# Decentralized DPCA Model for Large-Scale Processes Monitoring

A. Sanchez-Fernandez, M.J. Fuente, G.I. Sainz-Palmero
*Department of System Engineering*
*and Automatic Control, EII*
*University of Valladolid*
Valladolid, Spain
{alvsan, mjfuente, gresai}@eii.uva.es

***Abstract*—Monitoring large-scale processes is a crucial task to ensure the safety and reliability of the plants. This paper proposes an approach for decentralized fault detection in large-scale processes. The measured variables of the plant are divided into multiple and possibly overlapping blocks using different techniques based on data. Local monitoring methods are applied in each block using DPCA (Dynamic Principal Component Analysis) model. The local results are then fused by the Bayesian inference strategy. This paper also compares different techniques to decompose the plant looking for the best strategy from the point of view of the fault detection results. The proposed method was applied to the widely used benchmark Tennessee Eastman Process, showing its effectiveness when compared with a centralized method and another decentralized technique.**

***Index Terms*—Fault detection, Dynamic principal component analysis, Decentralized monitoring, Regression, Clustering.**

## I. INTRODUCTION

The complexity of monitoring systems in industrial plants has seen substantial growth over the last years, implying an increasing deployment of processing units and sensors. The consequence is an enormous amount of data that will be extremely useful to increase the knowledge about the plant and to develop better monitoring methods, in particular, data-based methods have been improved their effectiveness [1], [2].

Specifically, multivariate statistical process monitoring (MSPM) methods, like PCA, Partial Least Squares (PLS), Independent Component Analysis (ICA) and many others, have been gained great importance in the process monitoring field [3]–[6]. The main characteristic of these methods is their ability to handle large amount of data from the plant and to extract valuable information about the industrial process that can be used to perform the fault detection, without having any previous knowledge. Also, it is not required to train and adjust a first principles plant model, which can be many times unaffordable.

Furthermore, the variables are, usually, not only cross-correlated but auto-correlated, implying that the current state of these variables are the result of the past states of the plant. In other words, the system is dynamic and it is necessary to include this auto or time-correlation in the monitoring model. Some MSPM methods, like Canonical Variate Analysis (CVA) [5], [7], already include the analysis of past states, but other MSPM methods, like PCA, make use of the augmented data matrix, which includes time-lagged variables along with not lagged variables, to take in account the dynamic behaviour of the plant. This is the Dynamic PCA (DPCA) method [8].

The amount of data from the plant is growing significantly, with lot of sensors, control devices, etc. and this is a major problem as it is required to increase the data transmission capability and to have high computing capacity. Sometimes it is feasible to dispose a unique processor that receives all the data collected in the plant, but this is only possible when the plant size is reduced. In many other cases, for example, in chemical plants, this solution is not feasible and a decentralized or distributed approach has been considered by some authors [9]–[15]. A decentralized method creates some blocks that group the variables, using overlapping or not overlapping blocks, that is, a variable can be included or not in different blocks. Also, there is the possibility to do a complete decentralization, with one block per variable, or to develop a system with less blocks than variables [12]. Previous knowledge about the plant, process topology, etc. are different options to perform the plant division [16], [17], but a more practical alternative is to use data-driven decentralization methods [9], [10], [12]. These methods only need to collect data and analyse them to discover correlations between variables, which will be used to decompose the plant.

In a previous paper, the authors have explored the effectiveness of decentralized DPCA method using Neural Nets and Sparse Partial Least Squares to analyse the plant and perform the block division [18]. In that case, the fault detection method worked with a decentralized plant where every variable had its own block. Although that proposal had confirmed its effectiveness, it is not possible to work with a completely decentralized approach in very big plants because, as it is necessary to implement one processing unit with every variable, the economic costs as well as the transmission requirements will be unbearable. So, it is necessary to explore other decentralization methods that work with a reduced number of blocks.

This paper proposes some methods to find block distributions with different numbers of blocks for a decentralized

The authors would like to express their gratitude to the European Regional Development Fund (FEDER) and the Spanish Ministry of Economy and Competitivity for financial support through the project MASCONTROL (ref. MINECO/FEDER PI2015-67341- C2-2-R.)

fault detection method. In concrete, in this paper the plant decomposition is carried out using neural netwoks, mutual information and clustering methods. The objective is to test their effectiveness in an complex industrial plant, comparing them with other decentralization methods and with a centralized method, and looking for the best method from the point of view of fault detection results.

This document is organized as follows: Section II explains DPCA method, decentralized approach, with its variants, and Bayesian inference strategy based decision fusion technique. In Section III is detailed the decentralized proposal of this paper. Section IV contains the application of this proposal on the Tennessee Eastman Plant and a comparison with a centralized DPCA approach, as well as other decentralized method. The article finishes with Section V that presents the conclusions and the future work.

## II. PRELIMINARIES

### A. Dynamic PCA

*1) PCA:* PCA is a technique based on analysing the data measured in a plant. If there are $m$ sensors and $n$ measures are taken, the data matrix $\mathbf{X}(n \times m)$ can be constructed. This matrix is normalized, by columns, to zero mean and unit variance before the correlation matrix, $\mathbf{S}$ is obtained:

$$\mathbf{S} = \frac{1}{(n-1)}\mathbf{X}^T\mathbf{X} \tag{1}$$

$\mathbf{S}$ is decomposed, using singular value decomposition:

$$\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{2}$$

where $\mathbf{\Lambda}$ contains the eigenvalues in its diagonal, while the columns of $\mathbf{V}$ are the corresponding eigenvectors. The eigenvalues are the variance included in each principal component, $\mathbf{T}$, which are obtained using the loadings matrix, $\mathbf{P}$:

$$\mathbf{T} = \mathbf{X}\mathbf{P} \tag{3}$$

The loadings matrix, $\mathbf{P}$, is composed with the $a$ first columns of $\mathbf{V}$. With high values of $a$, more data variance is captured by PCA but the less dimensionality reduction is achieved. So, the selection of this parameter is a trade-off between a reduced number of principal components and information retained in the PCA model.

*a) PCA fault detection:* In order to detect faults, the statistics $T^2$ and $Q$ are used [6]. $T^2$ (also known as Hotelling's Statistic), for a new measure $\mathbf{x}$, is obtained as:

$$T^2 = \mathbf{x}^T\mathbf{P}\mathbf{\Lambda}_a^{-1}\mathbf{P}^T\mathbf{x} \tag{4}$$

And a fault is detected in measure $\mathbf{x}$ if $T^2$ is over its threshold $T_\alpha^2$. Also, the $Q$ statistic, which measures the goodness of fit and the system noise and disturbances, can be obtained, for a new measure $\mathbf{x}$, as:

$$Q = [(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x}]^T[(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x}] \tag{5}$$

where $\mathbf{I}$ is a square identity matrix.

A fault is detected in measure $\mathbf{x}$ if $Q$ overpasses its threshold $Q_\alpha$. The thresholds for $T^2$ and $Q$ can be found in [19], and their values are obtained for a certain significance level.

As the plant will suffer noise, disturbances, etc. it is expected to get some false positives when doing the fault detection. One way to reduce this problem is to require a certain number of consecutive fault detections to consider that there is a true fault in the system. This value must be set by the user reducing as much as possible the false alarms, but avoiding an excessive delay in the fault detection.

*2) DPCA:* In any industrial plant it is expected to find certain influence of past states into the current state. This is not taken in account by the standard PCA, but there is a modified version of this method that includes this time correlation in its analysis: the DPCA [8]. It follows the same steps as PCA but the initial data matrix is replaced by the augmented matrix, $\mathbf{X}_a$, which is constructed using current and delayed measures of the variables:

$$\mathbf{X}_a = \begin{bmatrix} \mathbf{X}_{l+1}^T & \mathbf{X}_l^T & \dots & \mathbf{X}_1^T \\ \mathbf{X}_{l+2}^T & \mathbf{X}_{l+1}^T & \dots & \mathbf{X}_2^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_n^T & \mathbf{X}_{n-1}^T & \dots & \mathbf{X}_{n-l}^T \end{bmatrix} \tag{6}$$

being $\mathbf{X}_t$ the vector of data at time $t$. The number of lags included are represented by $l$, taking in account that this parameter must be selected by the user choosing the value that achieves the best results. One way to do this is through Akaike Information Criterion (AIC) ( [5], [20]).

### B. Decentralized fault detection

The monitoring methods usually work with a unique model for the whole plant. So there exists one central processing unit that receive all the measures from all the sensors and uses them to determine the condition of the plant. This is the centralized approach, but there is the option to divide the plant in blocks of variables and implement a monitoring unit in each of these blocks. This is the decentralized approach.

But this approach needs to define how the decentralization is done and how to fuse the monitoring results obtained in the blocks, because a unique and global diagnosis is needed.

The decentralization can be done using previous knowledge about the plant, or analysing the available data [9]. The second option looks more feasible because it is not common to have a complete and detailed information about the installation. Some authors have explored this possibility using techniques as: Sparse PCA [12], Correlation [9], [10], etc. With these methods the variables are grouped according to the strength of the relations encountered by the method.

After the decentralization process the system will be divided in one of this two ways: Completely decentralized decomposition and Multi-block process decomposition [12]. In the first case, each sensor has its own block and no more variables are included. This option does not likely provide good results as it ignores the correlation between variables. In the second case, some blocks are created including different variables in each one. A variable can be grouped in various blocks and there is a possibility to have the same or less number of blocks than variables. In this case, the influence between variables are taken in account for the monitoring.

Also it is necessary to decide what conditions are needed to detect a fault in the decentralized plant. One option is to activate the fault alarm if a fault is detected in any group but this could lead to problems like false alarms. There are other options to fuse the results from the blocks in order to improve the performance of the monitorization , like Bayesian methods, weighted voting, etc. [21].

*1) Bayesian Inference Criterion:* The Bayesian Inference Criterion (BIC) [9] is able to fuse the results from different locations in a decentralized system, delivering a unique outcome. In PCA based monitoring model, each block will send two results: statistics $T^2$ and $Q$; so BIC method must be applied two times, one for each statistic.

For one statistic ($T^2$ or $Q$) in block $i$ (with $i = 1, 2, \dots, b$), the fault posterior probability is:

$$P_{(F|x_i)} = P_{(x_i|F)} P_{(F)} / P_{(x_i)} \tag{7}$$

and:

$$P_{(x_i)} = P_{(x_i|N)} P_{(N)} + P_{(x_i|F)} P_{(F)} \tag{8}$$

where $N$ is the normal system state while $F$ represents the abnormal state. The prior probabilities are: $P_{(N)}$ and $P_{(F)}$, for the normal and faulty state of the system, respectively. An $\alpha$ value is fixed for $P_{(N)}$ (with $\alpha \in [0, 1]$), while $1-\alpha$ is used for $P_{(F)}$. The values for $P_{(x_i|N)}$ and $P_{(x_i|F)}$ are obtained using the expressions:

$$P_{(x_i|N)} = e^{-ST/ST_{i,lim}}, P_{(x_i|F)} = e^{-ST_{i,lim}/ST} \tag{9}$$

where $ST_{i,lim}$ represents the corresponding threshold for $ST$ in the $i$-th group. After this calculations, the $BIC$ index is generated using the results form all the blocks:

$$BIC_{ST} = \sum_{i=1}^{b} \frac{P_{(x_i|F)} P_{(F|x_i)}}{\sum_{i=1}^{m} P_{(x_i|F)}} \tag{10}$$

When $BIC_{ST} > (1 - \alpha)$ a fault is detected with $ST$ statistic. As with PCA thresholds, the limit $(1 - \alpha)$ can be adjusted using faultless test data looking for a value that fixes the false alarms to the level demanded by the user.

### C. Neural Networks

A widely used option for modelling non-linear systems are the Artificial Neural Networks (ANN). This technique obtains one or more outputs processing some inputs. Between the inputs and outputs there are a set of interconnected processing elements called neurons. These neurons receive a set of inputs coming from other neurons, and calculate a linear combination of these inputs. Then, this linear combination is transformed with an activation function. The output of these neurons can feed another layer of neurons, or can be used to calculate the net output. All the connections between neurons are weighted, that is, each input of a neuron is multiplied by a weight, and the adjustment of these weights is done during the net training, which allows the net to “learn” patterns. A neural net with three inputs, one hidden layer with four neurons, and two outputs is shown in Figure 1.

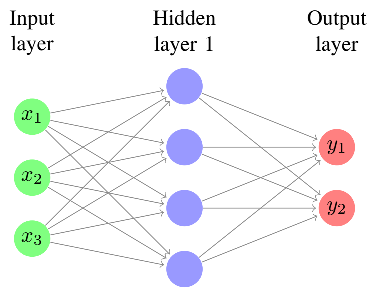


Figure 1. Neural network example

### D. Mutual Information

The Mutual Information (MI) is, in information theory, a measure that quantifies the mutual dependence between two variables [22]. Precisely, its value represents the “amount of information” that a random variable can provide about another random variable. MI can be understood as the reduction in uncertainty about one variable when another variable is known. High MI value means high uncertainty reduction, while low values means small uncertainty reduction. If MI is zero, the variables are not dependent.

MI for two random variables $x$ and $y$ is calculated as:

$$MI(x, y) = \sum_{y} \sum_{x} P_{(x,y)} log(\frac{P_{(x,y)}}{P_{(x)} P_{(y)}}) \tag{11}$$

where $P_{(x,y)}$ is the joint probability mass function of $x$ and $y$, and $P_{(x)}$ and $P_{(y)}$ are the marginal probability mass functions of $x$ and $y$, respectively. The MI value between two variables is relevant only if an upper $(1 - \alpha)\%$ critical threshold is overpassed ($\alpha$ must be defined by the user).

### E. Clustering

Clustering is a data mining technique which identifies in an automated way groups of elements (clusters) according to their similarity. The main objective is to find clusters so that: the average similarity between elements inside the same cluster will be high and the average similarity between elements from different clusters will be reduced.

There are many algorithms to do Clustering, but one of the most used is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [23]. In this algorithm the user must establish the parameter $\epsilon$: the radius of neighbourhood, which is used to do the clustering process: in each cluster must be *core* points that have, at least, a certain number of points placed at a distance $d < \epsilon$ from them; other points can be included in the same cluster if there are placed at a shorter distance than $\epsilon$. Any other point will be considered an outlier and it will not be included in any cluster.

## III. DISTRIBUTED FAULT DETECTION WITH DPCA

The proposal of this paper consists of three steps: Plant decentralization, Local DPCA development and Global fault detection. In the first one, Plant decentralization, the system will be divided in blocks of variables only analysing data. Then, the second is Local DPCA development, which trains a Dynamic PCA in each one of the created blocks. Finally, in Global fault detection step, a central processor will receive the results obtained by each local DPCA, fuse them and generate a global and unique result of the fault detection process. All this steps are detailed below:

*Step 1.- Plant decentralization*: three decentralization strategies were proposed: Neural Net, MI and Clustering. All of them were selected because they are non-linear and based on different concepts, specifically: regression, information and clustering, respectively. All of them only need to process data to divide the plant. Also, they can deliver diverse decentralizations sizes: Neural Net will create a full decentralized method (same number of blocks as the number of variables ), while Clustering allows to specify the number of blocks, giving the chance to obtain methods with a reduced number of blocks. MI based method will give a less decentralized approach than Neural Nets. This will be very useful to check how different decentralization sizes work.

*a) Neural Net:* This method was presented and explained in [18]. It creates a full decentralization model, so each variable has its own block. The process consists of creating a neural net model for each variable, where this variable is the output and the remaining variables are the inputs. Some net configurations (number of hidden layers and neurons) are tested and the one with the lowest value of rMSE is selected. After that, the variables with highest influence in the net output are included in the group along with variable which is the output of the net. As was verified in [18], this model is able to capture the non linear relations between the variables.

*b) Mutual Information:* This method is based on the analysis of the MI matrix. It consists in two steps, in the first one the plant is divided, getting one block per variable, and in the second step a reduction in the number of blocks is performed.The plant decomposition is done analysing the mutual information (MI) matrix, where in each $i$-th row there is the mutual information between the $i$-th variable with all the rest of variables. So, each $i$-th block is composed by the variables corresponding to the elements of that row that overpass a certain threshold. Then, each variable has its own block, but it is necessary to check if there exists blocks with a reduced number of elements. In that case, these small blocks are removed, and an extra block is created with the variables whose blocks have been eliminated. The minimum number of variables in each block is a parameter that must be selected by the user.

*c) Clustering:* Here the objective is to group variables that share any kind of relation. It is necessary to define which characteristics of the variables are going to be processed to discover these relations. In this proposal, a data matrix is created, where row $i$ contains the kurtosis and skewness of variable $i$, the mean and the variance value of the row $i$ from correlation matrix, and the mean and variance of row $i$ from MI matrix. Then, this data matrix is processed using DBSCAN algorithm, which generates blocks or clusters of variables that share more relation between them. The parameter of neighbourhood, $\epsilon$, must be chosen by the user, selecting this one that gives better result in the fault detection task.

*Step 2.- Local DPCA development*: Once the system is divided, a local fault detection method must be implemented in each block. This is done with Dynamic PCA, which processes an augmented matrix $\mathbf{X_a}$, composed of the variables that belong to the corresponding block as well as some delayed values of these variables. The number of lags, $l$, is selected between different values, as well as the value for $a$ parameter, which is set to a value that retains a certain percentage of data variance. The selected combination of both parameters is the one that achieves the best results in the fault detection task. Once the local DPCA are trained, the thresholds for $T^2$ and $Q$ are calculated. Then, these limits are tuned in order to obtain only $1\%$ of anomalous observations when analysing non faulty data.

After these two off-line steps, then comes the fault detection task:

*Step 3.- Global fault detection*. After the model training, new measures are taken and processed in each block by local DPCA models. These blocks deliver the current local values for $T^2$ and $Q$, as well as the respective thresholds, which are sent to a central processor. This processor fuses local statistics through BIC index (Section II-B1), obtaining global BIC values for $T^2$ and for $Q$, which are used to detect faults in the whole plant: if one or both BIC indexes overpass their corresponding thresholds, for a certain confidence level $\alpha$, a fault is detected. Also, as it is necessary to avoid false alarms, another condition is set: a fault will only be detected if a certain number of consecutive anomalous observations are found. This number will be set by the user looking to avoid false alarms but with a reduced fault detection delay.

The scheme of this method can be seen in Algorithm 1.

## IV. ILLUSTRATIVE EXAMPLE

Tennessee Eastman Process (TEP) [24] benchmark was used to evaluate the performance of the proposed approaches. Here we present the results of three different decentralizing strategies, including Neural Net, Mutual Information and Clustering based methods.

TEP plant has been widely applied to test monitoring methods [5], [12], [25]–[28] and it is a reference in this area. The available data for this plant are composed of measures of 52 variables, taken every 3 minutes, and faultless train and test datasets are included. Also train and test datasets from 21 different faults are available (see Table I) [24]. Each train dataset is formed by 500 samples while test datasets contain 960 samples.

**Algorithm 1** *Data based decentralization with DPCA*

**Off-line steps:**
Normalize train data (faultless)
**if** method: NeuralNet **then** ▷ **Step 1.a**
  **for** i=1 to m **do** ▷ For each variable
    Model Neural N. with varied parameters (layers, neurons)
    Select model with lowest rMSE
  **end for**
**end if**
**if** method: MI **then** ▷ **Step 1.b**
  Obtain MI matrix and thresholds
  **for** i=1 to m **do** ▷ For each variable
    Take row $i$ of MI: $\mathbf{MI}_i$ and thresholds: $\mathbf{MI}_i^{th}$
    Select variables $j$ that: $\mathbf{MI}_i(j) > \mathbf{MI}_i^{th}(j)$
  **end for**
  Remove small groups
  Create block with variables without group
**end if**
**if** method: Clustering **then** ▷ **Step 1.c**
  Create data matrix: kurtosis, skewness, etc.
  Apply Dbscan & Get clusters
**end if**
Generate blocks using previous results
Develop DPCA local models ▷ **Step 2**
**for** i=1 to b **do** ▷ For each block
  Develop local DPCA with different lags
  Select DPCA model with best results
**end for**
**On-line steps:** ▷ **Step 3**
**for** Each new measure **do**
  **for** i=1 to b **do** ▷ For each block
    Obtain $ST^i$ = { $T^2$, $Q$ } ▷ Block statistics
  **end for**
  **for** ST={$T^2$, $Q$} **do** ▷ For each statistic
    $\boldsymbol{BIC}_{ST} = f(ST_1, ST_2, \ldots, ST_m)$
    **if** $\boldsymbol{BIC}_{ST}$ overpass $(1-\alpha)$ **then**
      *Fault detection using ST*
    **else**
      *No fault using ST*
    **end if**
  **end for**
**end for**

Table I
TEP FAULTS

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

### A. Experimental setup

Three different methods were tested in this work, each one with its own parameters: number of lags, variance retained by DPCA, etc. These parameters were adjusted after some tests with train datasets looking to obtain the best monitoring model in terms of: lowest average fault detection time, highest number of faults detected and lowest number of faulty datasets with false alarms. After the parameters of each method were adjusted, the significance value $\alpha$ for BIC fusion is tuned using test faultless dataset looking to reduce or avoid the presence of false alarms.

For the first decentralization strategy, based on neural nets, different models were created (using various numbers of hidden neurons) and the one with the lowest value of rMSE error was selected, for each variable. With respect to the second strategy, MI based, the matrix MI data were obtained as well as the respective thresholds. Each row of the matrix were used to find which variables must be grouped together. Finally, Clustering based strategy was developed using information about Kurtosis, Skewness, and mean and variance values of correlation and mutual Information of each variable.

After the decentralization, local DPCA methods were trained try to get the best combination of parameters in order to obtain the best fault detection results. Finally, the methods were tuned to this values:

- Neural Net method used augmented matrices in local DPCA with 2 lags, the $\alpha$ value for BIC was 0.9, after 3 consecutive anomalous observations a fault is detected, the selected principal components in local DPCA retained 60% of variance, local thresholds in DPCA models were tuned for a confidence level of 90% and a variable was selected in each model when its coefficient is over the mean of the maximum and minimum values for all the variables inside the net. The value for BIC $\alpha$ was not readjusted after using test data.
- MI method needed 4 consecutive anomalous observations to detect a fault, local DPCA models were developed with 3 lags in augmented matrices, also, local DPCA retained 75% of variance in their principal components, and $\alpha = 0.9$ for BIC fusion. After some tests, this value was modified only for $T^2$: $\alpha = 0.865$.
- Clustering strategy was tuned to 5 consecutive anomalous observations to detect a fault, 4 lags in the local augmented matrices, 60% of variance included in principal components in DPCA models, the value for $\alpha$ in BIC was set to 0.99 and the clustering was done with $\epsilon = 0.8$. Using test datasets, $\alpha$ was set to: 0.987 for $T^2$ and 0.986 for $Q$.

### B. Results

In this proposal the objective was to develop decentralized monitoring using different plant decomposition methods based on data to looking the best one for the fault detection task. Also, the proposed methods are compared with a centralized DPCA approach, whose results are taken from [5] and with other decentralized method: Weighted Dynamic Decentralized PCA (WDDPCA), taken from [10], to see their effectiveness.

Some indexes were used to do the comparison: the Missed Detection Rate (MDR), that is, what percentage of faulty measures are classified as faultless data; the fault detection delay, which measures how many samples are needed to detect a fault after its occurrence; the False Alarm Rate (FAR), which represents the percentage of non-faulty samples that are classified as faulty; and, finally, the number of faults detected.

Table II shows the number of blocks implemented by each method. ANN works with one block per variable, so it has the highest grade of decentralization along with WDDPCA, so they are the methods which need more local processing units. Also, WDDPCA includes all variables in each block, complicating the local data processing task. Then comes MI based decentralization, with 42 blocks, so this technique is able to slightly reduce the number of blocks. And, finally, Clustering method can deliver a decentralized approach with a reduced number of blocks, in this case, only 4 blocks.

Also, in Table II, it is shown that all proposals were capable of detecting more faults than the centralized DPCA and WDDPCA, particularly, ANN, which could find all the 21 faults. Also, MI decentralization detected all the faults using $Q$ statistic. FAR index took values from 0 to 0.2 for the proposed methods, not too far from DPCA with $T^2$ and WDDPCA with $Q$, but clearly lower than central DPCA with $Q$.

Table II
BLOCKS, FALSE ALARMS RATES AND FAULTS DETECTED (IN %)

| | ANN | MI | Clustering | WDDPCA | DPCA |
|---|---|---|---|---|---|
| **Blocks** | 52 | 42 | 4 | 52 | 1 |
| **FAR $T^2$** | 0.2 | 0 | 0 | 2.41 | 0.6 |
| **FAR $Q$** | 0 | 0.2 | 0 | 0 | 28.1 |
| **Detected faults $T^2$** | 21 | 20 | 20 | 18 | 17 |
| **Detected faults $Q$** | 21 | 21 | 19 | 18 | 18 |

Tables III and IV contain the MDR results. This index gives an idea about the sensitivity of each method.

Tables V and VI show the delay results for each fault and each method. It should be taken into consideration that all methods in the comparison need to detect some consecutive anomalous observations, so the detection time shown in the Tables must be increased by the corresponding values specified in previous section, for each method.

Tables III and IV show that all decentralized methods were better than centralized DPCA in all index and with both statistics. Analysing the results, Neural Net method with $T^2$ delivered the best results of the comparison in MDR index (it is the best in 15 faults) results, while MI and WDDPCA were the best using $Q$, as they got the lowest value in 13 cases out of 21. When comparing MDR results with $Q$ only for MI against WDDPCA, MI is better in 14 faults while WDDPCA obtains the lowest MDR in 13 faults. Clustering method, which is the decentralization with the lowest number of blocks, delivered worse MDR values than the other decentralized methods, but it was able to deliver better results than central DPCA in 15 faults with $T^2$, and in 18 faults with $Q$.

Table III
MISSED DETECTION RATE (MDR), IN %. $T^2$

| Fault | ANN | MI | Clustering | WDDPCA | DPCA |
|---|---|---|---|---|---|
| 1 | 0.13 | **0** | 0.38 | 0.25 | 0.6 |
| 2 | 1.13 | 1.13 | **1.01** | 1.50 | 1.9 |
| 3 | **94.11** | 99.62 | 98.24 | 97.25 | 99.1 |
| 4 | 27.44 | 88.96 | 85.43 | **0.00** | 93.9 |
| 5 | **70.30** | 74.78 | 75.38 | 72.50 | 75.8 |
| 6 | **0** | **0** | 0.38 | 0.50 | 1.3 |
| 7 | **0** | **0** | **0** | **0** | 15.9 |
| 8 | **0.88** | 2.26 | 2.51 | 2.25 | 2.8 |
| 9 | **93.61** | 99.00 | 99.12 | 99.13 | 99.5 |
| 10 | **38.22** | 51.07 | 58.29 | 55.63 | 58.0 |
| 11 | 43.98 | 59.85 | 56.66 | **14.88** | 80.1 |
| 12 | **0.38** | 0.50 | 0.50 | 0.75 | 1.0 |
| 13 | **4.39** | 4.89 | 5.40 | 5.25 | 4.9 |
| 14 | **0** | **0** | **0** | **0** | 6.1 |
| 15 | **83.08** | 97.37 | 96.61 | 96.38 | 96.4 |
| 16 | **48.75** | 73.90 | 78.02 | 76.00 | 78.3 |
| 17 | **8.77** | 9.66 | 38.32 | 4.38 | 24.0 |
| 18 | **9.77** | 10.41 | 10.55 | 10.25 | 11.1 |
| 19 | 97.74 | 92.60 | 99.75 | **77.25** | 99.3 |
| 20 | **42.11** | 46.42 | 59.80 | 45.50 | 64.4 |
| 21 | 55.51 | 62.36 | 67.71 | **46.50** | 64.4 |

Table IV
MISSED DETECTION RATE (MDR), IN %. $Q$

| Fault | ANN | MI | Clustering | WDDPCA | DPCA |
|---|---|---|---|---|---|
| 1 | **0** | **0** | **0** | **0** | 0.50 |
| 2 | 1.25 | **0.88** | 1.51 | 1.50 | 1.50 |
| 3 | 98.50 | 97.74 | 98.99 | **97.63** | 99.00 |
| 4 | **0** | **0** | **0** | **0** | **0** |
| 5 | 75.56 | 71.39 | 66.96 | **0.00** | 74.80 |
| 6 | **0** | **0** | **0** | **0** | **0** |
| 7 | **0** | **0** | **0** | **0** | **0** |
| 8 | 2.01 | **1.88** | 2.51 | **1.88** | 2.50 |
| 9 | 98.37 | **96.99** | 99.50 | 99.38 | 99.40 |
| 10 | 38.72 | 47.93 | 59.30 | **31.00** | 66.50 |
| 11 | 8.90 | **3.76** | 13.44 | 9.63 | 19.30 |
| 12 | **0.5** | **0.5** | 0.63 | 0.63 | 2.40 |
| 13 | 4.76 | 4.39 | **4.15** | 5.13 | 4.90 |
| 14 | **0** | **0** | **0** | **0** | **0** |
| 15 | 96.24 | **95.11** | 95.73 | 96.88 | 97.60 |
| 16 | 54.39 | 54.83 | 60.05 | **31.75** | 70.80 |
| 17 | 2.88 | **2.01** | 3.89 | 2.25 | 5.30 |
| 18 | 9.65 | **9.16** | 9.42 | 9.50 | 10.00 |
| 19 | 75.31 | 25.22 | 72.99 | **24.25** | 73.50 |
| 20 | 38.97 | 31.49 | 46.23 | **30.50** | 49.00 |
| 21 | 47.49 | 47.43 | 52.39 | **46.88** | 55.80 |

Tables V and VI reveal that, again, ANN achieved the best detection time using $T^2$ (as it is the fastest in 15 faults), while, MI method was the best using $Q$, with the lowest detection delay in 14 faults. Clustering based decentralization was the best in 9 fault, the same result as WDDPCA. Also, Clustering get lower detection times than central DPCA, because, comparing only these two methods, Clustering was the fastest in 18 faults with $T^2$ and with $Q$.

Summarizing, using $T^2$, ANN achieved the best results in MDR and, also, in detection delay; while MI and WDDPCA decentralizations were the best using $Q$. It is known that $T^2$ monitors the model, while $Q$ processes the noise, disturbances, etc. [6], and, as ANN got better results with $T^2$, this method probably had captured the behaviour of the model better than

Table V
DETECTION DELAY, IN SAMPLES. $T^2$

| Fault | ANN | MI | Clustering | WDDPCA | DPCA |
|---|---|---|---|---|---|
| 1 | 1 | **0** | 3 | 2 | 6 |
| 2 | 9 | 9 | **8** | 12 | 16 |
| 3 | **40** | nd | 80 | nd | nd |
| 4 | **0** | 74 | 144 | **0** | 151 |
| 5 | **0** | **0** | **0** | **0** | 2 |
| 6 | **0** | **0** | 3 | 4 | 11 |
| 7 | **0** | **0** | **0** | **0** | 1 |
| 8 | **7** | 18 | 20 | 18 | 23 |
| 9 | **0** | 2 | 2 | nd | nd |
| 10 | **22** | 52 | 56 | 48 | 101 |
| 11 | 9 | 8 | **5** | **5** | 195 |
| 12 | **0** | **0** | **0** | 2 | 3 |
| 13 | **35** | 41 | 44 | 42 | 45 |
| 14 | **0** | **0** | **0** | **0** | 6 |
| 15 | **573** | 675 | 671 | nd | nd |
| 16 | **0** | 33 | 304 | 189 | 199 |
| 17 | 20 | **19** | 35 | 21 | 28 |
| 18 | **83** | **83** | 84 | 84 | 93 |
| 19 | **8** | 420 | nd | 17 | nd |
| 20 | 78 | **77** | 80 | 81 | 87 |
| 21 | 415 | 505 | 514 | **258** | 522 |

nd=not detected

Table VI
DETECTION DELAY, IN SAMPLES. $Q$

| Fault | ANN | MI | Clustering | WDDPCA | DPCA |
|---|---|---|---|---|---|
| 1 | **0** | **0** | **0** | 2 | 5 |
| 2 | 10 | **7** | 12 | 12 | 13 |
| 3 | **86** | 316 | nd | nd | nd |
| 4 | **0** | **0** | **0** | **0** | 2 |
| 5 | **0** | **0** | **0** | **0** | 2 |
| 6 | **0** | **0** | **0** | **0** | 1 |
| 7 | **0** | **0** | **0** | **0** | 1 |
| 8 | 16 | **15** | 16 | **15** | 21 |
| 9 | **3** | 359 | nd | nd | nd |
| 10 | 32 | 32 | 44 | **26** | 50 |
| 11 | 3 | **2** | **2** | 5 | 7 |
| 12 | **0** | **0** | **0** | 2 | 8 |
| 13 | 36 | 35 | **33** | 41 | 40 |
| 14 | **0** | **0** | **0** | **0** | 1 |
| 15 | 571 | **461** | 572 | nd | nd |
| 16 | 16 | 15 | 33 | **13** | 196 |
| 17 | 18 | **16** | 19 | 19 | 24 |
| 18 | 78 | 76 | **77** | **77** | 84 |
| 19 | 79 | 7 | 79 | **3** | 82 |
| 20 | 78 | **73** | 78 | 74 | 84 |
| 21 | 265 | **253** | 281 | 257 | 286 |

nd=not detected

the other methods, as it was stated in [18].

The evolution of $BIC_Q$ and $Q$ is shown in Figure 2 when Fault 11 is considered. All methods are able to detect the fault immediately after its emergence. But, this fault is a random variation (see Table I), so the fault appears and disappears continuously. And this behaviour is detected by the methods, going up and down of their thresholds. DPCA is the least sensitive as it has more observations below the limit. Also, the fault type explains why MDR rates are so high in some faults, because MDR considers as faulty observations all the samples after the fault appears. The same happened for $T^2$.

In any case, the proposed three decentralization strategies

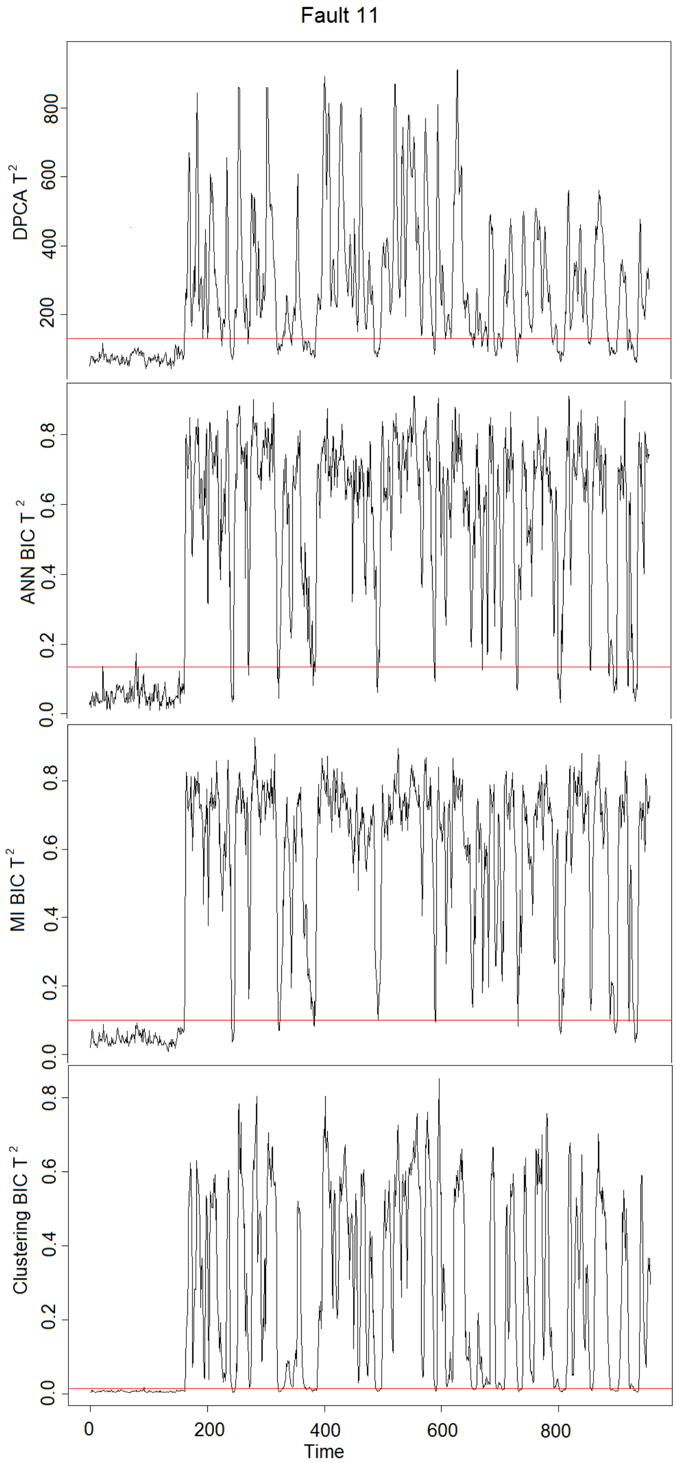


Figure 2. Fault 11. $Q$ and $BIC_Q$ indexes.

achieved better results with both statistics than the central DPCA, showing that decentralized approaches are preferable to centralized one. And, in view of the results, more blocks imply more faults detected and lower MDR and lower fault detection delay. Also, the proposed method ANN was able to improve the results of the other decentralization strategies, while MI method were able to get better, or at least, equal results than the WDDPCA method, with the advantage of having smaller number of blocks. The Clustering method, with the smallest number of blocks, worked better than the centralized approach and, also, it delivered results not too far from the remaining methods of the comparison.

After considering the results, it is clear that when working with big plants, which have a large number of sensors and the maximum number of blocks are restricted, it is possible to develop a distributed monitoring method with a reduced number of blocks and it will be assured that the results will be better compared with non distributed methods.

## V. CONCLUSIONS

Some decentralized monitoring methods were presented in this paper. All of them analyse the relation between the measured variables to decide how to group them without having previous knowledge about the plant. The decentralization were based on: neural nets, mutual information and clustering. After the decentralization, a DPCA model was developed in each block to process the measures and send the results to a central processor, which fuses all local results using BIC.

This research tried to find how different decentralization methods perform, and the results of the tests showed that two of the proposed methods, ANN and MI, worked better than the other decentralization strategy, WDDPCA, while all the three proposals were better than a centralized DPCA in terms of number of faults detected, MDR and detection delay. The proposals provide different options for decentralization: from a full decentralization (one block per variable) to a reduced decentralization, making it possible to work with different plant sizes or computational resources. Also, ANN, MI and Clustering included less variables per block than WDDPCA, making the data processing task easier and faster.

For future work, it will be interesting to use more decentralization methods trying to find more effective techniques. Also, it will be advisable to use different MSPM techniques to detect faults.

## REFERENCES


[1] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 16–25, 2017.

[2] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[3] S. Qin, "Data-driven fault detection and diagnosis for complex industrial processes," in *Proceedings of the 7th IFAC Symposium on Fault Detection and Supervision and Safety of Technical Processes*, pp. 1115–1125, 2009.

[4] M. Kano, S. Hasebe, I. Hashimoto, and H. Ohno, "Evolution of multivariable statistical process control: aplication of Independent Component Analysis and external analysis," *Computers & Chemical Engineering*, vol. 28, pp. 1157–1166, 2004.

[5] E. L. Russell, L. H. Chiang, and R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, 2000.

[6] T. Kourti and J. MacGregor, "Multivariate SPC methods for process and product monitoring," *Journal of Quality Technology*, vol. 28, pp. 409–428, 1996.

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

[8] W. Ku, R. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 30, no. 1, pp. 179–196, 1995.

[9] C. Tong and X. Shi, "Decentralized monitoring of dynamic processes based on dynamic feature selection and informative fault pattern dissimilarity," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 3804–3814, June 2016.

[10] C. Tong, T. Lan, and X. Shi, "Fault detection and diagnosis of dynamic processes using weighted dynamic decentralized PCA approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 161, no. Supplement C, pp. 34 – 42, 2017.

[11] A. Sanchez-Fernández, M. J. Fuente, and G. I. Sainz-Palmero, "Fault detection in wastewater treatment plants using distributed PCA methods," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–7, Sept 2015.

[12] M. Grbovic, W. Li, P. Xu, A. Usadi, L. Somg, and S. Vucetic, "Decentralized fault detection and diagnosis via sparse PCA based decomposition and maximum entropy decision fusion," *Journal of Process Control*, vol. 22, pp. 738–750, 2012.

[13] Z. Ge and Z. Song, "Distributed PCA model for plant-wide process monitoring," *Industrial and Engineering Chemistry Research*, vol. 52, pp. 1947–1957, 2013.

[14] Y. Zhang, H. Zhou, S. Qin, and T. Chai, "Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 3–10, 2010.

[15] W. Li, W. H. Gui, Y. F. Xie, and S. X. Ding, "Decentralised fault detection of large-scale systems with limited network communications [brief paper]," *IET Control Theory Applications*, vol. 4, pp. 1867–1876, September 2010.

[16] S. Qin, S. Valle, and J. Piovoso, "On unifying multiblock analysis with application to decentralized process monitoring," *Journal of Chemometrics*, vol. 15, pp. 715–742, 2001.

[17] G. A. Cherry and S. J. Qin, "Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 159–172, 2006.

[18] A. Sanchez-Fernandez, M. J. Fuente, and G. I. Sainz-Palmero, "Decentralized and dynamic fault detection using PCA and bayesian inference," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 800–807, Sep. 2018.

[19] J. Jackson and G. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 3, no. 21, pp. 341–349, 1979.

[20] W. E. Larimore, *Statistical methods in control and signal processing*. Marcel Dekker, 1997.

[21] R. S. Kaushik Ghosh, Yew Seng Ng, "Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods," *Computers and Chemical Engineering*, vol. 35, pp. 342–355, 2011.

[22] T. M. Cover and J. A. Thomas, *Elements of information theory*. Wiley John + Sons, 2006.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," pp. 226–231, AAAI Press, 1996.

[24] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, pp. 245–255, 1993.

[25] P. Odiowei and Y. Cao, "State-space independent component analysis for nonlinear dynamic process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 103, pp. 59–65, 2010.

[26] C.-Y. Chen and Y. Yao, "Robust process monitoring via stable principal component pursuit," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 617–622, 2015.

[27] Q. Zhu, Q. Liu, and S. J. Qin, "Concurrent monitoring and diagnosis of process and quality faults with canonical correlation analysis," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7999–8004, 2017.

[28] S. Heo and J. H. Lee, "Fault detection and classification using artificial neural networks," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 470–475, 2018.