Engineering Applications of Artificial Intelligence 122 (2023) 106145

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Integrating PCA and structural model decomposition to improve fault monitoring and diagnosis with varying operation points

D. Garcia-Alvarez [*,1], A. Bregon [1], B. Pulido [1], C.J. Alonso-Gonzalez [1]

Departamento de Informática, University of Valladolid, Valladolid, 47011, Spain

ARTICLE INFO

Keywords:
Fault diagnosis
Model decomposition
Possible Conflicts
Multivariate Statistical Process Control
Principal Component Analysis
Multiple Operation Points Systems

ABSTRACT

Fast and efficient fault monitoring and diagnostics methods are essential for fault diagnosis and prognosis tasks in Health Monitoring Systems. These tasks are even more complicated when facing dynamic systems with multiple operation points. This article introduces a symbiotic solution for fault detection and isolation, based on the integration of two complementary techniques: Possible Conflicts (PCs), a model-based diagnosis technique from the Artificial Intelligence (AI) community, and Principal Component Analysis (PCA), a Multivariate Statistical Process Control (MSPC) technique. Our proposal improves the PCA-based fault detection in systems with multiple operation points and transient states and provides a straightforward fault isolation stage for PCA. At the same time, the proposal increases the robustness for fault detection using PCs through the application of PCA to the residual signals. PCA has the ability to filter out residual deviations caused by model uncertainties that can lead to a high number of false positives. The proposed method has been successfully tested in a real-world plant with accurate fault detection results. The plant has noisy sensors and a system model without the same accuracy at each operation point and transient states.

1. Introduction

Complex dynamic systems, such as industrial and aerospace systems, require efficient fault diagnosis solutions to ease the prognosis and health monitoring tasks. Fault monitoring methods based on process data history have proven to be successful techniques for this job by directly collecting and analyzing the system's measurements (Venkatasubramanian et al., 2003). However, accurate and quick real-time monitoring using these techniques can be compromised when large amounts of highly correlated data are being managed. One widely used technique to overcome such a flaw is PCA (Jackson and Mudholkar, 1979; Wold et al., 1987).

PCA is a MSPC technique that has been used as a monitoring approach in many industrial processes (Kourti and MacGregor, 1996; Shlens, 2005; Ferrer, 2007). PCA is capable of finding the principal sources of variability in the space of the measured variables, thus allowing the dimensionality of the original space to be reduced in order to fit into a new one with the minimum number of uncorrelated variables required to explain the process trends (known as *latent variables* or *components*) (Camacho et al., 2009; Banguero et al., 2020). This reduction is carried out by searching linear combinations between the measured variables, and the transformation matrix procedure used to compute the components is known as the PCA model.

Nevertheless, PCA presents important limitations when dealing with continuous processes that go through different operation points. Changes in the current operation point can be caused by variations in product specifications, input variables or set-points of the system. In all these situations, the relationship between process variables will change, leading to wrong fault detections caused by discrepancies between the trends captured by the PCA model and the new process state. As pointed out by Hwang and Han (1999) and Tien et al. (2004), several solutions have been proposed to overcome such issues, such as building a PCA model for each operation point (Zhao et al., 2004, 2006), updating the model to reflect the changes, or developing a conventional PCA model considering all the operation points. However, these solutions could face difficulties when dealing with a large number of operation points due to the necessity of off-line PCA model fitting, transient monitoring, or online model reformulation, among others (Lane et al., 2003; Zarzo and Ferrer, 2004; De Ketelaere et al., 2015; Portnoy et al., 2016; Rafferty et al., 2016; Bakdi and Kouadri, 2017; Rezamand et al., 2020). Additionally, some of these solutions might exhibit poor detectability performance due to the difficulty for the PCA model to distinguish between the fault occurrence and the change in the operation point (García-Álvarez et al., 2012). Finally, PCA provides little support for fault isolation (Gertler et al., 1999).

* Corresponding author.
E-mail addresses: dieggar@infor.uva.es (D. Garcia-Alvarez), anibal@infor.uva.es (A. Bregon), belar@infor.uva.es (B. Pulido), calonso@infor.uva.es (C.J. Alonso-Gonzalez).

[1] This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación under grant PID2021-126659OB-I00.

https://doi.org/10.1016/j.engappai.2023.106145
Received 27 April 2022; Received in revised form 12 January 2023; Accepted 10 March 2023
Available online 22 March 2023

On the other hand, online fault diagnosis approaches based on analytical models (Blanke et al., 2006; Gertler, 1998) require quick and robust detection methods to determine significant deviations between observed and expected behavior. These deviations, known as residuals, are related to analytical redundancy derived from the system model. The structure of these residuals can be computed off-line. However, the current value of the residuals are computed on-line. Whenever the value of a residual exceeds a given threshold, a fault detection is performed and the set of constraints used to derive the analytical redundancy expression is considered to be non-consistent with the observations. After this process, fault isolation is straightforward, and a reduced set of faulty candidates can be easily computed.

Residuals can be computed using different methods, such as parity-equations, state-observers, or parameter estimation (Blanke et al., 2006; Gertler et al., 1999). In this work, we focus on Analytical Redundancy Relations (ARRs) (Blanke et al., 2006; Cordier et al., 2004) obtained through structural analysis. More precisely, our work uses PCs (Pulido and Alonso-González, 2004). The computation of the set of PCs is a model decomposition technique from the AI approach to model-based diagnosis. The set of PCs identify the whole set of subsystems with minimal analytical redundancy, similar to ARRs or Minimally Structural Overdetermined Sets (MSO) (Krysander et al., 2008). Each PC has a model that can be used to generate a residual. These residuals are used in a consistency-based diagnosis (CBD) approach and are able to automatically perform fault detection, isolation and identification (Bregon et al., 2012; Moya Alonso et al., 2013). CBD is the most used approach to model-based diagnosis within the AI community.

CBD with PCs, like any other model-based method based on analytical redundancy, requires detailed models of the process to compute residuals. In the absence of faults, residuals are theoretically zero. However, this is not usually the case. Even in simple processes, uncertainty in the models and noise in the sensors require the implementation of some kind of state observer to track the process, thus increasing the complexity of the fault diagnosis system. Still, residuals are usually noisy and some statistical decision-making process, for instance the z-test, is needed to assess a significant deviation of the residuals, which usually implies a compromise between false positive and false negative detections, plus some delay in the fault detection process. Additionally, the presence of nonlinearities poses a new set of problems. Each different operation point may require a different set of models or, at least, a new set of model and observer parameters. In both cases, it is necessary to design a bank of observers for each operation point of interest, which is a non-trivial task for non-linear systems and usually demands several properties from the system models. A related problem is tracking the system in transitory states, which usually generates false positive detections.

An alternative approach is to use the same models in different operation points, avoiding the implementation of state observers, to keep the diagnosis system as simple as possible (Moya Alonso et al., 2013; Pulido et al., 2015). Clearly, the residuals will be noisier and may show some bias, with non-zero means even in the absence of faults in some operation points. Our hypothesis is, that in the absence of faults, the correlations between the residuals will essentially be constant in each operation point, while the onset of a fault will modify the mentioned correlations, because each residual is only sensitive to an (ideally small) subset of faults. Hence fault detection will not be based on residual deviations from zero, but on the alteration of the residual's correlation structure. These alterations can be detected by PCA statistical process monitoring.

Summarizing, in this work, we propose to combine PCA together with CBD with Possible Conflicts to improve the overall diagnosis process for systems with multiple operation points. PC models can track the system state in different operation points (Moya Alonso et al., 2013; Pulido et al., 2015; Bregon et al., 2016). Our proposal is to use PCA to analyze the trends among residual signals, instead of using direct measurements from the system. The presence of faults will modify the correlations between residuals, leading to a fault detection. After this step, contribution analysis is used to determine those variables, i.e., residuals, responsible for such deviations. Deviated residuals will then be used to compute the set of faulty candidates using CBD.

The proposed combination of techniques has been tested in a real-world laboratory plant consisting of two connected tanks with liquid input and output and level control. This system presents two main weaknesses, the signals collected by sensors are significantly noisy and the fitted model does not represent the model with the same accuracy at each operation point. Our solution has proven to be successful in accurately detecting faults at different operation points and in designing a complete and straightforward fault detection and isolation solution for PCA. In addition, it has been observed that the deviations of the residuals due to lack of model accuracy were captured by the PCA model, leading to the reduction of false positives in the detection phase.

The summary of the main contributions of this proposal are: first, using CBD, which requires only correct behavior models, we can detect and isolate any type of failure, without imposing simplifying or unrealistic assumptions about fault models. Second, the method does not require faulty behavioral data, because PCA models can be effectively trained with correct behavioral data, and it can be used to perform robust fault detection by just analyzing the residual signals; moreover, the PCA model can be trained with a relatively small amount of data. Third, we are able to perform accurate and timely fault detection and isolation in different operation points, without the need of having dedicated or complex PCA models for each operation point. Last, but not least, for the case study the proposal is able to track the system during transients among different operating point regions without detecting false alarms using relatively simple analytical and PCA models.

The rest of the article is structured as follows. Sections 2 and 3 briefly present the theoretical background of PCA and PCs, respectively, as well as their use as fault detection and diagnosis techniques. Section 4 introduces the real-world case study used to apply the integration proposal. Section 5 describes the new approach where both techniques are integrated. Section 6 shows the experimental results obtained for the case study. Finally, Section 7 discusses the results of the approach and outlines its main conclusions.

2. Principal component analysis monitoring

The PCA model is fitted by using data in Normal Operation Conditions (NOC) arranged into a data matrix $\mathbf{X} \in \Re^{K\times J}$, where K is the number of time observations and J the number of measured variables in the process. The columns of matrix $\mathbf{X}$ are called variables ($\mathbf{x}_j$) and represent the values of every variable along time, having these variables (columns) means $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_J$ and variances $s_1, s_2, \dots, s_J$. The rows are called individuals or observations ($\mathbf{z}_k^T$). It is advisable to normalize every variable to zero mean and unit variance.

PCA can be computed using the decomposition of the covariance matrix ($\mathbf{S} = \frac{1}{K-1}\mathbf{X}^T\mathbf{X}$) and performing the Singular Value Decomposition (SVD) over it:

$$\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$$

where $\boldsymbol{\Lambda} \in \Re^{J\times J}$ is a diagonal matrix. $\boldsymbol{\Lambda}$ contains the eigenvalues (λ) of $\mathbf{S}$ in its diagonal sorted in decreasing order ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{rank(\mathbf{X})} \geq 0$).

Choosing A eigenvectors of $\mathbf{V}$ corresponding to the greatest A eigenvalues, matrix $\mathbf{P}_{1:A} \in \Re^{J\times A}$ is set. It transforms the space of the measured variables into the reduced dimension space of the latent variables or principal components:

$$\mathbf{T} = \mathbf{X}\mathbf{P}_{1:A}$$

Columns in $\mathbf{P}_{1:A}$ are called *loadings* $\mathbf{p}_a$ and $\mathbf{T} \in \Re^{K\times A}$ is the score matrix whose rows $\boldsymbol{\tau}_a^T = [\tau_1, \tau_2, \dots, \tau_A]^T \in \Re^{1\times A}$ are scores in the ath observation and columns the principal components $\mathbf{t}_a$. Scores can be transformed into the original measured variable space $\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}_{1:A}^T$ and the residual matrix $\mathbf{E}$ can be set as the differences between the original variables and the reconstructed variables $\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}}$. The PCA computation can also be performed by using the Non-linear Iterative Partial Least Squares (NIPALS) algorithm (Wold et al., 1987).

There are several methods proposed for selecting the number of principal components (Jackson, 2003; Weighell et al., 2001; Chiang et al., 2001). However, cross validation is considered one of the best ways to choose the number of principal components (Eastment and Krzanowski, 1982; Camacho, 2007; Bro et al., 2008; Camacho et al., 2010).

2.1. PCA as a fault detection technique

From the point of view of fault detection, two statistics are used to establish control charts to monitor processes (Chiang et al., 2001):

- **Hotelling's statistic** (T^2): for a new process observation $\mathbf{z}^T \in \mathfrak{R}^{1\times J}$, this statistic can be calculated as the sum of the squared A principal scores of the observation divided by the related eigenvalue:

  $$T^2 = \sum_{i=1}^{A} \frac{\tau_i^2}{\lambda_i}$$

  Using this statistic, the process is considered normal for a given significance level α if:

  $$T^2 \leq T_\alpha^2 = \frac{(K^2-1)A}{K(K-A)} F_\alpha(A, K-A)$$

  where $F_\alpha(A, K-A)$ is the critical value $(100(1-\alpha)\%$ percentile) of the Fisher–Snedecor distribution with A and $K-A$ degrees of freedom where α is the level of significance which usually takes values between 5% and 1%.
- **Squared prediction error (SPE) or Q statistic:** this statistic is a measurement of goodness of fit of the sample into the model and can be calculated for a normalized observation $\mathbf{z}^T$:

  $$Q = \mathbf{r}^T \mathbf{r}$$

  with $\mathbf{r} = (\mathbf{I} - \mathbf{P}_{1:A}\mathbf{P}_{1:A}^T)\mathbf{z}$, where $\mathbf{r}$ is the residual between the observation and its projection onto the reduced dimension space. In this case, the upper limit of this statistic can be set:

  $$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}}$$

  where:

  $$\theta_i = \sum_{j=A+1}^{J} \lambda_j^i \qquad h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$$

  c_α being the $100(1-\alpha)$ standardized normal percentile and λ_j^i the eigenvalues of the residual covariance matrix $\mathbf{E}^T\mathbf{E}/(K-1)$.

Although PCA does not provide much information to develop a fault isolation approach, Contribution Analysis (Kourti and MacGregor, 1996; Ferrer, 2007) is a technique which has been proposed as a first attempt for fault isolation. It computes the influence of each variable to the computed value of Q and T^2 statistics. These contributions (*cont*) can be plotted in a bar plot, which is shown together with lines marking the standard deviation and three times the standard deviation in order to detect abnormal values of the contributions.

- **Bar plots of normalized errors of the variables.** When an observation $\mathbf{z}^T = [z_1, z_2, \ldots, z_J]^T \in \mathfrak{R}^{1\times J}$ falls outside the limits in the Q statistic, the normalized error for each value z_j of $\mathbf{z}^T$ is calculated as:

  $$cont_{z_j} = (z_j - \bar{x}_j)/s_j$$
- **Bar plots of normalized scores.** If the T^2 statistic value falls outside the threshold set, the normalized scores can be plotted for the scores of a given observation outside the limits $\boldsymbol{\tau}^T$ as:

  $$cont_{\tau_a} = \tau_a/\lambda_a$$
- **Variable contributions to individual scores.** Bar plots of normalized scores do not provide information about the variables related to a fault. This contribution analysis allows us to plot the contribution of each observation z_j to the computation of the ath score τ_a (normally, those with a high normalized value):

  $$cont_{z_j,\tau_a} = z_j p_{j,a}$$

  where $p_{j,a}$ are elements of $\mathbf{P}_{1:A}$
- **Overall Average variable contributions.** Because it is very common that more than one score has a high value when a fault is detected, it is very useful to compute the overall average contribution per variable, instead of drawing a bar plot for every score with a high value (Kourti and MacGregor, 1996).

2.2. Limitations of PCA as a fault detection technique

PCA-based fault detection and process monitoring techniques are widely used in academia and industry. The reason for its success is due to several factors. In general, PCA-based monitoring techniques only require NOC data collected from the process, considering that the amount of data must be large enough to capture the principal trends of the process. In addition, the PCA model and the process monitoring statistic computation do not have a high computational cost. Furthermore, despite the fact that PCA-based monitoring techniques do not provide a complete fault isolation procedure, the contribution analysis can be used to identify the deviating variables related to monitoring plot triggers.

However, PCA-based monitoring approaches also present some drawbacks, particularly when the process has several operation points (Hwang and Han, 1999; Tien et al., 2004). This circumstance makes it necessary to consider modifications to the classical PCA approach, or even to discard the use of this technique when the number of operation points is high. Another drawback is directly related to the monitoring of transient states that some processes present when switching from one operation point to another. Transient state monitoring requires specific PCA-based techniques that are not as simple as classical PCA approaches (García-Álvarez et al., 2012).

The main target of this work is to improve PCA as a complete fault detection and isolation technique in order to give it the capability of isolating the possible source of a detected fault and also to improve the detection process, considering the mentioned drawbacks when a process presents several operation points and transient states. This improvement is done by using a model-based technique that we present next.

3. Model-based diagnosis using Possible Conflicts

As mentioned in Section 1, the set of Possible Conflicts (Pulido and Alonso-González, 2004) is a model decomposition method from the model-based AI community (usually known as DX community) (Reiter, 1987; Hamscher et al., 1992) that can find off-line the complete set of subsystems with minimal analytical redundancy. PCs can be used in a model-based approach to perform FDII (Fault Detection, Isolation and Identification). Additionally, we have shown (Armengol et al., 2009; Pulido and Alonso-González, 2004) that PCs are, under given assumptions, equivalent to the structure of the set of minimal ARRs (Cordier et al., 2004) and MSO sets of equations (Krysander et al., 2008), all used in structural approaches to FDII in the Control Engineering community (usually known as FDI[2] community).

[2] FDI stands for Fault Detection and Isolation.

The main idea of the DX approach is to link each equation (or component) in the model with a correctness assumption and to find differences between the measured variables and the model estimations. Whenever a difference is found between these two signals (a residual in FDI terminology) the fault isolation process searches for those minimal sets of equations used to estimate the conflicting value (a discrepancy in DX terminology), which are named minimal conflicts (Cordier et al., 2004) (formally, conflicts are the set of correctness assumptions linked to those sets of equations leading to a discrepancy (Hamscher et al., 1992)).

The computation of the set of PCs (Pulido and Alonso-González, 2004) is performed completely off-line in three steps: first, to obtain an abstract representation of the system as an hypergraph; then to search for the whole set of minimal over-determined subsets of equations; and finally to check if these sets of equations can be solved using local propagation alone.[3] These three stages are explained as follows:

1. Generating an abstract representation of the system model: The computation of the set of PCs needs an abstract representation of the system: an hypergraph (Murota, 2012). This type of representation only requires information about the constraints or equations in the model, and their relationship with the known and unknown variables in those equations. Thus, the hypergraph provides a representation of the set of equations from a structural point of view, i.e., it provides a structural model of the system.
2. Searching for the set of Minimal Evaluation Chains (MECs): In this step, a complete search algorithm finds the whole set of minimally overdetermined subsets (subhypergraphs) in the set of equations. Each subset is called a Minimal Evaluation Chain and represents a set of equations with exactly one more equation than unknown variables. But MECs do not have information about how the set of equations can be solved, if they can be solved at all, because they have no causality information.
3. Searching for the set of Minimal Evaluation Models (MEMs): this step works at the behavioral level in the model, by introducing causal information. In this stage, we consider all available information about causality for each constraint in a MEC. When a variable within a constraint can be solved assuming the rest of the variables are known, this is called an interpretation, i.e. a feasible causal assignment. For example, for the equation $a = b \times c$, there are three possible interpretations: $a := b \times c$, $b := a/c$, and $c := a/b$, if both b and c are different from 0. This stage searches for all the causally consistent interpretations for each constraint in a MEC, which is called the Minimal Evaluation Model. Each MEM represents a globally consistent causal assignment within a MEC and can be used to estimate the behavior of a part of the system. Each MEC can lead to none, one or multiple MEMs. One MEC can have no MEM if the system has no valid computational assignment (for instance, computing the flow in a pipe given pressure differences is feasible, but using the flows to estimate the pressure difference between both sides is almost impossible in real world applications). A Possible Conflict is a MEC which has at least one MEM.

   The results of the third stage provide not only the set of equations for each PC, but also a graphical model (directed hypergraph) which is the computational way to solve the set of equations (traversing the directed hypergraph from leaves to the root, which is the so-called discrepancy node, and also the variable used to compute the residual as the difference between their observed and estimated values).

[3] In the context of Consistency-based diagnosis, local propagation means solving one equation in one unknown in each computation step. This is imposed in order to be able to trace back which equations or component models were used to estimate a variable leading to a residual activation. If this is not possible, and several equations must be solved simultaneously, the set of equations is considered as one super-constraint, with its own correctness assumption, so the reasoning process is the same as before.

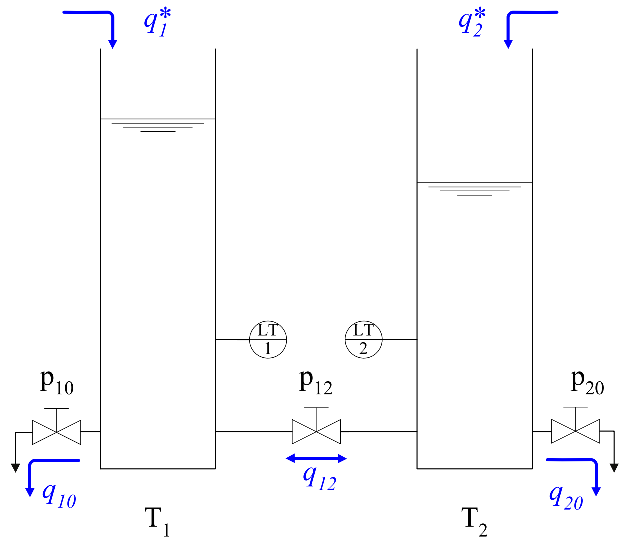

Fig. 1. Two tank system.

Software tools are available to compute the set of PCs[4] (Pulido et al., 2016) given a system model expressed as a set of Differential Algebraic Equations (DAEs), though other software tools do exist in the model-based diagnosis community to build similar executable models for different types of equations (Blanke and Lorentzen, 2006; Frisk et al., 2017).

The computational forms for each PC can be implemented in several ways: for instance, using the proposed steps directly as a simulation model, as in the DX approach (Pulido and Alonso-González, 2004), as a state-observer (Bregon et al., 2014), or as a particle-filter using a probabilistic model (Alonso-Gonzalez et al., 2011). However, they can also be used to build estimation models for fault identification (Bregon et al., 2012), or just to provide the structural model for building data-driven gray-box models (Pulido et al., 2019).

In the online stage, the computational model for each PC is fed with observations to produce an estimation, and consequently to compute a residual, which can be used for fault detection. If the residual is significantly different from zero, the *Possible Conflict* becomes a *minimal conflict* in the DX terminology, and the structural information – i.e., the correctness assumptions in DX terminology – can be used later to perform fault isolation. This stage is straightforward following Reiter's theory (Reiter, 1987): diagnosis candidates are obtained by computing the minimal hitting sets of the sets of equations in each PC. We illustrate this process in our case study next.

4. Generating Possible Conflicts for a case study

4.1. Case study

In order to illustrate the proposal of this paper, and later evaluate its performance in a real world situation, we use a laboratory plant that consists of two cylindrical tanks, T_1 and T_2, both with the same cross-sectional area. These tanks are connected by a cylindrical pipe with a manual valve. Fig. 1 shows the scheme of this plant.

[4] The tool for computing the set of PCs from an abstract representation of the system model and the software to perform fault diagnosis using PCs can be found at https://www.infor.uva.es/softwarepcs/

Tank levels are measured using two level sensors LT_1 and LT_2. The liquid is drained from the tanks T_1 and T_2 by means of two pipes, p_{10} and p_{20}, with flows q_{10} and q_{20}, respectively. The two tanks are connected by pipe p_{12}, whose flow is q_{12}. Both tank levels are controlled, but the control system is not considered in neither the plant model nor the fault detection technique. A detailed description of the hardware and software of this plant can be found in Fuente et al. (2008). In this system, we have considered typical faults, such as sensor faults, tank leakages, and clogged pipes. We deal with this issue later in this section.

This plant can be modeled using only Physics First Principles equations. The reader should notice that we provide only models of correct behavior for each equation. These models do not include additional (internal and non-observable) variables, which can be useful to model faulty behaviors, such as flows related to leakages in tanks, additional parameters to model potential pipe clogs, or potential bias in sensors. Additionally, our models obey the locality and no-function-in structure principles in Consistency-based Diagnosis (Reiter, 1987; Hamscher et al., 1992), and each equation is sensitive to exactly one fault.[5] Consequently, our behavior models are as follows.

First, the change in the level in each tank can be formulated according to the mass balances:

$$A\dot{h}_1(t) = q_1^*(t) - q_{12}(t) - q_{10}(t) \tag{eq_1}$$

$$A\dot{h}_2(t) = q_2^*(t) + q_{12}(t) - q_{20}(t) \tag{eq_2}$$

where A is the area of the cylindrical tanks ($A = 314$ cm^2), $h_1(t)$ and $h_2(t)$ are the liquid level in each tank ($\dot{h}(t)$ is the derivative), and $q_1^*(t)$ and $q_2^*(t)$ the input flow of each tank. Variables marked with an asterisk (*) are known variables, such as measured variables or known inputs. The maximum level of the tanks (h_{max}) is 36.3 cm.

Following Torricelli's law, flows q_{12}, q_{10} and q_{20} are calculated by:

$$q_{12}(t) = K_{12}\,\mathrm{sign}\left(h_1(t) - h_2(t)\right)\sqrt{2\,g\left|h_1(t) - h_2(t)\right|} \tag{eq_3}$$

$$q_{10}(t) = K_{10}\sqrt{2\,gh_1(t)} \tag{eq_4}$$

$$q_{20}(t) = K_{20}\sqrt{2\,gh_2(t)} \tag{eq_5}$$

where $K_{12} = 0.0499$ cm^2, $K_{10} = 0.097$ cm^2 and $K_{20} = 0.09665$ cm^2 are the cross sectional area of the pipes and g is the gravitational acceleration $g = 9.8$ m/s^2).

The observational model, which relates internal and measured variables, is given by the following equations:

$$h_1^*(t) = h_1(t) \tag{eq_6}$$

$$h_2^*(t) = h_2(t) \tag{eq_7}$$

where $h_1^*(t)$ and $h_2^*(t)$ are the sensor measurements corresponding to LT_1 and LT_2.

The relationship between the state variables ($\dot{h}_1$ and $\dot{h}_2$) and their derivatives is established through equations:

$$h_1(t+1) = \int \dot{h}_1(t)dt \tag{eq_8}$$

$$h_2(t+1) = \int \dot{h}_2(t)dt \tag{eq_9}$$

Flows q_1^* and q_2^*, provided by pumps P_1 and P_2, are considered as known inputs in this example. The model was fitted by estimating parameters K_{12}, K_{10} and K_{20} using real data collected from the plant. The process to fit these parameters was presented in García-Álvarez et al. (2011).

In order to test the approach presented in this work, a specific plant configuration has been set. It consists of the following behavior:

- **Tank 1** (T_1): the level reference is fixed at 30%.
- **Tank 2** (T_2): the level reference is fixed at 30% and then changed to 50% at time sample 800 s.

Both tanks are empty at the beginning of each experiment. The sampling time is 1 s and all performed experiments have a duration of 1600 s. This experiment allows the plant to run through 4 different states, two transient states and two steady states:

1. **Transient State 1**: both tanks are filled to reach the reference level.
2. **Steady State 1**: both tanks maintain the level at 30%.
3. **Transient State 2:** tank T_2 is filled to reach the reference 50%.
4. **Steady State 2**: T_1 maintains the level at 30% and T_2 maintains the level at 50%.

Fig. 2 shows the evolution of a nominal experiment considering the four states. Tank levels are expressed in percentages with respect to h_{max}.

Table 1
List of equations and related faults.

Equation	Variables	Component	Fault	Description
eq_1	$\{\dot{h}_1, q_1^*, q_{12}, q_{10}\}$	T_1	f_{T_1}	Leakage in tank T_1
eq_2	$\{\dot{h}_2, q_2^*, q_{12}, q_{20}\}$	T_2	f_{T_2}	Leakage in tank T_2
eq_3	$\{q_{12}, h_1, h_2\}$	p_{12}	$f_{p_{12}}$	Blockage of pipe/valve p_{12}
eq_4	$\{q_{10}, h_1\}$	p_{10}	$f_{p_{10}}$	Blockage of pipe/valve p_{10}
eq_5	$\{q_{20}, h_2\}$	p_{20}	$f_{p_{20}}$	Blockage of pipe/valve p_{20}
eq_6	$\{h_1, h_1^*\}$	LT_1	f_{LT_1}	Faulty sensor LT_1
eq_7	$\{h_2, h_2^*\}$	LT_2	f_{LT_2}	Faulty sensor LT_2
eq_8	$\{\dot{h}_1, h_1\}$	–	–	–
eq_9	$\{\dot{h}_2, h_2\}$	–	–	–

4.2. PCs for the case study

Regarding our case study, the relevant information for PCs computation can be found in Table 1: we only need to known the set of known and unknown variables for each equation, together with the physical component or the fault behavior which is being modeled. Additionally, Table 1 provides a brief description of the set of faults considered for the case study: leakages in both tanks, clogged pipes, and bias in both level sensors. As can be seen, each equation from eq_1 to eq_7 is linked to exactly one fault in the system. For instance, equations eq_1 and eq_2 could be used to model leakages in tanks T_1 and T_2 by merely subtracting the non-observable variables f_{leak_1} and f_{leak_2} to simulate the leakage flows f_{T_1} and f_{T_2}, respectively. In a similar way, the parameters K_{12}, K_{10}, K_{20}, would relate equations eq_3, eq_4, and eq_5 exactly to one fault: $f_{p_{12}}, f_{p_{10}}$, and $f_{p_{20}}$, respectively (just multiplying by the unknown faulty parameters $k_{f_{12}}, k_{f_{10}}, k_{f_{20}}$: if k_{f_i} is 1, there is no fault. If k_{f_i} moves closer to 0, we have a smaller or larger blockage). Finally, equations eq_6 and eq_7 can be used to model faults in the level sensors (bias) f_{LT_1} and f_{LT_2} in a similar way to leakages in equations eq_1 or eq_2. There is no faulty behavior modeled by equations eq_8, and eq_9, which only model the dynamic behavior of the system.

Using just the information in columns 1 and 2 in Table 1, the software used to compute the set of PCs follows the steps described in Section 3. We obtain 4 PCs that can be seen in Table 2, together with the set of faults related to these equations, the estimated variable for each PC, and the set of equations.

Fig. 3 provides the set of components, and the computational form for PC_1, which is represented by the directed hypergraph in the right

[5] This assumption could be removed in other systems. If the models have equations that can be related to more than one fault, we can still use them. Each PC containing those types of equations should be sensitive to those faults. This is neither relevant for PC calculation nor fault isolation. It would only be relevant for fault identification where we need to discriminate between those faults.

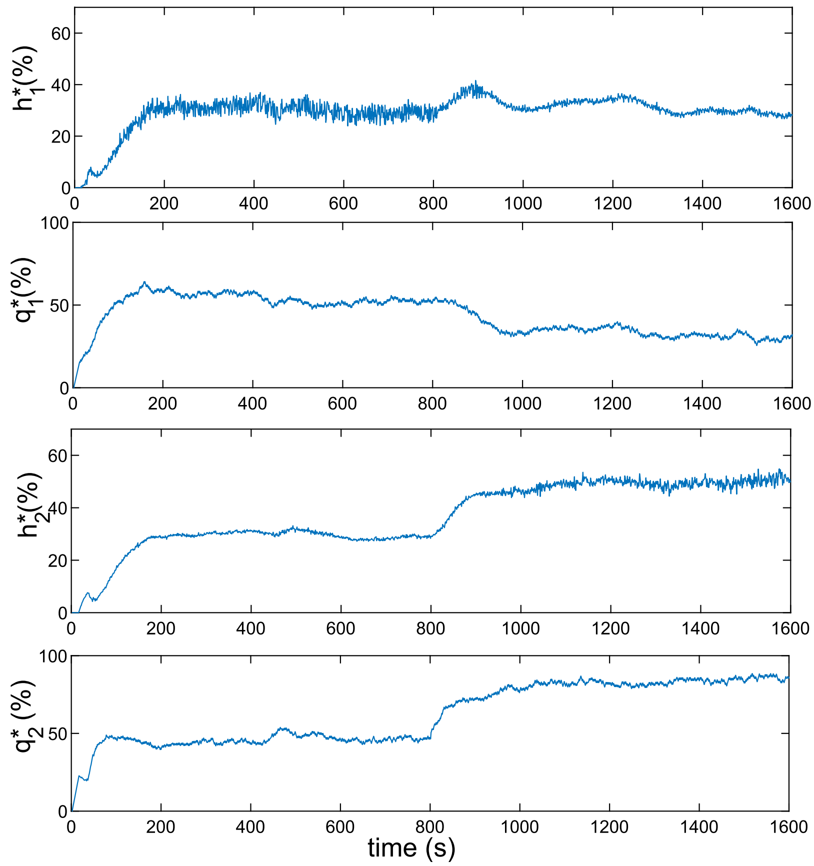

Fig. 2. Evolution of measured variables (h_1^* and h_2^*) and inputs (q_1^* and q_2^*).

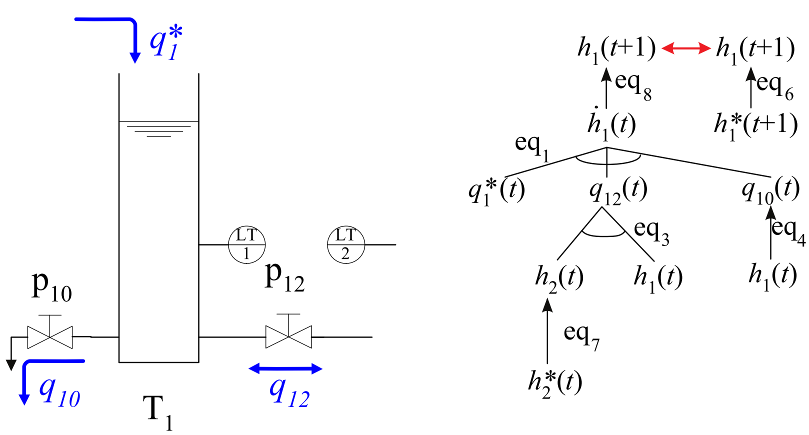

Fig. 3. Variables, measurements and components related to the directed hypergraph for PC_1.

Table 2
PCs and their related equations, estimated variables, and fault modes.

PC	Equations in the PC	Estimated variable	Faults in the PC						
			f_{T_1}	f_{T_2}	$f_{P_{12}}$	$f_{P_{10}}$	$f_{P_{20}}$	f_{LT_1}	f_{LT_2}
PC_1	$\{eq_1, eq_3, eq_4, eq_6, eq_7, eq_8\}$	h_1	1	0	1	1	0	1	1
PC_2	$\{eq_1, eq_2, eq_3, eq_4, eq_6, eq_8\}$	h_1	1	1	1	1	0	1	0
PC_3	$\{eq_2, eq_3, eq_5, eq_6, eq_7, eq_9\}$	h_2	0	1	1	0	1	1	1
PC_4	$\{eq_1, eq_2, eq_3, eq_5, eq_7, eq_8, eq_9\}$	h_2	1	1	1	0	1	0	1

hand side of the figure. Following the hypergraph, the equations are solved from the leaves to the root: h_1, which can be either measured, using LT_1, or estimated using the model. To estimate h_1, we need the value of its derivative, $\dot{h}_1$, which can be computed if q_1^*, q_{10}, and q_{12} are known. q_1^* is an input. q_{10} can be computed if h_1 is known. q_{12} needs h_2 (which is given by LT_2). The variable in the root is the only estimate for the PC, as expressed in Table 2. The difference between the real measurement and the estimation for that variable generates a residual signal for each PC. This residual could be used to perform fault detection.

As an example of the usage of PCs for FDII, let us assume that there is a fault in sensor LT_2, i.e., f_{LT_2} is present. Sooner or later, every PC related to f_{LT_2} will be activated. Let us assume that the first PC to be activated is PC_4. The CBD approach using PCs proceeds by analyzing Table 2 row-wise: faults related to the equations in the PC ($\{eq_1, eq_2, eq_3, eq_5, eq_7, eq_8, eq_9\}$) can be responsible for the fault.

Following CBD minimal conflicts are computed as minimal hitting sets from the sets of PCs. Hence, since there is no previous fault candidates the minimal hitting set for these equations provides the set of single fault candidates: $[[f_{T_1}], [f_{T_2}], [f_{p_{12}}], [f_{p_{20}}], [f_{LT_2}]]$.

If PC_1 is activated later, it would signal to the following equations as candidates for the fault: {eq_1, eq_3, eq_4, eq_6, eq_7, eq_8}. These equations are related to single fault candidates $[[f_{T_1}], [f_{p_{12}}], [f_{p_{10}}], [f_{LT_1}], [f_{L_2}]]$ However, this that must be consistent with the previous activation of PC_4. Hence, the set of consistent faults will be related to the minimal-hitting sets of the equations of both PCs: [[eq_1], [eq_3], [eq_7], [eq_2,eq_4], [eq_2,eq_6], [eq_4,eq_5], [eq_5,eq_6]]. These equations correspond to 3 single faults: $[[f_{T_1}], [f_{P_{12}}], [f_{LT_2}]$, and 4 double faults: $[f_{T_2}, f_{LT_1}], [f_{p_{10}}, f_{p_{20}}], [f_{p_{20}}, f_{LT_1}], [f_{T_2}, f_{p_{10}}]]$

Finally, when PC_3 is activated, we follow the same procedure, computing new minimal-hitting sets for the three PCs. The set of equations in the minimal-hitting sets are: [[eq_3], [eq_7], [eq_1,eq_2], [eq_1,eq_5], [eq_1,eq_6], [eq_2,eq_4], [eq_2,eq_6], [eq_4,eq_5], [eq_5,eq_6]]. The corresponding set of fault modes are: $[[f_{p_{12}}], [f_{LT_2}], [f_{T_1}, f_{T_2}], [f_{T_1}, f_{p_{20}}], [f_{T_1}, f_{LT_1}], [f_{T_2}, f_{p_{10}}], [f_{T_2}, f_{LT_1}], [f_{p_{10}}, f_{p_{20}}], [f_{p_{20}}, f_{LT_1}]]$.

As the reader can see, even without information about the effect of the fault modes, the fault in sensor LT_2 is among the potential single fault candidates. Usually, a fault identification stage is included in the CBD approach to help refine further the set of fault modes consistent with the observations. That is, however, beyond the scope of this work.

It must also be mentioned that the final result of single and double (or multiple) faults is the same, whatever the order of activation for the PCs. The procedure is automatic, and there is no assumption about the number of double or multiple faults. All of them are computed.

5. Integration approach and discussion

PCA-based fault detection techniques offer many advantages. These methods are easy and straightforward to implement and only require a good database containing the main trends in the system, regardless of whether the data are highly correlated. In addition, the detection task has a low computational cost, and can be completed by identifying which measured variables are directly or indirectly related to the fault thanks to contribution analysis.

However, as mentioned in previous sections these methods have some weaknesses too. Regarding fault isolation, classical PCA approaches do not provide tools to isolate faulty candidates when a fault is detected in the process. There are several authors who have proposed solutions for this flaw. Gertler et al. (1999) uses structured partial PCA models with the same isolation properties as the parity relations. Using that equivalence, Gertler et al. (1999) and Huang et al. (2000) decompose the original PCA model into smaller structured PCA models that guarantee the disturbance decoupling for the set of faults considered.

Another flaw that can be remarked is the use of classical PCA approaches in processes with several operation points. In general, these methods are used in processes without many operation points (García-Álvarez et al., 2012). When dealing with processes with several operation points, Hwang and Han (1999) and Tien et al. (2004) classify the different solutions into three different categories:

1. Building a PCA model for each operation point: This family of methods is known as Multiple PCA (MPCA) or MPLS (Multiple Projection to Latent Structures) in Zhao et al. (2004, 2006), where the authors propose using principal angles to distinguish the operating modes. Solutions falling into this category can be compromised if the process has many operation points, because they will need to fit a PCA model for each one of them. In addition, the complex task of monitoring transient states must be considered. A method to deal with transient states using PCA can be found in García-Álvarez et al. (2012) which is similar to the procedure used by PCA-based techniques for monitoring batch process (Nomikos and MacGregor, 1995; Zarzo and Ferrer, 2004). These methods will require additional PCA models for each of the transient states.
2. Updating the model to reflect the changes in the operation points: this category aims to adapt the PCA model to the current operation point. This family of methods can include such configurations as Exponentially Weighted PCA (EWPCA) (Lane et al., 2003), (Portnoy et al., 2016), Adaptive PCA (APCA) (Bakdi and Kouadri, 2017), Moving Window PCA (Rafferty et al., 2016), Dynamic PCA (DPCA) (De Ketelaere et al., 2015), and Recursive PCA (RPCA) (Rezamand et al., 2020). These proposals require the *on-line* reformulation of the PCA model with every change detection in the operation point. The fault detection system must detect the change in the current operating point to avoid its interpretation as a fault and reformulate the PCA model to capture the trends of the new point.
3. Developing a conventional PCA model to capture all such changes. One of the main drawbacks of methods of this category is that PCA could identify transient states between operation points as faults due to their non-linear behavior. Furthermore, if the number of operation points is high, the PCA model can capture an excessive number of tendencies. If the PCA model is fitted considering the tendencies of each case, the PCA monitoring technique can model too many tendencies in the nominal case, and the PCA-based fault detection method could exhibit poor detectability due to the difficulty to distinguish between faults and operation points.

Finally, there are hybrid solutions that combine data-driven and model-based approaches. These methods are capable of generating and/or processing residual signals obtained by using model-based fault detection. For instance, the work presented in Jiang et al. (2021), which proposes constructing optimized residual generators; Jung and Frisk (2018), where a residual selection approach is proposed using structural information and training data from different fault scenarios; and Jung and Sundström (2019), where a comparative study of model-based, data-driven and hybrid techniques in fault detection and diagnosis tasks is presented.

The integration approach presented in this work fits within this last group of techniques, because we propose to use a PCA model to track not the directed measurements from the system, but the residuals generated by PCs. This approach is based on two basic ideas: on the one hand, PCA-based fault detection approaches are straightforward to implement and the on-line detection presents a low computational cost. Using PCA, the process can be monitored by simply looking at the T^2 and Q statistics. On the other hand, the PCs technique goes one step further in the fault detection and isolation tasks by looking at the triggered residual signals.

5.1. How the integration approach works

Fig. 4 presents a graphical scheme of the integration approach presented in this work. This scheme can be divided into two main procedures. First, the *off-line* procedure, where the different models required for the detection and diagnosis tasks are established. This procedure is represented at the bottom of this figure in blue boxes. Second the *on-line* procedure, where the fault detection and isolation tasks are run. This process is represented at the top of this figure in orange boxes.

The ***off-line*** **procedure** consists of the following tasks:

1. Obtain two abstract descriptions from the process model (the non-directed hypergraph to build the set of MECs, and the directed hypergraph to build the set of MEMs for the set of MECs).
2. Compute the set of PCs (those MECs with at least one MEM) and build their computational models.
3. Analyze the fault information in each PC to build the theoretical Fault Signature Matrix (FSM).

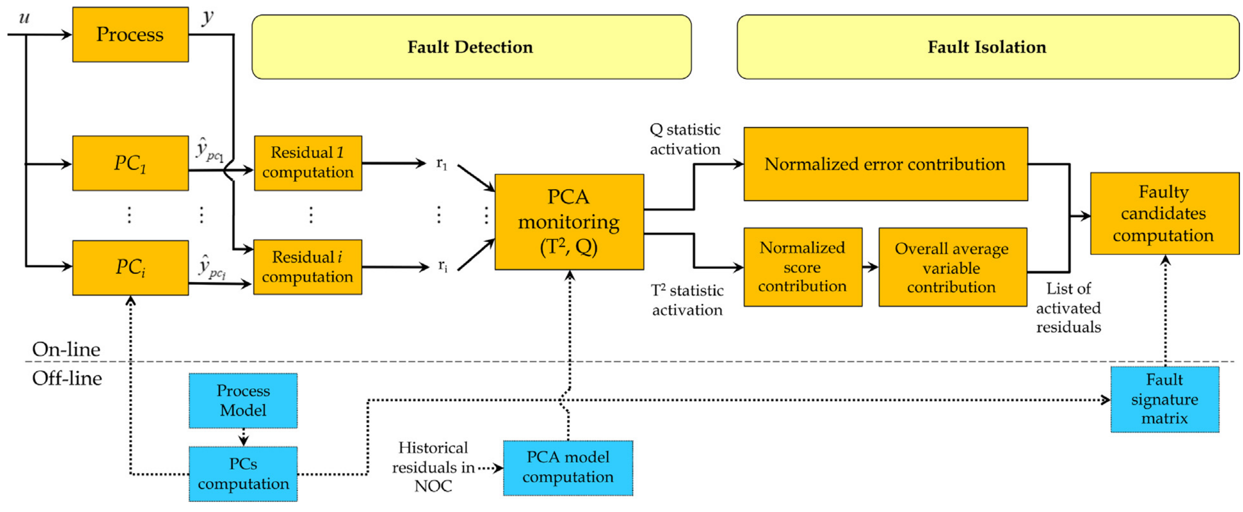

Fig. 4. Integration proposal. At the bottom of the figure, the off-line computation modules are represented in blue. At the top, the different modules which are run during the on-line process monitoring are shown in orange. The fault detection and fault isolation tasks are marked in the scheme in yellow boxes. Blocks *Normalized error contribution*, *Normalized score contribution* and *Overall average variable contribution* are computed through the contribution analysis explained in Section 2.

4. Fit the PCA model using the historical values of residual signals for the PCs using NOC data.

The ***on-line* procedure** consists of the following tasks:

1. Initialize the set of active fault candidates to empty.
2. While there are new observations available and there is at least one PC not activated do
   {Fault detection:}
   (a) For each PC_i do
   Compute its residual signal r_i using available observations.
   (b) Monitor the calculated residual signals by means of the T^2 and Q statistics.
   (c) If either the T^2 or Q statistics exceed their thresholds for a certain number of consecutive samples then confirm the activation of PC_i.
      - If the Q statistic is responsible for the detection, then
        compute the residual contribution using the Bar plot of normalized errors of the variables (See Section 2) and
        mark PCs with high related residuals as activated PCs.
      - If the T^2 statistic is responsible for the detection then
        compute the residual contribution using the Bar plot of normalized scores and the Overall average variable contribution of the highest scores (See Section 2) and
        mark PCs with high related residuals as activated PCs.

   {Fault isolation:}
   (d) If there is a new activated PC then:
   update the set of active fault candidates (analyzing the theoretical FSM row-wise and computing the minimal hitting-set).

The reader should notice that the activation of either the Q or the T^2 statistic will activate the corresponding PCs, because we are following the consistency-based diagnosis approach to find the fault candidates, and these fault candidates must be consistent with the whole set of activated conflicts.

5.2. Discussion

If we compare this proposal with those mentioned at the beginning of this section we can point out several relevant differences and some novelties:

- Considering the PCA-based methods mentioned above, this proposal does not need to fit different PCA models for different operation points as the MPCA-based methods (Zhao et al., 2004, 2006) propose. Related to PCA approaches that are able to adapt to the current operation point (Lane et al., 2003; Portnoy et al., 2016; Bakdi and Kouadri, 2017; Rafferty et al., 2016; De Ketelaere et al., 2015; Rezamand et al., 2020), the main difference is that we only require correct NOC data and we only need to build one PCA model, offline, instead of recomputing the model or adapting the monitoring task online when there is a change in the operation point.
- Regarding techniques which adapt the PCA model to new operation points, such as EWPCA (Lane et al., 2003; Portnoy et al., 2016), DPCA (De Ketelaere et al., 2015), Moving Window PCA (Rafferty et al., 2016) or RPCA (Rezamand et al., 2020), the main difference is again that we need only to fit one PCA model offline, using the NOC data concerning normal behavior. Since the PCs will track the changes in the operation point and the PCA model does not need to be aware of that, we do not need to compute the model again every time there is a change in the operation point.
- Regarding the set of approaches that combine data-driven models and model-based diagnosis (using mainly first principles or other analytical models): we use only one PCA model that analyzes residual signals for the whole system, instead of using Partial PCAs, created for structured residuals that are only sensitive to some faults, to direct measurements from the system or other derived signals (Gertler et al., 1999). Additionally, our proposal relies exclusively upon correct behavior models; thus, we only need correct behavior data and we do not need faulty behavior data (Jung and Frisk, 2018; Jung and Sundström, 2019). At the same time, using PCs we can track different operation points without considering different PCA models for each one of them. Regarding the work by Jiang et al. (2021), our proposal does not propose an optimization method to select the most sensitive residuals provided by parity relations (similar in many ways to ARRs). We use the residuals based on minimality with respect to the available model and use another statistical model, PCA. Last, but not least, PCs are used for fault isolation using the CBD

principles, which are different and more sound with respect to the fault isolation methods used by the FDI community (Cordier et al., 2004).

- Finally, we have tested the behavior of traditional Artificial Neural Networks (ANN) solutions (Multilayer perceptron) training two Neural Networks (NN) in cascade configuration, one for fault detection (that must avoid false positives due to operation point changes) and another one for fault isolation (capable to distinguish between two sensor faults). Samples are introduced into the first NN for fault detection. If a detection is established, the sample is processed by the second NN, which provides fault isolation information. Both NN solve a classification problem. We cope with the system dynamics using a semi-dynamical architecture, which requires feeding the network with current and past data for each variable, using a sliding window approach. We selected 80% of the data for training the models. The remaining 20% for testing the models. We made a stratified random partition, to ensure that both the training and test sets had at least one sample of each type of failure.
  We used a 5-folds stratified cross validation process on the training set to select the best ANN model. We tried different architectures (number and size of layers) epochs and batch sizes and build the final model with the best parameters with the full training set. The final model has 60 elements in the input layer, a hidden layer with 40 units and an output layer with one element. Experimental results shown that in the nominal experiments of the test dataset, the detection NN generate 8 false positives, which is not competitive with the PCA-PCs approach. The NN is sensitive to the first operation mode change, while the PCA+PCs is not.

6. Results

In this section, we illustrate how the proposed methodology can be used in our case study. First, we show how, by using PCs, we can obtain residuals from NOC historical data and compute a PCA model which drastically reduces the number of false positives in fault detection. We then show how this PCA model, built completely offline, can be used online to detect faults and successfully provide fault isolation results using structural information from the PCs with activated residuals.

6.1. Nominal case

In order to study the results obtained using the method proposed in this work, two PCA models were fitted. The first one consisted of a classical PCA approach. In this case, the PCA model was fitted using the measured variables (h_1^* and h_2^*) and the inputs (q_1^* and q_2^*) collected from 10 real experiments in NOC. Each experiment included the four described states (transient and steady). The PCA model was fitted with two principal components. The detection thresholds for the PCA model were established with a level of significance $\alpha = 95\%$. This configuration is denominated the PCA method in this work.

The second one was fitted following the integration approach presented in this paper. The PCA model of the integration approach, from now on the PCA+PCs method, was fitted using the data of the residual signals (r_1, r_2, r_3 and r_4) of each PC calculated for the same 10 real experiments used in the PCA method. This PCA model was also fitted with two principal components and the detection thresholds were also established with the same level of significance $\alpha = 95\%$. The PC model simulation and the PCA computation were developed with Matlab© and Simulink©.

It is interesting to notice the behavior of the residual signals (r_1, r_2, r_3, and r_4) related to each identified PC in NOC in Fig. 5. These residual signals were computed using the data of the experiment shown in Fig. 2. As Fig. 5 shows, the evolution of the different residuals can be affected by the current operation point or transient state. In this case, the deviation of the residual signals could be due to the fact that the estimated model parameters are more accurate in linear operation points than in transitory states, whose inner relationships between variables are typically non linear.

Table 3
Mean value and standard deviation of the alarms percentage obtained for 20 experiments under nominal conditions.

#	Alarm percentage	
	PCA	PC+PCA
1	7.37	2.06
2	7.80	3.25
3	7.06	1.69
4	6.99	0.68
5	7.93	3.44
6	7.06	3.43
7	7.25	1.12
8	7.81	1.37
9	7.93	1.99
10	7.62	1.87
11	8.62	3.74
12	8.06	0.74
13	7.62	1.18
14	7.99	1.18
15	7.75	1.99
16	7.43	3.31
17	8.81	1.18
18	7.62	1.37
19	7.81	2.18
20	8.50	2.43
Mean	7.75	2.01
Std	0.50	0.96
ANOVA data		
F-ratio	558.71	
P-value	0.000001	

The first contribution of this methodology can be seen by studying the behavior of the system in the nominal situation. Considering the monitoring and fault detection tasks, an alarm can be defined as a time sample where the value of T^2 or Q gives a value greater than its related threshold. In this context, one alarm is not enough to detect a fault; several alarms above the fixed threshold or a given ratio of alarms in a given period of time are two methods to determine a fault detection. In this work, the number of consecutive alarms that must occur for a fault to be detected has been empirically set to 15. When a fault is detected in a non-faulty situation, it is said that a false positive (FP) has been detected. In the MSPC context, it is normal to have alarms in the different control charts. However reducing the number of alarms can be a positive measure from the point of view of the cognitive overload of plant operators.

Table 3 shows the percentage of alarms detected with PCA and PCA+PCs methods. Looking at Table 3, the reduction in the number of alarms can be clearly seen when the PCA+PCs is used. After performing an analysis of variance (ANOVA) for comparing both mean values, the *P-value* of the *F-ratio* is lower than 0.05, so there exists a statistically significant difference between the means of the methods at the 95.0% confidence level.

Regarding the false positive detection, when the PCA method was applied, a false positive was detected in transitory state 1 in each of the 20 experiments studied. With the PCs alone (just paired with a z-test to identify significant deviations of the residual signals), at least one false positive was triggered in each experiment. However, when the PCA+PCs method is applied, no false positive was detected in any of the experiments. Table 4 shows a summary of the number of false positives detected by the PCA, PCs and PCA + PCs methods using the same NOC experiments as in Table 3.

Fig. 6 shows the evolution of the control charts based on the T^2 and Q statistics for the first experiment of Table 3. In this figure, it can be clearly seen that the start-up (when both tanks are being filled) is detected as a fault by both statistics when the PCA method is applied (Fig. 6(a)). However, when the PCA+PCs method is used, no fault is detected (Fig. 6(b)).

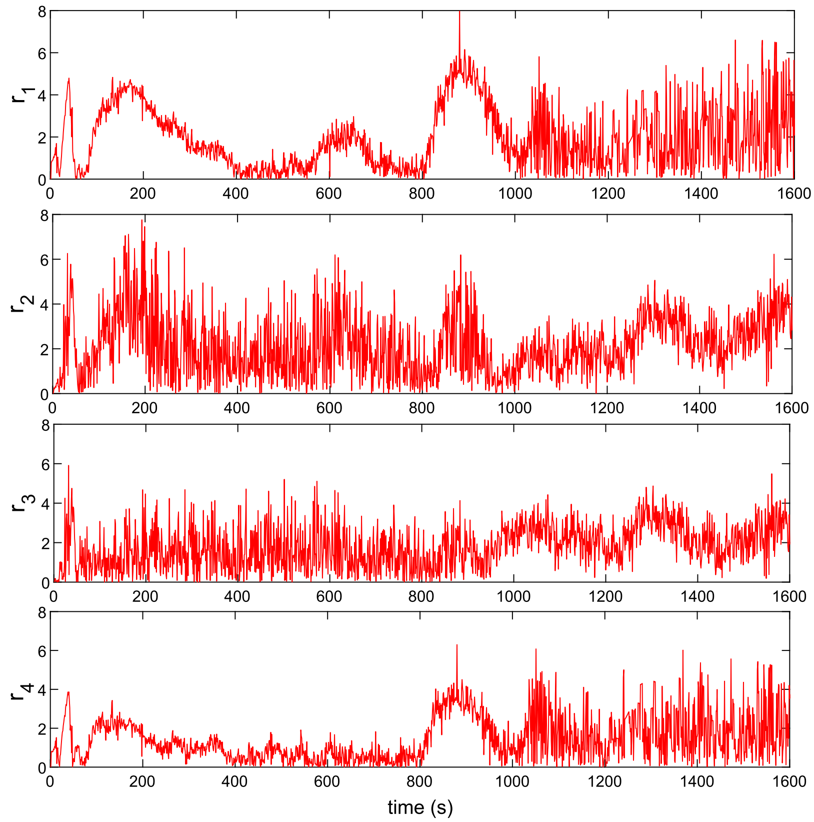

Fig. 5. Evolution of the residual signals (r_1, r_2, r_3 and r_4) related to each PC identified.

Table 4
False positives detected in 20 experiments under nominal conditions.

#	False positives				
	PCA		PCs		PCA+PCs
	Number of FP	Detection instant	Number of FP	First detection instant	Number of FP
1	1	15	3	479	0
2	1	15	4	804	0
3	1	15	4	854	0
4	1	15	3	802	0
5	1	15	3	776	0
6	1	15	4	472	0
7	1	15	4	647	0
8	1	15	2	692	0
9	1	15	3	759	0
10	1	15	3	581	0
11	1	15	1	1146	0
12	1	15	3	869	0
13	1	15	3	645	0
14	1	15	4	854	0
15	1	15	4	489	0
16	1	15	3	843	0
17	1	15	4	655	0
18	1	15	3	857	0
19	1	15	3	436	0
20	1	15	2	826	0

Table 5
Results for faulty situations in level sensor LT_1 when PCA is applied to PC residuals.

Statistic	T^2		Q	
Fault instant	Detection	Isolation	Detection	Isolation
40% fault size				
100	120	195	178	
500	515	521	557	
850	915	961		
1300	1315	1661	1364	
60% fault size				
100	116	193	176	
500	515	541	535	535
850	915	941	954	
1300	1315	1381	1379	

6.2. Fault detection and isolation

The second contribution of this methodology can be seen when we study fault detection and isolation performance in the presence of faults. For this propose, two additive faults were introduced in the level sensors LT_1 and LT_2. The fault sizes were 40% and 60%. These faults were triggered in four different situations, considering transitory and steady states:

- $t = 100$: Transitory state 1.
- $t = 500$: Steady state 1.
- $t = 850$: Transitory state 2.
- $t = 1300$: Steady state 2.

Tables 5 and 6 show the detection times using the T^2 and Q statistics (T^2 detection time and Q detection time), as well as the time instant after the fault detection when the approach is able to uniquely isolate the fault.

Considering the results presented in Tables 5 and 6, the approach proposed in this work is able to detect every considered fault. In the case of fault isolation, the isolation capability of the T^2 statistic was successful for all the experiments. Tables 5 and 6 only show an entry when faults are unequivocally isolated.

Figs. 7, 8, 9 and 10 give a graphical view of the fault detection and isolation behavior of a 40% fault in sensor LT_2 at time sample $t = 1300$ (steady state 2). Fig. 7 plots the time evolution of the four residuals

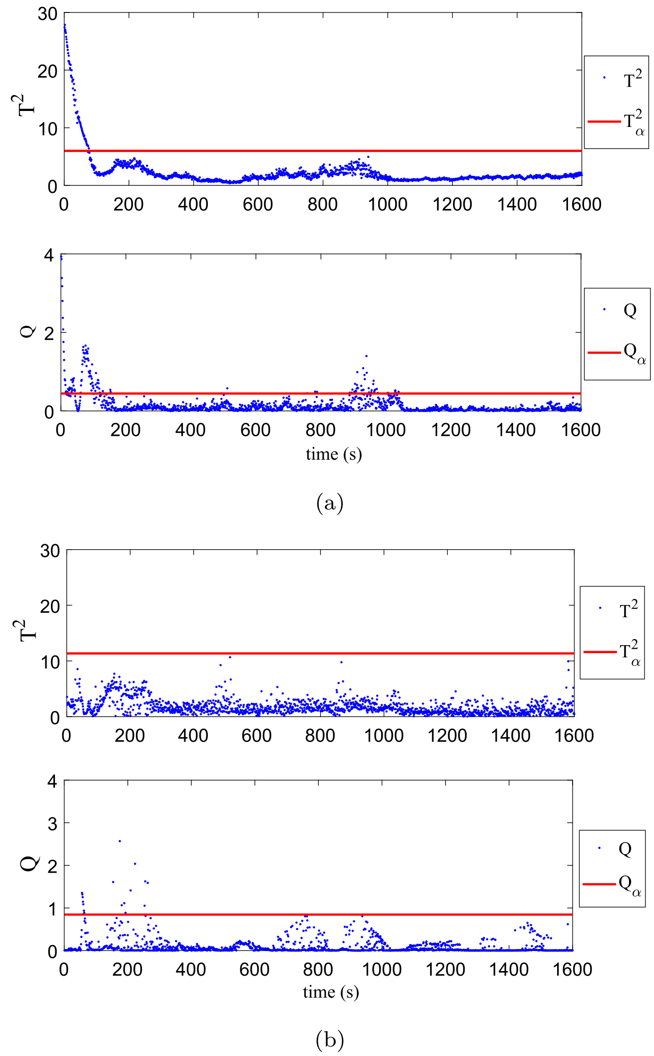

Fig. 6. Control charts for the T^2 and Q statistics in a nominal experiment monitored with the PCA method (a) and the PCA+PCs method (b).

Table 6
Results for faulty situations in level sensor LT_2 when PCA is applied to PC residuals.

Statistic	T^2		Q	
Fault instant	Detection	Isolation	Detection	Isolation
40% fault size				
100	115	161	1034	1020
500	515	521	1114	1140
850	915	961	983	
1300	1315	1361		
60% fault size				
100	115	141	174	
500	515	521	538	524
850	915	931	967	
1300	1315	1331	1476	1482

under the influence of the mentioned fault. The monitoring of these residual signals by means of the T^2 and Q monitoring statistics is given in Fig. 9, where the monitoring plots were able to detect the fault. Looking at this figure, it is clear that the T^2 based control chart detected the fault earlier than the Q chart. Fig. 8 shows the evolution of the consecutive alarms detected by the T^2 statistic and the fault detection, i.e., when 15 consecutive alarms are detected.

Additionally, and considering that the T^2 statistic was faster in the fault detection task, this statistic can be analyzed in order to

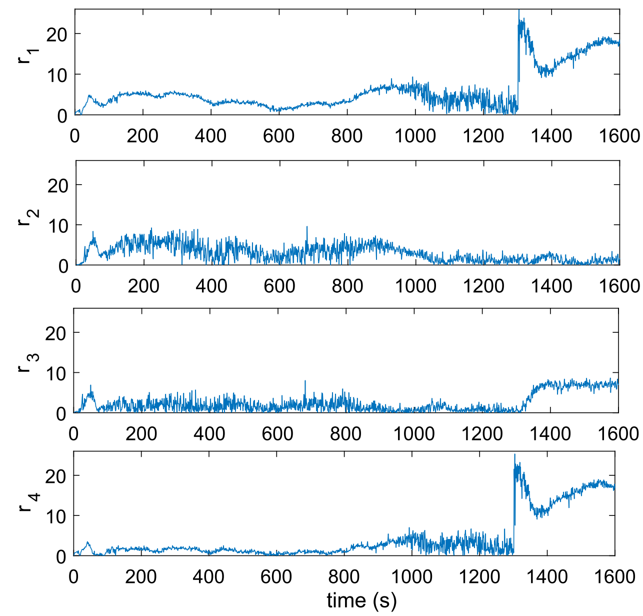

Fig. 7. Possible Conflict residuals for a 40% fault in sensor LT_2 at time sample $t = 1300$.

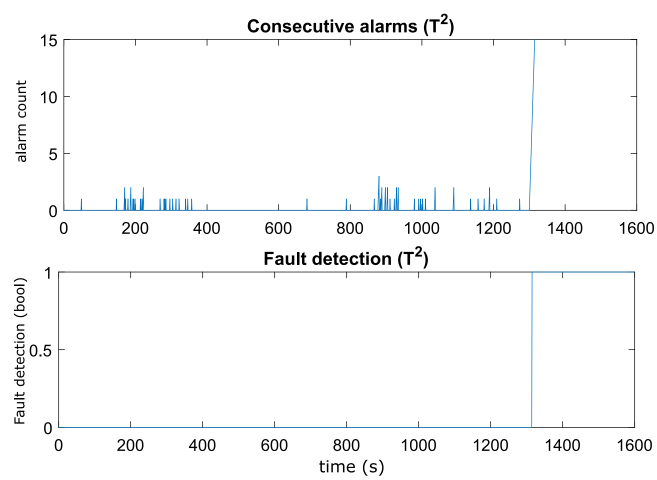

Fig. 8. Evolution of the consecutive alarms related to the monitoring of Fig. 7 and fault detection.

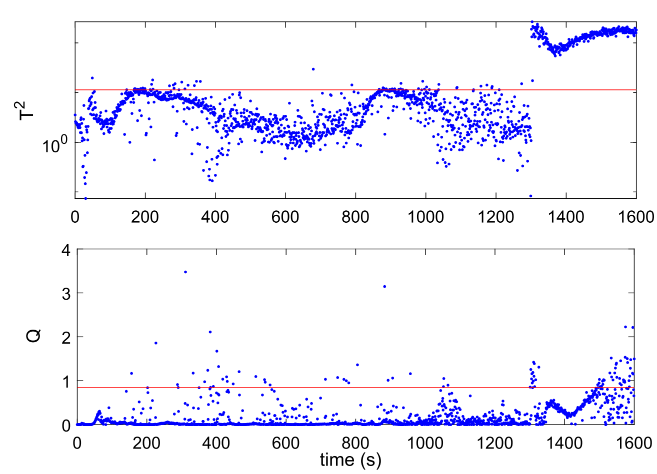

Fig. 9. Plots of the T^2 and Q statistics for the faulty experiment of Fig. 7 (logarithmic scale for the ordinate axis in T^2 statistic).

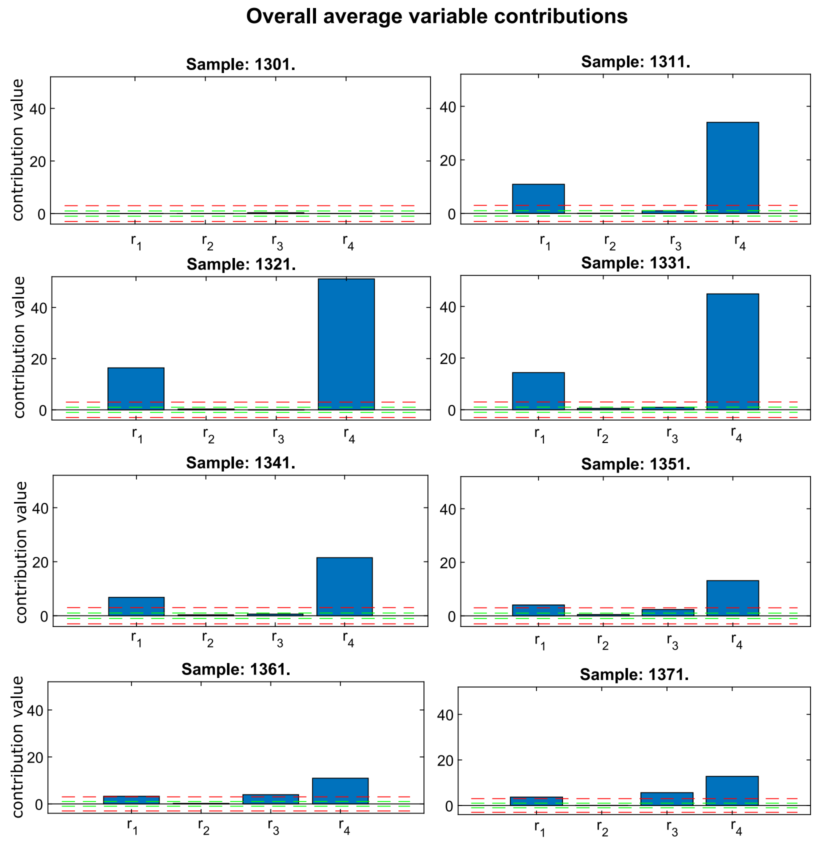

Fig. 10. Overall average variable contributions to the scores plot for the T^2 statistic for the faulty situation detected in Fig. 9. Green lines represent the standard deviation and red lines three times the standard deviations.

isolate the detected fault. When a fault is detected with the T^2 control chart, analyzing the normalized scores allows the principal components related to the fault to be found. However, in the procedure presented in this work, knowing the variables related to the detected fault is the key piece of information, because these variables are the PC residual signals that can be used to isolate the fault by analyzing the fault signature matrix. The plot of the overall average variable contributions to the scores with the highest value is shown in Fig. 10. This Figure shows the evolution of the variable contribution at samples 1301 s, 1311 s, 1321 s, 1331 s, 1341 s, 1351 s, 1361 s and 1371 s for the detected fault.

Looking at Fig. 10, it can be noted that when the fault is detected, no residual signal contribution shows an abnormal value, so no fault can be isolated at that sample time. At time samples 1311 s, 1321 s, 1331 s, 1341 s and 1351 s, it is clear that residual signals of PC_1 and PC_4 are related to the detected fault.

After computing the minimal hitting set for PC_1 and PC_4, we have 3 single fault candidates $[[f_{T_1}], [f_{p_{12}}], [f_{LT_2}]]$ and multiple double faults. When the contribution analysis is computed at time sample 1361 and later, it can be seen that the contribution of the residual signal PC_3 begin to show an abnormal value and that abnormal behavior is stable over time. So, the unique individual candidate in the fault signature matrix is a fault in the level sensor LT_2.

7. Conclusions

In this work, we have presented an integration approach of PCA models which is a MSPC technique, with CBD using PCs, which is a model based diagnosis approach.

The main advantage of PCA is that it does not need a model of the process and still provides a robust fault detection method when the process is working in a given stationary state. Its main disadvantage is that the basic PCA approach does not work well in different operations points, it cannot track the process in transitory states and it does not provide fault isolation capabilities.

The main advantage of CBD with PCs is that it provides fault detection and isolation capabilities even in transient states. Their main disadvantage is that PCs require detailed and accurate models of the process to track the system robustly. In the presence of noise, model uncertainties and nonlinearities, their behavior degrades significantly.

The integration approach presented in this paper has the potential to reduce the limitations of both methods, without introducing modifications to their basic formulation, while retaining their main advantages.

From the PCA point of view, the improvements from the integration are twofold. On the one hand, the solution improves the fault detection capabilities of the PCA models in situations with multiple operation points and transient states. On the other, it provides fault isolation capabilities.

From the PCs point of view, integration provides robustness against noisy measurements, model uncertainties and nonlinearities. In these circumstances, residuals are noisy and deviate from zero in case of input disturbances and operating point transients. This can lead to a high number of false positives and false negatives in fault detection and inaccurate fault isolation. The application of PCA to the residuals has the ability to filter out residual deviations related to the aforementioned causes, and which occur under NOC, allowing the detection of those deviations related to the presence of faults, which profoundly alter the correlations between residual values.

The main difference between our integration proposal and those related to Multiple PCA (Zhao et al., 2004, 2006; García-Álvarez et al.,

2012) is that we do not need to fit multiple PCA models for each operation point, because we only monitor the value of the PCs residuals, which absorb the changes due to operation point changes.

Regarding techniques which adapt the PCA model to new operation points, such as EWPCA (Lane et al., 2003; Portnoy et al., 2016), DPCA (De Ketelaere et al., 2015), Moving Window PCA (Rafferty et al., 2016) or RPCA (Rezamand et al., 2020), the main difference is again that we only need to fit one PCA model offline, using the NOC data concerning normal behavior. We do not need compute the model every time there is a change in the operation point.

Finally, regarding the set of approaches that combine data-driven models and model-based diagnosis (using physics first principles or other analytical models), such as Gertler et al. (1999), Jiang et al. (2021), Jung and Frisk (2018) and Jung and Sundström (2019), our proposal relies exclusively upon correct behavior models. Thus we only need correct behavior data avoiding the requirement of collecting data from faulty behavior.

Summarizing, the main contributions and novelties of this approach are:

- It is able to detect and isolate any type of failure, because we do not assume any predefined list of failures. This is due to the fact that Consistency-based Diagnosis with Possible Conflicts only uses correct behavioral models. These models are easier to develop than faulty behavior models, which often require the imposition of several simplifying and unrealistic assumptions.
- It does not require faulty behavioral data because PCA models can be effectively trained with just correct behavioral data, then used to perform robust fault detection on residual signals, whose value in the absence of faults is theoretically zero. This is a great advantage over conventional data-driven methods because faulty behavioral data are scarce, while correct behavioral data are readily available. Consequently, the system can be trained with a relatively small amount of easily obtainable data.
- Experimental results show that, for the case study, the proposal is able to perform accurate and timely fault detection and isolation in different operating regions, and contrary to other techniques, without the need of dedicated PCA models for each operating region.
- Finally, but far more important, for the case study, the proposal is able to track the system during transients to different regions of operation without detecting false alarms.

To the best of our knowledge, there is no similar work in the literature that addresses the problem with such a simple solution.

CRediT authorship contribution statement

D. Garcia-Alvarez: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **A. Bregon:** Conceptualization, Validation, Methodology, Software, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition. **B. Pulido:** Methodology, Writing – Original Writing – original draft, Writing – review & editing, Supervision. **C.J. Alonso-Gonzalez:** Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Alonso-Gonzalez, C., Moya, N., Biswas, G., 2011. Dynamic Bayesian Network factors from Possible Conflicts for continuous system diagnosis. In: Conference of the Spanish Association for Artificial Intelligence. Springer, pp. 223–232.

Armengol, J., Bregon, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., Olive, X., Pulido, B., Travé-Massuyès, L., 2009. Minimal structurally overdetermined sets for residual generation: A comparison of alternative approaches. IFAC Proc. Vol. 42 (8), 1480–1485.

Bakdi, A., Kouadri, A., 2017. A new adaptive PCA based thresholding scheme for fault detection in complex systems. Chemometr. Intell. Lab. Syst. 162, 83–93.

Banguero, E., Correcher, A., Pérez-Navarro, Á., García, E., Aristizabal, A., 2020. Diagnosis of a battery energy storage system based on principal component analysis. Renew. Energy 146, 2438–2449. http://dx.doi.org/10.1016/j.renene.2019.08.064.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2006. Diagnosis and Fault-Tolerant Control. Springer.

Blanke, M., Lorentzen, T., 2006. SATOOL-A software tool for structural analysis of complex automation systems 1, 2. IFAC Proc. Vol. 39 (13), 629–634.

Bregon, A., Alonso-González, C.J., Pulido, B., 2014. Integration of simulation and state observers for online fault detection of nonlinear continuous systems. IEEE Trans. Syst., Man, Cybern.: Syst. 44 (12), 1553–1568. http://dx.doi.org/10.1109/TSMC.2014.2322581.

Bregon, A., Alonso-González, C.J., Pulido, B., 2016. Faster and more accurate FDI for hybrid systems using hybrid possible conflicts. In: 2016 3rd Conference on Control and Fault-Tolerant Systems. SysTol, IEEE, pp. 499–504.

Bregon, A., Biswas, G., Pulido, B., 2012. A decomposition method for nonlinear parameter estimation in TRANSCEND. IEEE Trans. Syst., Man, Cybern. - Part A: Syst. Hum. 42 (3), 751–763. http://dx.doi.org/10.1109/TSMCA.2011.2170065.

Bro, R., Kjeldahl, K., Smilde, A.K., Kiers, H.A., 2008. Cross-validation of component models: A critical look at current methods. Anal. Bioanal. Chem. 390 (5), 1241–1251.

Camacho, J., 2007. New Methods Based on the Projection to Latent Structures for Monitoring, Prediction and Optimization of Batch Processes (Ph.D. thesis). Universidad Politécnica de Valencia. Spain.

Camacho, J., Picó, J., Ferrer, A., 2009. The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter? Anal. Chim. Acta 642 (1–2), 59–68.

Camacho, J., Picó, J., Ferrer, A., 2010. Cross-validation in Principal Component Analysis: Searching for the 'best' approach. In: Chemometrics in Analytical Chemistry. Antwerp (Belgium).

Chiang, L.H., Russell, E.L., Braatz, R.D., 2001. Fault Detection and Diagnosis in Industrial Systems. Springer, Nueva York.

Cordier, M.-O., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., Travé-Massuyès, L., 2004. Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. IEEE Trans. Syst. Man Cybern. B 34 (5), 2163–2177.

De Ketelaere, B., Hubert, M., Schmitt, E., 2015. Overview of PCA-based statistical process-monitoring methods for time-dependent, high-dimensional data. J. Qual. Technol. 47, 318–335.

Eastment, H.T., Krzanowski, W.J., 1982. Cross-validatory choice of the number of components from a principal component analysis. Technometrics 24 (1), 73–77.

Ferrer, A., 2007. Multivariate statistical process control based on principal component analysis (MSPC-PCA): Some reflections and a case study in an autobody assembly process. Qual. Eng. 19 (4), 311–325.

Frisk, E., Krysander, M., Jung, D., 2017. A toolbox for analysis and design of model based diagnosis systems for large scale models. In: IFAC World Congress. Toulouse, France.

Fuente, M.J., García, G., Sainz, G.I., 2008. Fault diagnosis in a plant using Fisher discriminant analysis. In: Proceedings of the Mediterranean Conference on Control and Automation Congress Centre. Ajaccio, France.

García-Álvarez, D., Bregon, A., Fuente, M.J., Pulido, B., 2011. Improving parameter estimation using minimal analytically redundant subsystems. In: Proceedings of the IEEE Conference on Decision and Control and European Control Conference. Orlando, USA.

García-Álvarez, D., Fuente, M.J., Sainz, G.I., 2012. Fault detection and isolation in transient states using principal component analysis. J. Process Control 22 (3), 551–563.

Gertler, J., 1998. Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, Inc..

Gertler, J., Li, W., Huang, Y., McAvoy, T., 1999. Isolation enhanced principal component analysis. AIChE J. 45 (2), 323–334.

Hamscher, W., Console, L., De Kleer, J., 1992. Readings in Model-Based Diagnosis. Morgan Kaufmann Publishers Inc..

Huang, Y., Gertler, J., McAvoy, T.J., 2000. Sensor and actuator fault isolation by structured partial PCA with nonlinear extensions. J. Process Control 10 (5), 459–469.

Hwang, D.H., Han, C., 1999. Real-time monitoring for a process with multiple operating modes. Control Eng. Pract. 7 (7), 891–902.

Jackson, J.E., 2003. A User's Guide to Principal Components. Wiley, p. 569.

Jackson, J.E., Mudholkar, G.S., 1979. Control procedures for residuals associated with Principal Component Analysis. Technometrics 21 (3), 341–349.

Jiang, Y., Yin, S., Kaynak, O., 2021. Optimized design of parity relation-based residual generator for fault detection: Data-driven approaches. IEEE Trans. Ind. Inform. 17 (2), 1449–1458. http://dx.doi.org/10.1109/TII.2020.2987840.

Jung, D., Frisk, E., 2018. Residual selection for fault detection and isolation using convex optimization. Automatica 97, 143–149.

Jung, D., Sundström, C., 2019. A combined data-driven and model-based residual selection algorithm for fault detection and isolation. IEEE Trans. Control Syst. Technol. 27 (2), 616–630.

Kourti, T., MacGregor, J.F., 1996. Multivariate SPC methods for process and product monitoring. J. Qual. Technol. 28 (4), 409–428.

Krysander, M., Åslund, J., Nyberg, M., 2008. An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. IEEE Trans. Syst., Man, Cybern. - Part A: Syst. Hum. 38 (1), 197–206. http://dx.doi.org/10.1109/TSMCA.2007.909555.

Lane, S., Martin, E.B., Morris, A.J., Gower, P., 2003. Application of exponentially weighted principal component analysis for the monitoring of a polymer film manufacturing process. Trans. Inst. Meas. Control 25 (1), 17–35.

Moya Alonso, N., Bregon, A., Alonso-González, C.J., Pulido, B., 2013. A common framework for fault diagnosis of parametric and discrete faults using possible conflicts. In: Conference of the Spanish Association for Artificial Intelligence. Springer, pp. 239–249.

Murota, K., 2012. Systems Analysis By Graphs and Matroids: Structural Solvability and Controllability, Vol. 3. Springer Science & Business Media.

Nomikos, P., MacGregor, J., 1995. Multivariable SPC charts for monitoring batch processes. Technometrics 37 (1), 41–59.

Portnoy, I., Melendez, K., Pinzon, H., Sanjuan, M., 2016. An improved weighted recursive PCA algorithm for adaptive fault detection. Control Eng. Pract. 50, 69–83.

Pulido, B., Alonso-González, C., 2004. Possible Conflicts: A compilation technique for consistency-based diagnosis. IEEE Trans. Syst., Man, Cybern. Part B: Cybern. 34 (5), 2192–2206. http://dx.doi.org/10.1109/TSMCB.2004.835007.

Pulido, B., Alonso-González, C., Bregon, A., Hernández, A., 2015. Characterizing and computing HBG-PCs for hybrid systems fault diagnosis. In: Conference of the Spanish Association for Artificial Intelligence. Springer, pp. 116–127.

Pulido, B., Bregon, A., Alonso-González, C.J., Hernández, A., Rubio, D., Villarroel, L.M., 2016. DX-PCs: A toolbox for model-based diagnosis of dynamic systems using possible conflicts. Progress Artif. Intell. 5 (2), 111–120.

Pulido, B., Zamarreño, J.M., Merino, A., Bregon, A., 2019. State space neural networks and model-decomposition methods for fault diagnosis of complex industrial systems. Eng. Appl. Artif. Intell. 79, 67–86. http://dx.doi.org/10.1016/j.engappai.2018.12.007.

Rafferty, M., Liu, X., Laverty, D.M., McLoone, S., 2016. Real-time multiple event detection and classification using moving window PCA. IEEE Trans. Smart Grid 7 (5), 2537–2548. http://dx.doi.org/10.1109/TSG.2016.2559444.

Reiter, R., 1987. A theory of diagnosis from first principles. Artificial Intelligence 32 (1), 57–95.

Rezamand, M., Kordestani, M., Carriveau, R., Ting, D.S.-K., Saif, M., 2020. A new hybrid fault detection method for wind turbine blades using recursive PCA and wavelet-based PDF. IEEE Sens. J. 20 (4), 2023–2033. http://dx.doi.org/10.1109/JSEN.2019.2948997.

Shlens, J., 2005. A Tutorial on Principal Component Analysis. Center for Neural Science, New York University.

Tien, D.X., Lim, K.W., Jun, L., 2004. Compartive study of PCA approaches in process monitoring and fault detection. In: The 30th Annual Conference of the IEEE Industrial Electronics Society.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003. A review of process fault detection and diagnosis. Part III: Process history based methods. Comput. Chem. Eng. 27 (3), 327–346.

Weighell, M., Martin, E.B., Morris, A.J., 2001. The statistical monitoring of a complex manufacturing process. J. Appl. Stat. 28 (3–4), 409–425.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2 (1–3), 37–52.

Zarzo, M., Ferrer, A., 2004. Batch process diagnosis: PLS with variable selection versus block-wise PCR. Chemometr. Intell. Lab. Syst. 73 (1), 15–27.

Zhao, S.J., Zhang, J., Xu, Y.M., 2004. Monitoring of processes with multiple operating modes through multiple principle component analysis models. Ind Eng Chem Res 43, 7025–7035.

Zhao, S.J., Zhang, J., Xu, Y.M., 2006. Performance monitoring of processes with multiple operating modes through multiple PLS models. 16, (7), pp. 763–772,