evaluate the benefits of using Kernel Density Estimators (KDE) for the estimation of health indicators' threshold in systems with important influence of non-linearity

This paper presents the results obtained after the application of CVA to experimental data acquired in the three-phase flow facility at Cranfield University, which is comparable to a small industrial multiphase flow separation process. In this case study, different faults were introduced deliberately in the system, simulating typical faults that can be expected in real plants such as blockages, incorrect system operation or non-conventional operating conditions. The data sets were acquired under changing operational conditions, modifying the flow rate set points to ensure that the fault detection was undertaken not only in the steady-state regime. The changing operational conditions and the non-linear nature of the multiphase flow process, together with the size and complexity of the test rig make this case study an ideal candidate as a benchmark case that can be used for the evaluation of novel multivariate process monitoring techniques based on real experimental data. The objective of the analysis of the acquired data in this investigation is to assess the performance of CVA as a method capable of detecting faults in systems working under variable operating conditions. Additional tests were carried out using data sets acquired under steady operational conditions, and the performance of CVA for the detection and diagnosis of faults was compared with other methodologies such as PCA, DPCA, PLS and DPLS using the data sets acquired under single or multiple steady operational conditions.

The paper is organized as follows: The methodology used is described in section 2, including the CVA application procedure (2.1) and the experimental set up for the benchmark case study (2.2). Section 3 shows and

discusses the results obtained from the analysis of the data sets. The parameter tuning and algorithm training is discussed in 3.1. A detailed discussion of the results obtained in terms of fault detection and diagnosis using CVA for two of the data sets is presented in 3.2. The results obtained for all the data sets produced using different process monitoring algorithms are analysed and compared in 3.3. Finally the study is concluded in section 4.

## 2. Methodology

### *2.1 CVA for fault detection in industrial processes*

The application procedure of CVA is similar to other multivariate algorithms for condition monitoring: it requires an initial data set acquired under normal operation conditions to train a model, which is used to calculate the transformation matrices and the thresholds for the health indicators. In an application, it is possible to monitor the process by simply converting the high-dimensional acquired data into the indicators using the transformation matrices. The value of these indicators compared with the threshold calculated during the training stage will determine the presence or absence of detected faults.

The objective of CVA is to find the linear combinations that maximize the correlation between two sets of variables. In order to take into account time correlations, the observation vector $\boldsymbol{y}$ is expanded at each time point $k$ by considering $p$ past and $f$ future measurements (each one containing $m$ variables), generating the past and future observation vectors $\boldsymbol{y}_{p,k}$ and $\boldsymbol{y}_{f,k}$ respectively:

$$\boldsymbol{y}_{p,k} = \begin{bmatrix} \boldsymbol{y}_{k-1} \\ \boldsymbol{y}_{k-2} \\ \vdots \\ \boldsymbol{y}_{k-p} \end{bmatrix} \in \Re^{mp} \qquad \boldsymbol{y}_{f,k} = \begin{bmatrix} \boldsymbol{y}_{k} \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+f-1} \end{bmatrix} \in \Re^{mf} \tag{1,2}$$

The data are normalized to 0 mean in each different variable to avoid domination of those variables with higher absolute values measured:

$$\hat{\boldsymbol{y}}_{p,k} = \boldsymbol{y}_{p,k} - \bar{\boldsymbol{y}}_{p,k} \tag{3}$$

$$\hat{\boldsymbol{y}}_{f,k} = \boldsymbol{y}_{f,k} - \bar{\boldsymbol{y}}_{f,k} \tag{4}$$

where $\bar{\boldsymbol{y}}_{p,k}$ and $\bar{\boldsymbol{y}}_{f,k}$ represent the sample means of $\boldsymbol{y}_{p,k}$ and $\boldsymbol{y}_{f,k}$ respectively. The optimal number of past and future lags ($p$ and $f$) considered in the analysis can be determined using the autocorrelation function of the summed squares of all measurements [3]. All the past and future vectors are arranged together in different columns generating the past and future matrices $\boldsymbol{Y}_p$ and $\boldsymbol{Y}_f$:

$$\boldsymbol{Y}_p = \left[\hat{\boldsymbol{y}}_{p,p+1}, \hat{\boldsymbol{y}}_{p,p+2}, \ldots \hat{\boldsymbol{y}}_{p,p+M}\right] \in \Re^{mp \times M} \tag{5}$$

$$\boldsymbol{Y}_f = \left[\hat{\boldsymbol{y}}_{f,p+1}, \hat{\boldsymbol{y}}_{f,p+2}, \ldots \hat{\boldsymbol{y}}_{f,p+M}\right] \in \Re^{mf \times M} \tag{6}$$

where $M=N-f-p+1$ for a data set of $N$ observations.

The covariance and cross-covariance matrices of past and future matrices can be estimated as follows:

$$\boldsymbol{\Sigma}_{pp} = \frac{1}{M-1} \boldsymbol{Y}_p \boldsymbol{Y}_p^{\boldsymbol{T}} \in \Re^{mp \times mp} \tag{7}$$

$$\boldsymbol{\Sigma}_{ff} = \frac{1}{M-1} \boldsymbol{Y}_f \boldsymbol{Y}_f^{\boldsymbol{T}} \in \Re^{mf \times mf} \tag{8}$$

$$\boldsymbol{\Sigma}_{fp} = \frac{1}{M-1} \boldsymbol{Y}_f \boldsymbol{Y}_p^{\boldsymbol{T}} \in \Re^{mf \times mp} \tag{9}$$

The canonical variates can be derived through the Singular Value Decomposition (SVD) of the scaled Hankel matrix:

$$\boldsymbol{H} = \boldsymbol{\Sigma}_{ff}^{-1/2} \boldsymbol{\Sigma}_{fp} \boldsymbol{\Sigma}_{pp}^{-1/2} = \boldsymbol{U} \boldsymbol{D} \boldsymbol{V}^{\boldsymbol{T}} \in \Re^{mf \times mp} \tag{10}$$

where:

$$\boldsymbol{U} = \left[\boldsymbol{u}_1, \boldsymbol{u}_2 \cdots \boldsymbol{u}_{mf}\right] \in \Re^{mf \times mf} \qquad \boldsymbol{V} = \left[\boldsymbol{v}_1, \boldsymbol{v}_2 \cdots \boldsymbol{v}_{mp}\right] \in \Re^{mp \times mp}$$

$$\boldsymbol{D} = \begin{bmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_x \end{bmatrix} \in \Re^{mf \times mp}$$

$\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices of singular vectors and $\boldsymbol{D}$ is a diagonal matrix, indicating that $\boldsymbol{U}$ and $\boldsymbol{V}$ are only pairwise correlated. The correlation between $\boldsymbol{U}$ and $\boldsymbol{V}$ is indicated by the diagonal elements $\gamma_i$ in $\boldsymbol{D}$. Reordering the elements in $\boldsymbol{D}$ ($\gamma_1 > \gamma_2 > \ldots > \gamma_{mp}$) and the corresponding singular vectors in $\boldsymbol{U}$ and $\boldsymbol{V}$, it is possible to select the first $r$ columns of $\boldsymbol{V}$, which correspond to $r$ largest singular values in $\boldsymbol{D}$, generating a new dimensionally reduced matrix $\boldsymbol{V_r} \in \mathbb{R}^{mf \times r}$.

The transformation matrices $\boldsymbol{J}$ and $\boldsymbol{L}$, which convert the $mp$-dimensional past measurements to the $r$-dimensional canonical variates and residuals, can be calculated as:

$$\boldsymbol{J} = \boldsymbol{V_r^T \Sigma_{pp}^{-1/2}} \in \Re^{r \times mp} \tag{11}$$

$$\boldsymbol{L} = \boldsymbol{(I - V_r V_r^T)\Sigma_{pp}^{-1/2}} \in \Re^{mp \times mp} \tag{12}$$

The canonical variates $\boldsymbol{z_k}$ and the residuals $\boldsymbol{\varepsilon_k}$ are calculated at each time point $k$ by projecting the $k^{\text{th}}$ normalised past measurement in (3). Left multiplying $\boldsymbol{J}$ and $\boldsymbol{L}$ to the past matrix, $\boldsymbol{Y}_p$ in (5) leads to:

$$\boldsymbol{Z} = \boldsymbol{J Y_p} \in \Re^{r \times M} \tag{13}$$

$$\boldsymbol{E} = \boldsymbol{L Y_p} \in \Re^{mp \times M} \tag{14}$$

where the columns of $\boldsymbol{Z}$ and $\boldsymbol{E}$ are $\boldsymbol{z_k}$ and $\boldsymbol{\varepsilon_k}$ respectively, which are the canonical variates and the residuals at different sampling points. The statistical indicators that provide information about the health of the system can be obtained converting the available past observations into a lower dimensional data. The statistical indicators more frequently used are the Hotelling $T^2$ statistic and $Q$ statistic.

$$T_k^2 = \sum_{i=1}^{r} z_{i,k}^2 \tag{15}$$

$$Q_k = \sum_{i=1}^{mp} \varepsilon_{i,k}^2 \tag{16}$$

where $z_{i,k}$ and $\varepsilon_{i,k}$ are the $(i,k)^{\text{th}}$ elements of $\boldsymbol{Z}$ and $\boldsymbol{E}$ matrices, respectively.

The upper control limits (UCL) for $T^2$ and $Q$ can be calculated for a given significance level $\alpha$ such that $P(T^2 < T^2_{UCL}(\alpha)) = \alpha$ and $P(Q < Q_{UCL}(\alpha)) = \alpha$ respectively. Normally these control limits are calculated by assuming that the measurements follow a Gaussian distribution. However, this is not the case for nonlinear systems, which makes this assumption invalid. Odiowei and Cao [3] developed a methodology to solve this issue by estimating the actual probability density function of the statistical indicators using Kernel Density Estimations (KDE). The probability of a random variable $x$ (with a probability density function $p(x)$) to be smaller than a certain value $s$ is defined as:

$$P(x < s) = \int_{-\infty}^{s} p(x)dx \tag{17}$$

where $p(x)$ can be calculated through the kernel function $K$:

$$p(x) = \frac{1}{Mh} \sum_{k=1}^{M} K\left( \frac{x - x_k}{h} \right) \tag{18}$$

$$K(g) = e^{-g^2/2} \Big/ \sqrt{2\pi} \tag{19}$$

where $h$ is the selected bandwidth (see [3]) and $x_k$ is the $k^{\text{th}}$ sample of $x$. By replacing $x_k$ with $T_k^2$ and $Q_k$ obtained from (15) and (16) respectively, it is possible to estimate the probability density function of $T^2$ and $Q$. The respective control limits for these statistics correspond to $s$ in (17), obtained by solving $P(x<s) = \alpha$.

The $T^2$ metric represents the total variation of the canonical variates indicating changes in the retained space. $Q$ is also known as the squared prediction error and represents the sum of the squared variation error in the

residual space. Both indicators are complementary; some faults can cause a variability increment in the retained space whilst others will be manifested as an increment in the residual space variability. For this case study the event of fault detection will be considered every time any of the indicators exceed the respective UCL. This makes the monitoring performance insensitive to *r*, the number of canonical variates retained [3].

Following this procedure, the values of the indicators calculated during the training period are representative for the normal system, i.e. in absence of any faults. The thresholds are calculated for a selected level of significance looking at the probability density functions of these indicators. Any deviation in the behaviour of the system caused by faults will cause an increment in the value of these indicators, reaching the threshold and producing an alarm. This means that the threshold has a statistic significance, as the indicator value is below the for the selected significance level, which represent normal conditions without any fault. If the training data is representative, the algorithm should ideally be able to accommodate the value of the indicator below the threshold if no faults are present. Certainly, if data of a new normal operation condition, which is not included in the train data, is obtained, then both the detection model and the threshold need to be updated. Such an update can be done off-line or on-line. The later one is the so called adaptive approach, however, it is beyond the scope of this work.

In the event of fault detection it is crucial to locate the source of the fault in order to understand the source of the problem and undertake the best maintenance action. Chiang et al. [10] suggested the use of contribution plots to estimate how much each one of the variables contributed to the final value of the statistical indicator. In CVA, the contribution *c* of each variable $\hat{y}_j$ to the final value of each canonical variate $z_i$ can be calculated as:

$$c_{i,j} = \frac{z_i}{\gamma_i} J_{i,j} \cdot \hat{y}_j \tag{20}$$

where $\gamma_i$ is the singular value corresponding to the loading vector $\boldsymbol{J}_i$. The total contribution *C* of the $j^{\text{th}}$ process variable $y_j$ can be obtained as:

$$C_j = \sum_{i=1}^{r} c_{i,j} \tag{21}$$

This technique has been successfully applied by other researchers to fault identification [36-39]. Once a fault has been detected, it is possible to prioritize the variables responsible for fault identification based on their individual contributions, and the plant engineers can use this information together with their plant knowledge to determine the origin of the fault.

*2.2 Description of the Three-phase Flow Facility*

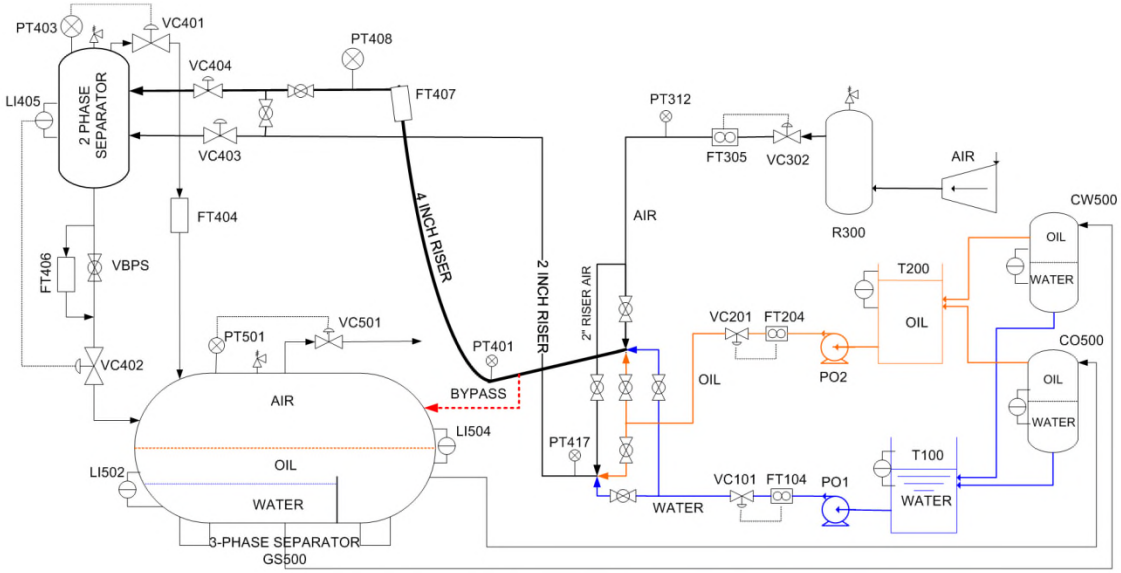



Fig. 1: Sketch of the three-phase flow facility

The Three-phase Flow Facility at Cranfield University is designed to provide a controlled and measured flow rate of water, oil and air to a pressurized system. Fig. 1 shows a simplified sketch of the facility. The test area consists of pipelines with different bore sizes and geometries, and a gas and liquid two-phase separator (0.5 m diameter and 1.2 m high) at the top of a 10.5 meters high platform. It can be supplied with single phase of air, water and oil, or a mixture of those fluids, at required rates. Finally the fluid mixtures are separated in an 11 $m^3$ horizontal three-phase separator at ground level (GS500). The air is returned to the atmosphere and the emulsions of oil and water are separated in their respective coalescers (CW500 and CO500, both having a capacity of 1.5 $m^3$ approximately) before returning to their respective storage tanks (T200 and T100). The capacity of both storage tanks is approximately 12.5 $m^3$.

Air is supplied by a union of two compressors which are capable to deliver a flow rate up to 0.4 $m^3/s$ at 0.7 MPa. The compressed air is received in an 8 $m^3$ vessel (R300) to dampen pressure fluctuations, then, it is filtered of droplets and particles and cooled before passing through the air flow meter FT305 and the pneumatic valve VC302 that controls the air flow. The water and oil are stored in tanks T100 and T200 and then supplied independently through multistage Grundfos CR90-5 pumps (PO1 and PO2), each one of which can provide up to 0.0278 $m^3/s$ at 1 MPa with their rotational speed controlled by frequency variable inverters. The water flow rate is measured by FT104 and oil flow rate by FT204. The water and oil flow rates are controlled by pneumatic valves VC101 and VC201, respectively.

After the mixture, the fluids can flow either through a 4” diameter flow loop which has a 55 m long and 2° downward inclined pipeline leading to a 10.5 m high catenary riser or via a 2” flow loop which is a 40 m long horizontal pipeline, connecting to a 10.5 m long vertical riser. Both flowlines are connected to the two phase separator but can be isolated from each other by manual valve manifolds in both ends of the flowlines. During all the experiments adopted in this benchmark case the 4” line was used exclusively, except for the study of a particular fault (Fault Case 6), where the 2” line was also involved. In the rig there is an alternative 4” line which can carry the flow directly from the mixing point to the 3 phase separator, bypassing the riser and the top separator. This line is labelled as “BYPASS” and marked in dashed red in Fig. 1.

There are sensors measuring pressure at the air supply line before the mixing point (PT312), and along the 4” line at the bottom of the riser (PT401), at the top of the riser (PT408), inside of the top separator (PT403) and inside of the three-phase separator (PT501). The flow rate in the 4” line at the top of the riser can be measured by FT407, which also provides measurements of the density and the temperature. FT406 provides the measurements of the mass flow rate, density and temperature at the bottom outlet of the top separator. The liquid level inside of the top separator is measured by LI405 and controlled by VC402. The pressure of the 3-phase separator is controlled by VC501, while the levels between the different phases are measured by LI502 and LI504 respectively. In addition, the pressure in the 2” line at the bottom of the riser is measured by PT417. The valves and sensors which control the level between phases in the coalescers have not been shown in Fig. 1 for

simplicity. The whole system is managed using Delta V [40] a Fieldbus based supervisory, control and data acquisition (SCADA) system supplied by Emerson Process Management. Time stamped data of different variables can be retrieved, processed and visualized. The data can also be saved for post-processing.

In this study, all the data was captured at a sampling rate of 1 Hz. The variables used include 24 different process variables (see Table 1) and two process inputs (air and water flow rate set point). Variable number 24 (pressure in mixture zone 2" line, PT417) was only included in the analysis of Faulty Case 6, while the first 23 variables were used in all the faulty cases studied as well as for algorithm training. Only air and water were used in all the experiments and the three phase separator was always pressurized to 0.1 Mpa.

Table 1: List of process variables used in this study

| Variable nr | Location | Measured Magnitude | Unit |
|---|---|---|---|
| 1 | PT312 | Air delivery pressure | MPa |
| 2 | PT401 | Pressure in the bottom of the riser | MPa |
| 3 | PT408 | Pressure in top of the riser | MPa |
| 4 | PT403 | Pressure in top separator | MPa |
| 5 | PT501 | Pressure in 3 phase separator | MPa |
| 6 | PT408 | Diff. pressure (PT401-PT408) | MPa |
| 7 | PT403 | Differential pressure over VC404 | MPa |
| 8 | FT305 | Flow rate input air | $Sm^3/s$ |
| 9 | FT104 | Flow rate input water | kg/s |
| 10 | FT407 | Flow rate top riser | kg/s |
| 11 | LI405 | Level top separator | m |
| 12 | FT406 | Flow rate top separator output | kg/s |
| 13 | FT407 | Density top riser | $kg/m^3$ |
| 14 | FT406 | Density top separator output | $kg/m^3$ |
| 15 | FT104 | Density water input | $kg/m^3$ |
| 16 | FT407 | Temperature top riser | °C |
| 17 | FT406 | Temperature top separator output | °C |
| 18 | FT104 | Temperature water input | °C |
| 19 | LI504 | Level gas-liquid 3 phase separator | % |
| 20 | VC501 | Position of valve VC501 | % |
| 21 | VC302 | Position of valve VC302 | % |
| 22 | VC101 | Position of valve VC101 | % |
| 23 | PO1 | Water pump current | A |
| 24 | PT417 | Pressure in mixture zone 2" line | MPa |

### *2.2 Description of the data sets*

#### ***2.2.1 Normal operation***

In order to obtain data representing normal operating conditions, three data sets (T1, T2 and T3) were acquired from the system. The set points of air and water flow rates were deliberately varied during the tests to obtain data from the process working under variable operating conditions. To ensure that the conditions during normal operation were representative, 20 different combinations of air and water flow rates (see Table 2) were tested for each one of the three training data sets. In each one of them, the flow conditions were changed, though not identically, in order to obtain a good variety of large, small, long and slow process changes happening in different directions (increment or decrement). The objective of this variety in the operational conditions is to ensure that the dynamics of the system are captured in all circumstances. These set points were selected in order to cover the available range of flow rate that the facility can supply through VC101 and VC302, avoiding excessively low flow rates to ensure that the conditions introduced will not generate slugging in the riser.

Table 2 represents the typical air and water flow rate set points selected during the experiments, while Fig. 2 shows the air and water flow rates measured during each one of the training data sets.

Table 2: Typical setpoint values for air and water flow rates

| | | | | | |
|---|---|---|---|---|---|
| Air flowrate ($Sm^3/s$) | 0.0208 | 0.0278 | 0.0347 | 0.0417 | |
| Water flowrate (kg/s) | 0.5 | 1 | 2 | 3.5 | 6 |

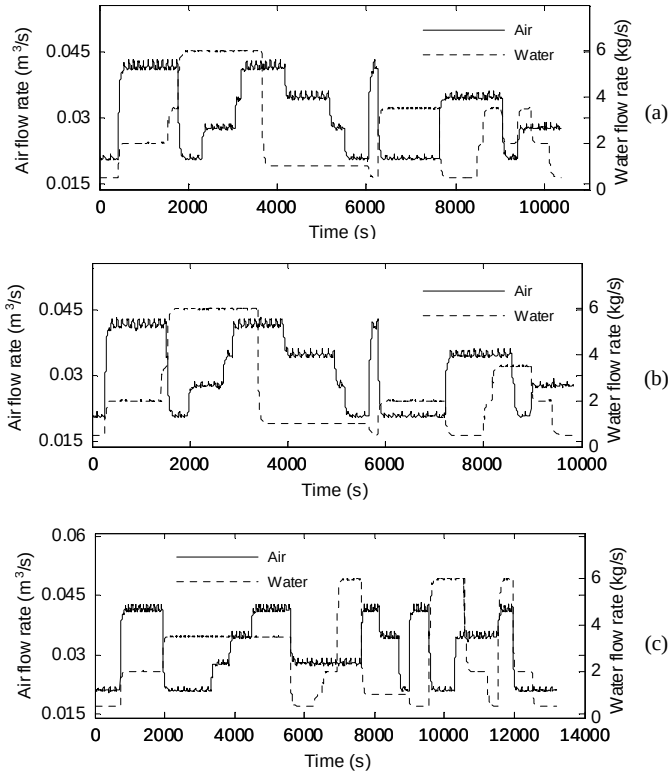



Fig. 2: Operational conditions for training data sets T1 (a), T2 (b) and T3 (c)

### *2.2.2 Operation with seeded faults*

In addition to the training data sets, different sets of data were acquired from the system working in faulty conditions after seeding six different faults into the system. These faults were used in order to simulate typical malfunctions that could be experienced in a real system such as blockages in the pipelines, wrong system operation or abnormal operating conditions. The faults were introduced after a certain time of normal operation in order to investigate changes in the health indicators. These faults were introduced gradually when possible, in order to observe how the severity of the fault affects the indicator. After reaching a certain level of severity the fault condition was removed, returning the system to normal conditions. Details about the fault start time and fault end time for the different data sets can be found in the description of the corresponding fault scenario. During these tests the flow rate conditions were changed in a similar way as was undertaken for normal operation. There are additional data sets acquired under constant flow rate conditions which allow the observation of the fault effects on the different measured variables without disturbances created by changes in the air and water flow rates. Table 3 summarizes the different faults tested. Each one of the subsections explains how a fault was introduced together with details about the duration of the data sets, flow rate conditions and the instants when faults were introduced or removed, which are summarized in tables.

Table 3: Summary of faults introduced

| Fault Case | Description | Type | Nr. of data sets (changing conditions) | Nr. of data sets (steady conditions) |
|---|---|---|---|---|
| 1 | Air line blockage | Incipient | 1 | 2 |
| 2 | Water line blockage | Incipient | 1 | 2 |
| 3 | Top separator input blockage | Incipient | 1 | 2 |
| 4 | Open direct bypass | Incipient | 1 | 2 |
| 5 | Slugging conditions | Intermittent | 2 | 0 |
| 6 | Pressurization of the 2” line | Abrupt | 2 | 0 |

*Fault Case 1: Air line blockage*

Just before the air, water and oil supply lines are joined at the mixing point, there are individual manual valves for each one of the lines. The fault introduced in this fault case was to gradually close the manual valve of the air line simulating an air blockage that develops over time. The valve angle was measured using a protractor

attached to the valve in order to record the fault evolution, but this measurement was not included in the measurement list for monitoring. At the beginning of the experiments the valve was fully open (normal conditions) and then, the valve was gradually closed to introduce the fault condition simulating the blockage. For this fault case three data sets were acquired, one set under changing operational conditions (1.1) and another two sets had constant air and water flow rates (1.2 and 1.3, see Table 4).

Table 4: Operational conditions for data sets in Fault Case 1

| Data set | Operating conditions | Duration (s) | Water flow rate (kg/s) | Air flow rate ($m^3/s$) | Fault start (s) | Fault end (s) |
|---|---|---|---|---|---|---|
| 1.1 | Changing | 5811 | Varying | Varying | 1566 | 5181 |
| 1.2 | Steady-State | 4467 | 2 | 0.0417 | 657 | 3777 |
| 1.3 | Steady-State | 4321 | 3.5 | 0.0208 | 691 | 3691 |

*Fault Case 2: Water line blockage*

This fault is exactly the same as the fault introduced in Fault Case 1, with the difference that in this fault case the water line valve was closed gradually to simulate the blockage instead of the air line valve. Despite the similarity of the fault to the air line blockage, the consequences are expected to be different. The main reason is that the physical properties of air and water (especially density and viscosity) are considerably dissimilar. For this fault case, three data sets were acquired, one under changing operational conditions (2.1) and two more sets with constant air and water flow rates (2.2 and 2.3, see Table 5).

Table 5: Operational conditions for data sets in Fault Case 2

| Data set | Operating conditions | Duration (s) | Water flow rate (kg/s) | Air flow rate ($m^3/s$) | Fault start (s) | Fault end (s) |
|---|---|---|---|---|---|---|
| 2.1 | Changing | 9192 | Varying | Varying | 2244 | 6616 |
| 2.2 | Steady-State | 3496 | 2 | 0.0278 | 476 | 2656 |
| 2.3 | Steady-State | 3421 | 3.5 | 0.0417 | 331 | 2467 |

*Fault case 3: Top separator input blockage*

In this fault case, the type of fault introduced is similar to the air and water line blockage. The main difference is that the valve involved in this fault case is VC404 (top separator input). The valve can be operated remotely from the control room and the angular position can be accurately observed. This allows a precise observation of how the fault severity affects the condition monitoring performance. Again this measurement was not included in the data analyses for fault detection and diagnosis. The fault was introduced in most cases with a fast evolution in a first instance due to the inherent behaviour of ball valves: in this type of valves the change in the pressure drop is almost insignificant for valve positions near the fully open position, but it is much more sensitive to changes near the fully closed position. For this fault case three data sets were acquired, one of them under changing operational conditions (3.1) and two more with constant air and water flow rates (3.2 and 3.3, see Table 6).

Table 6: Operational conditions for data sets in Fault Case 3

| Data set | Operating conditions | Duration (s) | Water flow rate (kg/s) | Air flow rate ($m^3/s$) | Fault start (s) | Fault end (s) |
|---|---|---|---|---|---|---|
| 3.1 | Changing | 9090 | Varying | Varying | 1136 | 8352 |
| 3.2 | Steady-State | 6272 | 2 | 0.0278 | 333 | 5871 |
| 3.3 | Steady-State | 10764 | 3.5 | 0.0208 | 596 | 9566 |

*Fault Case 4: Open direct bypass*

Under fault mode 4 a leakage at the bottom of the riser was simulated, causing a lack of flow in the test area. The 4” bypass line can direct the multiphase flow after the mixing point to the 3-phase separator, bypassing the riser. There are valves at the beginning and at the end of this alternative line to isolate it from the rest of the system, and in normal conditions they are always closed. These valves were opened gradually during the tests to direct the flow at the bottom of the riser through the bypass line, simulating a leakage at this point. For this fault case three data sets were acquired, one set under changing operational (4.1) conditions and other two with constant air and water flow rates (4.2 and 4.3, see Table 7).

Table 7: Operational conditions for data sets in Fault Case 4

| Data set | Operating conditions | Duration (s) | Water flow rate (kg/s) | Air flow rate ($m^3/s$) | Fault start (s) | Fault end (s) |
|---|---|---|---|---|---|---|
| 4.1 | Changing | 7208 | Varying | Varying | 953 | 6294 |
| 4.2 | Steady-State | 4451 | 2 | 0.0417 | 851 | 3851 |
| 4.3 | Steady-State | 3661 | 3.5 | 0.0208 | 241 | 3241 |

*Fault Case 5: Slugging conditions*

Slugging [41] is a transient phenomenon that can occur in risers with multiphase flow when the velocities of the gas and the liquid are relatively low. The liquid tends to accumulate in the base of the riser blocking the gas flow. Due to this blockage, the pressure builds upon the blockage until it is sufficient to flush the liquid (and the gas) out of the riser. After this surge, the liquid remaining in the riser falls down, creating a new blockage and starting the cycle again. This phenomenon is typical for offshore oil production systems, where multiphase hydrocarbon fluids travel from an oil filed along a long pipeline on the sea bed to a riser connecting to a separation process on an offshore oil rig. It can produce large amplitude fluctuations in the pressure and flow rates, which can affect and damage the equipment [42]. The fault was introduced by reducing the air and water flow rates to regimes where slugging is produced. During the test, the flow rate was varying at different points from normal to slugging conditions. For this fault case, 2 data sets were acquired; both of them under changing operational conditions (see Table 8). For data set 5.1 slugging conditions were introduced and removed twice during the experiment (referred in Table 8 as Fault 1 and Fault 2), while for data set 5.2 slugging was introduced at three different stages of the experiment (referred in Table 8 as Fault 1, Fault 2 and Fault 3).

Table 8: Operational conditions for data sets in Fault Case 5

| Data set | Operating conditions | Duration (s) | Water flow rate (kg/s) | Air flow rate ($m^3/s$) | Fault 1 start (s) | Fault 1 end (s) | Fault 2 start (s) | Fault 2 end (s) | Fault 3 start (s) | Fault 3 end (s) |
|---|---|---|---|---|---|---|---|---|---|---|
| 5.1 | Changing | 2541 | Varying | Varying | 686 | 1172 | 1772 | 2253 | - | - |
| 5.2 | Changing | 10608 | Varying | Varying | 1633 | 2955 | 7031 | 7553 | 8057 | 10608 |

*Fault Case 6: Pressurization of the 2" line*

In the fault cases presented previously the 4" line was used to carry the flow to the top of the riser. In these conditions, the 2" line is totally isolated from the rest of the system and thus it should not be pressurized. In the top of the riser, just before the input of the top separator there is a bridge with a valve that connects the 4" and the 2" line. The fault was introduced as a result of opening this bridge valve but keeping both sides of the 2" isolated from the rest of the installation. The objective of this fault condition is to simulate an unusual operation of the system that in a real process can cause degradation in the performance or in the output product quality or even safety issues. This fault should not affect the flow conditions in the 4" line or other parts of the rig, but it will pressurize the 2" line. In this particular fault case, an additional variable was measured and included in the analysis. This variable is the pressure measured in the bottom of the riser in the 2" line by PT417. For this fault case, 2 data sets were acquired (6.1 and 6.2), both of them under changing operational conditions (see Table 9).

Table 9: Operational conditions for data sets in Fault Case 6

| Data set | Operating conditions | Duration (s) | Water flow rate (kg/s) | Air flow rate ($m^3/s$) | Fault start (s) | Fault end (s) |
|---|---|---|---|---|---|---|
| 6.1 | Changing | 2800 | Varying | Varying | 1723 | 2800 |
| 6.2 | Changing | 4830 | Varying | Varying | 1037 | 4830 |

## 3. Results and discussion

The results provided by the application of CVA to the training and monitoring data sets introduced above are presented in this section. As an example, data sets 3.1 (a gradual degradation fault) and 5.2 (an abrupt fault) are analysed in detail, including plots of the operational conditions during the test (measured flow rates and fault evolution) as well as the results obtained from the CVA application ($T^2$ and $Q$ indicators and contribution plots at the moment of fault detection). The plots representing CVA fault detection results for each analysis contain the $T^2$ and $Q$ indicators plotted as solid black lines, while the threshold is plotted as a grey dashed line. The fault starting and ending points in each data set analysed are represented by a vertical dashed black line. The results obtained for all data sets are summarized in 3.3

### *3.1 Training data sets and selection of tuning parameters*

The first step for the application of CVA is to select the number of past and future lags considered (*p* and *f*) and the number of canonical variates retained (*r*). This will allow the construction of the past and future matrices and the calculation of the transformation matrices and UCL for the thresholds. The optimal number of past and future lags considered in the analysis can be determined by computing the autocorrelation function of the summed squares of all measurements [3]. This function measures the cross correlation between a signal and a delayed version of itself at different lags. In this way, it is possible to determine for how long the correlation of the signal with past lags is significant, and thus only the lags which are relevant are selected. Fig. 3 shows an example of autocorrelation function for the training data set T1 against a confidence bound of ±5 %.

For this study *p* and *f* were set to 15 according to the results obtained from the analysis of the autocorrelation function of the three training data sets.

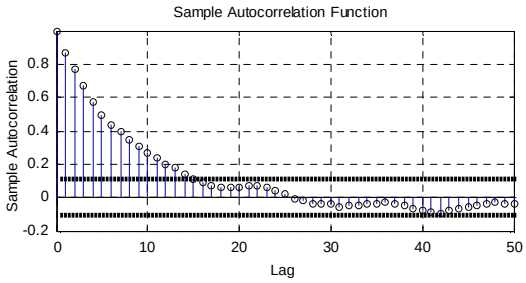



Fig. 3: Autocorrelation function of the summed squares of all measurements for data set T1

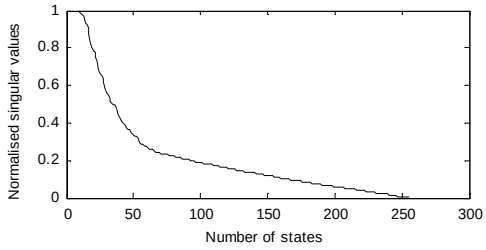



Fig. 4: Normalized singular values for T1

Different methodologies have been suggested to determine the optimal number of canonical variates retained *r*. Some of the most popular are those based on considering the dominant singular values in the matrix $\boldsymbol{D}$ [43] and methodologies based on the Akaike Information Criterion (AIC) [10]. Fig. 4 shows the normalized singular values obtained from (10). In this particular case, the singular values decrease slowly and setting the number of retained canonical variates based on the dominant singular values will derive in an unrealistic model [3]. However, the value of *r* does affect the false alarm rate. In this work, three normal condition data sets were jointly analysed to determine the best value of *r*, as described below, where a confidence bound of 99 % was considered for the calculation of the UCL.

Traditionally, process monitoring considers only a single operating mode. In this case, only one normal operation data set is required for off-line training. For monitoring under varying operating conditions, it is important to ensure that the monitoring model trained covers all normal operational conditions. However, in practice, a single data set may not be able to cover all possible normal conditions. In this work, in order to obtain a richer and reliable training set that covers all the spectrum of operational conditions, the data sets acquired under normal operational conditions were combined. The combination was done by calculating the past and future matrices individually for each data set according to (5) and (6) and then the matrices obtained were joined, producing a new matrix that combines the data of several sets. The original length of the data sets was 10372 s for T1, 9825 s for T2 and 13200 s for T3. A total of 23 variables (variables 1 to 23 in Table 1) were included in the algorithm for training in the study of Faulty Cases 1 to 5, while for the study of Faulty Case 6 variable 24 was added to the analysis. Additional details about these data sets were provided in section 2.2.1.

These three data sets were mixed in pairs generating 3 different combined sets. The objective of analysing these three different data sets is to see which training data set combination produces a lower false alarm rate when the remaining data set is used during the monitoring period. In order to select the optimal number of canonical variates retained r, CVA was performed for each one of the three combined training data sets using a range of values for this parameter with the objective of selecting the value that minimises the false alarm rate. For low values of r the number of false alarms is high because the retained space is not able to represent accurately the states of the system and consequently the number of the $T^2$ threshold violations increases. On the other hand if the number of canonical variates selected is too high it results in the model overfitting the data [10], increasing again the false alarm rate. The data set combination which produced lower false alarm rates was T2 and T3 (see Fig. 5). After several analyses testing different values for r for each combination of data sets finally *r*=25 was adopted which minimizes the false alarm rate in normal conditions according to the results represented in Fig. 5.

Table 10 summarises the results obtained from this analysis for the parameter *r*=25:

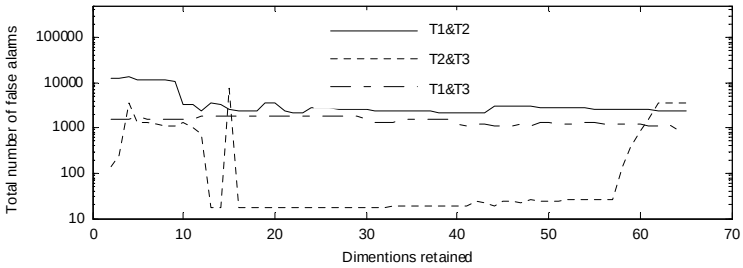



Fig. 5: Analysis of the influence of the number of states retained

Table 10: Summary training data sets analysis

| Training | Monitoring | $T^2$ Threshold | $Q$ threshold | False alarm rate $T^2$ | False alarm rate $Q$ |
|---|---|---|---|---|---|
| T1&T2 | T3 | 3036.81 | 734.94 | 0.04 % | 29.12 % |
| T1&T3 | T2 | 1620.63 | 1158.71 | 0.01 % | 7.96 % |
| T2&T3 | T1 | 1753.97 | 6940.73 | 0 % | 0.21 % |

These results show that the combination of T1 and T2 for training is not able to capture accurately all the systems variations that occurred during T3, obtaining a high number of points over the limit especially for the $Q$ indicator. The number of points over the threshold using the combination T1 and T3 for the training stage and T2 for monitoring produced slightly better results, but it is obvious that the combination which produced the best results capturing the system dynamics and consequently reducing the number of threshold violations was T2 and T3. For this reason, the combined T2 and T3 data set was finally adopted to derive the transformation matrices and upper control limits of $T^2$ and $Q$ for online monitoring.

*3.2 Analysis of results obtained for data sets 3.1 and 5.2*

In data set 3.1, the simulated blockage was introduced by manipulating the valve VC404 (top separator input) which is operated pneumatically. Fig. 6 shows the evolution of the fault (position of valve VC404) and operating conditions (air and water flow rate measurements) during the test.

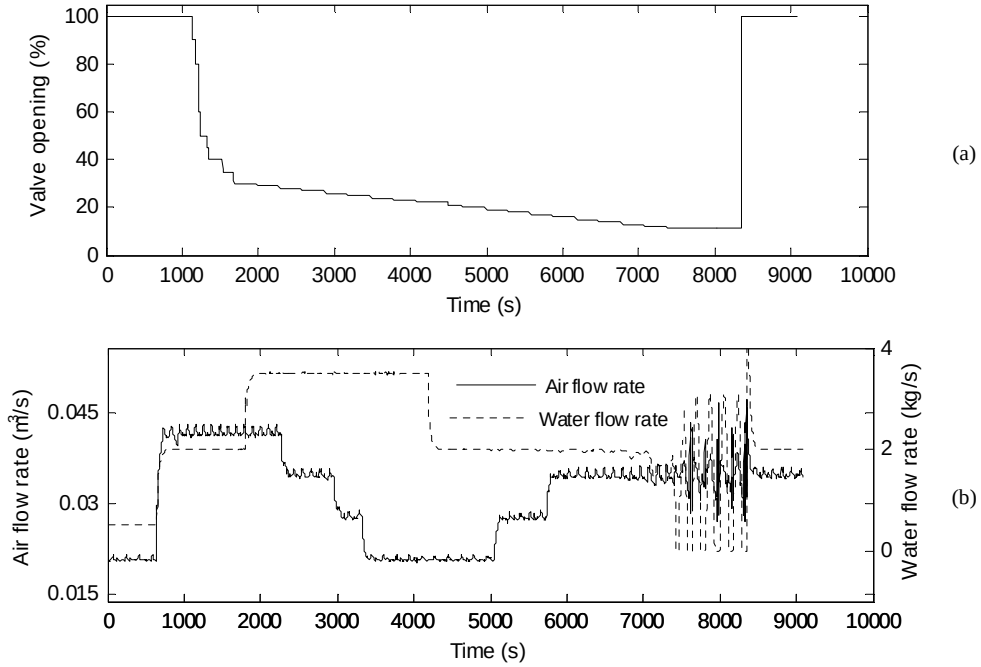



Fig. 6: Fault evolution (a) flow rate set points (b) and measured flow rate s (c) for data set 3.1

The results obtained in terms of fault detection and diagnosis for data set 3.1 are represented in Fig. 7.

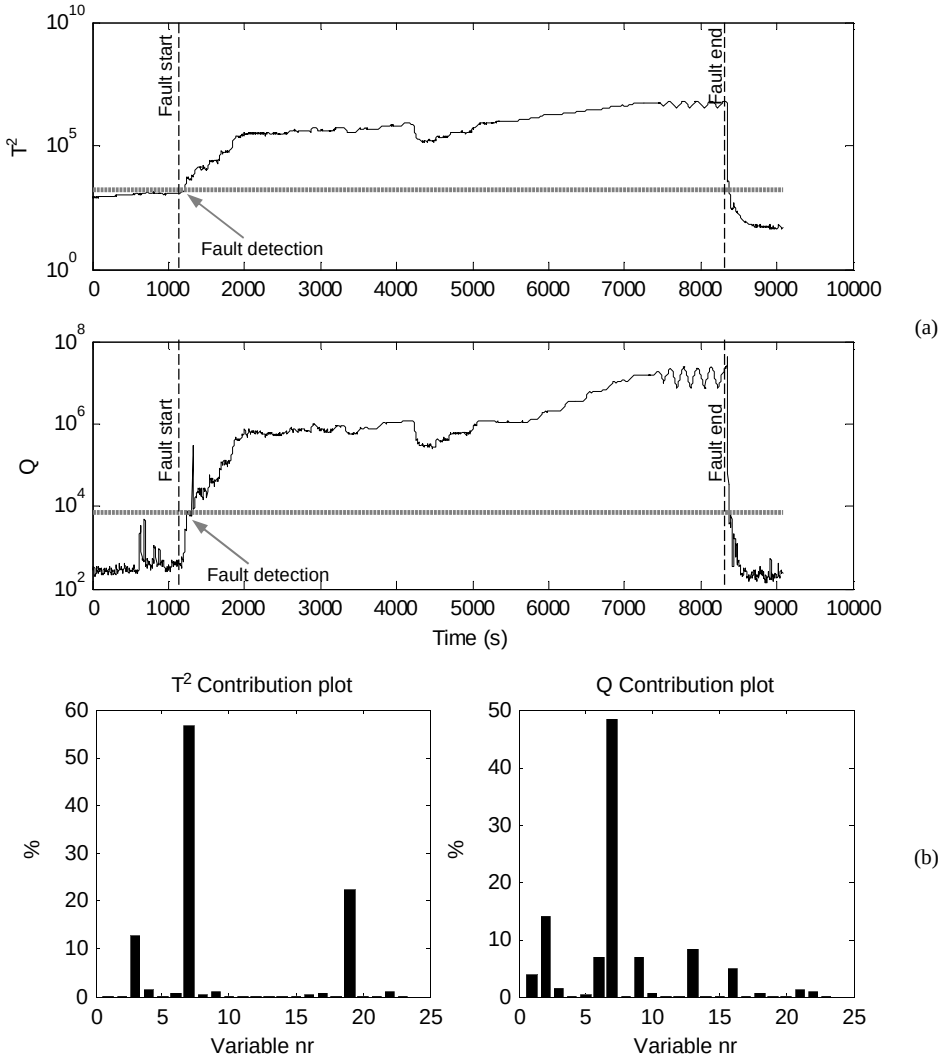



Fig. 7: Results from data set 3.1: $T^2$ and $Q$ indicators (a) and contribution plots at sample 1230 (b)

The first fault detection happens in sample 1230 for the $T^2$ statistical indicator and 1324 for the $Q$ indicator after 4 short false alarms. Those time points correspond to a valve opening of 60 % and 45 % respectively. Both indicators fall back to below the UCL values when the fault is removed by opening the valve completely. The contribution plots at the fault detection time (sample 1230) are represented in Fig. 7(b). In this case the variable contributing most to the final value of the $T^2$ statistical indicator is the differential pressure over the valve VC404 (PT408-PT403), pointing precisely to an excessive pressure loss in this valve. It is important to mention that the position of the valve VC404 was not included in the measurement list. The contribution of the level in the 3-phase separator (LI504) is also significant. For the $Q$ indicator, the most significant variables are the differential pressure over the valve VC404 (PT408-PT403), and the pressure measured at the bottom of the riser (PT401).

The fault seeded in data set 5.2 was introduced by reducing the air and water flow rates to regimes where slugging is produced. During the tests, the flow rate was varying at different points from normal to slugging conditions. Fig. 8 (a) represents the operational conditions selected for data set 5.2, where the flow rate combinations which are expected to produce slugging are shaded in grey. The pressure measured in the bottom of the riser (PT401) is represented in Fig. 8 (b), where it can be seen the pressure oscillations caused by slugging.

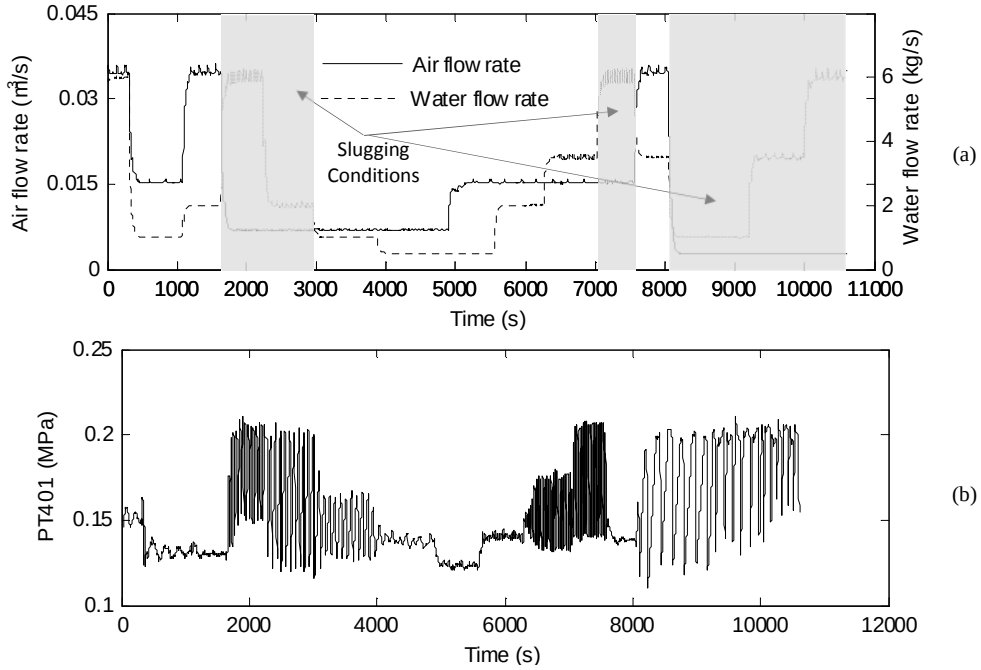



Fig. 8: Measured flow rates (a) and bottom riser pressure PT401 (b) for data set 5.2

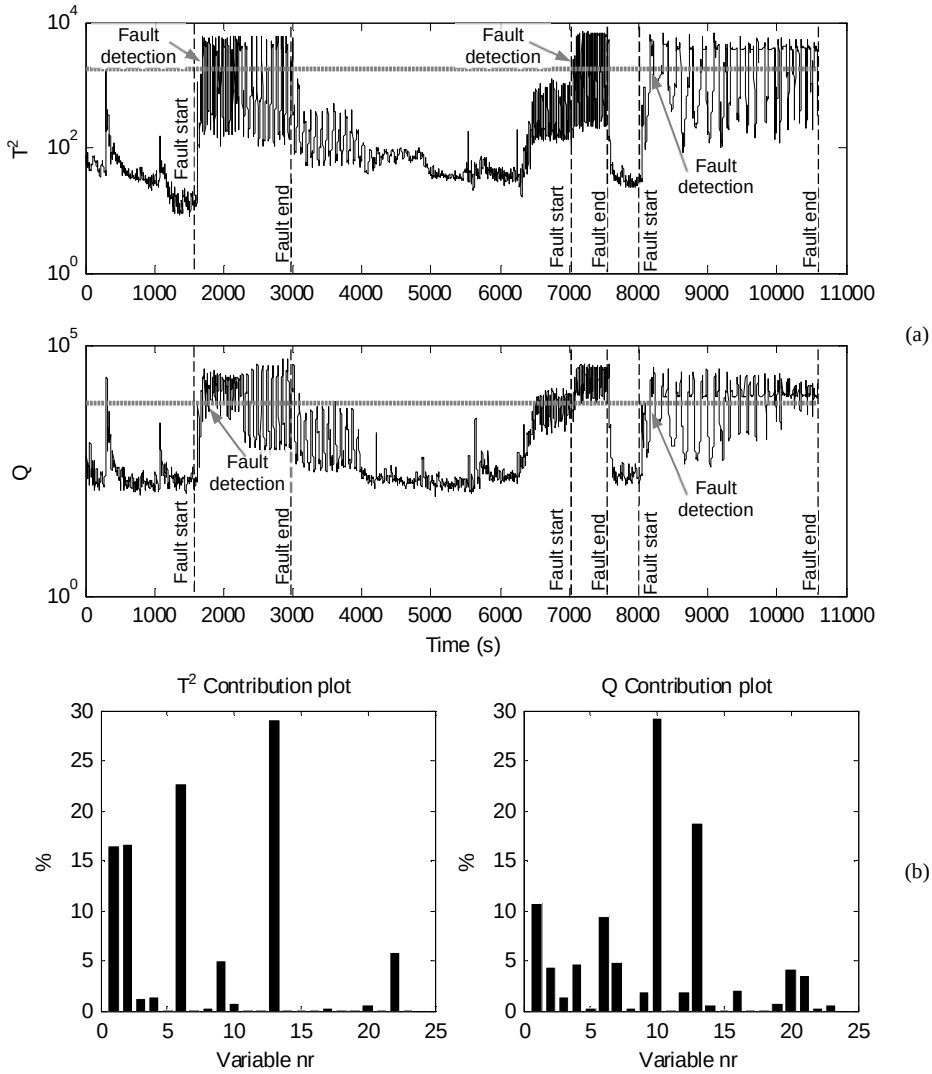



Fig. 9: Results from data set 5.2: $T^2$ and $Q$ indicators (a) and contribution plots (b)

Fault detection and diagnosis results for data set 5.2 are represented in Fig. 9. The fault detections happened at samples 1716, 7075 and 8171 for the $T^2$ statistical indicator and 1643 and 8103 for the $Q$ indicator after one short false alarm. The detection of the second period of slugging is complicated in the $Q$ indicator because it was already over the threshold limit before the introduction of slugging, where the flow rate conditions were in a transition zone between slugging and normal. In this particular case, both indicators fluctuate with time under slugging conditions due to the repetitive nature of the phenomenon, but the threshold is only reached under slugging conditions and also in transition zones in case of the $Q$ indicator. In this case the variables contributing more to the final value of the $T^2$ statistical indicator at the first detection point (sample 1643) are the density of the fluid at the top of the riser, differential pressure between the top and the bottom of the riser, and the bottom riser pressure (PT401). These are the key variables normally used to characterise slugging. The most significant variables for the value of the $Q$ indicator at this time point are the flow rate at the top of the riser and the density measured at the same point.

*3.3 Results summary*

This subsection summarizes the results obtained in terms of fault detection for all data sets analysed. The results obtained by the application of CVA with KDE are compared with the results obtained by the application of PCA, DPCA, PLS and DPLS with and without KDE for the calculation of the indicators' thresholds. For the PCA, DPCA, PLS and DPLS analyses, the number of retained components was set to 5 in order to cover 85 % of the covariance, and the number of lags considered in DPCA and DPLS was set to 5. For the PLS and DPLS analyses, the air and water set points were selected to construct the predictor matrix, while the measurements taken from the system were used in the predicted matrix. For the calculation of the UCL a confidence bound of 99 % was used in all cases.

Table 11 and Table 12 show the results obtained in terms of detection rate. This is calculated as the percentage of observations that produced indicator values over the threshold during each faulty case studied. The start and end points considered for each data set were described in section 2.2.2. The fact that most faults were introduced gradually makes it more difficult to identify the start and end of the fault. For this performance analysis it was established that the starting point of each fault will be the instant when the system was first manipulated to seed the fault. In the particular case of data sets 5.1 and 5.2 where the faulty conditions were introduced and removed 2 and 3 times respectively in each experiment, the start and end of the faulty conditions is considered as the instant when the flow rate set point expected to produce slugging was introduced or changed.

Those fault cases where the indicator produced no successful detection because the threshold was not reached consistently at any point between the fault start and end are indicated in the tables as "ND". This includes those fault cases where the indicator reached the threshold value only for short periods of time between the fault start and end, returning rapidly to normal values. The fault cases where the event of fault detection was impossible to determine because the indicator value was already over the threshold before the fault introduction are identified as "OT".

The false alarm rates presented in Table 13 and Table 14 are calculated as the percentage of observations outside the fault region when the indicator exceeds the threshold. The detection times presented in Table 15 and

Table 16 are calculated as the difference between the fault detection time and the fault starting time. The detection time is the first time instance followed a significant number of consecutive time instances with the $T^2$ and Q indicators above their thresholds correspondingly. The objective of this approach is to avoid short false alarms produced by an indicator before the actual detection of the fault.

***3.3.1 Detection rate***

The results from Table 11 about the detection rate observed using the $T^2$ statistic show that PCA and DPCA have a similar rate of detection success. The detection rate observed by adding KDE for the calculation of the indicator threshold is also very similar for both algorithms, and no significant improvement was found in comparison with the Gaussian assumption. The performance of PLS was slightly better than PCA and DPCA, but in most cases it was impossible to determine the fault detection event using DPLS due to the unrealistically low value obtained from the threshold estimation. The DPLS results were massively improved adding KDE for the calculation of a more realistic threshold, obtaining similar results than for the rest of algorithm s. Similarly, it was impossible in most cases to determine the success rate of CVA due to the low threshold value estimated using the Gaussian assumption, but the results were improved by adding KDE, obtaining better detection rates than with any other method.

It is important to mention that in the particular fault case 2, PCA and DPCA failed to produce consistent fault detections. The fault detections provided by PLS and DPLS produced a high false alarm rate (Table 13), consequently only CVA with KDE was able in this case to produce consistent and reliable fault detection.

Table 11: $T^2$ detection rate (%)

| Fault Case | Data set | *PCA* | *DPCA* | *PCA KDE* | *DPCA KDE* | *PLS* | *DPLS* | *PLS KDE* | *DPLS KDE* | *CVA* | *CVA KDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.1 | 22.65 | 23.10 | 14.72 | 14.69 | 36.49 | OT | 27.11 | 20.97 | OT | 60.66 |
| | 1.2 | 22.91 | 22.85 | 22.85 | 22.79 | 32.47 | OT | 22.91 | 22.98 | OT | 51.73 |
| | 1.3 | 19.63 | 19.60 | 19.56 | 19.56 | 31.2 | OT | 19.7 | 19.77 | OT | 39.73 |
| 2 | 2.1 | ND | ND | ND | ND | 1.87 | 59.42 | ND | ND | OT | 21.87 |
| | 2.2 | ND | ND | ND | ND | ND | 15.45 | ND | ND | OT | 16.88 |
| | 2.3 | ND | ND | ND | ND | ND | ND | ND | ND | OT | 29.21 |
| 3 | 3.1 | 98.37 | 98.44 | 97.69 | 97.81 | 99.36 | OT | 96.71 | 96.72 | OT | 98.69 |
| | 3.2 | 81.41 | 82.81 | 37.14 | 36.53 | 48.23 | 92.79 | 32.95 | 32.39 | OT | 45.75 |
| | 3.3 | 97.22 | 97.73 | 95.57 | 95.37 | 99.36 | OT | 96.21 | 96.62 | OT | 98.81 |
| 4 | 4.1 | 34.50 | 35.61 | 30.425 | 29.43 | 40.34 | OT | 30.94 | 30.08 | OT | 57.74 |
| | 4.2 | 9.5 | 9.87 | 6.4 | 4.70 | 15.07 | OT | 5.97 | 3.44 | 90.53 | 21.50 |
| | 4.3 | 18.43 | 17.23 | 15.47 | 12.3 | 25.43 | OT | 14.8 | 15.10 | OT | 29.03 |
| 5 | 5.1 | 70.63 | 71.87 | 53.98 | 54.91 | 94.72 | OT | 29.69 | 28.74 | OT | 25.33 |
| | 5.2 | 83.34 | 84.19 | 67.51 | 66.49 | 98.68 | OT | 57.73 | 52.77 | 100 | 43.17 |
| 6 | 6.1 | 99.72 | 99.53 | 99.72 | 99.44 | 67.13 | OT | ND | ND | OT | 99.91 |
| | 6.2 | 99.86 | 99.78 | 99.84 | 99.76 | ND | OT | ND | ND | 100 | 100 |

The results obtained from the analysis of the detection rate using the *Q* statistic (Table 12) indicate a similar trend to $T^2$. DPCA shows a better performance than PCA and the addition of KDE reduces slightly the detection rate of both methods due to the higher threshold value obtained. DPLS performance is better than PLS, but again this performance is slightly reduced by the addition of KDE, although the real benefit of KDE will be seen in the false alarm rate. The detection rate of CVA is better than the rate observed for other techniques. Again the addition of KDE to CVA reduces slightly the detection rate due to the higher threshold estimated, but this small decrement will be rewarded by a large reduction of the false alarm rate.

In the particular fault case 2, the *Q* statistic seems to have the inverse effect than $T^2$. In this case PCA and DPCA were able to produce consistent fault detection with relatively high detection rate, although the false alarm rate was also high for these algorithms (see Table 14). In the other hand, the *Q* statistic obtained from CVA with KDE was not able to detect this fault satisfactorily.

Table 12: *Q* detection rate (%)

| Fault Case | Data set | *PCA* | *DPCA* | *PCA KDE* | *DPCA KDE* | *PLS* | *DPLS* | *PLS KDE* | *DPLS KDE* | *CVA* | *CVA KDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.1 | 52.72 | 60.17 | 50.32 | 49.68 | 30.23 | 50.34 | 34.80 | 29.90 | 81.24 | 61.41 |
| | 1.2 | 42.92 | 43.33 | 42.46 | 42.48 | 32.56 | 32.69 | 32.59 | 32.63 | 83.75 | 61.13 |
| | 1.3 | 38.23 | 42.97 | 33.53 | 36.37 | 19.83 | 87.83 | 20.26 | 29.9 | 64.73 | 39.73 |
| 2 | 2.1 | 62.96 | 69.47 | 58.30 | 60.43 | ND | 6.84 | 1.21 | 3.79 | 46.75 | ND |
| | 2.2 | 24.86 | 45.22 | 18.67 | 18.35 | ND | 12.16 | ND | ND | 17.84 | ND |
| | 2.3 | 32.54 | 15.87 | 30.43 | 29.35 | ND | 12.68 | ND | ND | 34.59 | ND |
| 3 | 3.1 | 99.72 | 100 | 99.84 | 100 | 98.67 | 99.42 | 98.75 | 98.65 | 99.91 | 97.87 |
| | 3.2 | OT | OT | 96.87 | 99.89 | 88.01 | 93.82 | 96.17 | 56.62 | OT | 41.71 |
| | 3.3 | 99.65 | 99.70 | 99.63 | 99.61 | 99.25 | 99.88 | 99.26 | 99.38 | 99.60 | 96.55 |
| 4 | 4.1 | 92.64 | 94.14 | 92.14 | 91.98 | 43.02 | 57.59 | 45.22 | 42.85 | 92.53 | 82.32 |
| | 4.2 | 31.63 | 37.20 | 28.47 | 30.77 | 15.30 | 61.53 | 16.17 | 15.73 | 90.97 | 47.63 |
| | 4.3 | 94.86 | 99.43 | 92.23 | 94.1 | 22.67 | 54.1 | 25.00 | 25.83 | 92.3 | 58.43 |
| 5 | 5.1 | OT | OT | OT | OT | 58.63 | 96.48 | 65.77 | 78.80 | OT | 16.23 |
| | 5.2 | 89.41 | 93.28 | 87.34 | 89.57 | 71.51 | 97.64 | 74.13 | 85.89 | 100 | 67.49 |
| 6 | 6.1 | 99.81 | 99.81 | 99.81 | 99.81 | 99.81 | 99.72 | 99.81 | 99.72 | 99.91 | 99.91 |
| | 6.2 | 99.94 | 100 | 100 | 100 | 99.92 | 99.89 | 100 | 99.89 | 100 | 100 |

### *3.3.2 False alarm*

In terms of $T^2$ false alarms (Table 13), PCA and DPCA showed a similar performance, which really was improved by the introduction of KDE .This improved performance caused by the addition of KDE is even more evident in the case of PLS and DPLS, which show the best results together with CVA including also KDE for the estimation of the threshold. The results obtained from this analysis evidence the major improvements obtained when KDE is used to estimate the threshold of the indicators. Despite this improvement, fault case 2 was not detected by PCA and PLS based algorithms, which is an example of the superior performance of CVA.

Table 13: $T^2$ false alarm rate (%)

| Fault Case | Data set | *PCA* | *DPCA* | *PCA KDE* | *DPCA KDE* | *PLS* | *DPLS* | *PLS KDE* | *DPLS KDE* | *CVA* | *CVA KDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.1 | 5.91 | 5.69 | 3.09 | 3.18 | 23.99 | OT | 0.63 | 0.45 | OT | 1.14 |
| | 1.2 | 13.51 | 13.95 | 10.84 | 11.14 | 18.48 | OT | 5.79 | 1.78 | OT | 2.89 |
| | 1.3 | 9.53 | 10.14 | 6.73 | 6.89 | 12.11 | OT | 1.59 | 0.98 | OT | 2.42 |
| 2 | 2.1 | ND | ND | ND | ND | 9.58 | 95.71 | ND | ND | OT | 41.76 |
| | 2.2 | ND | ND | ND | ND | ND | 18.90 | ND | ND | OT | 1.67 |
| | 2.3 | ND | ND | ND | ND | ND | ND | ND | ND | OT | 1.47 |
| 3 | 3.1 | 15.95 | 13.32 | 10.62 | 10.83 | 22.62 | OT | 2.18 | 1.81 | OT | 2.13 |
| | 3.2 | 51.77 | 51.77 | 38.96 | 37.60 | 10.08 | 56.57 | 4.08 | 2.60 | OT | 3.54 |
| | 3.3 | 22.29 | 22.30 | 17.94 | 17.78 | 18.95 | OT | 2.28 | 1.89 | OT | 2.78 |
| 4 | 4.1 | 3.43 | 3.64 | 1.49 | 1.49 | 22.33 | OT | 1.50 | 1.50 | OT | 2.78 |
| | 4.2 | 9.17 | 9.65 | 6.96 | 6.96 | 14.81 | OT | 5.65 | 5.04 | 39.29 | 5.31 |
| | 4.3 | 29.35 | 29.65 | 21.48 | 21.33 | 32.22 | OT | 9.23 | 4.39 | OT | 13.31 |
| 5 | 5.1 | 70.08 | 69.82 | 65.31 | 65.05 | 50.25 | OT | 23.95 | 23.32 | ND | 24.46 |
| | 5.2 | 11.47 | 11.15 | 3.03 | 2.68 | 64.44 | OT | 3.02 | 3.10 | 57.91 | 0.48 |
| 6 | 6.1 | 1.27 | 1.45 | 0.29 | 0.29 | 0.29 | OT | ND | ND | OT | 0 |
| | 6.2 | 0 | 0 | 0 | 0 | ND | OT | ND | ND | 37.84 | 0 |

In the case of false alarms observed in the *Q* statistic (Table 14), PCA showed a better performance than DPCA, but those differences are minimal after the addition of KDE, which again caused a massive improvement in the false alarm rate. Similarly PLS presents lower rates of false alarms than DPLS and the latter one improved its performance remarkably with the addition of KDE. CVA shows a relatively high rate of false alarms, but this rate is reduced when adding KDE, obtaining the best results among all the algorithms studied although fault 2 was not detected by the *Q* statistic.

Table 14: *Q* false alarm rate (%)

| Fault Case | Data set | *PCA* | *DPCA* | *PCA KDE* | *DPCA KDE* | *PLS* | *DPLS* | *PLS KDE* | *DPLS KDE* | *CVA* | *CVA KDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.1 | 16.98 | 37.56 | 8.51 | 11.20 | 3.051 | 34.17 | 4.00 | 2.73 | 37.84 | 1.34 |
| | 1.2 | 35.26 | 50.26 | 25.24 | 34.04 | 11.14 | 12.28 | 12.03 | 8.04 | 32.74 | 3.34 |
| | 1.3 | 10.22 | 10.74 | 8.25 | 9.47 | 6.89 | 66.81 | 7.00 | 6.91 | 20.06 | 3.10 |
| 2 | 2.1 | 53.58 | 53.46 | 53.53 | 53.47 | ND | 22.80 | 0.91 | 1.57 | 60.49 | ND |
| | 2.2 | 23.48 | 56.91 | 11.02 | 14.06 | ND | 3.20 | ND | ND | 18.77 | ND |
| | 2.3 | 11.59 | 15.87 | 7.47 | 5.83 | ND | 8.89 | ND | ND | 30.66 | ND |
| 3 | 3.1 | 32.97 | 57.79 | 20.43 | 33.17 | 11.41 | 54.17 | 13.39 | 10.00 | 33.83 | 1.54 |
| | 3.2 | OT | OT | 71.93 | 86.64 | 48.36 | 51.50 | 52.17 | 16.43 | OT | 3.67 |
| | 3.3 | 12.31 | 15.73 | 9.98 | 9.92 | 16.94 | 35.59 | 19.40 | 15.92 | 25.25 | 5.51 |
| 4 | 4.1 | 47.30 | 50.51 | 38.29 | 47.93 | 1.93 | 56.93 | 2.25 | 2.30 | 46.27 | 2.78 |
| | 4.2 | 25.91 | 49.62 | 18.12 | 20.88 | 7.03 | 8.62 | 7.17 | 6.77 | 41.35 | 6.75 |
| | 4.3 | 42.97 | 60.96 | 31.32 | 39.93 | 18.46 | 75.34 | 20.12 | 19.06 | 41.30 | 12.10 |
| 5 | 5.1 | OT | OT | OT | OT | 35.77 | 73.06 | 41.99 | 41.11 | OT | 23.76 |
| | 5.2 | 22.53 | 26.77 | 19.90 | 61.64 | 7.86 | 32.34 | 9.02 | 21.73 | 40.97 | 5 |
| 6 | 6.1 | 5.97 | 8.87 | 3.19 | 4.87 | 0 | 0.81 | 0 | 0 | 8.067 | 0 |
| | 6.2 | 0.28 | 0 | 0.28 | 0 | 0 | 0.38 | 0 | 0 | 13.02 | 0 |

### *3.3.3 Detection time*

In terms of detection time using $T^2$ (Table 15), DPCA shows a faster response in some cases compared with PCA. The detection time is slightly increased for both algorithms after the addition of KDE due to the higher threshold estimated. Similarly, DPLS with KDE presents lower detection times than PLS with KDE in most cases, improving also the performance observed in DPCA, but the algorithm which showed the fastest reaction to the faults was CVA with KDE. In fault case 4, the differences in detection time between algorithms are minimal, which together with the low false alarm rates shows this fault is relatively easy to detect for all the algorithms tested.

Using the Q indicator the fault reaction time (

Table 16) is similar for PCA and DPCA, but DPCA produced slightly better results. The addition of KDE in these cases again increases the detection time. DPLS performs better than PLS but after the addition of KDE the performance of both algorithms is similar, and slower compared with the Gaussian assumption. CVA shows the fastest performance, which is slightly reduced by the addition of KDE.

Table 15: $T^2$ detection time (s)

| Fault Case | Data set | *PCA* | *DPCA* | *PCA KDE* | *DPCA KDE* | *PLS* | *DPLS* | *PLS KDE* | *DPLS KDE* | *CVA* | *CVA KDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.1 | 3080 | 3080 | 3083 | 3084 | 2296 | OT | 2698 | 3080 | OT | 1426 |
| | 1.2 | 2405 | 2407 | 2407 | 2409 | 2107 | OT | 2405 | 2407 | OT | 1506 |
| | 1.3 | 2411 | 2412 | 2413 | 2415 | 2105 | OT | 2409 | 2411 | OT | 1808 |
| 2 | 2.1 | ND | ND | ND | ND | 4378 | 4023 | ND | ND | OT | 3416 |
| | 2.2 | ND | ND | ND | ND | ND | 1849 | ND | ND | OT | 1812 |
| | 2.3 | ND | ND | ND | ND | ND | ND | ND | ND | OT | 1512 |
| 3 | 3.1 | 207 | 112 | 374 | 211 | 52 | OT | 399 | 235 | OT | 94 |
| | 3.2 | 3266 | 3219 | 3745 | 3733 | 3240 | 1405 | 3967 | 3788 | OT | 3086 |
| | 3.3 | 434 | 307 | 453 | 450 | 90 | OT | 451 | 307 | OT | 106 |
| 4 | 4.1 | 585 | 586 | 585 | 588 | 512 | OT | 584 | 585 | OT | 548 |
| | 4.2 | 493 | 494 | 507 | 443 | 457 | OT | 442 | 448 | 291 | 436 |
| | 4.3 | 400 | 393 | 416 | 404 | 375 | OT | 400 | 400 | OT | 416 |
| 5 | 5.1 (1) | OT | OT | OT | OT | OT | OT | 65 | 64 | OT | 69 |
| | 5.1 (2) | 40 | 37 | 47 | 45 | 28 | OT | 103 | 121 | 29 | 109 |
| | 5.2 (1) | 27 | 28 | 42 | 40 | 32 | OT | 50 | 47 | 2 | 83 |
| | 5.2 (2) | 54 | 53 | 54 | 182 | OT | OT | OT | OT | OT | 44 |
| | 5.2 (3) | 16 | 14 | 19 | 18 | 17 | OT | 36 | 86 | 83 | 114 |
| 6 | 6.1 | 3 | 1 | 3 | 2 | 356 | OT | ND | ND | OT | 1 |
| | 6.2 | 5 | 4 | 6 | 5 | ND | OT | ND | ND | 1 | 1 |

Table 16: *Q* detection time (s)

| Fault Case | Data set | PCA | DPCA | PCA KDE | DPCA KDE | PLS | DPLS | PLS KDE | DPLS KDE | CVA | CVA KDE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.1 | 1829 | 1830 | 1830 | 1831 | 2584 | 1834 | 2583 | 2585 | 644 | 1419 |
| | 1.2 | 1803 | 1804 | 1804 | 1804 | 2104 | 2104 | 2103 | 2106 | 590 | 1213 |
| | 1.3 | 2103 | 1809 | 2103 | 2101 | 2405 | 1713 | 2404 | 2107 | 1195 | 1808 |
| 2 | 2.1 | 1852 | 1783 | 2214 | 1857 | ND | 4073 | 4199 | 4249 | 3431 | ND |
| | 2.2 | 1816 | 1803 | 1828 | 1817 | ND | 1912 | ND | ND | 1791 | ND |
| | 2.3 | 1536 | 1536 | 1542 | 1537 | ND | 1918 | ND | ND | 1481 | ND |
| 3 | 3.1 | 17 | 0 | 74 | 0 | 99 | 46 | 94 | 101 | 34 | 188 |
| | 3.2 | OT | OT | 2067 | 1173 | 1834 | 1405 | 1409 | 2637 | OT | 3371 |
| | 3.3 | 34 | 34 | 50 | 35 | 93 | 31 | 92 | 65 | 22 | 309 |
| 4 | 4.1 | 426 | 422 | 436 | 428 | 480 | 461 | 467 | 476 | 455 | 578 |
| | 4.2 | 376 | 348 | 410 | 369 | 455 | 436 | 440 | 442 | 292 | 430 |
| | 4.3 | 302 | 162 | 313 | 302 | 379 | 371 | 378 | 3737 | 293 | 371 |
| 5 | 5.1 (1) | OT | OT | OT | OT | OT | OT | OT | OT | OT | 73 |
| | 5.1 (2) | OT | OT | OT | OT | 47 | 32 | 42 | 60 | 23 | 109 |
| | 5.2 (1) | 36 | 31 | 21 | 34 | 39 | 29 | 38 | 34 | 1 | 87 |
| | 5.2 (2) | OT | OT | OT | OT | OT | OT | OT | OT | OT | OT |
| | 5.2 (3) | 85 | 13 | 85 | 16 | 29 | 23 | 29 | 27 | 36 | 46 |
| 6 | 6.1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| | 6.2 | 2 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 1 | 1 |

## 4. Conclusion

Despite the success reported by many researchers in the application of different data driven methods for fault detection and diagnosis using computer simulated data and real data acquired in different experimental rigs and industrial facilities, there is a lack of a common benchmark case for the comparison of the performance of these methods using real process data. This work is presented with the aim of providing a benchmark case study to prove the effectiveness of different methodologies for the detection and diagnosis of faults in processes working under varying operational conditions. In this case study, many faults were introduced gradually in order to simulate the natural process of degradation expected in real systems, but also to provide data sets where the detection rate of different monitoring methods can be clearly compared. CVA was applied to sets of data obtained in a large scale test rig where different process faults were seeded under varying operational conditions. The faults were successfully detected using the $T^2$ and $Q$ metrics. Normally the fault detection happened earlier for the $Q$ statistic but on the other hand, $Q$ produced a higher number of false alarms than $T^2$. This indicates that the $Q$ statistic was slightly oversensitive for these particular conditions. This phenomenon has already been reported by other researchers [10] and it has been attributed to the fact that most faults tend to alter the underline model different from normal operation condition, hence causing residual errors increase rather than magnifying the variation in the model space determined based on normal operations. Contribution

plots were used to locate those measurements which are more severely affected by the fault in order to help in the process of fault diagnosis once a fault has been detected. The results obtained for the $T^2$ contribution plots were always related with the fault introduced and, combined with knowledge about the process, could have been used for fault identification. In the other hand the $Q$ contribution plots were not always as clear as the $T^2$ plots. This was probably caused by the mentioned oversensitivity of the $Q$ indicator; as a consequence any small variation in the residual space is represented in the contribution plots, sharing the contribution between several variables which complicates the fault detection. These results demonstrate that CVA can be effectively applied for the detection and diagnosis of faults in real complex systems working under variable operating conditions.

When comparing the performance of CVA including KDE with other methodologies it is clear that this method is superior to the rest in terms of detection rate, false alarm rate and detection time. For fault detection, CVA with KDE produced a higher detection rate in most cases and, when it was not the case, the higher rate obtained using other method had a much higher false alarm rate as a consequence (see for example data sets 3.2 and 4.2). It is noticeable that the use of KDE for the calculation of the thresholds improved the performance of all the algorithms in terms of false alarms rate. In some of the studied faulty cases it was not possible to use the results obtained for fault detection if the thresholds were calculated using the Gaussian assumption because the indicator value was higher than the threshold during the whole experiment. This evidences that the Gaussian assumption in this case was unrealistic due to high nonlinearities in the system, which made almost essential the use of KDE for threshold estimation. Despite the superior performance showed by CVA with KDE, the detection of the fault introduced in Fault Case 2 proved to be particularly challenging. The faults introduced in this case were not detected by the $Q$ indicator, and the detection rates provided by $T^2$ were relatively low, which evidences the need to develop improved algorithms for these particular fault scenarios in future.

**Acknowledgements**

Financial support from the Marie Curie FP7-ITN project "Energy savings from smart operation of electrical, process and mechanical equipment– ENERGY-SMARTOPS", Contract No: PITN-GA-2010-264940 is gratefully acknowledged.

**References**

[1] Ricker, N. L. , *Tennessee Eastman Challenge Archive*, available at: http://depts.washington.edu/control/LARRY/TE/download.html (accessed 21/03/2014).

[2] Isermann, R. (2006), "Fault detection with Principal Component Analysis (PCA)", in *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Springer, , pp. 267-284.

[3] Odiowei, P. -. P. and Cao, Y. (2009), "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations", *Computer Aided Chemical Engineering*, vol. 27, no. C, pp. 1557-1562.

[4] Yang, Y., Chen, Y., Chen, X. and Liu, X. (2012), "Multivariate industrial process monitoring based on the integration method of canonical variate analysis and independent component analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 116, pp. 94-101.

[5] Chen, R. Q. (2013), "Advances in data-driven monitoring methods for complex process", *3rd International Conference on Applied Mechanics, Materials and Manufacturing, ICAMMM 2013;* Vol. 423-426, 24 August 2013, Dalian; China, pp. 2448.

[6] Ku, W., Storer, R. H. and Georgakis, C. (1995), "Disturbance detection and isolation by dynamic principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 30, no. 1, pp. 179-196.

[7] Komulainen, T., Sourander, M. and Jämsä-Jounela, S. -. (2004), "An online application of dynamic PLS to a dearomatization process", *Computers and Chemical Engineering*, vol. 28, no. 12, pp. 2611-2619.

[8] Juricek, B. C., Seborg, D. E. and Larimore, W. E. (2004), "Fault Detection Using Canonical Variate Analysis", *Industrial and Engineering Chemistry Research*, vol. 43, no. 2, pp. 458-474.

[9] Russell, E. L., Chiang, L. H. and Braatz, R. D. (2000), "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 81-93.

[10] L.H. Chiang and E.L. Rusell and R.D. Braatz (2000), *Fault detection and diagnosis in industrial systems*, 1st ed, Springer, London UK.

[11] Kladis, G. P., Economou, J. T., Tsourdos, A., White, B. A. and Knowles, K. (2009), "Fault diagnosis with matrix analysis for electrically actuated unmanned aerial vehicles", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 223, no. 5, pp. 543-563.

[12] Yunus, M. Y. M. and Zhang, J. (2010), "Multivariate process monitoring using classical multidimensional scaling and procrustes analysis", *IFAC Proceedings Volumes (IFAC-PapersOnline)*, Vol. 9, pp. 165.

[13] Qi, Y. -., Wang, P., Fan, S. -., Gao, X. -. and Jiang, J. -. (2009), "Enhanced batch process monitoring using kalman filter and multiway kernel principal component analysis", *2009 Chinese Control and Decision Conference, CCDC 2009*, pp. 5289.

[14] Choi, S. W., Park, J. H. and Lee, I. -. (2004), "Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis", *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1377-1387.

[15] Ma, L. -., Wang, J. -. and Song, Y. (2004), "Process monitoring method based on multi-PCA models", *Beijing Ligong Daxue Xuebao/Transaction of Beijing Institute of Technology*, vol. 24, no. 1, pp. 64-68.

[16] Zhang, J., Martin, E. B. and Morris, A. J. (1997), "Process monitoring using non-linear statistical techniques", *Chemical Engineering Journal*, vol. 67, no. 3, pp. 181-189.

[17] Simoglou, A., Martin, E. B. and Morris, A. J. (2002), "Statistical performance monitoring of dynamic multivariate processes using state space modelling", *Computers and Chemical Engineering*, vol. 26, no. 6, pp. 909-920.

[18] Zhou, K. -., Li, Q. -. and Guo, R. -. (2012), "Improving monitoring accuracy of process based on SPC method", *Proceedings of the 2012 24th Chinese Control and Decision Conference, CCDC 2012*, pp. 1488.

[19] Fan, J., Qin, S. J. and Wang, Y. (2014), "Online monitoring of nonlinear multivariate industrial processes using filtering KICA-PCA", *Control Engineering Practice*, vol. 22, no. 1, pp. 205-216.

[20] Lau, C. K., Ghosh, K., Hussain, M. A. and Che Hassan, C. R. (2013), "Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS", *Chemometrics and Intelligent Laboratory Systems*, vol. 120, no. 0, pp. 1-14.

[21] Wang, Q. and Li, H. (2012), "A process singular value recognition based recursive PCA approach to mode-transition process monitoring", *Huagong Xuebao/CIESC Journal*, vol. 63, no. 9, pp. 2948-2952.

[22] Rato, T. J. and Reis, M. S. (2013), "Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR)", *Chemometrics and Intelligent Laboratory Systems*, vol. 125, pp. 101-108.

[23] Ma, H. -., Hu, Y., Yan, X. -. and Shi, H. -. (2012), "An improved PLS (IPLS) method utilizing local standardization strategy for multimode process monitoring", *Journal of Donghua University (English Edition)*, vol. 29, no. 4, pp. 288-294.

[24] Yi, H., Hehe, M. and Hongbo, S. (2013), "Fault detection for chemical process based on robust PLS", *2013 25th Chinese Control and Decision Conference, CCDC 2013*, pp. 4947.

[25] Stubbs, S., Zhang, J. and Morris, J. (2012), "Fault detection in dynamic processes using a simplified monitoring-specific CVA state space modelling approach", *Computers and Chemical Engineering*, vol. 41, pp. 77-87.

[26] Odiowei, P. -. P. and Cao, Y. (2010), "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations", *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 36-45.

[27] He, X. B., Yang, Y. P. and Yang, Y. H. (2008), "Fault diagnosis based on variable-weighted kernel Fisher discriminant analysis", *Chemometrics and Intelligent Laboratory Systems*, vol. 93, no. 1, pp. 27-33.

[28] Mori, J. and Yu, J. (2014), "A quality relevant non-Gaussian latent subspace projection method for chemical process monitoring and fault detection", *AIChE Journal*, vol. 60, no. 2, pp. 485-499.

[29] Jiang, Q. and Yan, X. (2012), "Multivariate statistical process monitoring using modified factor analysis and its application", *Journal of Chemical Engineering of Japan*, vol. 45, no. 10, pp. 829-839.

[30] Liu, J. and Chen, D. -. (2009), "Fault detection and identification using modified bayesian classification on PCA subspace", *Industrial and Engineering Chemistry Research*, vol. 48, no. 6, pp. 3059-3077.

[31] Verron, S., Li, J. and Tiplica, T. (2010), "Fault detection and isolation of faults in a multivariate process with Bayesian network", *Journal of Process Control,* vol. 20, no. 8, pp. 902-911.

[32] Lee, J. -., Yoo, C. K., Choi, S. W., Vanrolleghem, P. A. and Lee, I. -. (2004), "Nonlinear process monitoring using kernel principal component analysis", *Chemical Engineering Science,* vol. 59, no. 1, pp. 223-234.

[33] Liu, D., Jiang, D., Chen, X., Luo, A. and Xu, G. (2012), "Research on fault identification for complex system based on generalized linear canonical correlation analysis", *2012 IEEE International Conference on Automation Science and Engineering: Green Automation Toward a Sustainable Society, CASE 2012,* 20 August 2012 through 24 August 2012, Seoul, pp. 474.

[34] Borsje, H. J. (1999), "Fault detection in boilers using canonical variate analysis", *Proceedings of the 1999 American Control Conference (99ACC),* Vol. 2, 2 June 1999 through 4 June 1999, San Diego, CA, USA, IEEE, Piscataway, NJ, United States, pp. 1167.

[35] Cao, Y., "A Benchmark Case for Statistical Process Monitoring - Cranfield Multiphase Flow Facility", *MATLAB File Exchange*, http://www.mathworks.com/matlabcentral/fileexchange/50938-a-benchmark-case-for-statistical-process-monitoring-cranfield-multiphase-flow-facility.

[36] Deng, X. and Tian, X. (2011), "A new fault isolation method based on unified contribution plots", *Proceedings of the 30th Chinese Control Conference, CCC 2011,* pp. 4280.

[37] Liu, J. and Chen, D. -. (2012), "Multiple Sensor Fault Isolation Using Contribution Plots without Smearing Effect to Non-Faulty Variables", *Computer Aided Chemical Engineering,* vol. 31, pp. 1517-1521.

[38] Mnassri, B., El Adel, E. M., Ananou, B. and Ouladsine, M. (2009), "Fault detection and diagnosis based on PCA and a new contribution plots", *IFAC Proceedings Volumes (IFAC-PapersOnline),* pp. 834.

[39] Ramírez, A. W. and Llinàs, J. C. (2011), "Fault diagnosis of batch processes release using PCA contribution plots as fault signatures", *ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems,* Vol. 1 DISI, pp. 223.

[40] Emerson Electric Co. , *The DeltaV Digital Automation System* , available at: http://www2.emersonprocess.com/en-us/brands/deltav/pages/index.aspx (accessed 13 February 2014).

[41] Jansen, F. E., Shoham, O. and Taitel, Y. (1996), "The elimination of severe slugging - Experiments and modeling", *International Journal of Multiphase Flow,* vol. 22, no. 6, pp. 1055-1072.

[42] Ogazi, A. I., Cao, Y., Yeung, H. and Lao, L. (2010), "Slug control with large valve openings to maximize oil production", *SPE Journal,* vol. 15, no. 3, pp. 812-821.

[43] Negiz, A. and Çinar, A. (1998), "Monitoring of multivariable dynamic processes and sensor auditing", *Journal of Process Control,* vol. 8, no. 5-6, pp. 375-380.



https://dspace.org/

# Statistical process monitoring of a multiphase flow facility

**Ruiz Carcel, Cristobal; Cao, Yi; Harrison, David; Lao, Liyun; Samuel, Raphael**

**2015-06-16**
Elsevier
https://dspace.lib.cranfield.ac.uk/handle/1826/10650
Ruiz Carcel C et al. Statistical process monitoring of a multiphase ow facility, Statistical process monitoring of a multiphase ow facility, Volume 42, September 2015, Pages 74–88.

Downloaded from a DSpace repository. https://dspace.org/