IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 February 2023, accepted 4 March 2023, date of publication 13 March 2023, date of current version 21 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3256719

RESEARCH ARTICLE

Dynamic Decentralized Monitoring for Large-Scale Industrial Processes Using Multiblock Canonical Variate Analysis Based Regression

MARÍA JESÚS DE LA FUENTE, GREGORIO I. SAINZ-PALMERO, AND MARTA GALENDE-HERNÁNDEZ
Department of System Engineering and Automatic Control, School of Industrial Engineering, Universidad de Valladolid, 47011 Valladolid, Spain

Corresponding author: Gregorio I. Sainz-Palmero (gregorioismael.sainz@uva.es)

This work was supported by the Spanish Government through the Ministerio de Ciencia e Innovación (MICINN) / Agencia Estatal de Investigación (AEI) under Grant PID2019-105434RB-C32/AEI/10.13039/501100011033.

ABSTRACT Decentralized monitoring methods, which divide the process variables into several blocks and perform local monitoring for each sub-block, have been gaining increasing attention in large-scale plant-wide monitoring due to the complexity of their processes. In such methods, the dynamic nature of the process data is a relevant issue which is not usually managed. Here, a new data-driven distributed dynamic monitoring scheme is proposed to deal with this issue, integrating regression to automatically divide the blocks, a multivariate and dynamic statistical analysis (Canonical Variate Analysis, CVA) to perform local monitoring, and Bayesian inference to achieve the decision making. By constructing sub-blocks using regression, it is possible to identify the most commonly associated variables for every block. Three regression methods are proposed: LASSO (Least Absolute Shrinkage and Selection Operator), which forces the coefficients of the less relevant variables towards zero; Elastic-net, a robust method that is a compromise between Ridge and Lasso regression; and, finally, a non-linear regression method based on the Multilayer Perceptron Network (MLP). Then, the CVA model is implemented for each sub-block to consider the dynamic characteristics of the industrial processes and the Bayesian inference provides a global decision for fault detection. The Tennessee Eastman benchmark validates the efficiency and feasibility of the proposed method regarding some state-of-the-art methods.

INDEX TERMS Fault detection, canonical variate analysis, regression, decentralized process monitoring, Bayesian inference.

I. INTRODUCTION

Industrial plants are usually characterized by large-scale, multiple operation units and complex interactions that make them very prone to suffering anomalies. This means that the monitoring of such processes is an important issue, warning of faults or unexpected behaviors of the installation. This monitoring system must be fast enough to achieve early detection for any issue, allowing prompt action. This can decrease the chance of expensive breakdowns, work related accidents and product quality losses. Nowadays, given the extensive use of distributed control systems, process data have become abundant, and data-based monitoring methods, particularly multivariate statistical process monitoring (MSPM), have received significant interest [5], [9], [24], [38].

Of the various MSPM methods, Principal Component Analysis (PCA) and Partial Least Squares (PLS) are the most usual techniques for monitoring processes [1], [22], [39]. However, as industrial plants are normally non-linear and dynamic, some extensions of these methods have been developed to deal with these challenges, such as the Kernel methods (KPCA, KPLS) for non-linear data [4], [29]. Dynamic methods (DPCA, DPLS) using an augmented input matrix with time-lagged variables have been used to take

The associate editor coordinating the review of this manuscript and approving it for publication was Baoping Cai.

into account the process dynamics characterized by cross and auto-correlation [7], [14], [26], [36]. Moreover, several works use subspace modeling to monitor such dynamic processes, as Canonical Variate Analysis (CVA) [15], [16], where past data and future measurements are used to estimate the process state space model and build a fault detection scheme.

In large-scale plants, these centralized monitoring methods have to deal with a growing amount of data, as plants are fitted with more and more sensors, all of which means serious powerful processing needs. In addition this can usually imply the monitoring process will be slower and very sensitive to faults: if one variable is unavailable, the communication channel is blocked or the central processor stops working, then the whole monitoring system may stop functioning. So, a multiblock method is an alternative approach to manage these problems, as it provides an effective description for the large-scale process, dividing the system into multiple blocks of measured variables to reduce the complexity [10], [13]. This is a decentralized approach, where each block gathers measurements from a reduced number of sensors, processes them and delivers one local fault detection result. After that, one central processor collects all the local outcomes, analyzes them and decides whether there is a fault or not in the system.

Decentralized monitoring has been widely studied in recent years, showing advantages in monitoring for large-scale processes over centralized methods; especially in the reduction of the analysis complexity, the description of the multivariate models and the improvement of the monitoring performance [12], [13], [18], [28], [33]. However, the first challenge for these decentralized approaches is the decomposition process, i.e., the way in which the plant is divided is key. Traditional decomposition methods typically obtain blocks of variables based on prior knowledge or process topology [25], [28], [40]. However, in industrial plants, accurate knowledge of the block divisions is barely available. In these cases, data-driven process decomposition methods, which automatically divide the variables into overlapping or disjoint blocks, must be taken into account. So, these data-driven methods have two main advantages: they do not need knowledge of the process and perform the plant division automatically.

In this way, [12] proposes a decentralized fault detection and diagnosis method via sparse PCA-based decomposition and maximum entropy decision making. In [10], a distributed PCA is shown for plant-wide process monitoring through building blocks in different principal component directions. On the other hand, [19] uses a stochastic optimization algorithm based on performance-driven distributed monitoring for process decomposition and PCA for local monitoring; while [37] proposes minimal redundancy and maximal relevance (mRMR)-PCA based monitoring of plant-wide processes, [30] and [31] propose copula-correlation analysis, taking into account both the correlation degree and the correlation pattern for block division, also using PCA for local monitoring, and [13] develops a measurement which estimates the dependence and skewness of data to decompose the plant into blocks and also uses PCA for local monitoring. All these methods build sub-block monitoring models individually, but ignore the relevance of different blocks. Taking this issue into account, [35] proposes a modified MBPCA method that extracts block scores with respect to both the specificity in each block and the relevance of the different blocks.

However all these distributed approaches only consider the static process variation, ignoring the dynamic characteristics of the industrial processes due to changing demand or disturbances, which hinders their wider applications. More recently, taking into account the dynamic nature of the processes, [34] and [33] have developed dynamic decentralized PCA (DDPCA) and weighted dynamic decentralized PCA (WDDPCA) approaches, using the correlation between the variables in different time instants. Reference [41] has used the mRMR method to decentralize the plant using an augmented input matrix with time-lagged variables and the PCA for monitoring purposes in each block (mRMR-DDPCA), but the majority of these methods to decompose plants are linear and based only on the correlation degree between variables, which can lead to inaccurate partitioning that may to affect the monitoring performance.

In this work, we use linear and non-linear regression methods to decentralize the plant to generate models, identifying the most strongly linked variables and/or the least relevant ones, providing support to the sub-block division for monitoring by the CVA based approach, a multivariate dynamic monitoring method which takes into account the dynamic characteristics of the processes.

To sum up, the main contribution and goal of this paper is to propose a novel data-driven dynamic and decentralized fault detection method to monitor large-scale processes based on regression and CVA (R-DCVA) methods, without the need for knowledge regarding these processes. Firstly, three different regression methods are used to divide the plant variables into blocks: the regression based on neural networks (MLPR) for taking into account the non-linear nature of industrial plants, the LASSO regression (LR) method, which considers both the precision accuracy and the interpretability to build a model, and finally the elastic net regression (ENR) that selects variables such as LASSO and shrinks the coefficients according to Ridge, so it is more robust when the variables are highly correlated. These regression methods are supervised, so they permit the best fit and tuning for each block.

Following that decomposition, a local CVA algorithm is established for every sub-block for fault detection, taking into account the dynamic nature of the industrial processes. Then, the well-known Bayesian inference strategy [3], [11] is adopted to combine the different blocks monitoring outcomes in a global decision making. The majority of the decentralized methods considered in the literature [10], [13], [17], [28], [18], [30], [31], [34], [35], and [41] use Principal Component Analysis (PCA) for monitoring purposes. As far as we know, there are no references using the CVA model in distributed monitoring schemes.

On the other hand, this scheme permits the behavior and performance of the decomposition process to be evaluated in terms of monitoring results, showing that the proposed non-linear decomposition shows a better performance in most of the cases when a CVA model is implemented for each obtained sub-block to analyze its local dynamic behavior.

All this is carried out under an experimental cross-validation scheme for better robustness.

The rest of the paper is organized as follows. Section II provides some background knowledge of CVA, explains the different methods to perform the decentralization and details the Bayesian inference fusion technique. Section III elaborates the proposed R-DCVA method: a dynamic and decentralized fault detection method. The effectiveness of this proposal is tested on the Tennessee Eastman Plant and the results are summarized in Section IV, followed by the conclusions in Section V.

II. PRELIMINARIES

A. CANONICAL VARIATE ANALYSIS

Canonical Variate Analysis (CVA) is a dimensionality reduction technique in multivariate statistical analysis which maximizes the correlation between two selected sets of variables. CVA has been proposed for multivariate statistical analysis and was also developed to identify state-space models [21]. Given time series output data $\boldsymbol{y}(t) \in \Re^{m_y}$ and input data $\boldsymbol{u}(t) \in \Re^{m_u}$, the linear state space model is represented by [27]:

$$\begin{aligned} \boldsymbol{x}(t+1) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{e}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{e}(t) + \boldsymbol{w}(t) \end{aligned} \tag{1}$$

where $\boldsymbol{x}(t) \in \Re^{d}$ is the state vector, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{E}$ are matrices of coefficients and $\boldsymbol{e}(t)$ and $\boldsymbol{w}(t)$ are independent white noise processes.

The vector $\boldsymbol{p}(t)$ represents the past information:

$$\begin{aligned} \boldsymbol{p}(t) = [&\boldsymbol{y}^T(t-1), \boldsymbol{y}^T(t-2), \ldots, \boldsymbol{y}^T(t-l), \\ &\boldsymbol{u}^T(t-1), \boldsymbol{u}^T(t-2), \ldots, \boldsymbol{u}^T(t-l)]^T \end{aligned} \tag{2}$$

and the vector $\boldsymbol{f}(t)$ includes the present and future information about the output of the plant:

$$\boldsymbol{f}(t) = [\boldsymbol{y}^T(t), \boldsymbol{y}^T(t+1), \ldots, \boldsymbol{y}^T(t+h)]^T \tag{3}$$

For an assumed state order k, the CVA algorithm computes a constant matrix $\boldsymbol{J}_k$ that linearly relates the past vector $\boldsymbol{p}(t)$ to the reduced state vector $\boldsymbol{x}_k(t)\epsilon\Re^k$. The optimal matrix $\boldsymbol{J}_k$ is calculated via the singular value decomposition as:

$$\left(\boldsymbol{\Sigma}_{pp}\right)^{-1/2} \boldsymbol{\Sigma}_{pf} \left(\boldsymbol{\Sigma}_{ff}\right)^{-1/2} = \boldsymbol{U\Sigma V^T} \tag{4}$$

where $\boldsymbol{\Sigma}_{pp}$, $\boldsymbol{\Sigma}_{ff}$ and $\boldsymbol{\Sigma}_{pf}$ are the covariances of $\boldsymbol{p}(t)$, $\boldsymbol{f}(t)$ and the cross-covariance of $\boldsymbol{p}(t)$ and $\boldsymbol{f}(t)$, respectively. $\boldsymbol{\Sigma}$ is the diagonal matrix of non-negative singular values with descending order, while $\boldsymbol{U}$ and $\boldsymbol{V}$ are matrices of the right and left singular vectors. The matrix $\boldsymbol{J}_k$ is obtained by

$$\boldsymbol{J_k} \; \boldsymbol{\mathcal{D}} \; \boldsymbol{U_k^T} \left(\boldsymbol{\Sigma_{pp}}\right)^{-1/2} \tag{5}$$

where $\boldsymbol{U}_k$ contains the first k columns of $\boldsymbol{U}$. The value k is selected to be greater than or equal to the order of the minimal state-space realization of the plant, so the state vector $\boldsymbol{x}_k(t)$ is obtained as:

$$\boldsymbol{x}_k(t) = \boldsymbol{J}_k\boldsymbol{p}(t) = \boldsymbol{U}_k^T(\hat{\boldsymbol{\Sigma}}_{pp})^{-1/2}\boldsymbol{p}(t) \tag{6}$$

Finding the values for l and h, i.e., the lags to include in the input vectors, is not a trivial task. One option to solve this consists of fitting autoregressive models using different numbers of lags and selecting the values for l and h that minimize the Akaike Information Criterion (AIC) [21].

To use the CVA algorithm for statistical monitoring, there are two types of statistics: T_s^2, for the variations in the canonical subspace, and T_r^2, for the variations inside the residual subspace [27]:

$$\begin{aligned} T_s^2 &= \boldsymbol{x}_k^T(t)\boldsymbol{x}_k(t) \\ T_r^2 &= \boldsymbol{x}_r^T(t)\boldsymbol{x}_r(t) \end{aligned} \tag{7}$$

where $\boldsymbol{x}_r^T(t) = \boldsymbol{J}_r\boldsymbol{p}(t) = \boldsymbol{U}_r^T(\hat{\boldsymbol{\Sigma}}_{pp})^{-1/2}\boldsymbol{p}(t)$ and $\boldsymbol{U}_r$ are the remaining $l(m_u + m_y) - k$ columns of $\boldsymbol{U}$ after extracting $\boldsymbol{U}_k$. The state of the process is determined using the thresholds of these statistics [27]. If T_s^2 is over its limit, the states of the system are under abnormal conditions. Conversely, if T_r^2 passes its threshold, this indicates that the noise pattern has changed or new states have appeared.

Another possibility to detect faults is using the residual vector of the state space model:

$$r(t) = (\boldsymbol{I} - \boldsymbol{J}_k^T\boldsymbol{J}_k)\boldsymbol{p}(t) \tag{8}$$

which allows the statistic Q to be obtained:

$$Q(t) = r^T r \tag{9}$$

which measures the variation in the residual subspace.

As before, if Q passes its threshold [27], a fault is detected and that indicates the existence of new states or a different noise pattern.

B. MODEL DECOMPOSITION BASED ON REGRESSION

In this paper, three regression based methods to perform the model decomposition, i.e., to divide the plant variables into blocks, are considered: one non-linear (MLP network) and two linear (LASSO and Elastic net). All of them are experimented under a *k-cross validation* scheme for better robustness and significance of their results.

1) ARTIFICIAL NEURAL NETWORKS

Most industrial plants are non-linear and, for these cases, some non-linear modeling techniques, such as the Multilayer Perceptron (MLP) neural network [20], are the most suitable. This is a very popular supervised technique, which is capable of approximating any continuous function as accurately as necessary. The computing units of this network (neurons) are connected to each other by connections (weights) conforming the network which is able to learn patterns provided by the data over the training stage, updating the connection strength. These weights, which connect each input with the output through various neurons, perform the feature selection task [6].

2) LASSO METHOD

The LASSO (Least Absolute Shrinkage and Selection Operator) method was introduced by Tibshirani in 1996 [32] as an improvement for linear regressions. It penalizes some coefficients of the regression, making some of them zero. This makes the model simpler and more interpretable. This technique can also be used to perform feature selection, using only the most relevant regressors, which have non-zero coefficients [23]. Our proposal considers the LASSO method for performing variable selection.

A linear model with m predictors: $X(t) = (x_1(t), \ldots, x_m(t))$ and one response variable $y(t)$ can be expressed as follows:

$$y(t) = \beta_0 + \beta_1 x_1(t) + \beta_2 x_2(t) + \ldots + \beta_m x_m(t) + \varepsilon(t) \tag{10}$$

where β_i ($i = 1, \ldots, m$) are the regression coefficients and $\varepsilon(t)$ is the model error at time t. Using matrix notation:

$$Y = \mathbf{X}\beta + E \tag{11}$$

The LASSO method solves the problem:

$$\hat{\beta}(\lambda) = \underset{\beta}{\text{argmin}}\left(\frac{\parallel y - \mathbf{X}\beta \parallel_2^2}{n} + \lambda \parallel \beta \parallel_1\right) \tag{12}$$

where $\lambda \geq 0$. This λ controls the process: as its value increases, more coefficients are forced to be zero.

3) ELASTIC NET METHOD

The elastic net regression was introduced by [42] and can be seen as a compromise between Ridge and LASSO regression, i.e., it selects variables such as LASSO and shrinks the coefficients according to Ridge. So, the elastic net regression solves the $\hat{\beta}(\lambda, \delta)$ optimization problem as:

$$\underset{\beta}{\text{argmin}}\left(\frac{\parallel y - \mathbf{X}\beta \parallel_2^2}{n} + \lambda\left(\frac{1-\delta}{2} \parallel \beta \parallel_2^2 + \delta \parallel \beta \parallel_1\right)\right) \tag{13}$$

The elastic net is more flexible, and for $\delta = 1$, it gives the LASSO solution and for $\delta = 0$ the Ridge regression is obtained. A combination of penalizing both, i.e., when the value of δ is in the interval of $[0, 1]$, gives good results. A frequent strategy is to assign a big value to the $l1$ penalization in order to get a lower number of predictors, i.e., to put a value of δ near to 1, and gives a little weight to the $l2$ regularization to give some stability if some of the predictors are highly correlated.

C. BAYESIAN INFERENCE (BI)

While the fault detection method in a centralized approach returns one fault index for each statistic; in a decentralized method, each block returns its own fault indexes. In most cases of decentralized monitoring, a fault can be declared if it is observed at any location. However, it is important to stress that the fusion of multiple monitoring results would greatly influence the global monitoring performance. So, in order to obtain a global and outperformed result, a method which collects local results and fuses them to get a unique and global result per statistic is carried out. Various decision fusion strategies can be used, though the Bayesian Inference is the most popular one to fuse fault indexes [19], [30], [34], giving a single result for the whole plant.

For the statistic ST (ST can be T_s^2, T_r^2 or Q) in block i ($i = 1, 2, \ldots, B$), B being the number of blocks, the posterior fault probability is calculated as:

$$P_{(F|x_i)} = P_{(x_i|F)} P_{(F)} / P_{(x_i)} \tag{14}$$

where

$$P_{(x_i)} = P_{(x_i|N)} P_{(N)} + P_{(x_i|F)} P_{(F)} \tag{15}$$

Here, N and F are the normal and abnormal conditions of the plant, respectively. $P_{(N)}$ and $P_{(F)}$ are prior probabilities for the non-faulty and faulty system. $P_{(N)}$ is adjusted to an α value, which goes from 0 to 1, and $P_{(F)}$ is set to $1 - \alpha$. Also:

$$\begin{aligned} P_{(x_i|N)} &= exp(-ST/ST_{i,lim}) \\ P_{(x_i|F)} &= exp(-ST_{i,lim}/ST) \end{aligned} \tag{16}$$

where $ST_{i,lim}$ is the corresponding threshold for ST in block i.

The final statistic for ST in the whole plant can be determined to combine monitoring results from different blocks as follows [11] and [3]:

$$BII_{ST} = \sum_{i=1}^{B} \{ \frac{P_{(x_i|F)} P_{(F|x_i)}}{\sum_{i=1}^{B} P_{(x_i|F)}} \} \tag{17}$$

If the BII value for the statistic ST is over $(1 - \alpha)$, a fault is detected with this statistic.

III. REGRESSION-BASED DCVA SCHEME (R-DCVA)

In large-scale processes, the number of measured variables is usually very large, which usually means that the results of monitoring all these data is quite difficult to interpret, and the local behaviors of the system would not be sufficiently reflected. The aim of a decentralized method is to divide the process variables into several blocks, overlapped or not. Thus, the local system behavior can be better explained when a local monitoring method is carried out. The final decision depends on the local results for every block and a decision-making method is used to obtain a global decision about the state of the whole plant.

A critical step for decentralized methods is the division of the plant variables into blocks. If these blocks are obtained unreasonably, the monitoring results will be more difficult to interpret. So, a main contribution of this work is the division of the large-scale plant into fair blocks of variables using regression methods, which are supervised and permit the best variables for each one to be tuned. The monitoring method implemented for each block is based on CVA models, paying attention to the dynamic behavior of the industrial processes.

Suppose that a set of training data, $\boldsymbol{X} \in \Re^{n \times m}$, are collected from the large-scale plant in normal conditions, where n is the number of observations and m the number of variables. Now, these data are used to calculate a LASSO, an Elastic net and an MLP model per measured variable $Y = x_i$, $i = 1, \ldots m$, while the other variables x_j, $j = 1, \ldots, m$ and $j \neq i$ are used as predictors.

The three regression techniques: LASSO, Elastic net and MLP, need to tune hyperparameters, and a k-cross validation scheme has been used for this target. The LASSO regression technique requires a penalty value (λ) to be set for the regression process (see eq. 12). A range of different values must be tested to obtain different regression models for each variable using k-cross validation, then the best model, i.e., the λ value generating the model with the lowest average Root Mean Squared Error (rMSE), is selected. The Elastic net regression technique requires two penalty values (δ and λ) to perform the regression model (see eq. 13). So, a grid search for different values for those parameters is carried out for each variable, also using the k-cross validation; after that, the best regression model is selected. On the other hand, the regression models using MLP networks must be similarly tested by different values for the ordinary parameters of the MLP (number of hidden layers, number of neurons in each layer, etc.), choosing for the plant division the parametrized model that best fits the data (minimum average rMSE) through k-cross validation.

As a CVA method is used for monitoring purposes, i.e., a state space model is applied for every block to monitor the process, then the variables considered for each of these blocks must be the least correlated variables. So:

- In the LASSO method, a block is created for each measured variable x_i, $i = 1, \ldots, m$, including this variable and those others with zero coefficient in the respective LASSO regression model.
- In the Elastic net method, a block is created for each measured variable x_i, $i = 1, \ldots, m$, including this variable and those others that have obtained coefficients below a certain threshold in its respective Elastic-net regression model, i.e., the variables x_j with a coefficient $\beta_j < l_i$ (see eq. 10) have to be in the same block with x_i. This threshold l_i must be defined for each i-th model experimentally, in this case, the mean value of all the coefficients in the respective model.
- In the MLP model, the group for each variable x_i, $i = 1, \ldots, m$ is made up of this variable and those that have obtained the smallest score in their respective model. This score is calculated as the product of the synaptic weights that connect each input with the output through the neurons in the neural network [6], i.e.,

$$R_{ij} = \sum_{k=1}^{H} W_{jk} W_{ki} \tag{18}$$

where R_{ij} is the relative importance, or score, of the input variable x_j, $j = 1, \ldots, m$ and $j \neq i$ with respect to the output neuron, i.e., for the variable x_i that we are modeling, H is the number of neurons in the hidden layer, W_{jk} is the synaptic connection weight between the input neuron j and the hidden neuron k, and W_{ki} is the synaptic weight between the hidden neuron k and the output neuron.

The variables x_j with a coefficient $R_{ij} < Nl_i$ must be in the same block with x_i. This threshold Nl_i must be defined for each variable x_i experimentally, and, in this case, it is the mean value of all the coefficients R_{ij} in the respective model.

The plant division can be written as:

$$X = [X_1 X_2 \ldots X_B] \tag{19}$$

where B is the number of blocks and $\boldsymbol{X}_i \in \Re^{n \times m_i}$ ($i = 1, 2, \ldots, B$), where n is the number of observations and m_i the number of variables for every block.

Once the system has been divided into B blocks, a local fault detection method based on CVA is implemented for every block. In this algorithm, the user defines the number of lags used, l and h, i.e., the lags to include in the input vectors (eqs. 2 and 3). This can be done by testing with different values and choosing those that perform the best in terms of fault detection, false alarms, fault detection delay, etc., or by fitting auto-regressive models using different numbers of lags and selecting the values for l and h so as to minimize the Akaike Information Criterion (AIC).

A CVA method is implemented for each block and three statistics ($T^2_{s,i}$, $T^2_{r,i}$ and Q_i) are obtained for each one. To detect a fault in a block means that some of these statistics have to exceed their own threshold, so it is also necessary to calculate thresholds for every statistic and every block: $\{T^2_{s,\alpha,i}$, $T^2_{r,\alpha,i}$ and $Q_{\alpha,i}\}$ with ($i = 1, \ldots, B$).

Then, when a new observation, x, is acquired, this is turned into B blocks in accordance with the selection results provided by the previous stage, i.e., using the variables indicated by the MLP, the Elastic net or the LASSO regression models. Subsequently, a local CVA model calculates the corresponding values for the statistics $T^2_{s,i}$, $T^2_{r,i}$ and Q_i. To simplify the final decision of triggering a fault alarm, a global outcome for each statistic is obtained, fusing the local monitoring results using the Bayesian inference strategy, see Section II-C, to transfer them to a definitive probabilistic one. Finally, the process is considered faulty if any BII index ($BII_{T^2_s}$, $BII_{T^2_r}$ or BII_Q) is over its respective threshold for an α confidence level.

The monitoring procedure is schematically shown in Algorithm 1 and the details are also provided as follows:

Off-line Modeling Phase:

1) Acquire normal operating data-set X.
2) Implement a process decomposition method based on data:

a) Using the LASSO method to divide the plant. Normalize the data set X to zero mean and unit variance and for each variable x_i $(i = 1, \ldots, m)$:
   - Build a LASSO model using different λ parameters and the other variables x_j, $j = 1, \ldots, m$ and $j \neq i$ as predictors using k-cross validation.
   - Select the best λ for LASSO regression modeling with the lowest average rMSE value for each variable.
   - Generate B blocks. One for each variable, including the corresponding variable x_i in that block, together with the variables that have zero coefficients in the respective LASSO model choose in the previous step.

b) Using the Elastic net method to divide the plant. Normalize the data set X between 0 and 1, and for each variable x_i $(i = 1, \ldots, m)$:
   - Build an Elastic net model using different δ and λ parameters and the other variables x_j, $j = 1, \ldots, m$ and $j \neq i$ as predictors using k-cross validation.
   - Select the best δ and λ for Elastic net regression modeling with the lowest average rMSE value for each variable.
   - Generate B blocks. One for each variable, including the corresponding variable x_i in that block, together with the variables that have coefficients below the threshold, l_i, in its respective Elastic net model.

c) Using the MLP method to divide the plant. Normalize the data set X between 0 and 1, and for each variable x_i $(i = 1, \ldots, m)$:
   - Build an MLP model using different parameters (number of hidden layers, number of neurons in each layer, etc.) through k cross-validations, with the other variables x_j, $j = 1, \ldots, m$ and $j \neq i$ as predictors.
   - Select, for each variable, the parametrized MLP model with the lowest average rMSE value through cross validation.
   - Generate B blocks. One for each variable, including the corresponding variable x_i in that block, together with the variables with lower scores, i.e., the variables with scores R_{ij} below the threshold Nl_i, in the respective MLP model.

3) Normalize the data set X to zero mean and unit variance and construct a CVA-based monitoring model for each block. Calculate the thresholds for each statistic and each block: $\{T^2_{s,\alpha,i}, T^2_{r,\alpha,i}$ and $Q_{\alpha,i}\}$ for $i = 1, \ldots, B$

The on-line monitoring phase:

1) Get a new sample data set x and divide it into different blocks, as done in the training phase.
2) Calculate the three monitoring statistics, i.e., $T^2_{s,i}$, $T^2_{r,i}$, Q^2_i, for every block, $i = 1, \ldots, B$, according to equations 7 and 9.
3) Implement the Bayesian inference based decision fusion strategy to get just the three final monitoring indexes: $BII_{T_s^2}$, $BII_{T_r^2}$ or BII_Q.
4) When the values of $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q are below the control limit $(1 - \alpha)$, the system is operating in normal conditions. Otherwise, the monitored sample is abnormal and a fault triggered alarm is sent to the operator.

IV. CASE STUDY

This section introduces the main results from applying the proposed R-DCVA methodology to a well-known benchmark: the Tennessee Eastman Process [2], [8]. The performance of the R-CVA method has been evaluated by the following indexes:

- False Alarms Rate (FAR): percentage of non-faulty samples classified as faulty; i.e., this index takes into account the robustness of each statistic and it is determined by calculating the false alarm rate during normal operating conditions and comparing it against the level of significance upon which the threshold is based.
- Missed Detection Rate (MDR): percentage of faulty measures classified as faultless data, thus quantifying the sensitivity to possible faults.
- Fault Detection Delay (FDD): how many samples are needed to detect a fault after its occurrence.
- Number of Faults Detected (NFD): the success rate.

A. TENNESSEE EASTMAN PROCESS (TEP)

The Tennessee Eastman Process [8] and revisited by [2] is a well-known benchmark used to test the proposal; it is widely used to test fault diagnosis techniques [1], [7], [10], [13], [19], [30], [31], [33], [34], [41]. This benchmark is a model of a chemical plant which includes five major unit operations: a reactor, a condenser, a vapor-liquid separator, a recycle compressor and a stripper. A set of 52 variables can be measured, including 22 continuous variables, 12 manipulated variables, and 19 composition measurements. The schematic of this plan can be seen in Fig. 1. The available data, which can be downloaded from http://web.mit.edu/braatzgroup, are formed by faultless training and testing data, as well as faulty data, with 21 data-sets representing 21 different faults, where real faults and also disturbances that are important to detect are taken into account. The cited 21 faults are included in Table 1. Here, the faultless data used for tuning LASSO, Elastic net and MLP models are organized over a scheme of 3-cross validation, using 2/3 for training and 1/3 for testing.

B. EXPERIMENTAL SETUP

The proposed R-DCVA has been tested on this plant. According to the three different methods to divide the plant and to obtain the blocks for the CVA based monitoring, three monitoring approaches were set up: one based on LASSO regression (LR-DCVA), another on Elastic net regression (ENR-DCVA) and, finally, the last based on MLP modeling (MLPR-DCVA).

Algorithm 1 Decentralized Monitoring With CVA Fault Detection R-DCVA

```
Off-line steps:
Normalize data
Look for Blocks of Variables:
LASSO, Elastic net or MLP modeling using faultless data:
Generate $B$ blocks using LASSO decentralization:
for i=1 to NumberOfVariables do
    Tuning LASSO models through 3-cross validation
    Select $\lambda$ for LASSO modeling with lowest average rMSE
    Generate $B_i$ through the variables from LASSO model obtained by 3-cross validation.
end for
OR
Generate $B$ blocks using Elastic net decentralization:
for i=1 to NumberOfVariables do
    Tuning Elastic net models through 3-cross validation
    Select $\delta$ and $\lambda$ for Elastic net modeling with lowest average rMSE
    Generate $B_i$ through the variables from the best Elastic net model obtained.
end for
OR
Generate $B$ blocks using MLP decentralization:
for i=1 to NumberOfVariables do
    Create MLP models with different parameters through 3-cross validation
    Select MLP model with lowest rMSE
    Generate $B_i$ blocks using MLP decentralization
end for
CVA model for every $B_i$ Block & its statistics:
for i=1 to B do
    Develop local $CVA_i$ models tuning over different lags ($l$ and $h$)
    Select the best $CVA_i$ model
    Calculate statistics thresholds: $ST^j_{\alpha,i} = \{T^2_{s,\alpha,i}, T^2_{r,\alpha,i}, Q_{\alpha,i}\}$
end for
On-line steps:
Fault detection:
for New observation $x$ do
    Divide the observation $x$ in B blocks
    for i=1 to B do
        Obtain Local statistics: $ST_i = \{T^2_{s,i}, T^2_{r,i}, Q_i\}$
    end for
    Global Decision
    for s=$\{T^2_{s,i}, T^2_{r,i}, Q_i\}$ do
        $BII_s = f(ST_1, ST_2, \ldots, ST_B)$
        if $BII_s \geq (1-\alpha)$ then
            Fault detected with statistic s
        else
            Normal condition with statistic s
        end if
    end for
end for
```

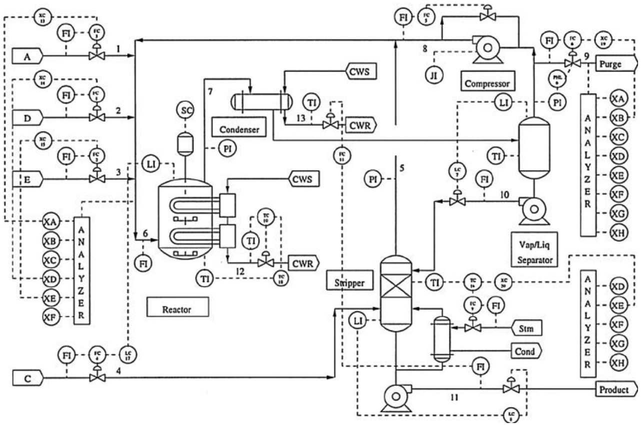


FIGURE 1. Tennessee eastman process diagram.

TABLE 1. Tennessee Eastman process faults.

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

These three alternatives were tested with different parameters for the CVA order (l and h), the number of consecutive anomalous observations to detect a fault and, also, the specific parameters of each method used to decompose the plant into blocks. Other parameters are similar for all the methods: such as the local thresholds used for the statistics of each block, adjusted to obtain a level of significance of 99%, as well as the threshold for the global decision, i.e., the threshold for the BII index that was adjusted to obtain a significance level of 95%, resulting in a value of $\alpha = 0.9$. For the rest of the parameters, after many tests, the best setup in terms of absence of false alarms, number of faults detected and best detection times was:

- LASSO regression: A LASSO based model was tuned for each variable of the system using 3-cross validation, looking for the minimum average rMSE. The parameter λ of eq. 12 was individually tuned for every variable. The order for the CVA models in each block, i.e., the parameters l and h in equations 2 and 3, were such that 4 and 5 consecutive anomalous observations are necessary to detect a fault.

- Elastic net regression: An Elastic net based model was tuned for every variable of the system using 3-cross validation, looking for the minimum average rMSE. The parameters δ and λ of eq. 13 were individually tuned for every variable in a grid search for both parameters. The block for a certain variable was made up of that variable and those others with coefficients of the respective model below a certain threshold calculated as the mean value of the coefficients in each model. The order for the CVA models in each block, i.e., the parameters l and h in equations 2 and 3, were such that 2 and 6 consecutive anomalous observations are necessary to detect a fault.
- Multilayer Perceptron network (MLP). A model was built for each variable using the MLP network. The MLP models were tuned (number of hidden layers, number of neurons in each layer, etc.) using 3-cross validation, selecting the tuning with the best average rMSE. The blocks were created using these models: the group for a certain variable was made up of that variable and those which obtained the smallest scores in the respective model, i.e., the variables with a score below a certain threshold. This threshold was calculated as the mean value of the relevance values of the variables in each model. For the remaining parameters: the order for the CVA models of each block were 5, and 5 consecutive anomalous observations are necessary to detect a fault.

C. RESULTS AND DISCUSSION

The proposed methods: LR-DCVA, ENR-DCVA and MLPR-DCVA have been tested to monitor this dynamic and complex plant. A comparison regarding other methods and approaches found in the literature ([13], [27], [31], [34], [41]) and applied over this benchmark and with the same dataset, has been carried out to illustrate the advantageous effectiveness and performance of the LR-DCVA, ENR-DCVA and MLPR-DCVA methods.

Two of the methods for comparison are centralized: the original dynamic PCA model (DPCA) and the CVA method [27], and four are decentralized ones: the DDPCA (Distributed DPCA) with a cut-off parameter of $\delta = 1.5$ proposed by [34]; the weighted copula-correlation-based multiblock PCA (WCMBPCA) proposed by [31]; a more recently proposed monitoring method that uses the minimal redundancy maximal relevance method to divide the plant and a dynamic PCA for local monitoring (mRMR-DDPCA) [41], and the Weighted dependence and skewness based multiblock PCA (WDSMBPCA) proposed by [13].

1) ALARM RATES

First of all, the false alarm rates are examined by monitoring a different normal dataset. The false alarm rate for the considered methods and the number of faults detected are shown in Table 2. It can be seen that LR-DCVA and MLPR-DCVA were able to detect more faults than the other methods, both centralized (DPCA and CVA) and decentralized ones. In particular, the LR-CVA, which found every one of the 21 faults with the T_s^2 statistic.

The fault alarm rate (FAR) for the test data, shown in the same Table, takes values from 0 to 0.4 for LR-DCVA and MLPR-DCVA, clearly outperforming the rest of the methods in the comparative. However the results for the ENR-DCVA are worse. This points to the fact that it is necessary to be careful to determine the control limits for the statistics. In the cases of LR-DCVA, ENR-DCVA and MLPR-DCVA, the control limit of each monitoring index is adjusted to obtain a significance level of 99%, and to avoid these false alarms in engineering practice, a fault alarm is triggered after detecting consecutively 5 abnormal samples. So, for the methods considered in this paper, taking this into account, the false alarm rate is always 0. Due to the high value of the FAR for the CVA and DPCA methods, the control limits of these statistics were modified experimentally in order to obtain the missing detection rate (MDR), as is explained in [27].

2) MISSING DETECTION RATES

The missing detection rates (MDR) for the 21 faults are listed in Table 3, as well as the mean of the MDR (MMDR) for 18 faults, as faults 3, 9 and 15 are not detected by nearly any method. The same Table shows the mean for the different statistics, T_s^2, T_r^2 and Q for the CVA-based methods, and T^2 and Q for the PCA-based models. This MDR index measures the percentage of faulty observations not detected as faults, indicating the fault sensitivity of the methods.

The proposed LR-DCVA method achieves the best monitoring results, generating the minimum missing detection rates for most of the cases and their three statistics. Furthermore, the performance of the other proposed methods, ENR-DCVA and MLPR-DCVA, are very high, showing similar results for the $BII_{T_s^2}$ and $BII_{T_r^2}$ statistics, but they are worse considering the BII_Q statistic. However, the performance of the centralized CVA method is also very good, but only regarding the T_r^2 statistic.

Table 3 clearly shows that the missed detection rates for faults 1, 2, 6, 8, 12, 13 and 14 are close to 0% for all the methods and for the three statistics, except for the centralized DPCA and the Q statistic for fault 13, and also for the centralized CVA with the statistic Q for fault 8. In contrast, faults 3, 9 and 15 are very difficult to detect and the missed alarm rate is very high for all the methods. However, in these faults for most of the statistics, the MDR value for LR-DCVA is usually the lowest. For the other faults, the results for the proposed methods and, in particular, for LR-DCVA and ENR-DCVA are better than the rest ones, especially for faults 5, 10, 11, 16, 17, 19 and 20, but better for LR-DCVA than for ENR-DCVA.

In order to obtain a more intuitive comparison, the average missing detection rates for the 18 faults produced by the different approaches (i.e, excluding faults, 3, 9 and 15 which are very difficult to detect by the listed approaches) are shown in Fig. 2 and Table 3. The last two rows of this table show the average of all the statistics for each method.

TABLE 2. Faults Alarm Rates (FAR) and faults detected.

Methods	DPCA [27]		CVA [27]			DDPCA [34]		WCMBPCA [31]		mRMR-DDPCA [41]		WDSMBPCA [13]		LR-DCVA			ENR-DCVA			MLPR-DCVA		
	T^2	Q	T_s^2	T_r^2	Q	BII_{T^2}	BII_Q	BII_{T^2}	BII_Q	BII_{T^2}	BII_Q	BII_{T^2}	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q
FAR	0.6	28.1	8.3	12.6	8.7	0.63	1.77	1.04	1.04	0.18	2.53	0.5	0.25	0	0	0.4	4.79	1.77	1.45	0	0	0
Detected faults	16	18	18	18	17	18	18	18	18	18	18	18	18	21	19	20	19	18	18	19	19	18

TABLE 3. Missed Detection Rate (MDR) in % for the 21 faults.

Methods	DPCA [27]		CVA [27]			DDPCA [34]		WCMBPCA [31]		mRMR-DDPCA [41]		WDSMBPCA [13]		LR-DCVA			ENR-DCVA			MLPR-DCVA		
Fault	T^2	Q	T_s^2	T_r^2	Q	BII_{T^2}	BII_Q	BII_{T^2}	BII_Q	BII_{T^2}	BII_Q	BII_{T^2}	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q
1	0.6	0.5	0.1	0	0.3	0.38	0.13	0.25	0.25	0.4	0.1	0.25	0.1	0	0.25	0.25	0.38	0.25	0.63	0.25	0.25	0.5
2	1.9	1.5	1.1	1	2.6	1.63	2.63	1.5	1.38	1.5	1.5	1.5	1.37	1.5	1.13	1.63	1.63	1.75	2.88	0.88	1.8	2.5
3	99.1	99	98.1	98.6	98.5	98.13	96.13							89.2	97.5	93.25	94.25	98.38	97.38	96.5	96.88	97.88
4	93.9	0	68.8	0	97.5	2.75	0	0	2.12	0	0	0	0	0.13	0.13	0.13	0.13	0.13	39.38	0.13	0.13	84.38
5	75.8	74.8	0	0	0	73.13	0	72	0	74	0	70.63	0	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
6	1.3	0	0	0	0	0.63	0	0.5	0	0	0	0.5	0	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
7	15.9	0	38.6	0	48.6	0	0	0.5	0	0	0	0	0	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	10.38
8	2.8	2.5	2.1	1.6	48.6	2.25	2.38	2.25	2.25	2.5	2	2	1.87	2.13	1.75	2.25	1.88	2.5	3.63	1.63	1.75	3.38
9	99.5	99.4	98.6	99.3	99.3	97.63	94.88							92.88	98.88	93.13	95.63	99	98	98.38	98.5	98.88
10	58	66.5	16.6	9.9	59.9	55.13	26.13	50.75	24.25	51.5	25	52.62	23.25	7.25	7	10	8.25	13.38	17.13	6.75	7.5	16.88
11	80.1	19.3	51.1	19.5	66.4	21.5	4.38	27.12	42.87	17.8	4.8	17.5	15.12	12.38	10.75	17.5	22.25	18.73	47.38	10.38	10.63	45.13
12	1	2.4	0	0	2.1	0.88	0.13	1.12	0.62	0.9	0.1	1.12	1	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
13	4.9	19.3	4.7	4	5.5	5.25	4.38	5	5	5.4	4.5	5.12	4.75	4.38	4.38	4.88	4.63	4.5	5.38	4	4.25	4.13
14	6.1	2.4	0	0	12.2	0	0	0	0.12	0	0	0	0	0.13	0.13	0.25	0.13	0.13	0.28	0.13	0.25	2.75
15	96.4	97.6	92.8	90.3	97.9	97.38	92.75							75.75	82	89.13	84.25	93	97.75	78.88	82.13	98.5
16	78.3	70.8	16.6	8.4	42.9	76.63	27.88	70.37	22	75	27	73.12	22.75	2.75	3.63	6.5	8.63	11	15.25	3.63	4.5	18.38
17	24	5.3	10.4	2.4	13.8	6.5	2.38	5.5	8.13	7.4	2.1	3.87	2.75	2.75	2.5	3.63	3.13	2.75	8.75	2.38	2.38	3.88
18	11.1	10	9.4	9.2	10.2	10.5	9.13	9.75	10.38	10.3	9.5	9.12	9.37	8.38	9.38	8.88	9.25	9.75	10.25	9.25	9.38	9.5
19	99.3	73.5	84.5	1.9	92.3	63.25	32.88	96	47.62	92.5	12.1	58	28	2.25	0.63	10.13	10.13	4.13	31.25	0.5	0.38	62.5
20	64.4	49	24.8	8.7	35.4	46.75	20.25	44.75	34.87	50.5	17	38.75	34.62	9.5	8.63	16.25	10.25	18.63	15.5	8.5	8.5	34.75
21	64.4	55.8	44	34.2	54.8	46.75	40.75	56	48	53.1	39.4	46.87	43.5	42.4	35.13	51.88	41.88	58.63	64.88	30.25	40.75	71.38
MMDR*	38	25.2	20.7	5.6	32.9	23	9.6	26.63	13.88	23.1	11.1	21.15	10.47	**5.4**	**4.78**	**7.5**	6.84	8.16	14.62	**4.4**	5.4	20.6
Mean	31.6		19.75			16.31		19.25		17.1		15.81		**5.89**			9.87			10.04		

MMDR = The mean value of MDR for 18 faults (without faults 3, 9 and 15)*

TABLE 4. Detection delay (samples) for the 21 faults.

Methods	DPCA [27]		CVA [27]			DDPCA [34]		LR-DCVA			ENP-DCVA			MLP-DCVA		
Fault	T^2	Q	T_s^2	T_r^2	Q	BII_{T^2}	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q	$BII_{T_s^2}$	$BII_{T_r^2}$	BII_Q
1	6	5	2	3	2	3	1	3	2	2	4	3	6	2	2	4
2	16	13	13	15	25	13	18	13	11	14	14	15	31	9	12	19
3								43								
4	151	2	462	1	0	0	0	1	1	1	2	2	79	1	1	249
5	2	2	1	1	0	0	0	1	1	1	2	2	2	1	1	1
6	11	1	1	1	0	5	0	1	1	1	2	2	2	1	1	1
7	1	1	1	1	0	0	0	1	1	1	2	2	2	1	1	1
8	23	21	20	20	21	18	16	18	17	18	17	21	23	17	18	21
9								5								
10	101	50	25	23	44	54	30	22	22	24	24	24	25	22	22	25
11	195	7	292	11	27	5	5	7	7	8	12	12	12	7	7	9
12	3	8	2	2	0	2	1	2	2	2	3	3	3	2	2	2
13	45	40	42	39	43	42	36	36	35	41	38	39	48	32	34	35
14	6	1	2	1	1	0	0	1	1	2	2	2	2	1	2	10
15			677					577	643		676			3	3	
16	199	196	14	9	11	36	11	9	9	10	10	9	10	9	10	10
17	28	24	27	20	23	22	19	19	20	21	20	20	24	19	19	21
18	93	84	83	79	84	85	17	77	76	80	80	80	85	76	76	79
19		82		11		17	12	3	2	12	12	12	13	3	3	19
20	87	84	82	66	72	80	74	71	66	71	68	66	65	67	66	74
21	522	286	273	511	302	256	240	255	257	276	256	490	561	207	287	489

These MMDR values show that, for the BII_Q statistic, the lowest value is for the LR-DCVA method (7.5), outperforming the other approaches. In addition, the $BII_{T_r^2}$ statistic shows the best result for the LR-DCVA method (4.78), but its performance for MLPR-DCVA, central CVA and ENR-DCVA is worse, although still very similar, i.e., values 5.1, 5.6 and 8.1 respectively. Finally, the best method for $BII_{T_s^2}$ is MLPR-DCVA (4.4), but with a very slight difference regarding the LR-DCVA and ENR-DCVA methods with a value of 5.4 and 6.8 respectively.

Furthermore, it is possible to see that although the centralized CVA method with the $BII_{T_r^2}$ statistics, and some of the decentralized methods (mRMR-DDPCA and WDSMBPCA) with the statistic BII_Q, win in some of the faults, such as faults 4, 5, 6, 7 and 14 with MDR=0 (however, very similar to the LR-DCVA with MDR=0.13); the mean value, taking into account all the faults, is 5.6 for centralized CVA with the statistic $BII_{T_r^2}$, and the proposed LR-DCVA has a better value, 4.78, with the same statistic. However, the mean values for the statistic BII_Q of the mRMR-DDPCA method is 11.1, and 10.47 for the WDSMBPCA method, however the value of the same statistic for LR-DCVA is better at 7.5.

On the other hand, if we consider the mean value for all the faults and statistics (the last row of Table 3), then the result for LR-DCVA is 5.89 while the rest of the methods show a bigger value: for ENR-DCVA, it is 9.87, this being the second best method, followed by MLPR-DCVA with 10.05, WDSMBPCA with 15.81, DDPCA with 16.31, mRMR-DDPCA with 17.1, WCMBPCA with 19.25, centralized CVA with 19.7 and, finally, the worst result is for the centralized DPCA with a mean value of 31.6.

3) DETECTION DELAY

Another important index to consider is the detection delay, which is required by the monitoring scheme to detect faults as soon as possible. Generally, a fault can be indicated only

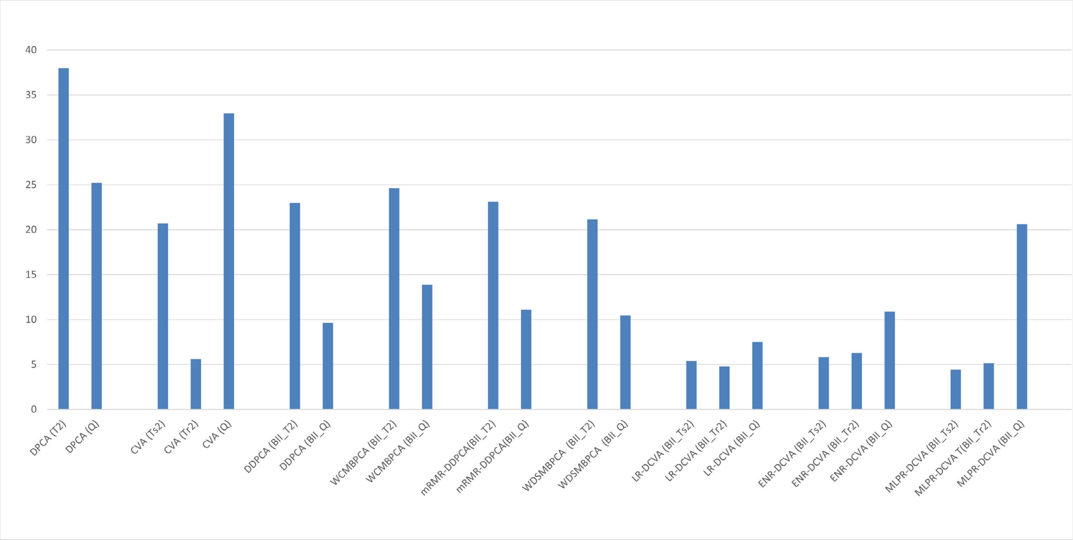


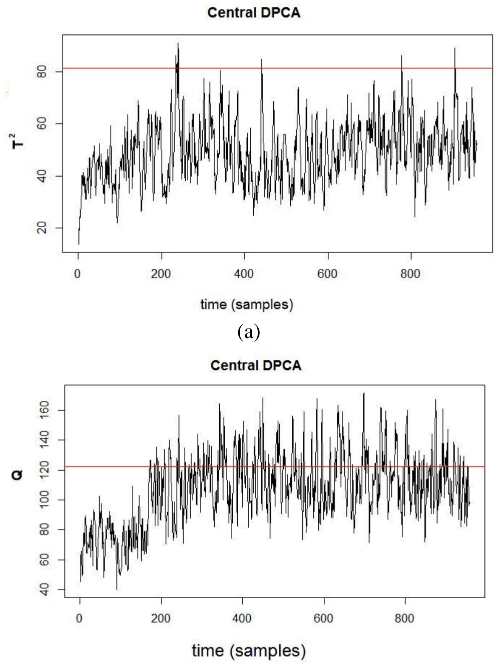
FIGURE 2. Comparison of average MDR (MMDR) for the different monitoring methods.

when several consecutive values of the statistics exceed their thresholds [27], [33].

In this paper, as mentioned above, 5 consecutive samples are needed to detect the fault, and the detection delay is recorded at the first time instance when the control limit is exceeded. Table 4 contains the results for detection delay. In this case, the comparison with the WCMBPCA, mRMR-DDPCA and WDSMBPCA methods are not shown because this index is not considered in [13], [31], and [41].

Table 4 shows that the four decentralized methods, DDPCA, LR-DCVA, ENR-DVCA and MLPR-DCVA, outperform the centralized DPCA and CVA methods regarding the detection delay for all of three statistics. The results are very similar for all the decentralized methods: MLPR-DCVA with the $BII_{T_s^2}$ statistic is the best approach for most of the faults. However, MLPR-DCVA with the BII_Q statistic is the worst one among the decentralized methods. The other three methods, DDPCA, LR-DCVA and ENR-DCVA are very similar, but LR-DCVA is a slightly better, specially for faults 3,9,10, 15, 16, 19 and 20.

Therefore, the conclusion drawn by all the indexes considered (faults detected, FAR (Table 2), MDR (Table 3) and detection delay (Table 4)) is that the proposed LR-DCVA method achieves the best monitoring results of all those considered. ENR-DCVA also gives good results for the three statistics, but worse than LR-DCVA. Finally, the third proposed method, MLPR-DCVA, also achieves very good results with the $BII_{T_s^2}$ and $BII_{T_r^2}$ statistics for both the MDR index and the detection delay index, but the results with the BII_Q statistic are worse than those for the LR-DCVA method. This may be due to the ability of neural networks to obtain a non-linear model of the processes, in addition to extracting the non-linear characteristics of the plant to build the blocks. The $BII_{T_s^2}$ statistic monitors the behavior of the model, while the BII_Q statistic monitors the residual space [27]. If the model is better, it is to be expected that the $BII_{T_s^2}$ will be more effective than the BII_Q.

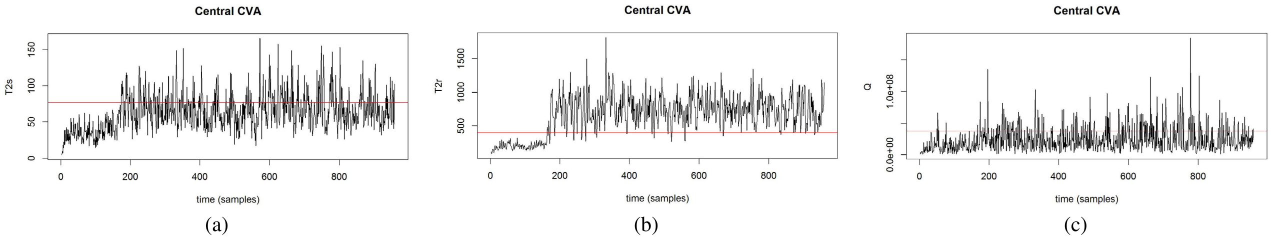


(a)

(b)

FIGURE 3. Monitoring results of TEP for fault 19 for DPCA: (a) T^2 (b) Q.

Fault 19 is a good example of these results. It is an unknown fault for the TEP process, which is difficult to detect for such centralized methods as PCA, DPCA, KPCA, not to mention other methods as shown in [1], and for some decentralized methods as those considered in the comparative, i.e., DDPCA, WCMBPCA, mRMR-DDPCA and WDSMBPCA, as shown in Table 3. The

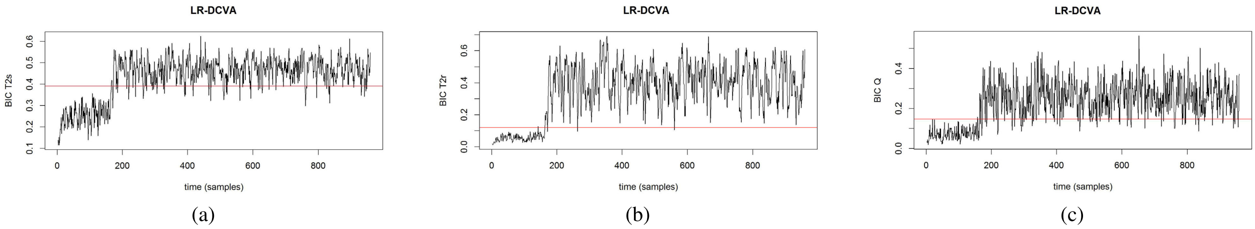


(a)

(b)

(c)

FIGURE 4. Monitoring results of TEP for fault 19 for CVA.

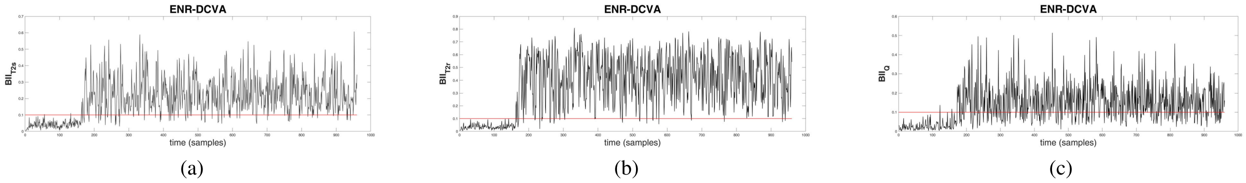


(a)

(b)

(c)

FIGURE 5. Monitoring results of TEP for fault 19 for LR-DCVA.

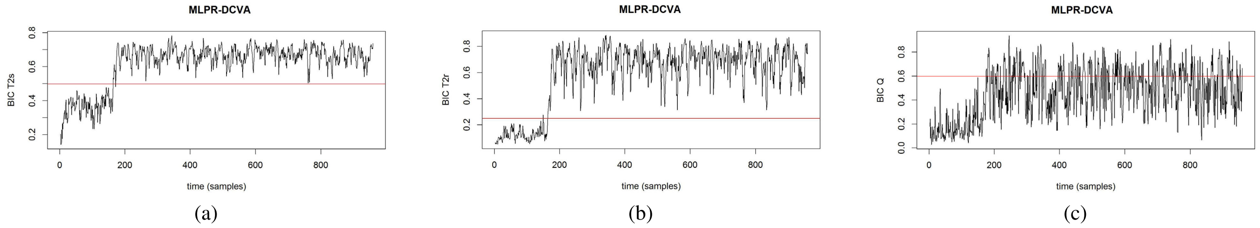


(a)

(b)

(c)

FIGURE 6. Monitoring results of TEP for fault 19 for ENR-DCVA.



(a)

(b)

(c)

FIGURE 7. Monitoring results of TEP for fault 19 for MLPR-DCVA.

monitoring results for this fault using DPCA, CVA, LR-DCVA, ENR-DCVA and MLPR-DCVA are shown in Figs. 3-7.

Fig. 3 sets out the results of the centralized DPCA model, which is unable to detect this fault for either of the two statistics. The same for the CVA method with the T_s^2 and Q statistics, as shown in Fig. 4, but CVA with T_r^2 can detect the fault. However, LR-DCVA can detect this fault by any the three statistics ($BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q (see Fig. 5)). ENR-DCVA detects the fault by any of three statistics, but the BII_Q statistic has a worse result (see Fig. 6). Finally, MLPR-DCVA detects the fault by $BII_{T_s^2}$ and $BII_{T_r^2}$ statistics and, as commented previously, the results for BII_Q are worse, while the percentage of missed alarms increases, as it is possible to see in Fig. 7.

V. CONCLUSION

This paper proposes a new dynamic and decentralized process monitoring framework for large-scale processes based on regression and canonical variate analysis (R-DCVA). This new approach uses regression methods to divide the plant-wide processes into blocks, obtaining the most defining variables for each block and its CVA based monitoring.

Three regression methods are considered: LASSO regression (LR-DCVA), Elastic net regression (ENR-DCVA) and Multilayer Perceptron Network (MLPR-DCVA). MLPR-DCVA, in comparison, the other two methods, gives a non-linear regression model to capture the non-linear relationship of the process variables. After the division, a local fault detection method based on the Canonical Variate Analysis

(CVA) approach is performed for every block to take into account the dynamical nature of the industrial processes.

The regression methods, as supervised approaches, permit the best tuning for capturing the best relationship between the variables as they can divide the plant variables into blocks more reasonably, so the R-CVA reflects better local behaviors of the process: the faults can be more easily detected and the monitoring results can be explained better. Finally, the fault detection results of each block are managed by the Bayesian Inference Criterion to obtain a global fault detection outcome.

In order to check the proposed decentralized approaches and to test the performance of the R-DCVA fault detection method, the proposals were compared to other published works on a well-known benchmark plant: the Tennessee Eastman Plant. The results show that the LR-DCVA method proposed in this work outperforms the rest of the comparison methods for most of the considered indexes (number of detected faults, false alarm rate, detection delay and missing detection rates considering the $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q statistics), all of which show its effectiveness.

However, the results for the other proposed methods, ENR-DCVA and MLPR-DCVA are also very good with the statistics $BII_{T_s^2}$ and $BII_{T_r^2}$, and better than the other methods over the comparison except for LR-DCVA, but their results with the BII_Q statistic are worse.

Finally, it should also be said that these methods consider a very large number of blocks, one for each variable, which implies that the number of blocks increases greatly when the dimensions of the plant are high. So, in future work, two solutions can be studied to deal with this kind of problems: first, to study another kind of plant decomposition using fewer blocks, or to use a reduction of the dimensionality of the plant and regression-based analysis to divide the plant in that lower dimensionality space. This is the work in progress.

REFERENCES

[1] A. Bakdi and A. Kouadri, "A new adaptive PCA based thresholding scheme for fault detection in complex systems," *Chemometrics Intell. Lab. Syst.*, vol. 162, pp. 83–93, Mar. 2017.

[2] A. Bathelt, N. L. Ricker, and M. Jelal, "Revision of the Tennessee Eastman process model," in *Proc. 9th IFAC Symp. Adv. Control Chem. Process (ADCHEM)*, 2015, pp. 309–314.

[3] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[4] C. Chakour, M. Harkat, and M. Djeghaba, "New adaptive kernel principal component analysis for nonlinear dynamic process monitoring," *Appl. Math. Inf. Sci.*, vol. 9, no. 4, pp. 1833–1845, 2015.

[5] Z. Chen, C. Liu, S. X. Ding, T. Peng, C. Yang, W. Gui, and Y. A. W. Shardt, "A just-in-time-learning-aided canonical correlation analysis method for multimode process monitoring and fault detection," *IEEE Trans. Ind. Electron.*, vol. 68, no. 6, pp. 5259–5270, Jun. 2020.

[6] J. de Oña and C. Garrido, "Extracting the contribution of independent variables in neural network models: A new approach to handle instability," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 859–869, Sep. 2014.

[7] Y. Dong and S. J. Qin, "A novel dynamic PCA algorithm for dynamic data modeling and process monitoring," *J. Process Control*, vol. 67, pp. 1–11, Jul. 2018.

[8] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.

[9] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometrics Intell. Lab. Syst.*, vol. 171, pp. 16–25, Dec. 2017.

[10] Z. Ge and Z. Song, "Distributed PCA model for plant-wide process monitoring," *Ind. Eng. Chem. Res.*, vol. 52, no. 5, pp. 1947–1957, 2013.

[11] Z. Ge, M. Zhang, and Z. Song, "Nonlinear process monitoring based on linear subspace and Bayesian inference," *J. Process Control*, vol. 20, no. 5, pp. 676–688, 2010.

[12] M. Grbovic, W. Li, P. Xu, A. K. Usadi, L. Song, and S. Vucetic, "Decentralized fault detection and diagnosis via sparse PCA based decomposition and maximum entropy decision fusion," *J. Process Control*, vol. 22, no. 4, pp. 738–750, 2012.

[13] C.-C. Hsu, P.-C. Shih, and F.-C. Tien, "Integrate weighted dependence and skewness based multiblock principal component analysis with Bayesian inference for large-scale process monitoring," *J. Taiwan Inst. Chem. Engineers*, vol. 119, pp. 6–22, Feb. 2021.

[14] J. Huang and X. Yan, "Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and Bayesian inference," *Chemometric Intell. Lab. Syst.*, vol. 148, pp. 115–127, Nov. 2015.

[15] B. Jiang, D. Huang, X. Zhu, F. Yang, and R. D. Braatz, "Canonical variate analysis-based contributions for fault identification," *J. Process Control*, vol. 26, pp. 17–25, Feb. 2015.

[16] B. Jiang and R. D. Braatz, "Fault detection of process correlation structure using canonical variate analysis-based correlation features," *J. Process Control*, vol. 58, pp. 131–138, Oct. 2017.

[17] Q. Jiang and X. Yan, "Plant-wide process monitoring based on mutual information–multiblock principal component analysis," *ISA Trans.*, vol. 53, pp. 1516–1527, Sep. 2014.

[18] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 377–386, Jan. 2016.

[19] Q. Jiang and B. Huang, "Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method," *J. Process Control*, vol. 46, pp. 75–83, Oct. 2016.

[20] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY, USA: Springer-Verlag, 2013.

[21] W. E. Larimore, *Statistical Methods in Control and Signal Processing*. New York, NY, USA: Marcel Dekker, 1997, pp. 83–120.

[22] Z. Luo, Y. Wang, S. Lu, and P. Sun, "Process monitoring using a novel robust PCA scheme," *Ind. Eng. Chem. Res.*, vol. 60, no. 11, pp. 4397–4404, 2021.

[23] R. Muthukrishnan and R. Rohini, "LASSO: A feature selection technique in predictive modeling for machine learning," in *Proc. IEEE Int. Conf. Adv. Comput. Appl. (ICACA)*, Oct. 2016.

[24] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Ind. Eng. Chem. Res.*, vol. 58, no. 29, pp. 12899–12912, Jul. 2019.

[25] S. J. Qin, S. Valle, and M. J. Piovoso, "On unifying multiblock analysis with application to decentralized process monitoring," *J. Chemometrics*, vol. 15, no. 9, pp. 715–742, 2001.

[26] T. J. Rato and M. S. Reis, "Fault detection in the Tennessee eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR)," *Chemometric Intell. Lab. Syst.*, vol. 125, pp. 101–108, Jun. 2013.

[27] 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 Intell. Lab. Syst.*, vol. 51, no. 1, pp. 81–93, 2000.

[28] A. Sanchez-Fernandez, M. J. Fuente, and G. I. Sainz-Palmero, "Fault detection in wastewater treatment plants using distributed PCA methods," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–7.

[29] Y. Si, Y. Wang, and D. Zhou, "Key-performance-indicator-related process monitoring based on improved kernel partial least squares," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2626–2636, Mar. 2021.

[30] Y. Tian, T. Hu, X. Peng, W. Du, and H. Yao, "Decentralized monitoring for large-scale process using copula-correlation analysis and Bayesian inference–based multiblock principal component analysis," *J. Chemometrics*, vol. 33, no. 8, pp. 1–18, 2019.

[31] Y. Tian, H. Yao, and Z. Li, "Plant-wide process monitoring by using weighted copula–correlation based multiblock principal component analysis approach and online-horizon Bayesian method," *ISA Trans.*, vol. 96, pp. 24–36, Jan. 2020.

[32] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[33] C. Tong, T. Lan, and X. Shi, "Fault detection and diagnosis of dynamic processes using weighted dynamic decentralized PCA approach," *Chemometrics Intell. Lab. Syst.*, vol. 161, pp. 34–42, Feb. 2017.

[34] C. Tong and X. Shi, "Decentralized monitoring of dynamic processes based on dynamic feature selection and informative fault pattern dissimilarity," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3804–3814, Jun. 2016.

[35] C. Tong and X. Yan, "A novel decentralized process monitoring scheme using a modified multiblock PCA algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1129–1138, Apr. 2007.

[36] K. Wang, J. Chen, and Z. Song, "Performance analysis of dynamic PCA for closed-loop process monitoring and its improvement by output oversampling scheme," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 378–385, Jan. 2019.

[37] C. Xu, S. Zhao, and F. Liu, "Distributed plant-wide process monitoring based on PCA with minimal redundancy maximal relevance," *Chemometric Intell. Lab. Syst.*, vol. 169, pp. 53–63, Oct. 2017.

[38] S. Yin, S. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 16, no. 11, pp. 6418–6428, Nov. 2014.

[39] S. Yin, X. Zhu, and O. Kaynak, "Improved PLS focused on key-performance-indicator-related fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1651–1658, Mar. 2015.

[40] Y. Zhang, H. Zhou, S. J. Qin, and T. Chai, "Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 3–10, Feb. 2010.

[41] K. Zhong, M. Han, T. Qiu, B. Han, and Y.-W. Chen, "Distributed dynamic process monitoring based on minimal redundancy maximal relevance variable selection and Bayesian inference," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 2037–2044, Sep. 2020.

[42] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc., B (Stat. Methodol.)*, vol. 67, no. 2, pp. 301–320, 2005.

MARÍA JESÚS DE LA FUENTE received the Ph.D. degree in physical sciences from the University of Valladolid, Spain, in 1994. She is currently a Full Professor with the Department of Systems Engineering and Automatic Control, School of Industrial Engineering, University of Valladolid. She was a leader in many research projects. She is the coauthor of many peer-reviewed articles in prestigious journals and international conferences. Her research interests include fault detection and identification (FDI) in large industrial processes and plants. She is a member of the Spanish Research Agency. She has participated on numerous occasions in the evaluation and accreditation of research activities.

GREGORIO I. SAINZ-PALMERO received the Ph.D. degree in industrial engineering from the University of Valladolid, Spain, in 1997. He is currently an Associate Professor with the Department of Systems Engineering and Automatic Control, School of Industrial Engineering, University of Valladolid. He was the Director of the Intelligent Systems Division, CARTIF Technology Center, from 1998 to 2015. His research career was largely oriented toward applied research, training of researchers, and technology transfer to companies, public institutions, and society in general. He has participated in many research projects and contracts, most of them in cooperation with companies or public entities. His research interests include the development of intelligent systems and their industrial application.

MARTA GALENDE-HERNÁNDEZ received the Ph.D. degree from the University of Valladolid, in 2014. She is currently an Associate Professor with the Department of Systems Engineering and Automatic Control, School of Industrial Engineering, University of Valladolid. In 1998, she was a Researcher with CARTIF Technology Center, Boecillo, Spain. She has participated in many research and development projects and contracts with companies and public institutions. She is the coauthor of numerous peer-reviewed articles in international journals and conferences. Her research interests include computational intelligence and its application in industrial environments.