# Fault Detection and Diagnosis of Nonlinear Dynamical Processes through Fractal-based Dynamic Kernel PCA

Wahiba Bounoua[a,*]; Azzeddine Bakdi[b]

[a] *Signals and Systems Laboratory, Institute of Electrical and Electronics Engineering, University M'Hamed BOUGARA of Boumerdes, Avenue of independence, 35000 Boumerdès, Algeria*

[b]*Department of Mathematics, University of Oslo, 0851 Oslo, Norway*. bkdaznsun@gmail.com

** Corresponding author. E-mail address: w.bounoua@univ-boumerdes.dz , wb.bounoua@gmail.com. (Wahiba Bounoua).*

**Abstract**

This paper presents a novel Dynamic Kernel PCA (DKPCA) method devoted to process monitoring in nonlinear dynamical systems. It focuses on the optimal dynamical structure constructed through the powerful mathematical tool in fractal theory, the Fractal Dimension (FDim). DKPCA offers a generic data-driven framework for treating nonlinear dynamical systems and enables modelling the possible dynamic behaviour. The Fractal Dimension (FDim) provides an accurate intrinsic measure of the complexity of a data set and it is used to explore the nonlinear dynamics and the chaotic behaviour. This paper integrates the two strategies to overcome the shortcomings associated with classical methods based on linear dynamic PCA and DKPCA exhibiting linearity assumptions on the dynamical structure. These limitations are alleviated through the proposed Fractal-based DKPCA (FDKPCA) framework. An additional contribution is a new approach for selecting the exact number of Principal Components as the main phase in dimensionality reduction based on FDim. The novel fault detection and diagnosis method is verified through seven applications using the Process Network Optimization (PRONTO) benchmark with real heterogeneous data, FDKPCA showed superior performance compared to contemporary approaches.

**Keywords**: Fault Detection and Diagnosis; Dynamic Kernel PCA; Fractal Analysis; Correlation Dimension; Intrinsic Dimension; Process Network Optimization (PRONTO) Benchmark.

## 1. Introduction

Pre-empting failures in industrial processes via Fault Detection and Diagnosis (FDD) techniques has undergone considerable expansion in recent decades [1]. Indeed, FDD frameworks have been integrated into a variety of process control systems due to their increased complexity encompassing the modern technology. Such techniques typically involve the model-based approaches [2, 3], relying on a priori knowledge about the mathematical and physical relationships of the system. Such approaches include filters [4, 5], observers, and parameter estimation [6, 7]. These methods might be intricate because it is still difficult to set up the most adequate mathematical model. Data-driven methods [8], on the other hand, have become more widespread and mostly adopted from researchers; requiring only the minimum a priori knowledge, they rely on fully exploiting the largely available historical data [8, 9]. These methods, however, are under continuous development aiming to facilitate and optimise their implementation. Multivariate Statistical Process Control (MSPC) techniques [10] are widely used for process monitoring. Principal Component Analysis (PCA) [11], Partial Least Squares (PLS) [12], Fisher Discriminant Analysis (FDA) [13], Support Vector Machines (SVM) [13], Independent Component Analysis (ICA) [14], and Slow Feature Analysis (SFA) [15] are common MSPC methods in FDD.

PCA has received tremendous attention and diverse PCA extensions have been introduced to cope with process dynamics, nonlinearities, and unknown disturbances. Nonlinear PCA (NLPCA) [16] and Kernel PCA (KPCA) [17] map the process data into a high dimensional feature space using integral operators and nonlinear basis functions. Additionally, dynamic, recursive, moving-window, and multiple-mode PCA variants are originated from process dynamics and non-stationarities [18-20] with a paid cost of increased complexity. PCA and its descendants still, however, exhibit major shortcomings: PCA theory does not guarantee an optimal static PCA model exists for particular data; besides, an optimal PCA model is not well-defined in the existing literature; furthermore, the optimal structure that explains correctly the nonlinear dynamical process functionality in high dimensional settings remains an issue in research. Solving for the appropriate Number of PCs (NPC) to retain in

a static PCA model is an unsolved dilemma where many methods are proposed in the literature as pointed in [21]. Common methods include Kaiser's rule [22], Cumulative Percent Variance (CPV) [23], Parallel analysis (PA) [24], scree test [25], cross-validation [26], and Variance of the Reconstruction Error (VRE) [27]. These methods are summarized and discussed in [28] where each of the criteria is derived from a particular rule of thumb and results in vague decisions for different PCA models and performance. The first contribution of this work will be to consider this issue through a completely new approach based on the rich and powerful theory of Fractal Dimension (FDim). It was reported in [8] that the major drawbacks associated with standard PCA are due to the covariance matrix which is scale variant and it is obtained through Pearson's dependence measure which is sensitive to outliers. Instead, [29] proposed to form the coefficients of the rank correlation matrix through Spearman's or Kendall tau's coefficients between all the pairs of process variables where sequential eigenvector extraction lead to a sparse static PCA model. [30] combined PCA's $T^2$ statistic and Bayesian network to improve PCA's capabilities of fault diagnosis and overcome the limitations of multivariate contribution plot. Alternative approaches to FDD include machine learning methods such as risk-based fault indicators using self-organizing map [31] which require larger amounts of data and computational resources and dynamic Bayesian network based control chart [32] for FDD in small to medium scale processes.

The existing Dynamic PCA (DPCA) is a direct approach integrating dynamic dependency information in a linear static model extracted by PCA. The cardinality of each of the variables in a raw data matrix is expanded to the $l$ previous values such that $\widetilde{\boldsymbol{X}}\boldsymbol{b} = \boldsymbol{0}$ is solved to obtain the null space of the augmented data matrix $\widetilde{\boldsymbol{X}}$ [33]. Besides its major limitation to linear dependencies, the determination of a proper lag structure is a key challenge. This was described in [18] through the null-space dimension based on linear PCA. Similarly, [34] determined the number of lags by performing Singular Value Decomposition (SVD) of the extended data matrix. Vanhatalo et al. [35] proposed another method based on the the sample and partial autocorrelation matrices of the original data extended at different lags but also focusing on the eigenvalues. The aforementioned DPCA methods are not definite; extra-sensitive to the different criteria issue; and limited to linear relations through linear PCA where the process is approximated by a basic linear model. DPCA extension to nonlinear processes through Dynamic Kernel PCA (DKPCA) was firstly proposed in [36], integrating the potentials of KPCA and the dynamical properties of DPCA. DKPCA has been modified to batch DKPCA in [37], and incorporated with FDA (DKPCA-FDA) in [38] for fault isolation. These methods aim at capturing nonlinear static dependencies, however, their dynamical structure and the number of PCs are still identified using linear approximations that fail to address nonlinear dynamics.

Henceforth, a data-driven algorithm that addresses properly nonlinear dynamical systems is highly demanded in the literature to limit design costs, model dimension, complexity, and improve performance potentials. This paper introduces a novel and powerful data-driven approach whose subject matter is to optimally model the nonlinear dynamics exposed in the lag structure based on the actual Fractal Dimension (FDim). FDim [39] theory is effectively rich and not limited to time-series. Characteristically, large multivariate m-dimensional data lie intrinsically in a topological manifold of dimension $d < m$; this lower dimension $d$ is the Intrinsic Dimension (IDim) defined practically as the number of independent variables required to describe the data set with minimal loss of information [40]. Moreover, the Correlation Dimension (CDim) [41], the most practical fractal dimension definition, identifies the number of the independent variables required to model the dynamical system [42]. Noticing that this definition states the ultimate objective of PCA except that it is the intrinsic characteristic of a particular data without a single assumption. FDim herin overcomes the reliance of PCA, DPCA, and DKPCA on simple SVD and rule-of-thumb techniques. For many decades, extensive research has been conducted to set up and analyse methods for determining the IDim. An inclusive survey on the most relevant IDim estimation techniques is provided in [43], emphasising the fractal-based methods as the most robust estimators.

Fractal theory gives a mathematical tool to deal with complex systems. The phase space of a system, defined as the space of variables that specify the state of the system, may attract the initial conditions to some subset called the attractor. For simple systems, the attractor could be a point of dimension zero or a closed curve of dimension one. For most real dynamical systems, the attracting set is more complex and chaotic in nature with a non-integer dimension. These are known as strange attractors characterized by the underlying dynamics that have space-filling properties expressed as the fractal dimension [44]. Exploring these chaotic dynamics of nonlinear systems is the main area where considerable progress has been made through the fractal theory, particularly for strange attractors [45, 46]. FDim was originally derived for image processing where the main objective was to estimate the topological characteristics of geometrical objects [47]. It was proved useful as a tool in machine learning applications to quickly select the most important attributes [48, 49]. FDim used in the literature of FDD is very limited to model-based techniques and small-scale applications such as temperature sensor [50]; crack and spalling failure detection [51]; induction motors faults [52]; and bearing faults [53, 54]. However, fractal theory was not investigated before in large-scale high-dimensional systems. Theoretically, no study has considered phase space reconstruction based on autocorrelations and cross-correlations associated with MSPC methods through FDim, especially for optimal dynamical structure stemming from constructing an attractor space composed of each independent degree of freedom which is often unknown for chaotic dynamical systems.

This paper underlines these aspects and thorough analysis is left for future work from possibly even pseudo-phase-space perspective. The proposed methods determine correctly the number of nonlinear correlation relationships in the extended dynamical model that ensures optimal phase space reconstruction under nonlinearities. The dynamic analysis accomplished using the FDim for KPCA so as to detect the nonlinear relationships remains coherent under nonlinear modelling via a nonlinear IDim estimator. The proposed approach is automatic and assumption-free that provides a sharp limit i.e. an important property of not being actually enforced to be an integer number. To address these challenges, the Fractal-based Dynamic Kernel PCA (FDKPCA) is proposed as an inclusive optimal monitoring framework for nonlinear dynamical systems. The novel concept is explained and proved in this article, it is then validated through seven applications using real heterogeneous data from the Process Network Optimization (PRONTO) benchmark.

The remainder of this paper is organized as follows: The different techniques, methodologies, and the new proposed methods are presented in Sections 2, 3, and 4. Next in Section 5, the proposed fractal-based DKPCA framework is presented in details. Afterward, applications are implemented to analyse the monitoring results in Section 6. Finally, the conclusions are drawn in section 7.

## 2. Fractal dimension:

The Correlation Dimension (CDim) is selected in this paper to compute the FDim of the data set $\boldsymbol{X}$ due to its effectiveness in providing an unbiased estimator of the intrinsic dimension by means of a computationally affordable algorithm [46]. CDim is computed based on the Grassberger-Procaccia algorithm [41, 55] on the basis of the probability that two points on the attractor are at a distance $\varepsilon$ apart.

The point correlation function for the normalized data matrix $\boldsymbol{X} \in \Re^{N\times m}$ is the number of neighbours in a $m$-dimensional ball of radius $\varepsilon$ around a reference point $\boldsymbol{x}_i$, it is given as:

$$C_i(\varepsilon) = \frac{1}{N-1}(\ the\ number\ of\ points\ \boldsymbol{x}_j\ within\ \varepsilon\ of\ a\ reference\ point\ \boldsymbol{x}_i) \tag{1}$$

The average of this point correlation functions over the $N$ points gives the radial correlation function:

$$C(\varepsilon) = \frac{1}{N}\sum_{i=1}^{N} C_i(\varepsilon) \tag{2}$$

Based on the previous relations, the correlation sum can be defined as:

$$C(\varepsilon) = \lim_{N\to\infty} \frac{2}{(N(N-1))} \sum_{i=1}^{N}\sum_{j=i+1}^{N} I(\ \delta_{ij} \le \varepsilon) \tag{3}$$

The correlation sum is characterized by $N(N-1)/2$ pairwise similarities, where $\delta_{ij} = \left\|\boldsymbol{x}_j - \boldsymbol{x}_i\right\|^2$ is the Euclidian distances between $ij$ rows to the input matrix, and $I$ represents an indicator function defined as:

$$I\left(\delta_{ij} \le \varepsilon\right) = \begin{cases} 1 & if\ \delta_{ij} \le \varepsilon \\ 0 & otherwise \end{cases} \tag{4}$$

In other words, $I$ is 1 if the condition holds and 0 otherwise.

As the distance $\varepsilon$ gets smaller, the correlation sum is found to follow a power low as:

$$C(\varepsilon) \sim \varepsilon^{\mathrm{CDim}} \tag{5}$$

In the limit of $\varepsilon$ and taking the logarithm of both sides, the correlation dimension CDim is obtained as:

$$\mathrm{CDim} \equiv \lim_{\varepsilon\to 0} \frac{log\left(C(\varepsilon)\right)}{log(\varepsilon)} \tag{6}$$

Assuming a sufficient number of points have been acquired lying closely in the underlying space, then the slope of the linear part of the $log - log$ plot of $C(\varepsilon)$ versus $\varepsilon$ represents the correlation dimension $CDim$.

## 3. PCA-based Modelling

Considering the data matrix $\boldsymbol{X} = \{\{\boldsymbol{x}_i\}_{i=1:m}\} \in \Re^{N\times m}$, collected from a process in normal operation with $N$ samples of $m$ variables, the data is firstly normalized to zero mean and unit variance. PCA transforms this data matrix into the score matrix $\boldsymbol{\mathcal{T}} = \{\{\boldsymbol{t}_i\}_{i=1:m}\} \in \Re^{N\times m}$ through the projection:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{XP} \tag{7}$$

where $\boldsymbol{P}$ is the loading matrix obtained by an orthogonal transformation of the covariance matrix $\boldsymbol{\Sigma}$ that can be computed through Singular Value Decomposition (SVD) as:

$$\boldsymbol{\Sigma} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T \tag{8}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_m)$ is the diagonal eigenvalues matrix. The data dimensionality reduction is to split $\boldsymbol{P}$ into modelled and non-modelled variations, $\widehat{\boldsymbol{P}} \in \Re^{m\times d}$ for PCs and $\widetilde{\boldsymbol{P}} \in \Re^{m\times(m-d)}$ for residuals. A challenging task arises when trying to reduce the dimensionality from $m$ to $d \ll m$ is to select the appropriate criterion to derive $d$. Most of the criteria proposed so far, Kaiser's rule, CPV, PA, the scree test, cross-validation, VRE, and others rely mainly on the evaluation of variances which becomes misleading when the raw data dimensionality is high. In this paper, we propose a novel criterion to obtain the number of PCs based on FDim.

The number of PCs extraction through FDim aims at finding a subset of orthogonal variables $\boldsymbol{Z} \in \{Principal\ PCA\ model\ projection\ of(\boldsymbol{X})\}$ such that $\boldsymbol{Z}$ can thoroughly describe $\boldsymbol{X}$. Hence, $\boldsymbol{Z}$ is the informative subset of the score matrix $\boldsymbol{\mathcal{T}}$ i.e. $\boldsymbol{Z} = \{\{\boldsymbol{t}_i\}_{i=1:d}\} \in \mathcal{R}^{N\times d} \subset \boldsymbol{\mathcal{T}}$ and $d$ is its cardinality investigated through FDim. Subsequently, the number of PCs is extracted on two bases to identify the exact number of independent PCs:

1. If $\mathrm{FDim}(\boldsymbol{X}) = d_{\boldsymbol{X}}$ and $\mathrm{FDim}(\boldsymbol{\mathcal{T}}) = d_{\boldsymbol{\mathcal{T}}}$ and $d_{\boldsymbol{\mathcal{T}}} \leq d_{\boldsymbol{X}}$, then $\{\text{the number of PCs}\} = \lceil d_{\boldsymbol{X}} \rceil$ and $\boldsymbol{Z} \in \mathcal{R}^{N\times\lceil d_{\boldsymbol{X}}\rceil}$ to guarantee that these PCs tightly cover the original data.
2. If $\mathrm{FDim}(\boldsymbol{\mathcal{T}}) = d_{\boldsymbol{\mathcal{T}}} > d_{\boldsymbol{X}}$, then $\{\text{the number of PCs}\} = \lceil d_{\boldsymbol{\mathcal{T}}} \rceil$ and $\boldsymbol{Z} \in \mathcal{R}^{N\times\lceil d_{\boldsymbol{\mathcal{T}}}\rceil}$ in order to account for the informative subspace in raw data and in PCA-based model projections concurrently since each principal component reflects the maximum amount of variance in the observed variables that was not taken into account by the preceding component [56].

These conditions ensure that the least important component is not missed and hence increasing the model precision in accordance. The same exact procedure is applied to models derived using DPCA and DKPCA. The proposed algorithm is summarized in Table 1.

**Table 1.** Pseudo-code of FDim-based number of PCs extraction

| **Algorithm** |
|---|
| **Input:** Data set $\boldsymbol{X}$ (either original data for PCA or augmented data matrix for DPCA and DKPCA) |
| **Output:** Number of PCs |
| Step 1: Compute CDim of the raw data set $\boldsymbol{X}$ and set $\mathrm{FDim}(\boldsymbol{X}) = \mathrm{CDim}(\boldsymbol{X}) = d_{\boldsymbol{X}}$ |
| Step 2: Compute the score matrix components $\boldsymbol{\mathcal{T}} = \boldsymbol{XP}$ |
| Step 3: Compute CDim of $\boldsymbol{\mathcal{T}}$ and set $\mathrm{FDim}(\boldsymbol{\mathcal{T}}) = \mathrm{CDim}(\boldsymbol{\mathcal{T}}) = d_{\boldsymbol{\mathcal{T}}}$ |
| Step 4: Compare $d_{\boldsymbol{X}}$ and $d_{\boldsymbol{\mathcal{T}}}$: |
| If $d_{\boldsymbol{X}} \geq d_{\boldsymbol{\mathcal{T}}}$ then $\{\text{the number of PCs}\} = \lceil d_{\boldsymbol{X}} \rceil$, |
| If $d_{\boldsymbol{X}} < d_{\boldsymbol{\mathcal{T}}}$ then $\{\text{the number of PCs}\} = \lceil d_{\boldsymbol{\mathcal{T}}} \rceil$. |

## 4. Nonlinear Dynamic Process Monitoring

Large scale industrial processes exhibit nonlinear and/or dynamical behaviour. The application of static linear methods such as the standard PCA, that make heavy assumptions of linearity and/or steady-state conditions, may be insufficient to model the intrinsic structure of the process. Thereafter, more developed algorithms may be integrated in order to take into account the nonlinearity as well as the dynamical structure of the underlying process states.

### *4.1. Dynamic PCA*

The dynamical structure represents an explicit description of the auto-/cross-correlations that exist within the process variables. An augmented matrix is composed to include delayed replicate of the original data matrix variables [18]. Based on the original raw data matrix $\boldsymbol{X} = \{\{\boldsymbol{x}_i\}_{i=1:m}\} \in \Re^{N\times m}$, the augmented matrix is obtained as:

$$\widetilde{\boldsymbol{X}} \triangleq [\boldsymbol{X}(k)\ \boldsymbol{X}(k-1)\ \cdots\ \boldsymbol{X}(k-l)] \in \boldsymbol{\Re}^{(N-l)\times m(l+1)} \tag{9}$$

where $\boldsymbol{X}(k-j) = \{\boldsymbol{x}(i-j)\}_{i=1:m}^{j=0:l} \in \boldsymbol{\Re}^{(N-l)\times m}$

In an attempt to investigate the linear autocorrelation in the augmented data matrix, Ku et al. [18] developed a method in which the number of lags $l$ needed to encapsulate the dynamics within the data is selected based on linear PCA modelling. This approach is highly eigen-analysis and linear correlations reliant on; it uses the conventional techniques for measuring dependencies in linear systems applied in the standard PCA. The dimensionality $d$ of $m$-dimensional data set ($d < m$) is estimated by calculating the eigenvalues and eigenvectors

of the covariance or the correlation data matrix. This technique is restricted since it is based on a linear transformation which clearly deteriorates for nonlinear systems [40]. In realistic applications, chaotic behaviour can arise from even the simplist nonlinear dynamical systems where the linear theory cannot hold to any further extent. Systems exhibiting such chaotic behaviour naturally tend to have a stange attractor in the phase space which is typically explained by the fractal dimension. The FDim of the strange attractors is the most robust estimator of the intrinsic dimension. In this paper, an unconventional method based on FDim as a nonlinear correlation measure [42] is proposed to search the nonlinear relations in the extended data matrix.

*4.2. The Proposed New Method of Optimal Dynamical Structure Construction*

Having $\boldsymbol{X} = \{\{\boldsymbol{x}_i\}_{i=1:m}\}$ a data set based on a process and $\boldsymbol{x}_i$ are its intercorrelated variables. A subset $\boldsymbol{\mathcal{S}} = \{\{\boldsymbol{x}_i\}_{i=1:d}\} \in \boldsymbol{X}$ is a set of independent variables whose total or partial combinations define the relations expressing each variable in $\boldsymbol{\mathcal{R}}$ such that $\boldsymbol{\mathcal{R}} = \boldsymbol{X} - \boldsymbol{\mathcal{S}} = \{\{\boldsymbol{x}_i\}_{i=1:(m-d)}\}$ i.e. there exists a mapping $f: \boldsymbol{\mathcal{S}} \to \boldsymbol{\mathcal{R}}$ that gives rise to linear/nonlinear relations from variables in $\boldsymbol{\mathcal{S}}$ to every variable in $\boldsymbol{\mathcal{R}}$. Hence referring that FDim provides the exact number of independent variables, the objective is to find the true dimension of $\boldsymbol{X}$ that is the cardinality of $\boldsymbol{\mathcal{S}}$ using the exact FDim estimator and therefore the number of relations as the main aspect in the dynamical structure construction. Based on this statement, the proposed procedure to determine the optimal lag structure consists of the following steps:

1. A dynamical structure is constructed through attractor reconstruction of phase space. The dimension of this space is selected by finding relatively independent pieces of information. The common mode is to choose a variable $\boldsymbol{x}(t)$ and its $l$ derivatives to span its $\{l+1\}$-dimensional space. For data points separated by a time interval $\Delta t$, the derivatives are substituted with measurable differences as:

$$\left.\begin{aligned} \frac{d\boldsymbol{x}}{dt} &\approx \frac{\boldsymbol{x}(t) - \boldsymbol{x}(t-\Delta t)}{\Delta t} \\ \frac{d^2\boldsymbol{x}}{dt^2} &\approx \frac{\boldsymbol{x}(t) + 2\boldsymbol{x}(t-\Delta t) - \boldsymbol{x}(t-2\Delta t)}{\Delta t^2} \\ &\vdots \end{aligned}\right\} \tag{10}$$

Practically, higher order derivatives are corrupted by noise. An unassuming and yet better way is to use a set of time-delay variables, where the points in the phase space are $(l+1)$-tuple of consecutive values of the coordinates: $\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), \boldsymbol{x}(t-2\tau), \cdots, \boldsymbol{x}(t-l\tau)$ [57]. Extending this form to multivariate data, the phase space reconstruction is then: $\boldsymbol{X}(t), \boldsymbol{X}(t-\tau), \boldsymbol{X}(t-2\tau), \cdots, \boldsymbol{X}(t-l\tau)$. For general use, $\tau$ is selected as the spacing interval $\Delta t$. This forms the generic phased space reconstruction representing a dynamical structure as $\widetilde{\boldsymbol{X}} \triangleq [\boldsymbol{X}(t)\ \boldsymbol{X}(t-\Delta t)\ \boldsymbol{X}(t-2\Delta t)\ \cdots\ \boldsymbol{X}(t-l\Delta t)] \in \Re^{(N-l)\times m(l+1)}$ which is indeed equivalent to Eq. (9) in the sense of sampled data sets. Hence, this phase space is formed to captures the dynamics of the normal operating conditions attractor.

2. The fractal dimension is evaluated for the temporal data on the basis that it has the property of defining the linear and nonlinear correlations between the phase space variables. Using Eqs. (3), (5) and (6), FDim is computed through the CDim algorithm providing the cardinality of $\boldsymbol{\mathcal{S}}$ related to $\widetilde{\boldsymbol{X}}$. In these equations, the parameter $\varepsilon$ is substituted with values from the set of distances $\{\delta_{ij}\}_{i=1:N-l}^{j=1:N-l}$ sorted in ascending order. Taking all the possible values of distances separating the points in $\widetilde{\boldsymbol{X}}$ can handle the limitation of the required number of samples encountered when estimating the correlation dimension [42], since there will be $\frac{N-l(N-l-1)}{2}$ pairs of points $\left(log(\delta_{ij}),\ log\left(C(\delta_{ij})\right)\right)$ in the $log-log$ plot holding a sufficient amount of information that can be explored about the structural properties of the data.

3. The number of linear and/or nonlinear static and/or dynamical relations $r$ is the number of variables in the phase space minus the fractal dimension which is, in fact, the cardinality of subset $\boldsymbol{\mathcal{R}}$:

$$r = m(l+1) - \mathrm{CDim} \tag{11}$$

4. The number of new relations $r_n$ is computed as in [18], the number of relations when $l = 0$ is the number of static relations $r(l=0)$. The number of new relationships is strictly related to the correlation dimension of the data matrix under analysis. The inclusion of lagged features reveals new linear and/or nonlinear dynamical relations in the phase space at each increasing degree of dynamics. Hence, the number of new relations is computed by omitting the effect of the previously exposed relationships. The number of new relations is then given as:

$$r_n = r(l) - \sum_{i=0}^{l-1}(l-i+1)r_n(i) \tag{12}$$

$r_n$ is a decreasing function with respect to the number of lags. The process of exposing new raltionships is continued until all the important are incorporated in the augmented matrix. At this stage, the optimal dynamical structure is constructed with $l = l_{opt}$ as:

$$\widetilde{\boldsymbol{X}}_{opt} = \left[\boldsymbol{X}(t)\, \boldsymbol{X}(t-\Delta t)\, \cdots\, \boldsymbol{X}\left(t - l_{opt}\Delta t\right)\right] \in \Re^{N_D \times m(l_{opt}+1)} \tag{13}$$

with $N_D = N - l_{opt}$ is the new size of the sampling points attained after forming the optimal dynamical structure.

This optimal dynamical structure is related to attractor reconstruction in the phase space based on time-delayed coordinates and computation of FDim in accordance. The simplified procedure is provided in Table 2.

**Table 2.** Pseudo-code of fractal-based dynamical structure construction

**Algorithm**

**Input:** Raw input data $\boldsymbol{X} = \{\{\boldsymbol{x}_i\}_{i=1:m}\}$
**Output:** The optimal dynamical structure $\widetilde{\boldsymbol{X}}_{\boldsymbol{opt}} \in \Re^{\boldsymbol{N_D \times m(l_{opt}+1)}}$

Step 1: Normalise the input data features.
Step 2: Set $l = 0$
**Repeat**
  Step 3: Construct the augmented data matrix $\widetilde{\boldsymbol{X}} \triangleq [\boldsymbol{X}(t)\ \boldsymbol{X}(t-\Delta t)\ \boldsymbol{X}(t-2\Delta t)\ \cdots\ \boldsymbol{X}(t-l\Delta t)] \in \Re^{(N-l)\times m(l+1)}$
  Step 4: Compute $\mathrm{CDim}(\widetilde{\boldsymbol{X}})$.
  Step 5: Compute the number of relationships $r = m(l+1) - \mathrm{CDim}$ using Eq. (11).
  Step 6: Compute the number of new relationships $r_n(l)$ using Eq. (12).
**Until** $r_n(l) \leq 0$ **if not** set $l = l + 1$
Step 7: Set $\boldsymbol{l_{opt}}$ and construct the optimal dynamical structure $\widetilde{\boldsymbol{X}}_{\boldsymbol{opt}} \in \Re^{\boldsymbol{N_D \times m(l_{opt}+1)}}$

### *4.3. Kernel PCA*

Standard PCA is a rotation of the original axes by finding the orthogonal vectors that coincide with directions of maximum variance. The new axes are then a linear combination of the original axes executed through a linear transformantion. Therefore, standard PCA is inadequate to extract the nonlinear patterns incorporated in the complex realistic processes. Consequently, a nonlinear extension has been introduced to PCA through the use of kernel functions [33] that is the Kernel PCA (KPCA) [17]. KPCA maps the original data $\boldsymbol{X}^T = \{\{\boldsymbol{x}_i\}_{i=1:N}\} \in \Re^{m\times N}$ from the input space $\mathcal{X}$ to an implicit kernel-induced feature space $\mathcal{H}$ through a nonlinear mapping $\boldsymbol{\Phi}$ as:

$$\boldsymbol{\Phi}\colon \mathcal{X} \longrightarrow \mathcal{H}, \boldsymbol{x} \longrightarrow \boldsymbol{\Phi}(\boldsymbol{x}) \tag{14}$$

Assuming that the points in the feature space are centred i.e.

$$\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}(\boldsymbol{x}_i) = 0 \tag{15}$$

The covariance matrix in the feature space $\mathcal{H}$ is a $N \times N$ matrix given by:

$$\boldsymbol{C} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}(\boldsymbol{x}_i)\,\boldsymbol{\Phi}(\boldsymbol{x}_i)^T \tag{16}$$

Detecting nonlinear relations in the input space $\mathcal{X}$ is reduced to extracting the linear relations in the nonlinear feature space $\mathcal{H}$ by means of the standard PCA applied to the new feature data points. This can be achieved by the eigendecomposition of the covariance matrix as follows:

$$\boldsymbol{C}\boldsymbol{p}^{\hbar} = \lambda^{\hbar}\boldsymbol{p}^{\hbar} \tag{17}$$

Where the eigenvectors $\boldsymbol{p}^{\hbar}$corresponding to the $\hbar^{th}$ eigenvalue $\lambda^{\hbar}$ for all $N$ eigenvalues $\left(\lambda^{\hbar} \neq 0\right)_{\hbar=1}^{N}$ are in the span of the feature space data $\boldsymbol{p}^{\hbar} \in \mathrm{span}\left\{\{\boldsymbol{\Phi}(\boldsymbol{x}_i)\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}\right\}$ weighted by the coefficients $\alpha_i^{\hbar}$ as:

$$\boldsymbol{p}^{k} = \sum_{i=1}^{N} \alpha_i^{k} \boldsymbol{\Phi}(\boldsymbol{x}_i) \tag{18}$$

Substituting Eq. 16 and Eq. 18 in Eq. 17, we get:

$$\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\Phi}(\boldsymbol{x}_i)\, \boldsymbol{\Phi}(\boldsymbol{x}_i)^T \sum_{i=1}^{N} \alpha_i^{k} \boldsymbol{\Phi}(\boldsymbol{x}_i) = \lambda^{k} \sum_{i=1}^{N} \alpha_i^{k} \boldsymbol{\Phi}(\boldsymbol{x}_i) \tag{19}$$

Defining the kernel function:

$$K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)\,; \qquad \kappa\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big) = \langle \boldsymbol{\Phi}(\boldsymbol{x}_i), \boldsymbol{\Phi}\big(\boldsymbol{x}_j\big) \rangle = \boldsymbol{\Phi}(\boldsymbol{x}_i)^T \boldsymbol{\Phi}(\boldsymbol{x}_j) \tag{20}$$

The kernel matrix is then denoted as $\boldsymbol{K} = \big\{K_{ij}\big\}_{i=1:N}^{j=1:N}$ with entries $K_{ij}$

Using the last two equations, the condensed form to solve for the coefficients $\alpha_i^{k}$ based on the kernel function is:

$$\boldsymbol{K}\boldsymbol{\alpha}^{k} = N\lambda^{k}\boldsymbol{\alpha}^{k} \tag{21}$$

$\boldsymbol{\alpha}^{k}$ is the N-dimensional column vector of the weighting coefficients $\alpha_i^{k}$:

$$\boldsymbol{\alpha}^{k} = [\alpha_1^{k}\ \alpha_2^{k}\ \cdots \alpha_N^{k}]^T \tag{22}$$

These weighting vectors are normalized to ensure that the eigenvectors are of unit-length as:

$$\hat{\boldsymbol{\alpha}}^{k} = \frac{\boldsymbol{\alpha}^{k}}{\sqrt{\lambda^{k}}} \tag{23}$$

Hence, the unit-length eigenvectors are:

$$\boldsymbol{p}^{k} = \frac{1}{\sqrt{\lambda^{k}}} \sum_{i=1}^{N} \alpha_i^{k} \boldsymbol{\Phi}(\boldsymbol{x}_i) \tag{24}$$

Based on the radial basis function, the unnormalised Gaussian kernel is chosen with kernel width $\sigma$ as:

$$\kappa\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big) = \exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 \Big/ 2\sigma^2\right) \tag{25}$$

Similar to PCA, the feature space data $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ must be centred. Since the computational cost of this requirement is usually prohibitive, the calculations in the feature space $\boldsymbol{\Phi}$ are implemented in an implicit form in the lower-dimensional kernel matrix. This can be achieved by substituting the kernel matrix with the Gram matrix , the Gram matrix is then calculated as follows:

$$\widehat{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{I}_N - \boldsymbol{I}_N\boldsymbol{K} + \boldsymbol{I}_N\boldsymbol{K}\boldsymbol{I}_N \tag{26}$$

where $\boldsymbol{I}_N$ is an $N \times N$ matrix with all entries equal to $1/N$.

The kernel principal components $\boldsymbol{t}^{k}$ (or the feature scores) are calculated based on the weighting vectors $\boldsymbol{\alpha}^{k}$ obtained through the solution of Eq. (21). For the input space sample $\boldsymbol{x}$, the feature space sample $\boldsymbol{\Phi}(\boldsymbol{x})$ is projected onto the eigenvector $\boldsymbol{p}^{k}$ as:

$$t^{k} = \langle \boldsymbol{p}^{k}, \boldsymbol{\Phi}(\boldsymbol{x}) \rangle = \frac{1}{\sqrt{\lambda^{k}}} \sum_{i=1}^{N} \alpha_i^{k} \langle \boldsymbol{\Phi}(\boldsymbol{x}_i), \boldsymbol{\Phi}(\boldsymbol{x}) \rangle = \frac{1}{\sqrt{\lambda^{k}}} \sum_{i=1}^{N} \alpha_i^{k} \kappa(\boldsymbol{x}_i, \boldsymbol{x}) = \sum_{i=1}^{N} \hat{\alpha}_i^{k} \kappa(\boldsymbol{x}_i, \boldsymbol{x}) \tag{27}$$

The number of retained principal components $\gamma$ is then selected to construct the lower $\gamma$-dimensional principal scores $\boldsymbol{t} = \big(t^{k}\big)_{k=1}^{\gamma}$ for $\gamma < N$.

## 5. Fractal-based DKPCA Proposed framework

The proposed FDKPCA monitoring procedure consists of two stages, namely, the offline modelling and the online monitoring. The complete procedure is illustrated as a flowchart in Fig. 1, and is summarized as follows:

### *5.1. Fault Detection Phase*

- **The training stage and explatory data analysis (Stage I):**

Given the input data matrix $\boldsymbol{X} = \{\{\boldsymbol{x}_i\}_{i=1:m}\} \in \Re^{N\times m}$ with sampling point $\boldsymbol{x}_i$. All the input features are normalized to a standard scale ha zero mean and unit variance. The optimal dynamical structure is constructed using the new procedure proposed in Subsection 4.2, obtaining the optimal augmented matrix $\widetilde{\boldsymbol{X}}_{opt} = \left[\boldsymbol{X}(t)\ \boldsymbol{X}(t-\Delta t)\ \cdots\ \boldsymbol{X}\left(t-l_{opt}\Delta t\right)\right] \in \Re^{N_D\times m(l_{opt}+1)}$. Having constructed the optimal dynamical structure, KPCA is implemented in an implicit way whose pseudo-code is provided in Table 3.

**Table 3.** Pseudo-code of KPCA on the augmented matrix

| **Algorithm** |
|---|
| **Input:** Augmented data matrix $\widetilde{\boldsymbol{X}}_{opt}^{T} \in \boldsymbol{\Re}^{\boldsymbol{m(l_{opt}+1)\times N_D}}$ |
| **Output:** Nonlinear Feature score matrix $\boldsymbol{T}_{ref} \in \boldsymbol{\Re}^{\boldsymbol{N_D\times\gamma}}$ |
| Step 1: Construct the kernel matrix $\boldsymbol{K}$ (Eq. (20)) using Gaussian kernel (Eq. (25)). |
| Step 2: Compute the Gram matrix (Eq. (26)) to obtain the centred kernel matrix $\widehat{\boldsymbol{K}}$. |
| Step 3: Compute $\left(\boldsymbol{\alpha}^{\hbar}\right)_{\hbar=1}^{p}$ (Eq. (21)) for the eigenvectors corresponding to the first $p(< N_D)$ eigenvalues s.t. $\lambda^p \geq 10^{-10}$ (a limit defined to consider an eigenvalue nonzero in practical calculations). |
| Step 4: Normalize the coefficients $\left(\boldsymbol{\alpha}^{\hbar}\right)_{\hbar=1}^{p}$ using Eq. (23). |
| Step 5: Replace $\boldsymbol{x}$ with $\boldsymbol{x}_j$ in Eq. (27) and calculate the normal nonlinear feature scores $\boldsymbol{t}_{ref}^{j} = \left(t_{ref,j}^{\hbar}\right)_{\hbar=1}^{p}$. |
| Step 6: Repeat Step 5 for all vectors $\left\{\boldsymbol{x}_j\right\}_{j=1}^{N_D}$ in the augmented training set to obtain the feature score matrix $\boldsymbol{T}_{ref} = \left\{\boldsymbol{t}_{ref}^{j}\right\}_{j=1}^{N_D} \in \Re^{N_D\times p}$. |
| Step 7: Calculate the correlation dimension $\mathrm{CDim}(\boldsymbol{T}_{ref})$ to estimate the NPC γ. |
| Step 8: The retained feature score matrix for the training data set is $\boldsymbol{T}_{ref} = \left\{\boldsymbol{t}_{ref}^{j}\right\}_{j=1}^{N_D} \in \Re^{N_D\times\gamma}$. |

From the nonlinear principal scores and residuals, the two reference statistics, $T_{ref}^2$ and $SPE_{ref}$, are derived for the in-control training data set as follows [58]:

$$T_{ref,j}^2 = \left[t_{ref,j}^1, t_{ref,j}^2, \cdots, t_{ref,j}^{\gamma}\right]\boldsymbol{\Lambda}^{-1}\left[t_{ref,j}^1, t_{ref,j}^2, \cdots, t_{ref,j}^{\gamma}\right]^T \tag{28}$$

And

$$SPE_{ref,j} = \sum_{\hbar=1}^{p}\left(t_{ref,j}^{\hbar}\right)^2 - \sum_{\hbar=1}^{\gamma}\left(t_{ref,j}^{\hbar}\right)^2 \tag{29}$$

where $\boldsymbol{\Lambda}$ is the diagonal eigenvalue matrix with diagonal entries $\left(\lambda^{\hbar}\right)_{\hbar=1}^{\gamma}$.

The control limits (or thresholds) based on the above in-control statistics are defined as $J_{th,T^2}$ and $J_{th,SPE}$ at pre-defined significance level $\mathcal{I}$ (type I error). Eventually, a detection rule is established to indicate the status of the process.

- **The testing stage and online process monitoring (Srage II):**

Once the reference model and the thresholds are established in the training stage (Stage I), they are employed in the online testing stage in order to monitor the process underevaluation. In concise, the $N_{tst}$ testing samples represented as a data set $\boldsymbol{X}_{tst} = \left\{\boldsymbol{x}_{tst}^{j}\right\}_{j=1}^{N_{tst}} \in \Re^{N_{tst}\times m}$ are pre-processed; this data is normalised and shifted by $l_{opt}$ lags according to the dynamical structure constructed in the training stage as:

$$\widetilde{\boldsymbol{X}}_{tst} = \left[\boldsymbol{X}_{tst}(t)\ \boldsymbol{X}_{tst}(t-\Delta t)\ \cdots \boldsymbol{X}_{tst}\left(t-l_{opt}\Delta t\right)\right] \in \Re^{N_{tst,D}\times m(l_{opt}+1)} \tag{30}$$

with, $N_{tst,D} = N_{tst} - l_{opt}$.

For each new observation $\boldsymbol{x}_{tst}^{j}$ from the above testing data set, the testing kernels are computed with reference to the training data points $\widetilde{\boldsymbol{X}}_{opt} = \{\boldsymbol{x}_i\}_{i=1}^{N_D}$ according to Eq. (20) as:

$$K_{ij}^{tst} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{tst}^{j}) \tag{31}$$

The corresponding testing Gram matrix is then computed as:

$$\widehat{\boldsymbol{K}}_{tst} = \boldsymbol{K}_{tst} - \boldsymbol{K}_{tst}\boldsymbol{I}_{N_D} - \boldsymbol{I}_{N_{tst,D}}\boldsymbol{K} + \boldsymbol{I}_{N_{tst,D}}\boldsymbol{K}\boldsymbol{I}_{N_D} \tag{32}$$

where $\boldsymbol{I}_{N_{tst,D}}$ is $N_{tst,D} \times N_D$ matrix with all entries equal to $1/N_D$. Henceforth using Eq. (27), the testing kernel scores are calculated as follows:

$$t_{tst,j}^{\hbar} = \frac{1}{\sqrt{\lambda^{\hbar}}}\sum_{i=1}^{N_D} \alpha_i^{\hbar}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{tst}^{j}) \text{ for } \hbar = 1{:}\gamma \tag{33}$$

From these kernel feature scores, the detection statistics of the Hotelling's $T^2$ and $SPE$ for $\boldsymbol{x}_{tst}^{j}$ are calculated by:

$$T^2 = \left[t_{tst,j}^{1}, t_{tst,j}^{2}, \cdots, t_{tst,j}^{\gamma}\right]\boldsymbol{\Lambda}^{-1}\left[t_{tst,j}^{1}, t_{tst,j}^{2}, \cdots, t_{tst,j}^{\gamma}\right]^{T} \tag{34}$$

$$SPE = \sum_{\hbar=1}^{p}\left(t_{tst,j}^{\hbar}\right)^2 - \sum_{\hbar=1}^{\gamma}\left(t_{tst,j}^{\hbar}\right)^2 \tag{35}$$

These two control charts are therefore checked against the thresholds ($J_{th,T^2}$ and $J_{th,SPE}$). To affirm that the process is in-control the two detection statistics have to remain below the thresholds. Alternatively, if any point gets beyond the thresholds, then the process is considered out-of-control and an abnormal behaviour exists in the process measurements.

### *5.2. Fault Diagnosis Phase*

The faults detected using the proposed framework can be diagnosed in order to extract the root causes of the fault by determining which variables are directly related to the occurrence of the abnormal behaviour. This can be achieved through the analysis of the contribution plots based on both control charts $T^2$ and $SPE$. Contribution plots [59] have been successfully applied to fault diagnosis purposes [60, 61]. The kernel-based contribution plots used are based on the method proposed by [62] with substituting the kernels with dynamic kernels derived in this work as:

$$CT^2(x_i) = x_i\frac{\partial T^2}{\partial x_i} \tag{36}$$

$$CSPE(x_i) = x_i\frac{\partial SPE}{\partial x_i} \tag{37}$$

$x_i$ is the observed value of the $i^{th}$ variable at time t ($\boldsymbol{x}_t = (x_i)_{i=1}^{m}$). From Eqs. (27), (34), and (35), $T^2$ and $SPE$ can be rewritten as:

$$T^2 = \boldsymbol{k}_{x_t}^{T}\widehat{\boldsymbol{A}}_{\gamma}\boldsymbol{\Lambda}^{-1}\widehat{\boldsymbol{A}}_{\gamma}^{T}\boldsymbol{k}_{x_t} \tag{38}$$

$$SPE = \boldsymbol{k}_{x_t}^{T}\left(\widehat{\boldsymbol{A}}_{p}\widehat{\boldsymbol{A}}_{p}^{T} - \widehat{\boldsymbol{A}}_{\gamma}\widehat{\boldsymbol{A}}_{\gamma}^{T}\right)\boldsymbol{k}_{x_t} \tag{39}$$

where, $\widehat{\boldsymbol{A}}_{\gamma} = \left\{\widehat{\boldsymbol{\alpha}}^{\hbar}\right\}_{\hbar=1}^{\gamma}$, $\widehat{\boldsymbol{A}}_{\gamma} = \left\{\widehat{\boldsymbol{\alpha}}^{\hbar}\right\}_{\hbar=1}^{p}$, and $\boldsymbol{k}_{x_i} = \left[\kappa(\boldsymbol{x}_1, \boldsymbol{x}_t)\ \kappa(\boldsymbol{x}_2, \boldsymbol{x}_t)\ \cdots \kappa\left(\boldsymbol{x}_{N_D}, \boldsymbol{x}_t\right)\right]^{T}$ is already normalized according to the Gram-based scaling (Eq. (26)).

Consequently, Eqs. (36) and (37) becomes:

$$CT^2(x_i) = x_i \frac{\partial}{\partial x_i}\left(\boldsymbol{k}_{x_t}^T \widehat{\boldsymbol{A}}_\gamma \boldsymbol{\Lambda}^{-1} \widehat{\boldsymbol{A}}_\gamma^T \boldsymbol{k}_{x_t}\right) = 2x_i \frac{\partial \boldsymbol{k}_{x_t}^T}{\partial x_i} \widehat{\boldsymbol{A}}_\gamma \boldsymbol{\Lambda}^{-1} \widehat{\boldsymbol{A}}_\gamma^T \boldsymbol{k}_{x_t} \tag{40}$$

$$CSPE(x_i) = x_i \frac{\partial}{\partial x_i}\left(\boldsymbol{k}_{x_t}^T \left(\widehat{\boldsymbol{A}}_p \widehat{\boldsymbol{A}}_p^T - \widehat{\boldsymbol{A}}_\gamma \widehat{\boldsymbol{A}}_\gamma^T\right) \boldsymbol{k}_{x_t}\right) = 2x_i \frac{\partial \boldsymbol{k}_{x_t}^T}{\partial x_i} \left(\widehat{\boldsymbol{A}}_p \widehat{\boldsymbol{A}}_p^T - \widehat{\boldsymbol{A}}_\gamma \widehat{\boldsymbol{A}}_\gamma^T\right) \boldsymbol{k}_{x_t} \tag{41}$$

Computing the testing contributions relative to the normal operating contributions as:

$$RCT^2(x_i) = \frac{CT^2(x_i) - mean(C^n T^2(x_i))}{std(C^n T^2(x_i))} \tag{42}$$

$$RCSPE(x_i) = \frac{CSPE(x_i) - mean(C^n SPE(x_i))}{std(C^n SPE(x_i))} \tag{43}$$

The contribution plots are derived for the normal operating conditions using Eqs. (40) and (41) with $\boldsymbol{x}_t$ are the observed vectors in the healthy data set. The contributions of each variable to the specific control chart are then obtained as the summation of the individual contributions in absolute value at each sample time in the desired region of analysis. For the sake of simplicity, the percentage of these contribution is used to depict the contribution plots.

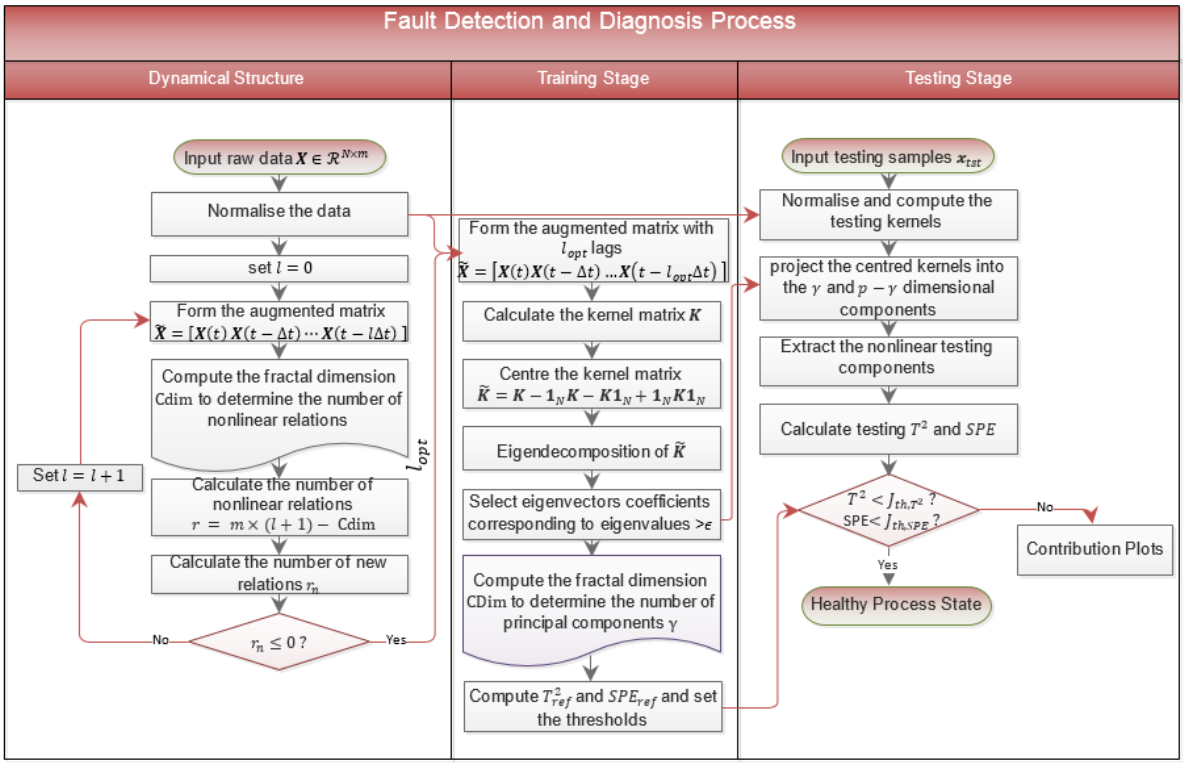


**Fig. 1. Schematic diagram of the proposed** Fractal-based DKPCA for fault detection and diagnosis framework

## 6. Application

The performance efficiency of fractal-based DKPCA is appraised by investigating three fault detection indices. These indices are: (i) the False Alarm Rate (FAR) quantifying the robustness and stability of the FDim system as the probability of type I error, (ii) Missed Detection Rate (MDR) for sensitivity evaluation as the probability of type II error, and (iii) fault detection Time Delay (TD). For each data set, a false alarm is triggered

whenever any of the control charts signals an out-of control observation while the process is in the fault-free status ($F = 0$), whereas a missed detect is to affirm that the process is in-control while it is operating in a faulty condition ($F \neq 0$):

$$\text{FAR}\% = \mathbb{P}(\text{Control chart} > \text{Threshold}|F = 0) \times 100\% \tag{44}$$

$$\text{MDR}\% = \mathbb{P}(\text{Control chart} < \text{Threshold}|F \neq 0) \times 100\ \% \tag{45}$$

$$\text{TD} = t_d - t_o \tag{46}$$

where $t_d$ is the fault detection time and $t_o$ is the fault occurrence time.

### *6.1. Case study 1: Simulated* $2 \times 2$ *nonlinear system*

Firstly, FDim-based number of PCs extraction is testified using a $2 \times 2$ system defined by nonlinear relations as follows [32]:

$$\begin{bmatrix} \boldsymbol{y}_1(k) \\ \boldsymbol{y}_2(k) \end{bmatrix} = \begin{bmatrix} (\boldsymbol{u}_1(k) + \boldsymbol{u}_2(k))^2 \\ (\boldsymbol{u}_1(k) - 2\boldsymbol{u}_2(k))^2 \end{bmatrix} \tag{47}$$

where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are measured process outputs, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the inputs taken from a white Gaussian noise $N(0,0.1)$. Clearly, the number of nonlinear relations in this system is two and the number of free variables is two as well which is the dimensionality of the data set. 100 samples are generated for $\boldsymbol{X} = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \boldsymbol{y}_1\ \boldsymbol{y}_2]$.

Fig. 2(a) shows the $log - log$ plots of the raw original data. It is readily discernible through the graphs and the estimated dimensionality that the correlation dimension provides a perfect estimation of the intrinsic dimension as the CDim computation yields CDim=1.8943 approaching 2 that is the real dimensionality of the system under analysis. An additional independent variable was added to attest the ability of FDim to detect the exact number of independent variables. The new system becomes 3×2 with new variable following $\boldsymbol{u}_3 \sim N(0,0.1)$ and $\boldsymbol{y}_1(k) = (\boldsymbol{u}_1(k) + \boldsymbol{u}_2(k) + \boldsymbol{u}_3(k))^2$, the dimension of the data is found to be CDim=2.8713 approximating 3, again with perfect estimation. Pursuing PCA modelling, the number of PCs is obtained using FDim evaluated on $\boldsymbol{\mathcal{T}}$ revealing $\text{FDim}(\boldsymbol{\mathcal{T}}) = 1.8970 \cong \text{FDim}(\boldsymbol{X}) = 1.8943$ for the $2 \times 2$ system hence $\{\text{the number of PCs}\} = \lceil d_{\boldsymbol{X}} \rceil = 2$; and for the $3 \times 2$ system, $\text{FDim}(\boldsymbol{\mathcal{T}}) = 2.8813 \cong \text{FDim}(\boldsymbol{X}) = 2.8713$ resulting in $\{\text{the number of PCs}\} = \lceil d_{\boldsymbol{X}} \rceil = 3$. The results are shown in Figs. 2(b-c) for the two systems. To show superiority, FDim-based extraction of the number of PCs is compared to Kaiser's rule [22], CPV [23], PA [24], and VRE [27] and the results are summarized in Table 2. It is obvious that CPV always overestimates the number of PCs while Kaiser's rule taking eigenvalues greater than 1, PA with random profile generation and VRE looking for a minimum of reconstructions provide vague estimations unrelated to the true relations for which the thresholds, assumptions, random parallel profiles, and considering specific disturbances associated with these methods limit the uniqueness and optimality of the number of PCs estimation. Whereas, FDim estimations present an exact analysis both in raw data and PCA-based model projections. A further comparison based on performance potentials is summarized in Table 3.

**Fig. 1.** Fractal-based DKPCA for fault detection and diagnosis procedure

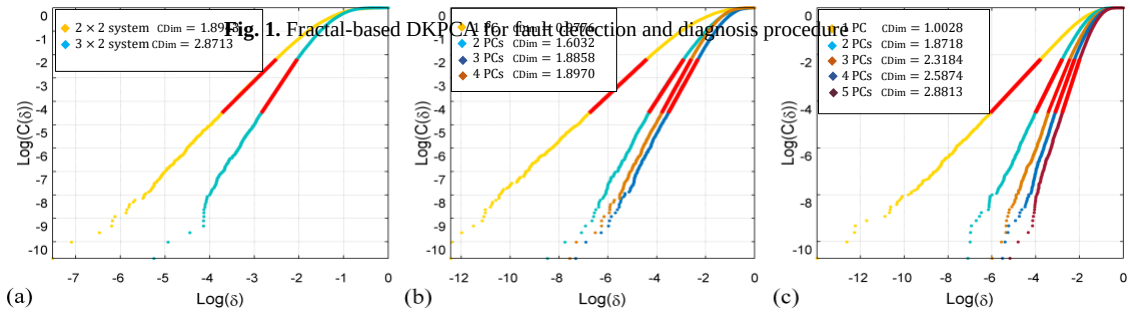


**Fig. 2.** CDim evaluation of (a) raw data sets having 2 and 3 independent variables (b) components in $\boldsymbol{\mathcal{T}}$ for the $2 \times 2$ system and (c) components in $\boldsymbol{\mathcal{T}}$ for the $3 \times 2$ system.

**Table 2.** Comparison results of FDim-based extraction of the number of variables and CPV, PA, and VRE.

| | FDim (raw data) | FDim (PCA projection) | Kaiser's rule | CPV (95% PCA projection) | PA (PCA projection) | VRE ((PCA projection) |
|---|---|---|---|---|---|---|
| 2 × 2 system | 1.8943 | [1.8970] = 2 | 1 | 3 | 1 | 1 |
| 3 × 2 system | 2.8713 | [2.8813] = 3 | 2 | 4 | 1 | 1 |

**Table 3.** Comparison of the different criteria for selecting the number of PCs

| | FDim | Kaiser's rule | CPV | PA | VRE |
|---|---|---|---|---|---|
| Reliable | ☑ | ☒ | ☒ | ☒ | ☒ |
| Unthresholded | ☑ | ☒ | ☒ | ☒ | ☑ |
| Eigenvalue-independent | ☑ | ☒ | ☒ | ☒ | ☒ |
| Uniqueness | ☑ | ☒ | ☒ | ☒ | ☒ |
| User-independent | ☑ | ☒ | ☒ | ☑ | ☑ |
| Unbiased IDim estimator | ☑ | ? | ? | ? | ? |
| Nonlinearity-accounted | ☑ | ☒ | ☒ | ☒ | ☒ |

☑≡approved, ☒≡not approved, ?≡not assessed

*6.2. Case study 2: PRONTO heterogeneous benchmark*

The proposed FDKPCA method is then validated through seven applications including six real faults using the Process Network Optimization (PRONTO) benchmark which is a real process made recently available for research in 2019. The PRONTO heterogeneous benchmark is a multiphase flow facility located at the Process System Engineering Laboratory of Cranfield University [63]. The multiphase flow is an automated industrial-scale process with high pressure operating conditions. For main processing, air, water, and oil flow rates are controlled and measured for transportation purposes. In this facility, air and water are mixed at the mixing zone, the mixed two-phase flow is pulled through the horizontal pipeline into the three-phase separator to separate the mixed flow into air and water. Consequently, water is reinjected to the system's water storage tank (T100) through the water coalescer by means of the water coalescer outlet valve (LVC502), and air is exhausted to the atmosphere through the air outlet valve three-phase separator (PIC501 corresponding to valve VC501) [64]. An in-depth description of the facility and its schematic can be found in [65]. In this work, the benchmark data sets are adopted to validate the proposed framework for fault detection and diagnosis. The benchmark data sets are collected for testing multiple monitoring algorithms. 17 variables are used to build the FDD framework as listed in Table 4.

**Table 4.** PRONTO process variables

| Variable | Description | unit |
|---|---|---|
| FT305/FT302 | Input air flow rate | $Sm^3/h$ |
| FT305-T | Input air temperature | ◦C |
| PT312 | Air delivery pressure | bar(g) |
| FT102/104 | Input water flow rate | Kg/s |
| FT102-T | Input water temperature | ◦C |
| FT102-D | Input water density | $kg/m^3$ |
| PT417 | Pressure in the mixing zone | bar(g) |
| PT408 | Pressure at the riser top | bar(g) |
| PT403 | Pressure in the 2-phase separator | bar(g) |
| FT404 | 2-phase separator output air flow rate | $m^3/h$ |
| FT406 | 2-phase separator output water flow rate | kg/s |
| PT501 | Pressure in the 3-phase separator | bar(g) |
| PIC501 | Air outlet valve 3-phase separator | (%) |
| LI502 | Water level 3-phase separator | (%) |
| LI503 | Water coalescer level | (%) |
| LVC502 | Water coalescer outlet valve | (%) |
| LI101 | Water tank level | m |

The measurements were taken at a sampling rate of 1 Hz. Normal operating conditions are implemented by varying the process inputs set point which are the air and water flow rates. For FDD, the faulty scenarios are induced at only two operating conditions which are:

-Operating condition A: 120 $Sm^3/h$ air, with 0.1 kg/s water flow rates;

-Operating condition B: 150 $Sm^3/h$ air, with 0.5 kg/s water flow rates.

Three types of incipient faults (air blockage, air leakage, and diverted flow) are manually injected to simulate the real malfunctions occurring in the process. The air blockage faults are induced manually by gradually closing

valve V11 whereas gradually opening valve V10 and U39 to simulate the air leakage and diverted flow faults, respectively.

**Table 5.** Application faults, fault-free and faulty data sets, and their sizes

| | Data set | | Samples (s) |
|---|---|---|---|
| Operating condition A | F0_A | Normal | 1961 |
| | F1 | Air blockage | 3843 |
| | F2 | Air leakage | 3181 |
| | F3 | Diverted flow | 7620 |
| Operating condition B | F0_B | Normal | 1100 |
| | F4 | Air leakage | 2880 |
| | F5 | Diverted flow | 4499 |
| Operating condition C | F0_C | Normal | 500 |
| | F6 | Slugging | 800 |

The multiphase flow facility also suffers from slugging presented at seven operating conditions as listed in table 3 in [65]. Slugging occurs when large bubbles comprised of the lighter fluid flows in riser isolated from the heavier fluid. Slugging slows down the fluids flow rates and results in oscillations in the pressure, flow rate, and density through the riser. In this paper, the slugging tested is taken in the operating condition C that corresponds to 100 $Sm^3$/h air, with 0.5 kg/s water flow rates.

The number of samples used for each testing data set with their corresponding normal operating data sets are presented in Table 5. where the data are selected s.t. the fault is induced at the 301$^{st}$ sample (after 301 s) in each faulty set. [63, 65] include the full details of these faults injection and data collection procedures as well as the severity characteristics and impact of each fault on the process.

6.2.1. Fault Detection results

Firstly, the optimal dynamical structure is constructed for the normal data sets extracted from the PRONTO heterogeneous benchmark based on the proposed fractal dimension algorithm. The results of the analysis described in in Subsection 4.2 on the healthy training data set are presented in Table 6. As mentioned earlier, the fractal dimension FDim is computed using the correlation dimension algorithm CDim based on the $log - log$ plots as shown in Fig. 3(a). Because of the mutual dependencies between process variables, it is a difficult task to determine the number of degrees of freedom of the system, however, since the FDim provides an unbiased IDim estimator, the optimal number of lags is determined adequately. Table 5 indicates that 3.3541 uncorrelated features and 13.6459 static nonlinear relations are present for the original data without lags for operating condition A (normal data set F0_A) along with 3.9429 uncorrelated features and 13.0571 static nonlinear relations for the normal data set (F0_B) in operating condition B. Since CDim provides the exact number of independent variables there will be no need to check the autocorrelation and crosscorrelation plots at each stage as suggested in [18]. When adding 3 lags ($l = 3$) to F0_A, the number of new nonlinear relations becomes 0.0102, implying that weak dynamical relations have been added to the data as Eq.25 limits the effects of old relations at each increasing order of dynamics. Additionally, the fractal dimension is tending to a steady value of around 4.3 for $l \geq 3$. Hence, the optimal dynamical structure is selected for $l_{opt_{\mathrm{F0\_A}}} = 2$ for 0.3307 new nonlinear relations. For operating condition B, in like manner, 3 lags ($l_{opt_{\mathrm{F0\_B}}} = 3$) are selected to express the nonlinear dynamics of the process with 0.1174 new nonlinear relations.

Kernel PCA was applied to the optimal augmented matrices with two and three lags for F0_A and F0_B, respectively. The number of PCs was selected using FDim as shown in Figs. 3(b) and 4(b), leading to $\mathrm{FDim}\left(\boldsymbol{T}_{ref_{\mathrm{A}}}\right) = 1074.6325$ and $\mathrm{FDim}\left(\boldsymbol{T}_{ref_{\mathrm{B}}}\right) = 608.6708$. Since $d_{\tilde{\boldsymbol{X}}_{\mathrm{A}}} = 3.9579 < d_{\mathcal{T}_{\boldsymbol{A}}}$ and $d_{\tilde{\boldsymbol{X}}_{\mathrm{B}}} = 5.1110 < d_{\mathcal{T}_{\boldsymbol{B}}}$, hence $\gamma = \left\lceil d_{\boldsymbol{T}_{ref_{\mathrm{A}}}} \right\rceil = 1075$ and $\left\lceil d_{\boldsymbol{T}_{ref_{\mathrm{B}}}} \right\rceil = 609$ PCs are retained out of 1950 and 1089 components corresponding to the nonzero eigenvalues ($> \epsilon = 10^{-10}$) for operating conditions A and B respectively. The control charts ($T^2_{ref}$ and $SPE_{ref}$) were evaluated and the corresponding thresholds based on 99% Confidence Level (CL) were evaluated and the corresponding thresholds were established statistically based on the chosen significance level $\mathcal{I} = 0.01$. It is designed in such a manner that the desired FAR is not exceeded

through tuning the thresholds in the training statge (Stage I) . For an assumptive FAR ($\mathcal{I} = 0.01$) $\times 100\% = 1\%$ in the reference data set, the threshold is selected such that FAR% $= \mathbb{P}(\text{Control chart} > \text{Threshold}|\text{Training data set}) \times 100\% \leq 1\%$. This can be achieved by computing the $(1 - \mathcal{I}) \times 100\%$ percentiles of the corresponding distribution using the percentile approach for limits correction [29].

**Table 6.** Fractal analysis on training data sets (F0_A and F0_B) of the PRONTO heterogeneous benchmark

| | Operating condition A | | | | Operating condition B | | | |
|---|---|---|---|---|---|---|---|---|
| $l$ | $m(l+1)$ | CDim | $r(l)$ | $r_n(l)$ | $m(l+1)$ | CDim | $r(l)$ | $r_n(l)$ |
| 0 | 17 | 3.3541 | 13.6459 | 13.6459 | 17 | 3.9429 | 13.0571 | 13.0571 |
| 1 | 34 | 3.8214 | 30.1786 | 2.8869 | 34 | 4.6030 | 29.3970 | 3.2827 |
| 2 | 51 | 3.9579 | 47.0421 | 0.3307 | 51 | 4.9157 | 46.0843 | 0.3475 |
| 3 | 68 | 4.0843 | 63.9157 | 0.0102 | 68 | 5.1110 | 62.8890 | 0.1174 |
| 4 | 85 | 4.1605 | 80.8395 | 0.0501 | 85 | 5.2416 | 79.7584 | 0.0647 |
| 5 | 102 | 4.2226 | 97.7774 | 0.0142 | 102 | 5.2321 | 96.7679 | 0.1402 |
| 6 | 119 | 4.2738 | 114.7262 | 0.0109 | 119 | 5.2586 | 113.7414 | -0.0361 |
| 7 | 136 | 4.3113 | 131.6887 | 0.0137 | | | | |
| 8 | 153 | 4.3099 | 148.6901 | 0.0388 | | | | |
| 9 | 170 | 4.3300 | 165.6700 | -0.0215 | | | | |

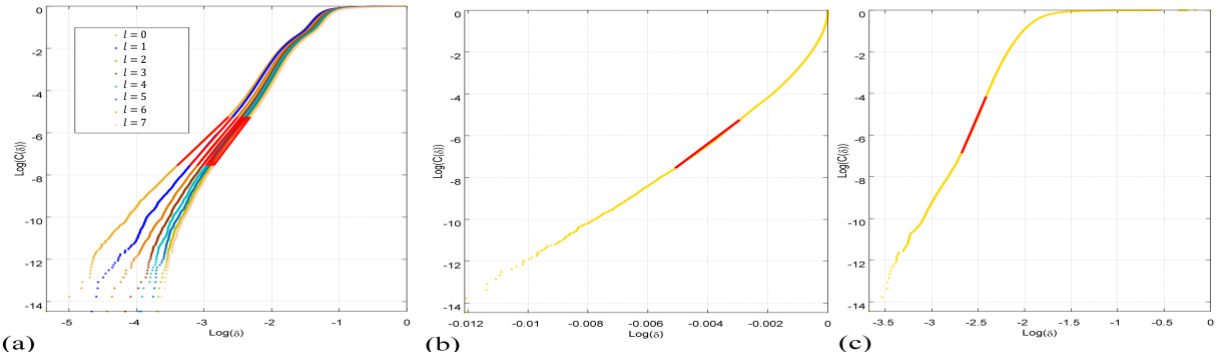


**Fig. 3.** $Log - Log$ plots of (a) fractal analysis of the training data F0_A with different lags, (b) number of PCs extraction in FDKPCA Dim $\left(\boldsymbol{T}_{ref_{\mathrm{A}}}\right) = 1074.6325$, and (c) number of PCs extraction in FDPCA FDim$(\boldsymbol{\mathcal{T}}_{\mathrm{A}}) = 10.2668$.

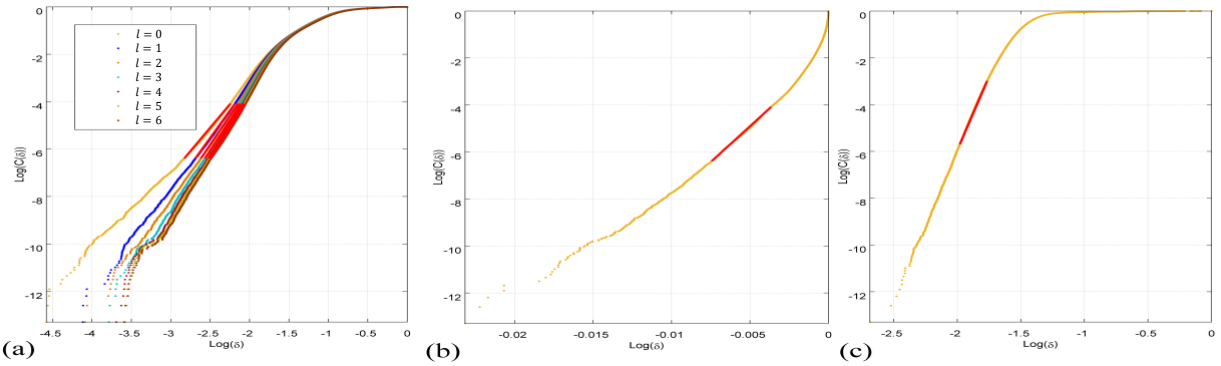


**Fig. 4.** $Log - Log$ plots of (a) fractal analysis of the training data F0_ B with different lags, (b) number of PCs extraction in FDKPCA FDim $\left(\boldsymbol{T}_{ref_{\mathrm{B}}}\right) = 608.6708$, and (c) number of PCs extraction in FDPCA FDim$(\boldsymbol{\mathcal{T}}_{\mathrm{B}}) = 12.6314$.

Table 7 lists the results of evaluating FAR, TD, and MDR for the different faulty scenarios using FDKPCA and compared to PCA, KPCA, DPCA, and FDPCA. The lag structure in the DPCA method was constructed based on the algorithm proposed in [18] using PA for selecting the number of PCs to retain. This method proposed augmenting the training matrices using two lags in time and the retained PCs was given as 5 and 6 for operating conditions A and B, respectively. The lag structure in FDPCA must be the same as for FDKPCA, while the number

of retained PCs selected using FDim are shown in Figs. 3(c), 4(c) where $\mathrm{FDim}(\boldsymbol{\mathcal{T}}_{\mathrm{A}}) = 10.2668$ and $\mathrm{FDim}(\boldsymbol{\mathcal{T}}_{\mathrm{B}}) = 12.6314$. Henceforth, $\lceil d_{\mathcal{T}_{\mathrm{A}}} \rceil = 11$ PCs and $\lceil d_{\mathcal{T}_{\mathrm{B}}} \rceil = 13$ PCs were retained since $d_{\tilde{\mathbf{X}}_{\mathrm{A}}} = 3.9579 < d_{\mathcal{T}_{\boldsymbol{A}}}$ and $d_{\tilde{\mathbf{X}}_{\mathrm{B}}} = 5.1110 < d_{\mathcal{T}_{\boldsymbol{B}}}$. Table 7 reflects that the proposed FDKPCA method outperforms the other methods by reaching the optimal performance for most of the faults where the best results are highlighted in red. These results prove the proficiency of the proposed FDKPCA in detecting various range of faults of different types in comparison with the other methods. The proposed FDPCA is yet a powerful paradigm which is still less proper in detecting accurately the anomalies introduced in nonlinear processes but outperforms the parallel analysis-based DPCA proposed by Ku et al. [18]. This provides further evidence that FDim provides the ideal number of PCs needed to interpret the data and PCA-based projections in an optimal way. However, the generic proposed nonlinear dynamical framework FDKPCA accomplished the best monitoring results. This can be seen also from the monitoring plots depicted in Figs. 5 and 6 showing the air blockage and air leakage faults in operating condition A and Fig. 7 visualizing the diverted flow fault in operating condition B using FDPCA and FDKPCA, where the horizontal green line presents the thresholds and the faulty region is highlighted in light red. The faults are induced at 301 s while the valves V10 (F1) and U39 (F5) are opened gradually whereas valve V11 (F2) is closed. The proposed FDPCA instantaneously detects the changes in the systems' dynamics while opening or closing a valve. The correlation dimension integrated with DKPCA possesses the potential to detect the correlated measurements and locate the changes in nonlinear dynamics. This substantial integration is not introduced in the remaining methods where the delay of detection is misleadingly large. Small angles of valves opening or closing cannot be recognised and the fault could not be detected so that maintaining faulty process operation. The monitoring charts $T^2$ and $SPE$ obtained through FDKPCA are precise in detecting such small incipient faults soothingly that recognizes all the faults with a best false alarms-detection time delay to detection rate trade-off.

**Table 7.** FAR%, TD(Samples in seconds), and MDR% for PCA, KPCA, DPCA, FDPCA, and FDKPCA

| | | Operation condition A | | | | | | Operation Condition B | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Air blockage | | Air leakage | | Diverted flow | | Air leakage | | Diverted flow | |
| | | $T^2$ | SPE | $T^2$ | SPE | $T^2$ | SPE | $T^2$ | SPE | $T^2$ | SPE |
| PCA | FAR (%) | 0 | 0 | 0 | 0 | 6.66 | 6 | 0 | 0.66 | 3.66 | 3 |
| | TD | 2995 | 319 | 635 | 480 | 7289 | 563 | 1071 | 12 | 3387 | 5 |
| | MDR (%) | 84.5 | 55.2 | 37.9 | 21.7 | 99.5 | 21.2 | 60.1 | 12.7 | 98.9 | 12.4 |
| KPCA | FAR (%) | 19.6 | 0 | 0 | 1.66 | 27.3 | 35.6 | 13.3 | 24.3 | 29.6 | 4.33 |
| | TD | 112 | 1293 | 625 | 595 | 608 | 328 | 481 | 15 | 444 | 43 |
| | MDR (%) | 67.0 | 77.9 | 21.6 | 21.8 | 85.0 | 91.1 | 20.6 | 25 | 32.5 | 68.2 |
| DPCA | FAR (%) | 0 | 0 | 0 | 2.33 | 4.66 | 1.33 | 0 | 1 | 1.66 | 1 |
| | TD | 2996 | 1293 | 636 | 626 | 7290 | 635 | 1074 | 278 | 3389 | 252 |
| | MDR (%) | 84.5 | 67.3 | 37.9 | 21.7 | 99.5 | 29.9 | 60.2 | 16.0 | 98.9 | 16.0 |
| FDPCA | FAR (%) | 0 | 0.66 | 0 | 1.33 | 4.66 | 3 | 0 | 2 | 1.66 | 1.33 |
| | TD | 2996 | 0 | 636 | 88 | 7290 | 0 | 1072 | 0 | 3389 | 1 |
| | MDR (%) | 84.5 | 55.3 | 34.3 | 21.9 | 99.5 | 22.4 | 59.3 | 21.2 | 98.9 | 13.0 |
| FDKPCA | FAR (%) | 0 | **3.42** | 2.39 | **0.68** | 2.05 | **1.36** | 2.74 | **0.34** | **0** | 3.09 |
| | TD | 2 | **0** | 1 | **0** | 1 | **0** | 1 | **0** | **1** | 0 |
| | MDR (%) | 11.0 | **0** | 16.6 | **0.45** | 5.23 | **0** | 0.34 | **0** | **0.00** | 0 |

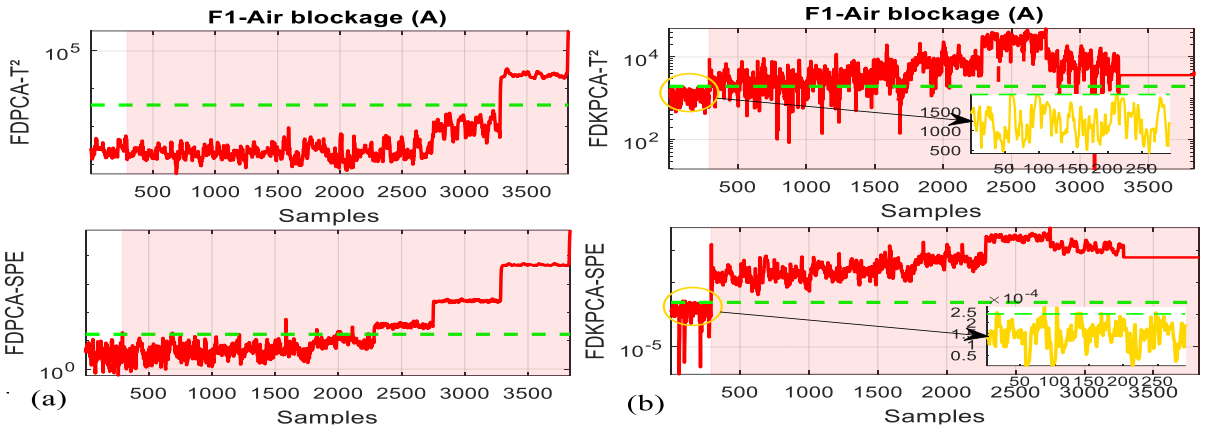


**Fig. 5.** Monitoring results of (a) FDPCA and (b) FDKPCA using $T^2$ and SPE at 99% CL of F1.

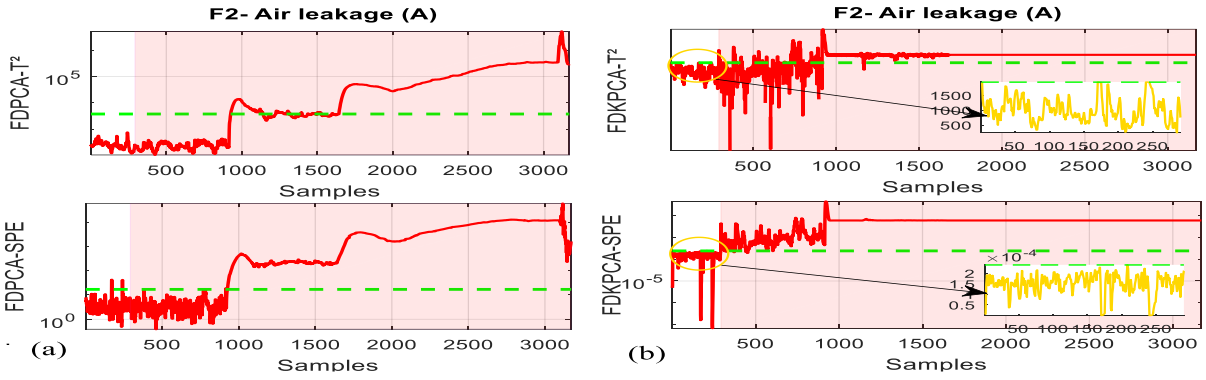


**Fig. 6.** Monitoring results of (a) FDPCA model and (b) FDKPCA using $T^2$ and SPE at 99% CL of F2

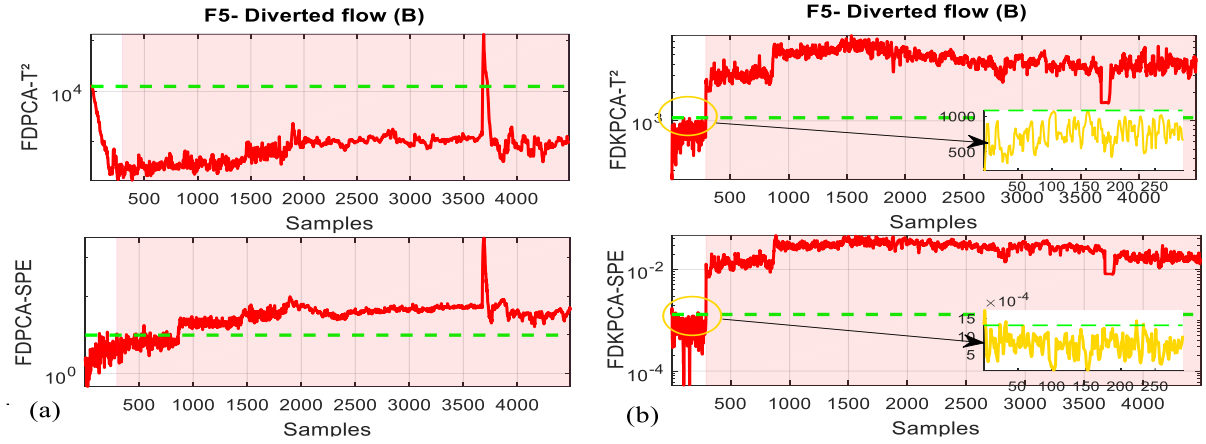


**Fig. 7.** Monitoring results of (a) FDPCA model and (b) FDKPCA using $T^2$ and SPE at 99% CL of F5

Slugging detection is also investigated in this work. Because slugging is just an unstable multiphase flow regime, its effect is not as severe as faults that generally lead to process failures if not detected and managed adequately. The slugging condition is detected using the generic proposed nonlinear dynamical framework FDKPCA easily and accomplished satisfactory monitoring results shown in Fig.8 (3.006% MDR after fault detected within 14s from occurrence (TD) and with 0% FAR before fault using $T^2$ and 2.20% MDR corresponding to 10 s TD with 7% FAR using $SPE$), hence capturing the systems nonlinear dynamics and oscillations simultaneously.


**Fig. 8.** Monitoring results of FDKPCA model using $T^2$ and SPE at 99% CL of slugging condition

**Table 8.** Comparison of computational time of the faulty scenarios

| | Operating condition A | | | Operating condition B | |
|---|---|---|---|---|---|
| Algorithm | F1 | F2 | F3 | F4 | F5 |
| PCA | 1.193 s | 1.015 s | 1.277 s | 1.381 s | 1.452 s |
| KPCA | 4.938 s | 4.825 s | 7.245 s | 2.660 s | 3.395 s |
| DPCA | 3.614 s | 3.361 s | 3.721 s | 2.654 s | 2.809 s |
| FDPCA | 6.325 s | 6.165 s | 6.548 s | 3.220 s | 3.278 s |
| FDKPCA | 21.779 s | 21.360 s | 32.627 s | 5.623 s | 6.925 s |

The computational time complexity is assessed to the five faults using the different methods compared in this work. Table 8. Shows the computational time in seconds from the training stage to the testing of each faulty scenario. Incontestably, more complex algorithms suffer from increased time complexity. PCA as the standard static linear method has the advantage of low complexity and therefore fast data processing. Kernel and dynamic methods rise the complexity as their processed features are highly increased. The computational time depends on the size of the data since the data sets have different number of observations (see Table 5.). Faulty sets with larger number of samples tested on larger training normal observations exhibit an increasing computational time. The computational time of FDKPCA was slightly increased (still within seconds) as the proposed method merges a dynamic kernel method with fractal dimension computations which are both complex algorithms, however, with substantially higher accuracy in detecting the abnormal events.

### 6.2.2. Fault Diagnosis results

After the faults are effectively detected using FDKPCA, the monitoring process passes into the diagnosis phase using both $T^2$ and $SPE$ statistics. Fig. 8 illustrates $CT^2$ and $CSPE$ after the fault is detected by the corresponding statistic according to the nonlinear contribution plots strategy stated in Subsection 5.2. From the diagnosis results, shown in Fig.8., the five abnormal events present changes to the process variables according to the main cause of each fault. Through the contribution plots, the root cause and influence of each fault is observed. Variable 3 (the air delivery pressure PT312) has the largest contribution to the air blockage fault (F1), air leakage (F2 and F4) implies the largest influence on the air outlet valve 3-phase separator (Variable 13 PIC501) as air leaks out to the atmosphere, PIC501 will close to maintain air as the input air reduces. The diverted flow faults (F3 and F5) are induced by gradually opening valve U39 which allowed the flow to be led straight to the riser, the contribution plots shows that this influence the variables 3 and 16 ( air delivery pressure PT312 and water coalecser outlet valve LVC502) and variable 17 ( water tank level LI101), respectively. It is noticed that both $CT^2$ and $CSPE$ can locate variables contributing to each fault up to immediately after the fault occurrence depending on the nonlinear components derived using the developed fractal-based DKPCA model.

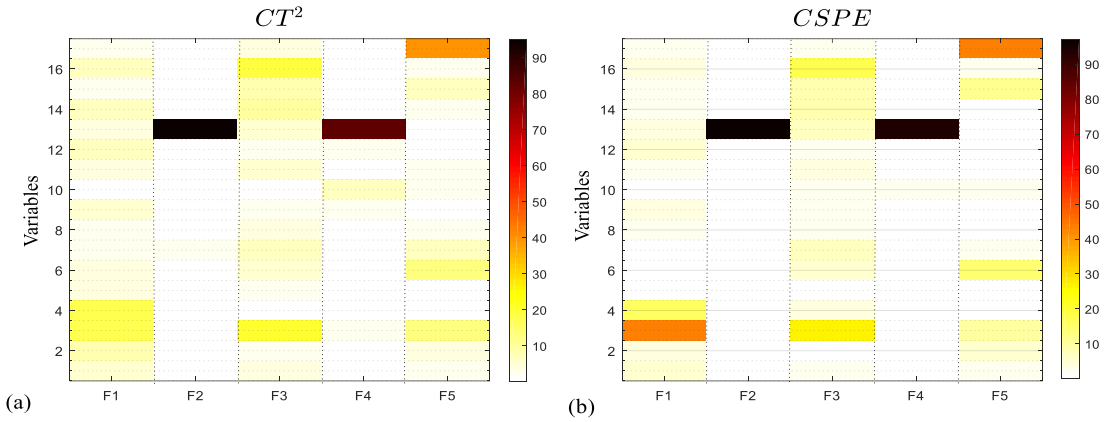


**Fig. 9.** Contribution plots for faults F1, F2, F3, F4, and F5 with (a) $T^2$ and (b) $SPE$

## 7. Conclusion

It is important, in complex systems, to design a monitoring framework that accurately detects component faults and abnormal behaviour precisely and diagnose its source in order to assure the correct operation conditions of the system and maintain high level of safety, efficiency, and reliability. This paper deals with nonlinear dynamical systems monitoring using the proposed Fractal-based DKPCA (FDKPCA) with a new methodology to construct the optimal dynamical structure based on the fractal dimension. It was further used as a nonlinear feature parameter to select the number of PCs as an unbiased estimator of important components in PCA, KPCA, DPCA, and DKPCA and proved its efficiency against the traditional linear methods such as CPV, PA, and VRE through analytical systems. The application of the proposed FDKPCA was used as a generic nonlinear dynamical monitoring system for FDD on the industrial benchmark multiphase flow process. Through the implementation, it was shown that the proposed methodology has satisfactorily proven its capability in extracting the nonlinear relationships between time-lagged duplicates and furthermore its proficiency and efficacy in detecting the different types of faults. The real (manually injected) incipient faults were successfully detected at the earliest stage of occurrence, whereas, the remaining mentioned methods failed to detect these faults until the degree of fault severity gets considerably increased. Consequently, it provided the lowest performance indices, FAR, TD, MDR. To extend the monitoring process, the fault diagnosis stage was carried out using the nonlinear contribution plots. Guided by the previous fault diagnosis ivestigations, the severity of all the faults was diagnosed accordingly using the model obtained through FDKPCA.

## References

[1] J. Gertler, (1998), "Fault Detection and Diagnosis in Engineering Systems", 1st ed., Routledge, New York.

[2] J. Chen and R.J. Patton, (1999), "Robust Model-Based Fault Diagnosis for Dynamic Systems", 1st ed., Springer, US.

[3] S. Ding, (2013), "Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms and Tools", 2nd ed., Springer-Verlag, London.

[4] N. Olivier-Maget, et al., (2008), "Integration of a failure monitoring within a hybrid dynamic simulation environment", Chem. Eng. Process. 47 (1942-1952). DOI: https://doi.org/10.1016/j.cep.2007.12.009.

[5] G. Zhuang, Y. Li, and J. Lu, (2014) , "Fuzzy fault-detection filtering for uncertain stochastic time-delay systems with randomly missing data", T. I. Meas. Control. 37 (242-264), DOI: https://doi.org/10.1177/0142331214538088.

[6] A. A. Barresi, et al., (2009) "Monitoring of the primary drying of a lyophilization process in vials", Chem. Eng. Process. 48(408-423), DOI: https://doi.org/10.1016/j.cep.2008.05.004.

[7] J. Yu, (2019), "A selective deep stacked denoising autoencoders ensemble with negative correlation learning for gearbox fault diagnosis", Comput. Ind. 108 (62-72), DOI: https://doi.org/10.1016/j.compind.2019.02.015.

[8] W. Bounoua, A. B. Benkara, A. Kouadri, and A. Bakdi, (2019), "Online monitoring scheme using principal component analysis through Kullback-Leibler divergence analysis technique for fault detection", T. I. Meas. Control. 42(1225-1238), DOI: 10.1177/0142331219888370.

[9] C. Wu, P. Jiang, C. Ding, F. Feng, and T. Chen, (2019), "Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network", Comput. Ind. 108(53-61), DOI:

https://doi.org/10.1016/j.compind.2018.12.001.
[10] E.B. Martin and A.J. Morris, (1996), "An overview of multivariate statistical process control in continuous and batch process performance monitoring", T. I. Meas. Control. 18(51-60), DOI: https://doi.org/10.1177/014233129601800107.
[11] A. Bakdi, W. Bounoua, S. Mekhilef, and L. M. Halabi, (2019), "Nonparametric Kullback-divergence-PCA for intelligent mismatch detection and power quality monitoring in grid-connected rooftop PV", Energy. 189 (116366), DOI: https://doi.org/10.1016/j.energy.2019.116366.
[12] J.L. Godoy, J.R. Vega, and J.L. Marchetti, (2013), "A fault detection and diagnosis technique for multivariate processes using a PLS-decomposition of the measurement space", Chemom. Intell. Lab. Syst. 128(25-36), DOI: https://doi.org/10.1016/j.chemolab.2013.07.006
[13] J.E. Jackson, (1991), "A User's Guide to Principal Components". John Wiley & Sons, Inc.
[14] J. Storck and G. Deco, (1997), "Nonlinear independent component analysis and multivariate time series analysis", Physica D. 108(335-349), DOI: https://doi.org/10.1016/S0167-2789(97)00010-9.
[15] N. Zhang, et al., (2015), "Process fault detection based on dynamic kernel slow feature analysis", Comp. Electr. Eng. 41(9-17), DOI: https://doi.org/10.1016/j.compeleceng.2014.11.003.
[16] Z. Geng and Q. Zhu, (2005), "Multiscale Nonlinear Principal Component Analysis (NLPCA) and Its Application for Chemical Process Monitoring", Ind. Eng. Chem. Res. 44(3585-3593), DOI: 10.1021/ie0493107
[17] B. Scholkopf and A.J. Smola, (2001), "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond", MIT Press.
[18] W. Ku, R.H. Storer, and C. Georgakis, (1995), "Disturbance detection and isolation by dynamic principal component analysis", Chemom. Intell. Lab. Syst. 30(179-196), DOI: https://doi.org/10.1016/0169-7439(95)00076-3.
[19] J. C. Jeng, (2010), "Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms", J. Taiwan Inst. Chem. E. 41(475-481), DOI: https://doi.org/10.1016/j.jtice.2010.03.015.
[20] S. J. Zhao, et al., (2004), "*A multiple PCA model based technique for the monitoring of processes with multiple operating modes"* In: Computer Aided Chemical Engineering. A. Barbosa-Póvoa and H. Matos (Eds.), Elsevier, (865-870). DOI: https://doi.org/10.1016/S1570-7946(04)80210-4.
[21] W. R. Zwick and W. F. Velicer, (1986), "Comparison of five rules for determining the number of components to retain", Psychol. Bull. 99(432-442), DOI: https://doi.org/10.1037/0033-2909.99.3.432.
[22] H. F. Kaiser, (1960), "The Application of Electronic Computers to Factor Analysis". Educ. Psychol. Meas. 20(141-151), DOI: https://doi.org/10.1177/001316446002000116.
[23] Malinowski, E. R., (1991) "*Factor Analysis in Chemistry"*. New York: Wiley Interscience.
[24] J.L. Horn, (1965), "A rationale and test for the number of factors in factor analysis", Psychometrika. 30(179–185), DOI: https://doi.org/10.1007/BF02289447.
[25] R. B. Cattell, (1966), "The Scree Test for The Number of Factors", Multivariate Behav. Res. 1(245-276). DOI: https://doi.org/10.1207/s15327906mbr0102_10.
[26] S. Wold, (1978), "Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models", Technometrics. 20(397-405). DOI: 10.2307/1267639
[27] S. J. Qin and R. Dunia, (2000), "Determining the number of principal components for best reconstruction", J. Process Contr. 10(245-250), DOI: https://doi.org/10.1016/S0959-1524(99)00043-8.
[28] S. Valle, et al., (1999), "Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods". Ind. Eng. Chem. Res. 38(4389-4401). DOI: https://doi.org/10.1021/ie990110i.
[29] H. Yu, F. Khan, and V. Garaniya, (2016), "A sparse PCA for nonlinear fault diagnosis and robust feature discovery of industrial processes", AIChE Journal. 62 (1494-1513), DOI: 10.1002/aic.15136.
[30] M. T. Amin, S. Imtiaz, and F. Khan, (2018) , "Process system fault detection and diagnosis using a hybrid technique", Chem. Eng. Sci. 189 (191-211), DOI: https://doi.org/10.1016/j.ces.2018.05.045.
[31] H. Yu, F. Khan, and V. Garaniya, (2015), "Risk-based fault detection using Self-Organizing Map", Reliab. Eng. Syst. Safe. 139(82-96), DOI: https://doi.org/10.1016/j.ress.2015.02.011.
[32] M. T. Amin, F. Khan, and S. Imtiaz, (2019), "Fault detection and pathway analysis using a dynamic Bayesian network", Chem. Eng. Sci.. vol. 195(777-790), DOI: https://doi.org/10.1016/j.ces.2018.10.024.
[33] C. Aldrich and L. Auret, , 2013, "Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods", first ed., Springer-Verlag, London.
[34] T.J. Rato and M.S. Reis, (2013), "Defining the structure of DPCA models and its impact on process monitoring and prediction activities", Chemom. Intell. Lab. Syst. 125(74-86), DOI: https://doi.org/10.1016/j.chemolab.2013.03.009
[35] E. Vanhatalo, M. Kulahci, and B. Bergquist, (2017), "On the structure of dynamic principal component analysis used in statistical process monitoring", Chemom. Intell. Lab. Syst. 167(1-11), DOI: https://doi.org/10.1016/j.chemolab.2017.05.016.
[36] S.W. Choi and I.-B. Lee, (2004), "Nonlinear dynamic process monitoring based on dynamic kernel

PCA", Chem. Eng. Sci. 59(5897-5908), DOI: https://doi.org/10.1016/j.ces.2004.07.019.
[37] M. Jia, et al., (2010) , "On-line batch process monitoring using batch dynamic kernel principal component analysis" Chemom. Intell. Lab. Syst. 101(110-122), DOI: https://doi.org/10.1016/j.chemolab.2010.02.004.
[38] S. Z. Gao, et al., (2016), "Fault Diagnosis Method on Polyvinyl Chloride Polymerization Process Based on Dynamic Kernel Principal Component and Fisher Discriminant Analysis Method", Math. Probl. Eng. 2016, DOI: http://dx.doi.org/10.1155/2016/7263285.
[39] B.B. Mandelbrot, (1977), "Fractals, Form, Chance and Dimension", 1st ed., W.H.Freeman & Company, San Fransisco.
[40] K. Fukunaga, (1982), "*15 Intrinsic dimensionality extraction"*, in Handbook of Statistics, Elsevier. p. 347-360.
[41] P. Grassberger and I. Procaccia, (1983), "Measuring the strangeness of strange attractors", Physica D. 9(189-208), DOI: https://doi.org/10.1016/0167-2789(83)90298-1.
[42] S.K. Nayak, et al., (2018), "A Review on the Nonlinear Dynamical System Analysis of Electrocardiogram Signal". J. Healthc. Eng. 6920420. 19, DOI: https://doi.org/10.1155/2018/6920420.
[43] F. Camastra and A. Staiano, (2016), "Intrinsic dimension estimation: Advances and open problems", Inform. Sciences. 328(26-41), DOI: https://doi.org/10.1016/j.ins.2015.08.029.
[44] J. Feder, Fractals, first ed., Springer, US, 1988.
[45] A. Gusso, et al., (2019), "Nonlinear dynamics and chaos in micro/nanoelectromechanical beam resonators actuated by two-sided electrodes". Chaos Soliton. Fract. 122(6-16), DOI: https://doi.org/10.1016/j.chaos.2019.03.004.
[46] C. Grebogi, E. Ott, and J.A. Yorke, (1987), "Chaos, Strange Attractors, and Fractal Basin Boundaries in Nonlinear Dynamics", Science. 238(632), DOI: https://doi.org/10.1126/science.238.4827.632.
[47] B. Mandelbrot, (1967), "How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension", Science. 156(636), DOI: https://doi.org/10.1126/science.156.3775.636.
[48] D. Mo and S.H. Huang, (2012), "Fractal-Based Intrinsic Dimension Estimation and Its Application in Dimensionality Reduction", IEEE Trans. Knowl. and Data Eng. 24(59-71), DOI: 10.1109/TKDE.2010.225.
[49] C. Jr, et al., ( 2001), "Fast Feature Selection Using Fractal Dimension", in Proceedings of the XV Brazilian Symposium on Databases (SBBD).
[50] X. B. Yang, et al., (2011), "A novel model-based fault detection method for temperature sensor using fractal correlation dimension". Build. Environ. 46(970-979), DOI: https://doi.org/10.1016/j.buildenv.2010.10.030.
[51] R. Ma and Y. Chen, (2012), "Research on the dynamic mechanism of the gear system with local crack and spalling failure", Eng. Fail. Anal. 26 (12-20), DOI: https://doi.org/10.1016/j.engfailanal.2012.05.022.
[52] J.P. Amezquita-Sanchez, et al., (2016), "Fractal dimension-based approach for detection of multiple combined faults on induction motors". J. Vib. Control. 22(3638-3648). DOI: 10.1177/1077546314565685.
[53] A. Ziaja, T. Barszcz, and W. Staszewski, (2012), "*Fractal Based Signal Processing for Fault Detection in Ball-Bearings"*. In: Fakhfakh T., Bartelmus W., Chaari F., Zimroz R., Haddar M. (eds), *Condition Monitoring of Machinery in Non-Stationary Operations*. Springer, Berlin, Heidelberg , DOI: https://doi.org/10.1007/978-3-642-28768-8_41.
[54] N. Upadhyay and P.K. Kankar, (2018), "Diagnosis of bearing defects using tunable Q-wavelet transform". J. Mech. Sci. Technol. 32(549–558), DOI: https://doi.org/10.1007/s12206-018-0102-8.
[55] P. Grassberger and I. Procaccia, "Characterization of Strange Attractors". Phys. Rev. Lett. 50(1983), 346-349, DOI: https://doi.org/10.1103/PhysRevLett.50.346.
[56] N. O'Rourke, et al. , (2008), "A Step-by-Step Approach to Using SAS for Univariate & Multivariate Statistics", SAS Institute.
[57] G. L. Baker and J. P. Gollub . (1996 ), "Chaotic Dynamics: An Introduction". Cambridge, Cambridge University Press.
[58] J.-M. Lee, et al., (2004), "Nonlinear process monitoring using kernel principal component analysis". Chem. Eng. Sci. 59(223-234), DOI: https://doi.org/10.1016/j.ces.2003.09.012.
[59] J.A. Westerhuis, S.P. Gurden, and A.K. Smilde, (2000), "Generalized contribution plots in multivariate statistical process monitoring". Chemom. Intell. Lab. Syst. 51(95-114), DOI: https://doi.org/10.1016/S0169-7439(00)00062-9.
[60] R. Vitale, O.E. de Noord, and A. Ferrer, (2015), "Pseudo-sample based contribution plots: innovative tools for fault diagnosis in kernel-based batch process monitoring". Chemom. Intell. Lab. Syst. 149(40-52), DOI: https://doi.org/10.1016/j.chemolab.2015.09.013.
[61] A.K. Conlin, E.B. Martin, and A.J. Morris, (2000), "Confidence limits for contribution plots" J. Chemometr. 14(725-736), DOI: https://doi.org/10.1002/1099-128X(200009/12)14:5/6<725::AID-CEM611>3.0.CO;2-8.
[62] X. Deng, X. Tian, and S. Chen, (2013), "Modified kernel principal component analysis based on local

structure analysis and its application to nonlinear process fault diagnosis". Chemom. Intell. Lab. Syst. 127(195-209), DOI: https://doi.org/10.1016/j.chemolab.2013.07.001.
[63] A. Stief, R. Tan, Y. Cao, and J. R. Ottewill, (2018), "Analytics of Heterogeneous Process Data: Multiphase Flow Facility Case Study", IFAC-PapersOnLine. 51 (363-368), DOI: https://doi.org/10.1016/j.ifacol.2018.09.327.
[64] M. Lucke, A. Stief, M. Chioua, J. R. Ottewill, and N. F. Thornhill, (2020), "Fault detection and identification combining process measurements and statistical alarms", Control Eng. Pract. 94 (104195), DOI: https://doi.org/10.1016/j.conengprac.2019.104195.
[65] A. Stief, R. Tan, Y. Cao, J. R. Ottewill, N. F. Thornhill, and J. Baranowski, (2019), "A heterogeneous benchmark dataset for data analytics: Multiphase flow facility case study", J. Process Contr. 79 (41-55), DOI: https://doi.org/10.1016/j.jprocont.2019.04.009.