People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Power and Control

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

MASTER

In **Electrical and Electronic Engineering**

Option: **Control**

Title:

PCA-Based Approach for Fault Detection In Cement Rotary Kiln.

Presented by:

- **ARIBI Yacine. (Control)**
- **GUERMI Hamza. (Power Engineering)**

Supervisor:

Dr. KOUADRI Abdelmalek.

Registration Number:……/2016

To our Families, Friends and Teachers.

Acknowledgment

At the end of this work, we want to express our deep gratitude and appreciation to our supervisor, Dr. KOUADRI Abdelmalek, for the valuable suggestions, support and guidance. We are sincerely grateful to all the staff of the Institute of Electrical and Electronics Engineering of the University of M'Hamed Bougara- Boumerdes, Teachers and workers, especially the Power and Control Department members, for the great help and support during the last five years. A special thanks to all of our teachers, we have learned a lot from your knowledge and moralities.

Abstract

Principal component analysis (PCA) is a well-known data dimensionality technique that is widely used in industrial processes detection fault. Dynamic PCA is acknowledged for its capability to cope with autocorrelation in time-series. False indications impose one of the greatest problems in the monitoring of many processes. A study is carried in the first part of this work in order to compare the performances of Static and Dynamic PCA approaches in cement rotary kiln. The issue of false indications in fault detection systems is investigated in the second part. A monitoring approach based on constant false alarms rate (CFAR) is proposed to reduce false detections. A piecewise constant threshold is developed for T^2 and Q-statistics to limit the rate of false alarms to a given percentile at each time instant. A control chart constructed using the rate of false alarms per window is used to monitor the process. Finally, the proposed monitoring technique is tested on SPCA to confirm the ability of the proposed approach to result in zero false detections without imposing high delays or misdetections.

Key words: principal component analysis (PCA), dynamic PCA, Time-series, Static PCA, constant rate of false alarms (CFAR).

Table of contents

List of Tables

List of Figures

Nomenclature

Abbreviations:

ACF	:	Autocorrelation Function.
$ARIMA$	:	Autoregressive Integrated Moving Average.
$ARMA$	:	Autoregressive Moving Average.
$ASPC$	:	Average Squared Partial Correlation.
BS	:	Brocken Stick Method.
CC	:	Control Chart.
$CFAR$	:	Constant False Alarms Rate.
CI	:	Confidence Interval.
$CPBD$	:	Contribution Plots Based Diagnosis.
CPV	:	Cumulative Percent of Variance.
CL	:	Central Line.
CV	:	Cross-Validation.
DID	:	Delay in Detection.
$DPCA$	:	Dynamic Principal Component Analysis.
FA	:	Factor Analysis.
FAR	:	False Alarms Rate.
$FARW$	:	False Alarms Rate per Window.
EOF	:	Empirical Orthogonal Functions.
FDA	:	Fisher Discrimination Analysis.
FDI	:	Fault Detection and Isolation.
FDD	:	Fault Detection and Diagnosis.
GFC	:	Goodness-of-Fit Criteria.
GFP	:	Goodness-of-Fit Parameter.
$GPCA$	:	Global Principal Component Analysis.
ICA	:	Independent Component Analysis.
$J7$	:	Kaiser Greater-than-0.7 Criteria.
$K1$	:	Kaiser Greater-than-one Criteria.
$KFCV$	:	K-Folds Cross-Validation.
KSS	:	Karlis-Saporta-Spinaki rule.
KSV	:	Key Singular Value.
$KSVR$	:	Key Singular Value Ratio.
LCL	:	Lower Control Limit.
LEV	:	Log-Eigenvalues Test.
$LOOCV$	:	Leave-One-Out Cross-Validation.
$LPCA$	:	Local Principal Component Analysis.
MAP	:	Minimum Average Partial.
$MCDS$	:	Model-Construction Data set
MSE	:	Mean Squared Error.
$MWPCA$	:	Moving-Window Principal Component Analysis.
$NIPALS$	:	Nonlinear Iterative Partial Least Squares.
PA	:	Parallel Analysis.
PCA	:	Principal Component Analysis.
PDC	:	Poorly Determined Components.
PLS	:	Partial Least Squares.

$PRESS$	:	Prediction Residuals Sum of Squares.
$PWCT$	:	Piece-Wise Constant Threshold.
QTA	:	Qualitative Trend Analysis.
$ROBPCA$	:	Robust Principal Component Analysis.
$RPCA$	:	Recursive Principal Component Analysis.
SDG	:	Signed Directed Graph.
SNR	:	Signal-to-Noise Ratio.
SPC	:	Statistical Process Control.
$SPCA$	:	Static Principal Component Analysis.
SPM	:	Statistical Process Monitoring.
SVD	:	Singular Value Decomposition.
SVM	:	Support Vector Machines.
UCL	:	Upper Control Limit.
WER	:	Western Electric Rules.

Symbols:

Δt_f	:	Time between the appearance and the detection of the fault.
Δt_{max}	:	Maximum delay in detection
μ_u	:	The expected value (mean) of the random variable u.
σ_u	:	The standard deviation of the random variable u.
σ_{uv}	:	The covariance of the random variables u and v.
ρ_{uv}	:	The correlation of the random variables u and v.
$\sigma_Y \in \mathbb{R}^{m\times m}$	:	The covariance matrix of the matrix $Y \in \mathbb{R}^{n\times m}$, also written $Cov(Y)$.
$\rho_Y \in \mathbb{R}^{m\times m}$	:	The correlation matrix of the matrix $Y \in \mathbb{R}^{n\times m}$, also written $Corr(Y)$.
Ξ, Σ	:	The singular values matrix.
ξ_{ij}	:	The j^{th} singular value from the i^{th} experiment in a randomly generated data set.
η	:	Forgetting factor.
γ	:	FAR upper control limit.
λ_i	:	The i^{th} Eigenvalue of a given matrix.
$\chi_\alpha(\beta)$	:	The value of the Chi-square distribution with β degrees of freedom, at $1-\alpha$ confidence level.
$\|.\|_2$	:	The Euclidean norm.
a	:	Number of retained components by PCA model.
c	:	Failure Class.
d	:	Decision Variable.
$diag(A)$	:	The diagonal elements of the matrix A, or a diagonal matrix with the vector A forms its diagonal and the off-diagonal values are zero.
E	:	Error Matrix.
$F_\alpha(\beta,\gamma)$	:	The value of the Fisher-distribution with β and γ degrees of freedom, at $1-\alpha$ confidence level.
I_m	:	The $m \times m$ identity matrix.
l	:	Number of lags in DPCA.
$\mathcal{L}_k$	:	The length of the k^{th} stick in a BS model.
m	:	Number of measured properties.
n	:	Number of observations.
N	:	Noise.
$P \in \mathbb{R}^{m\times m}$	:	Loading matrix, Orthogonal Projection Matrix.

$P_a \in \mathbb{R}^{m \times a}$	:	Principal Loadings Matrix.
Q	:	Q-statistic (Squared Prediction Error).
r	:	Residuals Vector.
s_i	:	The i^{th} singular value.
$T \in \mathbb{R}^{n \times m}$	:	Matrix of Scores.
$T_a \in \mathbb{R}^{n \times a}$	:	Matrix of principal scores, also written $\hat{T}$.
T^2	:	Hotelling's T-square Statistic.
t_a	:	Appearance time of the fault.
t_d	:	Detection time of the fault.
$U(t)$	:	Input signal.
$X \in \mathbb{R}^{n \times m}$	:	Original Data Matrix.
$Y(t)$	:	Output signal.
Z_{ij}	:	The element in the i^{th} row and the j^{th} column of the matrix Z.
Z^T	:	The transpose of the rectangular matrix Z.
Z^{-1}	:	The inverse of the invertible, square-matrix Z.
Z_0	:	Matrix with zero-mean columns.
Z_n	:	Standardized matrix, i.e., zero-mean and unity standard deviation columns.
z_α	:	The value of the standardized normal distribution at $1 - \alpha$ confidence level.

Introduction

The increasing need for high performance, efficient, safe, and reliable operations within different industrial applications results in a large growing in the complexity of the systems. With a complex processes overtaking industries, it becomes more difficult and more critical to detect the presence of abnormal operation before complete failures occur. Early fault detection and quick and accurate diagnosis was acknowledged for its ability to prevent catastrophic damages, avoid defects, and improve the quality. The ancestor of fault detection and diagnosis, statistical process control (SPC), dates back to the early 1920s works of Dr. Walter. A. Shewhart of the Bell telephone laboratories. In 1970s, with the advent of the computer and its increasing application in decentralized process automation systems since 1975 was the beginning of computationally more involved and soft-based fault detection algorithms [1]. In particular, Japan had the first successful application of SPC methods. In 1970s, American industry suffered extensively from the Japanese competition; that has led, in turn, to renewed interest in SPC methods. Since then, many U.S. companies have begun extensive programs to develop and implement these methods in their manufacturing, engineering, and other business organizations. Over the last two decades, Fault Detection and Diagnosis (FDD) algorithms and their applications to a wide range of industrial processes have been the subject of intensive research over the past two decades [2,3,4]. Current FDD methods can be classified into model-based and data driven-based methods. The use of neural networks, artificial intelligence, and statistical methods is widely common as data-driven approaches for fault detection.

Due to the complexity of industrial processes, a large trend to the application of data driven methods can be observed from the amount of research carried in that particular field [5]. Furthermore, within data driven approaches, principal component analysis (PCA) and partial least squares (PLS) are the two standard methods in data driven fault detection and diagnosis. The use of statistical methods to construct control charts usually leads to a considerable amount of false detections. This last problem rises as a natural consequence of the presence of background noise and the statistical uncertainties. Many efforts have been spent in the field in order to increase the sensitivity so that small deviations can be detected, ameliorate the detection time, decrease the delay in deciding about the presence of the fault, and decreasing the rate of false alarms in the control chart.

The aim of this work is to study the ability of PCA-based SPM to detect the presence of faults in complex industrial system, where a cement rotary kiln is taken as an application. Due to the presence of outliers in statistical process control charts, the final objective of our work is to develop a monitoring scheme to eliminate the existence of false detections without deteriorating the detection time and the sensitivity to the different deviations.

This report is organized as follows, first, the field of fault detection and diagnosis is introduced along with its terminology and the different methods existing are listed to familiarize ourselves with the notations and the methods. The aim of the first chapter is to make the choice of data driven methods and particularly PCA clearer. In the second chapter, the mathematical foundation of PCA as a dimensionality reduction technique is explored and the application of PCA as a multivariate statistical process monitoring technique for complex processes is demonstrated. In the third chapter, principle component analysis is studied through applying SPCA to a multivariate data set of a cement rotary kiln. Along with the different features, the

sensitivity and the detectability of PCA-based fault detection is explored. Finally, the last chapter is dedicated to the development and application of a CFAR-based monitoring scheme. The proposed monitoring method is carried on the fault indicators of PCA in order to eliminate the problem of false detections.

Chapter 1:

Fault Detection and Diagnosis

1.1 Introduction

Fault detection and diagnosis is an extremely important task in any engineering and industrial field. Its importance emerges from the fact that early detecting a fault while the system is still operating avoids abnormal events and losses. It gives an opportunity and time to avoid catastrophic damages in the system, it can save human's life and reduce the economical losses. There are several ways to detect faults and they can be divided into two classes: *model based* and *date driven*. For the model based; one has to extract a mathematical modal to describe the process where data driven is based on the availability of the amount data for many industrial process. The basic aim of this chapter is getting general and clear idea about faults and their classifications, also to get enough knowledge and information in order to be able to select and use the best fault detection and diagnosis approach or model.

1.2 Terminology

First of all, it is highly important to be more familiar with the terminology of the field of fault detection and diagnosis, thus making it easy to understand and to compare the different approaches and methods which are used in this field.

1. *Fault*: A fault is a *state* within the system. It is an unpermitted deviation of at least one characteristic property (*feature*) of the system from the acceptable, usual, standard condition. Where the unpermitted deviation is the difference between the fault value and the violated threshold of a *tolerance zone* for its usual value [1]. It can be also defined as a departure from an acceptable range of an observed variable or a calculated parameter associated with a process [6].
2. *Failure:* A failure is an *event*. It is defined as a permanent *interruption* of a system's ability to perform a required function under specified operating conditions [1]. There are different types of failure and they can be classified according to:
    - number of failures: single, multiple.
    - predictability: random failure (unpredictable, e.g. statistically independent from operation time or other failures), deterministic failure (predictable for certain conditions), systematic failure or causal failure (dependent on known conditions).
3. *Malfunction:* A malfunction is an *event*. it is defined as an *intermittent irregularity* in the fulfillment of a system's desired function [1]. It arises after the start of the operation or by increasingly stressing the system. Both failures and malfunctions are events caused by faults. In other words, we can say that a fault appears as failure or a malfunction.
4. *Error:* It is a deviation between a measured or computed value of an output variable and its true or theoretically correct one.
5. *Disturbance:* It is an unknown and uncontrolled input acting on a system.
6. *Residual:* A fault indicator, based on deviation between measurements and model equation based computations.

7. *Symptom:* A change of an observable quantity from normal behavior [7].
8. *Fault detection:* It is the process of finding if there is any fault in the system and also the time of that fault [8]. Fault detection amounts to determining whether the supervised process is working in a normal (or *healthy*) operating mode. It can be given by a model of the process (model based) or a sequence of measured process input and output (data driven based). The quality of a fault detection system is measured in terms of *detection delay* and time between *false alarms*. A typical objective is to minimize the mean delay for detection of a change subject to a fixed false alarm rate before the change time [9].
9. *Fault isolation:* It is the determination of the kind, location and time of detection of fault follows fault isolation.
10. *Fault identification:* It is the determination of the process variables that are responsible for the fault and thus the information can be used to diagnose the fault. It helps to focus on the subsystems where the fault occurred in a large plant [7].
11. *Fault diagnosis:* The whole process of determining the kind, size, location of the fault including fault isolation and identification is called as fault diagnosis. Based on the observed analytical and heuristic symptoms and the heuristic knowledge of the process, the diagnostic procedure is outlined. The knowledge from physical laws or quantitative measurements and observations is analytical diagnostic knowledge. The knowledge that is not clearly written or described, but is obtained as a result of learning through experimental methods is heuristic diagnostic knowledge [6,7].
12. *Monitoring:* The continuous real task of determining the conditions of a physical system by recording information, recognizing and indication anomalies in the behavior is the process of monitoring.
13. *Supervision:* It is the processes of monitoring and taking appropriate actions to maintain the operation in the case of fault [7].
14. *Reliability:* Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time.
15. *Safety:* Ability of a system not to cause danger to persons or equipment or the environment.
16. *Availability:* Probability that a system or equipment will operate satisfactorily and effectively at any period of time [1].
17. *Dependability:* The term dependability seems not to be clearly defined. Therefore, different meanings are cited:
    - A form of availability that has the property of always being available when required (and not at any time). It is the degree to which a system is operable and capable of performing its required function at any randomly chosen time during its specific operating time, provided that the system is available at the start of the period. This definition excludes non-operation related influences.
    - Dependability is a property of a system that justifies placing one's reliance on it. It covers reliability, availability, safety, maintainability and other issues of importance in critical systems.
18. *Integrity:* The integrity of a system is the ability to detect faults in its own operation and to inform a human operator [1].

1.3 Fault classification

A suitable modelling of faults is important for the right functioning of fault-detection methods. A realistic approach presupposes the understanding between the real physical faults and their effect on the mathematical process models. This can usually only be provided by the inspection of the considered real process, the understanding of the physics and a fault-symptom-tree analysis. There are many reasons for the appearance of faults. They stem for examples from:

1. wrong design, wrong assembling.
2. wrong operation, missing maintenance.
3. ageing, corrosion, wear during normal operation.

With regard to the operation phase they may be already present or they may appear suddenly with a small or large size or in steps or gradually like a drift; they can be considered as deterministic faults. Especially disadvantageous are usually *intermittent* faults which appear as stochastic faults [1]. Fault can be classified into:

1. *Additive process faults:* These are known inputs acting on the plants which are normally zero and when present can cause a change in the plant output independent of the known inputs [10]. It can be described mathematically as:

$$Y(t) = Y_u(t) + f(t) \tag{1.1}$$

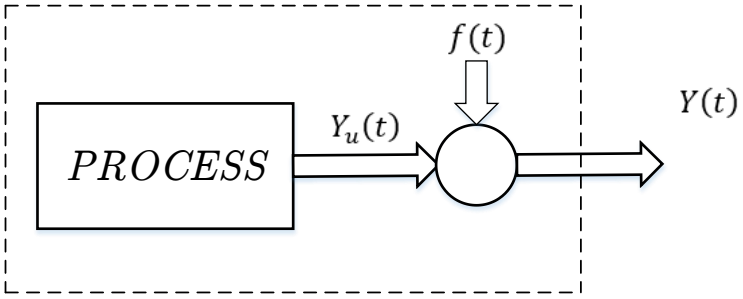

Figure 1. 1. Additive fault model for an output signal.

In the case of the additive fault, the detectable change $\Delta Y(t)$ of the variable is independent from any other signal and depends only on the fault, i.e.,

$$\Delta Y(t) = f(t) \tag{1.2}$$

2. *Multiplicative process faults:* They are changes in some plant parameters. They can change the plant outputs which depend on the magnitude of the known inputs. Those faults describe the damages and malfunction of the plant equipment. It is described mathematically as follow:

$$Y(t) = \big(\alpha + \Delta\alpha(t)\big)U(t) \tag{1.3}$$

And this can be seen as:

$$Y(t) = Y_u(t) + f(t).U(t) \tag{1.4}$$

For the multiplicative fault, the detectable change of the output $\Delta Y(t)$ depends on the input signal $U(t)$.

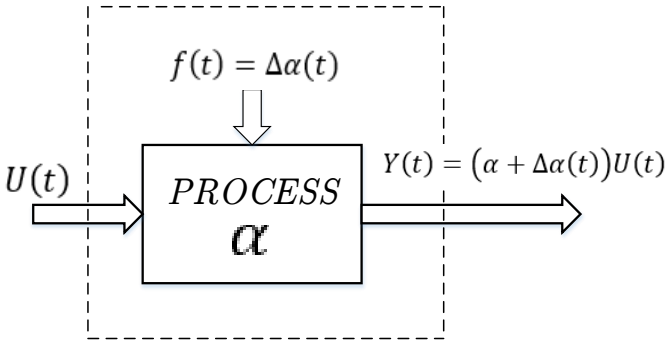

Figure 1. 2. Multiplicative process faults for an output signal.

Also faults that occur in process plant can be classified into three types:

1. *Sensor faults:* They represent incorrect reading from the sensors this can be due to broken wires, lost contact with the surface etc. In which case the reading shown by the sensor is not related to the value of the measured physical parameter. This can for instance; be a gain reduction, a biased measurement or increased noise the following figure is shown the sensor faults [10,8].
2. *Actuator faults:* They represent partial or complete loss of control action. Total actuator fault can occur; for instance; as a result of a breakage cut or burned wiring short cuts; or the presence of outer body in the actuator. In case of partially failed actuator only part of the normal actuation is produced; it can be result of hydraulic pneumatic leakage reduced input voltage or increased resistance.
3. *Component faults:* These faults are those faults that we are not able to considered them as sensor faults or actuator faults. They occur due to structural damages of the components. The dynamical behavior of the system can be changed because of these faults [1].

And these are the most frequently encountered types in fault family to deal with.

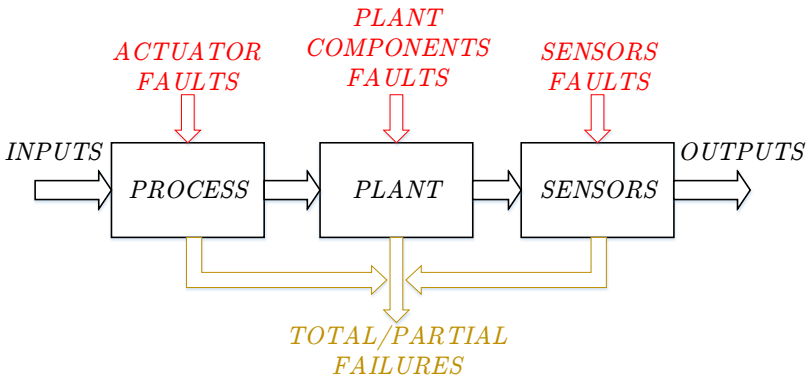

Figure 1. 3. Fault models based on the faulty components.

According to the time dependency, faults can be classified into:

1. *Abrupt faults:* They occur instantaneously often as a result of hardware problems (failures).
2. *Incipient faults:* They represent slow in time parametric changes often as a result of aging they are more difficult to detect.
3. *Intermittent faults:* They are faults that appear and disappear repeatedly for instance; due to a partially damaged wiring.

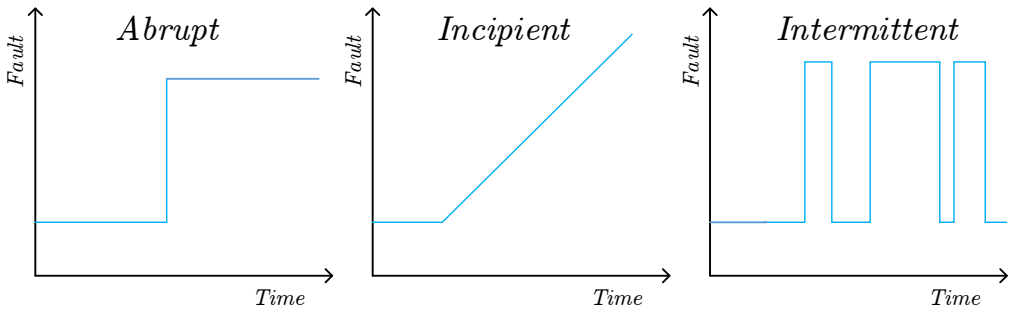

Figure 1. 4. Time Characteristics of Faults.

1.4 Desirable characteristics of a fault diagnostic system

It is useful to a fault detection and diagnosis system to have desired set of characteristics to be acknowledged as efficient methodology. These set of traits are known as desirable characteristics. They are used in order to compare the different diagnostic approaches. It though there are several characteristics that are considered in FDD, these desired characteristics are:

1. *Quick detection and diagnosis:* Typically, it is desirable to reduce the delay between the starting time of the fault and its appearance time in the process monitoring. Furthermore, the delay between the appearance time and the detection time of the fault due to the decision rule has to be minimized [9]. The FDD system should respond as fast as possible to detect a failure or a malfunction and must be quick in identifying the roots of the abnormal process state.
2. *Isolability:* It refers to the ability of the diagnostic system to distinguish between different failures [6]. Isolability of a fault depends on the way the fault affects system outputs. Various sources of uncertainties (modeling uncertainty, errors and system disturbances) pose a difficulty to achieve a good isolability.
3. *Robustness:* It is the ability of the system to resist the environmental conditions like dust or humidity. A robustness according to measurements noise, system disturbances, and modeling uncertainties is a desirable attribute of a diagnostic system intended for practical implementations.
4. *Novelty identifiability:* The ability of the system to distinguish normal functionality from abnormal functionality especially when an unknown malfunction is present (novel abnormality).
5. *Adaptability:* In general processes operation changes according to the change of the operating conditions which are changed due disturbances and environmental conditions (including changes in production quantities). Thus, the diagnostic system should be adaptable to changes.
6. *Classification error estimate:* error measures would be useful to project confidence and reliability on the diagnostic decisions made by the system. This characteristic makes the diagnostic system more recommended.
7. *Explanation facility:* It is an important factor in designing of FDD system to provide explanation on how the fault originated and propagated to the current situation, which requires the ability to reason about cause and effect relationships in a process.
8. *Modelling requirements:* for fast and easy deployment of real time diagnostic classifiers, the modelling effort should be as minimal as possible.

9. *Storage and computational requirements:* Quick and real time solution require algorithms and implementation which are computationally less complex, but might entail high storage requirements. Often, an FDD system with the ability to balance on these requirement is highly needed [6].
10. *Multiple fault identifiability:* This refers to the ability of a diagnostic system to identify and correctly classify multiple faults that may coexist in the process [9]. Identifying multiple faults is important and difficult task since in nonlinear systems the interactions would be synergistic and FDD can model the combined effect of the faults.

1.5 Transformations of measurements in diagnostic systems

It is essential to identify the different transformations that process measurements go through before the final diagnostic decision is made. The most essential *spaces* where the diagnostic process can be at any given time are:

1. *Measurement space:* This is considered as the inputs of FDD system. They can be represented as $x_1, x_2, \dots, x_m$ where "m" refers to the number of variables (measurements) with no *priori* problem knowledge relating this variables (measurements).
2. *Feature space:* in this space the measurements are analyzed and combined with the help of a prior process knowledge to extract useful features about the process behavior to help diagnosis where the feature extraction is the process of understanding the relationship between the variable in the measurement space using prior knowledge. The space can be seen as $y = (y_1, y_2, \dots, y_i)$ where y_i is the feature obtained as function of the measurements by using a priori problem knowledge.
3. *Decision space:* this space is obtained by subjecting the feature space to meet on objective function which could be some kind of discriminant or simple threshold function. This space can be expressed as space of point $d = (d_1, d_2, \dots, d_e)$ where e is the number of decision variables.
4. *Class space:* is a set of integers $c = (c_1, c_2, \dots, c_j)$ where j is the number of failure classes and normal class of data to any of which a given measurements pattern may belong [6,7].

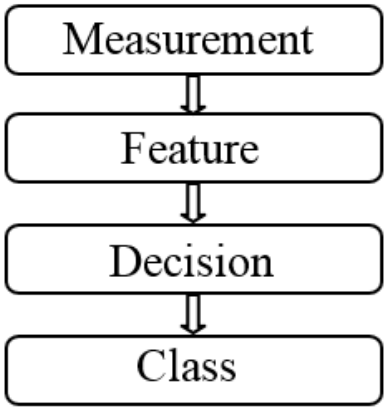

Figure 1. 5. Transformations in a diagnostic system.

1.6 Classification of fault diagnosis methods

FDD methods can be classified into *model-based* methods and *data-driven* methods. They use simulation models and measurement data. We can distinguish between them by the knowledge used to diagnose the cause of faults. The main components in a diagnosis classifier are the type of knowledge and the type of diagnostic search strategy. Diagnostic search strategy is usually a very strong function of the knowledge representation scheme which in turn is largely influenced by the kind of a priori knowledge available. Hence, the type of a priori knowledge used is the most important distinguishing feature in diagnostic systems [6].

The basic priori knowledge that is needed for fault diagnosis is the set of failures and the relationship between the observations (symptoms) and the failures. A diagnostic system may have them explicitly, or it may be inferred from some source of *domain knowledge*. The priori domain knowledge may be developed from a fundamental understanding of the process using *first-principles knowledge*. Such knowledge is referred to as *deep*, *causal* or *model-based* knowledge. On the other hand, it may be gleaned from past experience with the process, this knowledge is referred to as *shallow, compiled, evidential* or *process history-based* knowledge. The model-based methods use priori-knowledge to identify the differences between model simulation results and actual operation measurement. They are divided into qualitative and quantitative modeling methods. The models are developed based on some physical knowledge related on the process which is necessary to be understood. In quantitative models this understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of the system (like transfer function). The qualitative models use rule-based methods developed based on priori-knowledge. Qualitative models use the qualitative rule relationships to detect and diagnose faults instead of quantitative mathematical equations. The rules are derived from expert knowledge, process history data and quantitative models simulation data. Expert knowledge is normally summarized to a database in the form of *if-then statements*. Data-based models are divided into qualitative and quantitative data based. There are different ways in which data can be transformed and presented as a priori knowledge to a diagnostic system [10,11,14,16]. The following chart summarize the different methods which are used in FDD:

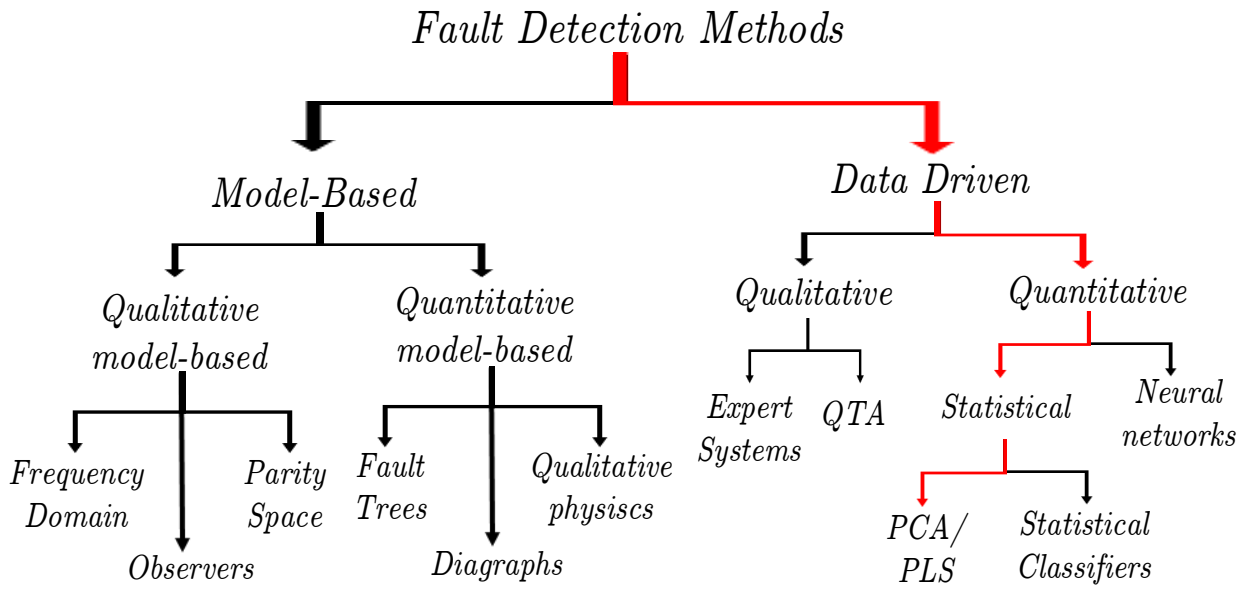

Figure 1. 6. Classification of Fault Detection Methods.

1.6.1 Model-based Fault Detection Methods

Model based methods are based on concept of *analytical redundancy*. The essence of this concept is the comparison of the actual outputs of the monitored system with the outputs obtained from a (redundant. i.e. not physical) analytical mathematical model. It involves two stages: residual generation and residual evaluation. This approach assumes that the structure and the parameters of the model are precisely known [11].

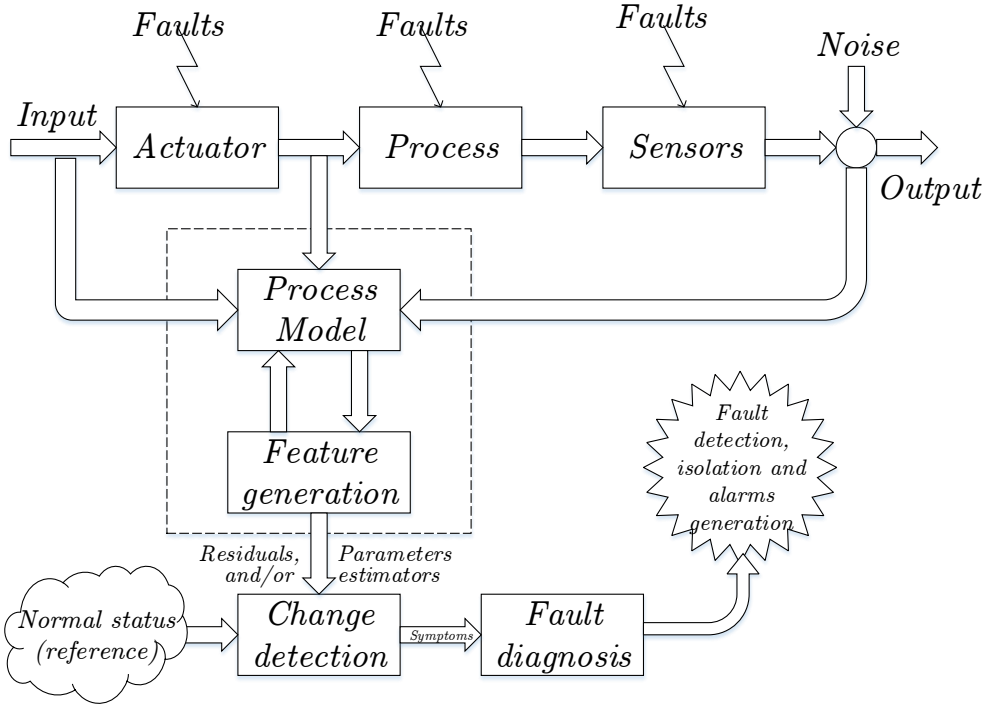

Figure 1. 7. General scheme of process model-based fault-detection and diagnosis.

1.6.1.1 Quantitative Model-based Methods

The quantitative model-based approaches have been based on using general input-output and state space models to generate residuals. These approaches can be classified into observers, parity space and frequency domain approaches.

1. *Observer or filter-based:* The idea of the observer or *filter-based* approaches is to estimate the states or outputs of the system from the measurements. The freedom in the design of the observer can be used to enhance the residuals for isolation. In recent years, several model-based methods have been developed and especially observer-based methods have been given more attention. Geometric approach and adaptive control are combined successfully with observer-based fault detection and diagnosis techniques [10,11,12].
2. *Parity space approaches:* This method compares the process behavior with a process model describing nominal, i.e., non-faulty behavior. The key idea is to check the parity (consistency) of the mathematical equations of the system (analytical redundancy relations) by using the actual measurements. The concept of this approach is to rearrange the model structure to get the best fault isolation. Redundancy provides freedom in the design of residual generating equations so that further fault isolation can be achieved. Fault isolation requires the ability to generate residual vectors which are *orthogonal* to

each other for different faults. Parity relations and the observer-based methods produce identical residuals if the generators are designed for the same specification [10,11,12].

3. *Frequency domain approaches:* Residuals are also generated in the frequency domains via factorization of the transfer function of the monitored system.

Fault detection and diagnosis based on quantitative models is often used in applications because of the fact that models are based on sound physical or engineering principles and they provide the most accurate estimators of output when they are well formulated. Moreover, a detailed models based on first principles can model both normal and faulty operations; therefore, faulty operation can be easily distinguished from normal operation without any excessive work. Furthermore, the transients in a dynamic system can only be modeled with detailed physical models. In the other side, Weaknesses of FDD based on quantitative models include: [10,13]

1. They can be complex and computationally intensive.
2. The effort required to develop a model is significant.
3. These models generally require many inputs to describe the system.
4. Extensive user input creates opportunities for poor judgment or input errors that can have significant impacts on results.

FDD based on detailed physical models is unlikely to emerge as the method of choice in the near future because of the weaknesses listed above, but simplified physical models will continue to make inroads into FDD applications [13].

1.6.1.2 Qualitative Model-Based Methods

Qualitative models enable conclusions to be reached about the state of a system with incomplete or uncertain knowledge of the physical process. They are based on various forms of qualitative knowledge used in fault diagnosis, qualitative model-based approaches can be classified into:

1. *Causal model approaches using digraphs:* A Signed Directed Graph or Signed Digraph (SDG), as a qualitative model, effectively and graphically represents a process system. Cause-effect relations or models can be represented in the form of signed digraphs. A digraph is a graph with *directed arcs* between the *nodes* and SDG is a graph in which the directed arcs have a positive or negative sign attached to them. The directed arcs lead from the cause nodes to the effect nodes. SDGs provide a very efficient way of representing qualitative models graphically and have been the most widely used form of causal knowledge for process fault diagnosis. SDG can be obtained from differential algebraic equations for the process. The issue of conditional arcs in SDG is addressed, also the idea of SDG was extended to include *five-range patterns* instead of the usual three-range pattern used in the standard SDG. Partial system dynamics, statistical information about equipment failure, and digraphs to represent the failure propagation network for identifying fault location are used. Rule-based methods using SDG have been used for fault diagnosis [10].
2. *Fault tree approach:* Fault trees are used in analyzing system *reliability* and *safety*. Fault tree approaches are top-down analysis technique that describes the relationship between basic events (failures, human errors, etc.), intermediate conditions (operating modes, environmental conditions, etc.) and top events (incidents, accidents). The relationship is

modeled in a tree-like structure with logical AND/OR gate. The numeric fault tree can then be analyzed for minimal cut sets (combination of basic events and conditions that cause the top event) and used for determining the probability of the top event.

3. *Qualitative physics approaches:* The detailed physical models are based on detailed knowledge of the physical relationships and characteristics of all components in a system. Using this detailed knowledge for mechanical systems, a set of detailed mathematical equations based on mass, momentum, and energy balances along with heat and mass transfer relations are developed and solved. Detailed models can simulate both normal and faulty operational states of the system. Qualitative physics knowledge in fault diagnosis has been represented in mainly two approaches, the first approach is to derive qualitative equations from the differential equations termed as *confluence equations*. The other approach in qualitative physics is the derivation of qualitative behavior from the *Ordinary Differential Equations* (ODEs). These qualitative behaviors for different failures can be used as a knowledge source [10].

The main benefits of the qualitative models for fault detection and diagnosis are listed here:

1. Well suited for data-rich environments and noncritical processes.
2. These methods are simple to develop and apply.
3. Their reasoning is transparent, and they provide the ability to reason even under uncertainty.
4. They possess the ability to provide explanations for the suggested diagnoses because the method relies on cause-effect relationships.
5. Some methods provide the ability to perform FDD without precise knowledge of the system and exact numerical values for inputs and parameters.

In the other side, this approach suffers from the fact that it is Process (or system) specific approach. Although these methods are easy to develop, it is difficult to ensure that all rules are always applicable and to find a complete set of rules, especially when the system is complex. Consequently, as new rules are added to extend the existing rules or accommodate special circumstances, the simplicity is lost. In addition, these models, to a large extent, depend on the expertise and knowledge of the developer in order to build a reasonable cause and effect chain.

Qualitative methods provide shortcuts and may offer the most expedient way to meet analytical needs where more rigorous approaches are time or cost prohibitive [13].

1.6.2 Data-Driven Fault Diagnosis Approaches

There are different ways in which data can be transformed and presented as a priori knowledge to a diagnostic system. This is known as feature extraction. This extraction process can be either qualitative or quantitative in nature. Two of the major methods that extract qualitative history information are the expert systems and trend modelling methods.

1.6.2.1 Qualitative Data-Driven Methods

There are two of important methods that employ qualitative feature extraction, which are the expert systems and trend modelling approaches.

1. *Expert system approaches:* An expert system is the system that solves problems in a narrow domain of expertise. The essential components in this system development are: knowledge acquisition, choice of knowledge representation, the coding of knowledge in a knowledge base, the development of inference procedures for diagnostic reasoning and the development of input/output interfaces. Its advantages in the development for diagnostic problem-solving are: ease of development, transparent reasoning, the ability to reason under uncertainty and the ability to provide explanations for the solutions provided [10,14].
2. *Qualitative trend analysis approaches:* Trend analysis and prediction are important components of process monitoring and supervisory control. Trend modelling can be used to explain the various important events happening in the process, do malfunction diagnosis and predict future states. From a procedural perspective, in order to obtain a signal trend not too susceptible to momentary variations due to noise, some kind of filtering needs to be employed. For example, time series representations assume, a priori, certain behavior as they are identified using a known process behavior. Alternatively, one may simply use a *filter* (such as an auto-regressive filter) with a priori chosen filter coefficients (specifying the required degree of smoothing). Both types of filters suffer from the fact that they cannot distinguish well between a transient and true instability [14].

1.6.2.2 Quantitative Data-Driven Methods

The quantitative approaches essentially formulate the diagnostic problem-solving as a *pattern recognition* problem. The goal of pattern recognition is the classification of data points to, in general, pre-determined classes. *Statistical* methods use knowledge of a priori class distributions to perform classification. Methods that extract quantitative information can be broadly classified as non-statistical or statistical methods. *Neural networks* are an important class of non-statistical classifiers. *Principal component analysis* (PCA), *partial least squares* (PLS) and statistical pattern classifiers form a major component of statistical feature extraction methods [10,14].

1. *Multivariate statistical approaches:* Multivariate statistical techniques are powerful tools capable of compressing data and reducing its *dimensionality* so that essential information is retained and easier to analyze than the original huge data set; and they are also able to handle noise and correlation to extract true information effectively [14]. The successful applications of multivariate statistical methods to fault diagnosis such as Principal Component Analysis (PCA) and Partial Least Squares (PLS) have been extensively reported in the literature. Principal component analysis (PCA) is one of the most intensively studied methods for fault detection of plants [15].

    - *Factor analysis* is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. The aim of factor analysis is to explain the outcome of m variables in the data matrix X using fewer variables, the so-called factors. Ideally all the information in X can be reproduced by a smaller number of factors.
    - PCA transforms a number of possibly correlated variables in a dataset into a smaller number of *uncorrelated pseudo* or *latent variables*. This is done by a

bilinear decomposition of the covariance matrix of the dataset. The uncorrelated (orthogonal) variables obtained are called the *principal components* and they represent the axes obtained by rotation of the original co-ordinate system along the direction of maximum variance. The main assumptions in this method are that the data follows a Gaussian distribution and that all the samples are independent of one another [16]. Since the PCA approach is adopted in this work, a full discussion concerning the different approaches of Principal Component Analysis is provided in the second chapter.

- *Partial Least Squares* (PLS) is a dimensional reduction as well as a *regression* technique that finds a new set of latent variables which maximize the covariance between the input data matrix X and the output data matrix Y. The main objective here is to approximate X and Y into reduced dimensional forms as well as to model a linear relationship between them.
- *Independent Component Analysis* (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals [17]. ICA can be used to extract useful information from original signals, which provides more abundant and useful information for the fault diagnostic system. In real process, the variable distribution characteristics are complicated, thus some variables are Gaussian distributed and others are not. Separating Gaussian variables from non-Gaussian variables and building monitoring models can take advantages of both models and then achieve good monitoring performance [18]. The data are represented by the random vector $x = (x_1, \dots, x_m)$ and the components as the random vector $s = (s_1, \dots, s_a)$. The task is to transform the observed data x using a linear static transformation W into maximally independent components s measured by some independence function $F(s_1, \dots, s_a)$.

2. *Statistical classifier approaches:* Fault diagnosis is essentially a classification problem and hence can be cast in a classical statistical pattern recognition framework. Fault diagnosis can be considered as a problem of combining, over time, the instantaneous estimates of the classifier using the knowledge about the statistical properties of the failure modes of the system [14].
3. *Neural network approaches:* Neural networks do not require specific knowledge of process structure. They can serve as black-box models of general nonlinear, multivariable static and dynamic systems. They contain many parameters, but these parameters are generally not suitable for physical interpretation of the modeled system. In general, neural networks that have been used for fault diagnosis can be classified along two dimensions:
    - The architecture of the network such as sigmoidal, radial basis and so on.
    - The learning strategy such as supervised and unsupervised learning [10,12,14,17].

These methods are well suited to problems for which theoretical models of behavior are poorly developed or inadequate to explain observed performance and where training data are plentiful or inexpensive to create or collect. This approach provides black-box models, which are easy to develop and do not require an understanding of the physics of the system being modeled with a generally manageable computational requirement. Last but not least, there is a wealth of documented information available on the underlying mathematical methods. Beside all the advantages listed earlier, the most significant drawbacks of this approach emerge from the fact

that most of the models cannot be used to extrapolate beyond the range of the training data and a large amount of training data is needed, representing both normal and faulty operation. The models are specific to the system for which they are trained and rarely can be used on other systems. Process data-based methods are suitable where no other methods exist. Some are applicable for virtually any kind of pattern recognition problems [13].

1.7 Comparison of Various Diagnostic methods

Latterly, we have seen the two conceptually different frameworks for process fault diagnosis. In this section, we provide a comparative evaluation of these different frameworks against a common set of desirable characteristics. The evaluations are summarized in *Table1.1*. where a comparison of various methods is given in terms of the desirable characteristics of diagnostic systems. In the table only some representative methods in each of the two approaches (model-based, data based) are chosen for comparison. A check mark would indicate that the particular method satisfies the corresponding desirable property. A cross would indicate that the property is not satisfied and a question mark would indicate that the satisfiability of the property is case dependent [14].

Table 1. 1. Comparison of some diagnostic methods.

	Observer	Digraphs	Expert systems	QTA	PCA	Neural networks
Quick detection and diagnosis	✓	?	✓	✓	✓	✓
Isolability	✓	✗	✓	✓	✓	✓
Robustness	✓	✓	✗	✓	✓	✓
Novelty identifiability	?	✓	✗	?	✓	✓
Classification error	✗	✗	✗	✗	✗	✗
Adaptability	✗	✓	✓	?	✗	✗
Explanation facility	✗	✓	✓	✓	✗	✗
Modelling requirement	?	✓	✓	✓	✓	✓
Storage and computation	✓	?	✓	✓	✓	✓
Multiple fault identifiability	✓	✓	✗	✗	X	✗

✓ ≡ *suitable*, ✗ ≡ *not suitable*, ? ≡ *not assesed*.

1.8 Statistical Process Monitoring

In any industrial process, regardless of how well the process is designed and maintained, a natural inherent variability will always exist in the system as a background noise. The background noise is a result of many small, unavoidable causes. This kind of natural variability is often called *stable system of chance causes* or simply, *Chance causes*. Chance causes defines the acceptable variability when the system is under *statistical control* (healthy state). Any other variability that drives the system *out-of-control* are called *assignable causes*. A process operating

in the presence of any kind of assignable causes is exhibiting some kind of abnormal or faulty operation that have to be detected and removed effectively in minimum time.

SPM is a key feature in the long-term reliable operation of any automated controlled process. The main objective of the so called *statistical process monitoring (SPM)* is to provide a compact chart to monitor the process. *Control charts (CC),* are graphical representations for the process quality characteristics that has been measured or computed from an acquired data and using certain classifier and used to determine whether the process is under statistical control (healthy state) or not. For any control chart to be useful, it must satisfy the following objectives:

1. The process has to be monitored using the minimum possible number of graphs; i.e., for multivariate processes, all the measurement types have to be summarized into, preferably, one fault indicator. The use of multiple graphs makes the monitoring process more difficult and messy.
2. In order to be able to decide about the state of the process from the control chart, it is necessary to have some *control limits* for the chart. Usually, *Upper control limit (UCL), Lower control limit (LCL)* and *Central Line (CL)* are used to define the range within which any variability is considered to be due to chance causes only. Consequently, ant point that lies beyond the control limits is treated as a result of an assignable cause (fault) [19].

A typical control chart is shown in *Figure 1.8*. The control limits are shown along with the central line representing the average value. The control limits are defined based on the empirical distribution of the monitored quality. Therefore, the control limits are determined with a given confidence level; i.e., we select *Control Limit*s to ensures that $1-\alpha$ percent of the points expressing chance causes are lying within the control limits.

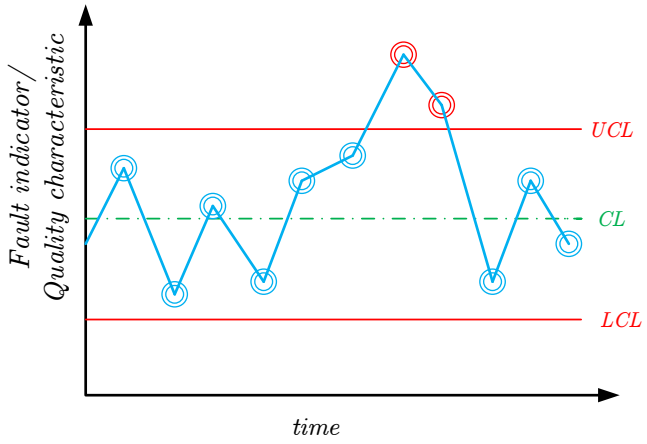

Figure 1. 8. Typical Control Chart for SPM.

It is customary to say that a point within the control limits indicates normal process state while a point outside the control limits indicates abnormal behavior. Nevertheless, due to the use of confidence intervals, two situations are always present:

1. A *run* or *persistence* of values indicates a strange behavior that might be faulty; since it is often expected that the control graph resulting from chance causes is randomly distributed around the central line. Therefore, a point within the control limits is not necessarily *in-control.*

2. A point beyond the control limits does not necessarily indicates an out-of-control state. A control chart based on confidence level of $1-\alpha$ does merely means that there is a probability equals to α that an element lying beyond the control limits is due to a chance causes. This fact points out the existence of *false indications* or *false alarms* in any SPM system which cannot be avoided.

The control limits are usually determined based on the *k-significance* rule:

$$CL = \mu_f \quad (1.5)$$

$$UCL = CL + k.\sigma_f \quad (1.6)$$

$$LCL = CL - k.\sigma_f \quad (1.7)$$

μ_f and σ_f are the mean and the variance of the fault indicator signal. Interestingly, the control limits, often called *detection threshold*, can be either constant or varying (*adaptive*). An adaptive thresholding scheme appears naturally when the process is not stationary; i.e., when the process has a moving average and non-constant variance. Adaptive thresholding is more efficient and better in terms of reducing the rate of false alarms and reducing the fault appearance time. Furthermore, it is usually due to the fault classifier and the method used for treating the data; the monitoring system is acceptable or not in terms of the false alarms rate and the fault detection time.

SPM is an endless loop that have to be always acting on any system, the following figure summarized the Process monitoring loop [20].

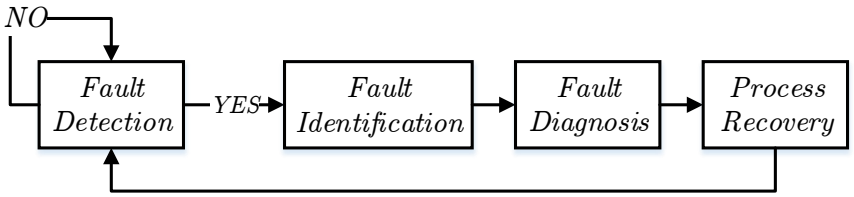

Figure 1. 9. Process monitoring loop.

1.9 Conclusion

In this chapter, the basic terminology of fault detection and diagnosis was presented. The different approaches used for fault detection was presented briefly as they appear in the literature. Many definition and aspects on FDD and SPM was investigated in order to achieve a background in the field. Furthermore, due to the vastness of FDD methods, a deep investigation for each method is a time expensive process. From this chapter, the use of data driven methods for fault detection can be justified based on the relative simplicity and the acceptable efficiency.

Chapter 2:

Principal Component Analysis

2.1 Introduction

For the task of Fault detection and diagnosis in industrial systems, many methods have been investigated. Throughout the long studies, data driven methods have proven themselves due to the availability of the process data and the easiness, applicability and the good results given by the data-driven methods. Furthermore, among the data driven methods existing in literature, the principal component analysis has been the most studied method. This chapter is dedicated to the exploration of the different PCA approaches and their applicability to the field of Fault detection and Diagnosis.

2.2 PCA: Mathematical Basis

Principal Component Analysis or *Empirical Orthogonal Functions* (EOF) is a multivariate statistical method for data analysis that aims to re-express a given multidimensional data as a set of orthogonal (uncorrelated) components by projecting the original dataset to a new axis (change of basis) aiming to extract the important information from the original dataset [21]. However, the question that arises by the moment is: What does "an important information" mean?

The term "*important*" is defined based on the assumption that the direction of the maximum variance defines the direction containing the maximum amount of information about the process, and the directions having a minimum variance along them contains almost no information to retain. Furthermore, if the directions with minimum variance is seen as a *white noise* carrier, PCA has the advantage to separate the data from the noise. This reasoning allows us to define the main objective of PCA as: "to transform a set of dependent variable to a set of uncorrelated variable, called Principal components, which are order so that the first few components contains most of the information in the original dataset" [22]. Since a change of basis is applied, PCA is clearly assuming a linear relationship between the original data. The main goals of Principal component analysis are:

1. Reducing the data dimensionality, i.e., representing the data with less number of random variables.
2. Removing data dependencies, i.e., transforming the data to a set of uncorrelated variables.

Figure 2.1. represents an illustration for PCA with two dimensional data. x and y represent the originally dependent data, while $\{u, v\}$ represent the Principal Components. In this figure, the new co-ordinate system can describe the data in a better way. The v-coordinates are very close to zero and can be neglected without any intensive loss of information. Moreover, one can assume that the v-coordinates are non-zero only because of the *random noise* and the uncertainties affecting the measurements.

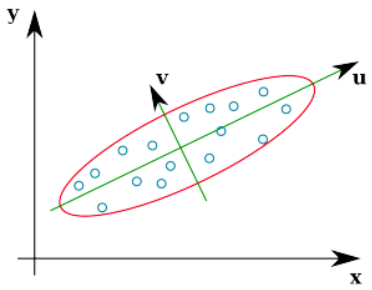

Figure 2. 1. Illustration of PCA in 2-D data.

Assume a set of data $X \in \mathbb{R}^{n\times m}$ where m is the number of variables (sensors) and n is the number of observations (instances). The PCA problem can be seen as the problem of finding an appropriate transformation $P \in \mathbb{R}^{m\times m}$ when applied to X results in a new data set $T \in \mathbb{R}^{n\times m}$ i.e.

$$T = X.P \tag{2.1}$$

The first thing we are seeking in T is to have an uncorrelated set of variables, which can be addressed via the *correlation* or the *covariance* matrices of the new data set T.
The covariance of two random variables $u, v \in \mathbb{R}^n$ is defined as:

$$Cov(u,v) = \sigma_{uv} = E((u - \mu_u)(v - \mu_v)) \tag{2.2}$$

Where E denotes the expectation and μ denotes the mean. The covariance can be calculated as:

$$\sigma_{uv} = \frac{\sum_{i=1}^{n}(u_i - \mu_u)(v_i - \mu_v)}{n} \tag{2.3}$$

The covariance as represented in (2.3) is a *biased estimator* [23,24]. While the *unbiased estimator* for the covariance is given by:

$$\sigma_{uv} = \frac{\sum_{i=1}^{n}(u_i - \mu_u)(v_i - \mu_v)}{n-1} \tag{2.4}$$

The correlation is then defined as:

$$\rho_{uv} = \frac{\sigma_{uv}}{\sigma_u.\sigma_v} \tag{2.5}$$

Provided that $\sigma_u^2 = \sigma_{uu}$ is the variance of the random variable u. For a data matrix

$$X = [x_1; x_2; \dots; x_m] \tag{2.6}$$

x_i denotes a measurement type (column) of X. The covariance and the correlation matrices are defined as in (2.7) and (2.8) respectively:

$$\sigma_X = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1x_2} & \cdots & \sigma_{x_1x_m} \\ \sigma_{x_2x_1} & \sigma_{x_2}^2 & \cdots & \sigma_{x_2x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_mx_1} & \sigma_{x_mx_2} & \cdots & \sigma_{x_m}^2 \end{bmatrix} \tag{2.7}$$

$$\rho_X = \begin{bmatrix} 1 & \rho_{x_1x_2} & \cdots & \rho_{x_1x_m} \\ \rho_{x_2x_1} & 1 & \cdots & \rho_{x_2x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{x_mx_1} & \rho_{x_mx_2} & \cdots & 1 \end{bmatrix} \tag{2.8}$$

Back to the PCA problem, forcing T to have uncorrelated columns is mathematically equivalent to *diagonalizing* the covariance (correlation) matrix of T. Before dealing with this problem, the original dataset has to be normalized, i.e.,

1. Each measurement type has a zero mean; this simplifies the computation of the covariance (correlation) matrix to:

$$\sigma_X = \frac{X_0^T . X_0}{n-1} \tag{2.9}$$

With:

$$X_0 = [x_1 - \mu_{x_1};\ x_2 - \mu_{x_2}; \dots;\ x_m - \mu_{x_m}] \tag{2.10}$$

2. In case of using the covariance matrix, each measurement type should have a unity variance in order to unify the units of the different measurements. We define the normalized data set:

$$X_n = \left[\frac{x_1 - \mu_{x_1}}{\sigma_{x_1}};\ \frac{x_2 - \mu_{x_2}}{\sigma_{x_2}}; \dots;\ \frac{x_m - \mu_{x_m}}{\sigma_{x_m}}\right] \tag{2.11}$$

The intended transformation is changed to:

$$T_n = X_n . P \tag{2.12}$$

In order for T_n to be normalized, P have to be *orthonormal*. The covariance matrix of T_n is then:

$$\sigma_{T_n} = \frac{T_n^T . T_n}{n-1} = P^T \frac{X_n^T . X_n}{n-1} P \tag{2.13}$$

Thus:

$$\sigma_{T_n} = P^T . Cov(X_n) . P \tag{2.14}$$

By noticing that the covariance (correlation) matrix is *symmetric* and by taking the assumption that P is *orthonormal*, we can write:

$$Cov(X_n) = P . \sigma_{T_n} . P^T \tag{2.15}$$

Equation (2.15) transforms the PCA problem to the simpler problem of finding a proper factorization for the matrix σ_{X_n}that takes the form of equation (2.15) with P is an $m \times m$ orthonormal matrix and σ_{T_n} is a diagonal matrix. Interestingly, the form of equation (2.15) can be simply found by applying the so-called *singular value decomposition* (refer to Appendix A).
T_n is called the *Scores* or *Features* matrix, while P is called the *Loading* or the *Principal components* matrix. The columns of the loading matrix are the principle components which are the eigenvectors of the matrix σ_{X_n}, this last can be proven based on the *Eigen-structure* equation of the covariance matrix:

$$\sigma_{X_n} . p_i = \lambda_i . p_i \tag{2.16}$$

Assembling these equations, leads to:

$$\sigma_{X_n} . [p_1\ \ p_2\ \ \dots p_m] = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} . [p_1\ \ p_2\ \ \dots p_m] \tag{2.17}$$

Where the eigenvectors can be selected to be orthogonal, equation (2.17) can be arranged as

$$\begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} = [p_1 \ p_2 \ \ldots p_m]^T . \sigma_{X_n} . [p_1 \ p_2 \ \ldots p_m] \quad (2.18)$$

The *eigenvector* associated with the largest *eigenvalue* describes the direction of the largest variance and so on. Moreover, it can be shown that the *singular value* λ_i represents the variance in the direction of the corresponding principal vector p_i. The columns of P (the principal vectors) are ordered according to the decreasing order of the singular values of σ_{X_n}. In many cases, due to redundancy between the variables, fewer components are sufficient to represent the data. Thus, using $a < m$ of the components, one can obtain a-dimensional scores by the following relationship: [24]

$$\hat{T}_n = X_n . P_a \quad (2.19)$$

Where P_a contains only the first a columns of P. a is called the PCA model *dimension*. In fact, the number of retained components is the most critical factor that affects the *sensitivity* of any PCA model. By the assumption that the orthogonal projection will compress and separate the information and the noise, overestimating the number of retained components will increase the randomness in the model which affects the quality of the detection and increases the *false alarms*. In the other side, underestimating the number of components results in a loss of information [26,27,28].

As a summery, the construction of a PCA model means:

1. Determining the loadings P.
2. Determination of the number of components to retain.
3. Calculation of the features (scores).

While a PCA model would looks like (2.20), with E is the error matrix.

$$X_n = T_n . P_a^T + E \quad (2.20)$$

As we have seen previously, the *loading* matrix and the *scores* can be calculated using *SVD*. A more commonly used method for calculating the principal components of a data set is presented under the name NIPALS: *Nonlinear Iterative Partial Least Squares*. This algorithm was developed first for PCA and have been extended to PLS. NIPALS gives more numerically accurate results when compared with the SVD of the covariance matrix, but it is slower to calculate [29,30].

2.3 Model-Dimension Selection

The selection of the number of retained components plays a major role in the goodness of the PCA model. Many researches have been carried out in order to determine the optimal number of principle components. However, a method to estimate the number of retained components that is guaranteed to peak the best number of PCs still out of reach; even though a considerable number of methods are already presented in literature. Some of the widely used methods are presented in this section.

2.3.1 Kaiser criteria

The so called Guttman-Kaiser criteria was found in the early 1954, and still being the most used method for the estimation of the number of components in PCA and factor analysis.

Kaiser criteria, also known as *k1*, consists of picking the components corresponding to those eigenvalues larger than one (the average eigenvalue). The first investigation on the method lead to the result that the *k1* criteria provides a lower band for the number of retained components while other studies shows that the criteria tends to overestimate the number of retained components. Furthermore, picking the components with singular values greater than one makes the criteria so discussable, since a component with singular value equals to 1.01 is considered as significant and informative and, therefore, should be retained while a component with singular value equals 0.99 is not significant [31]. Another major problem was reported by many studies, *k1* always retains between $\frac{1}{3}$ and $\frac{1}{5}$ or $\frac{1}{6}$ of the total components. A modification to the Kaiser criteria was suggested in [22] In order to deal with the significance problem, a reasonable choice is to select a cut-off less than 1, the value proposed in [32] was 0.7 which will be referred to as *J7* criteria throughout this text. The difference between the number of picked components using Kaiser's and Jolliffe's methods can be dramatic.

Another modification to *k1* is proposed in [33], a *distribution free* method is proposed. This method sets the significance limit to a value greater than 1 using the formula:

$$\hat{\lambda} = 1 + 2\frac{\sqrt{m-1}}{\sqrt{n-1}} \tag{2.21}$$

This last rule is known as the KSS rule (Karlis-Saporta-Spinaki).

2.3.2 Scree Plot

The scree test was first proposed in [34] as a method that consists of plotting the eigenvalues of the covariance/correlation matrix in decreasing order then exploring the resulting graph to determine the point where the last drop takes place and the graph starts to be smooth. The reasoning behind this approach can be seen as if the elbow point is dividing the important (major) components from the insignificant (minor) components. This test is simple to apply, but it may be hard to interpret due to the fact that a graphical method without any systematic rule may turns out to be highly subjective. Furthermore, the graph itself might be misleading due to the ambiguity of the elbow due to the gradual sloop of the graph or the existence of more than one elbow [22,35,36]. Usually, a cumulative eigenvalue proportion graph is used in parallel with the scree test in order to increase the confidence in the selection made by the method. The test can be carried out in terms of the *logarithmic eigenvalue test* (LEV), this test does extend the scree test by plotting the logarithms of the eigenvalues ($\log(\lambda_i)\ vs.\ i$) instead of the eigenvalues. This approach can increase the interpretability of the plot. In a comparative study carried in [35], 90% of scree test estimation errors were found to be *underestimates*.

2.3.3 Cumulative Percent of Variance (CPV)

It is well known that the variance is a good measure for the importance of a given principal dimension and for how much of information could be exist in that dimension compared to the others. Thus, retaining a number of components that corresponds to a certain percentage of the total variance is reasonable. The amount of variance preserved by a-dimensions out of a total of m available measurement types can be calculated based on the eigenvalues of the covariance/correlation matrix as:

$$CPV\% = \frac{\sum_{i=1}^{a}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \times 100 \tag{2.22}$$

Even though CPV is one of the most used methods, a major problem emerges from the fact that the percentage of the cumulative variance have to be set in advance. Usually, a percentage between $80-85\%$ is used in some context while some references encourages the use of percentages greater than 85%, but the optimal selection that guarantees the best representation for the data is a data-dependent and still out of reach. The optimal value of the CPV that best represents the data may vary widely according to the amount of noise present in the measurement and the redundancy between the different variables. Due to the absence of a clear rule, CPV turns out to be subjective and the fact that the cumulative percentage increases with the increase of the number of components makes the method ambiguous. Furthermore, while we would like to conserve as much as possible of the variance, we want to retain as few principal components as possible and to keep the noise out of interference. Therefore, the decision becomes a matter of trading-off the amount of conserved variance and the number of retained components [37].

2.3.4 Broken Stick method (BS)

The reasoning behind this method is based on considering the variance shared by the principal axes to be embedded in a *stick* of a unit length. If we have a stick of unit length, broken randomly to m segments, it can be shown that the expected length of the k^{th} longest segment would be:

$$E(\mathcal{L}_k) = \frac{1}{m}\sum_{i=k}^{m}\frac{1}{i} \tag{2.23}$$

If we consider that the PCA model divides the variance randomly on all the m-dimensions, the fraction of the variance explained by each dimension would be the same as the relative lengths obtained by breaking the sticks randomly into m pieces [38]. Then it would be meaningless to retain those components that explain variance less than or equal that predicted by the broken stick model. BS can be carried in two ways after putting the eigenvalues and the BS predictions in decreasing order:

1. Compare individual eigenvalues with individual BS predictions.
2. Compare cumulative eigenvalues with cumulative BS predictions.

Some studies showed that broken stick model has a tendency to underestimate the number of retained components, and a major problem with this model is its independence from the sample size [39]. Nevertheless, BS model is easy to calculate and it showed a good performance with acceptable accuracy, at least when the studied data is highly correlated [40].

2.3.5 Minimum Average Partial (MAP)

The minimum average partial is a method based on the *matrix of partial correlation* that was developed basically for component analysis and first presented in [41]. This method was claimed to be exact in its foundation. MAP involves a complete PCA followed by the examination of the matrices of partial correlations [42]. The average of squared partial correlations is used as a *Goodness-of-fit* measure, when the minimum average squared partial correlation is reached, the *residual matrix* resembles an identity matrix and no further components are extracted. The original MAP algorithm is presented in *Figure 2.2*. The statistic:

$$f_0 = \sum_{\substack{i=1 \\ i \neq j}}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} \frac{\rho_{ij}^2}{m(m-1)} \tag{2.24}$$

Was proposed for comparative purposes. If $f_0 > f_1$, then no component would be extracted. Nevertheless, this later statistic is of no use in our work.

In this method, the systematic variance starts decreasing as more significant components are partialled out, then the unsystematic variance causes the *average statistic of partial correlations* (ASPC) to increase again [42]. Minimum average partial method was investigated in [35], it was found that MAP is more accurate than scree test and *k1*, quite accurate and shows less variability and not affected by the sample size. In the other side, MAP tends to neglect non-trivial components when they have small loading (Poorly Determined Components; PDCs), this fact means that MAP tends to underestimate the number of components under the presence of PDCs and it was concluded in [35] that 90% of the errors made by the method are underestimations.

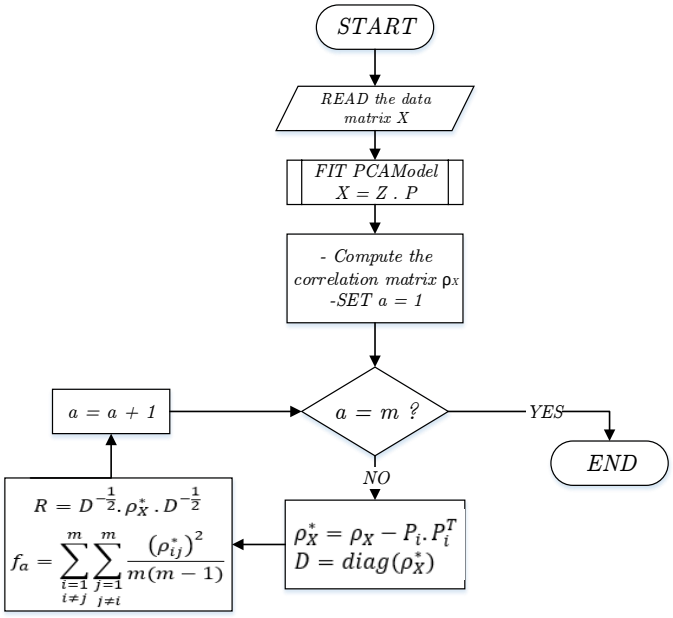

Figure 2. 2. Flowchart of the MAP algorithm.

2.3.6 Parallel Analysis (PA)

Parallel analysis is a sample-based adaptation for the population-based *k1* rule, which implies *Monte Carlo* simulation process [43]. The methodology is based on comparing the eigenvalues of the Covariance/Correlation matrix of the original data set with those of a randomly sampled data set from normally distributed population. A component is considered to be significant once the associated eigenvalue is larger than the one generated from the random sample. Based on the fact that a method based on random sampling may exhibit a large variability, it is usually required to repeat the generation of the pseudo-eigenvalues for *k*-times. Having a set of ordered pseudo-eigenvalues for each experiment, one could increase the estimation by taking either the average eigenvalue for each component or by assessing the empirical distribution of the eigenvalues associated with a given component to determine an upper confidence limit for each pseudo-eigenvalue with a pre-selected percentile [44].

The algorithm of PA as stated in [43] is summarized in the following few lines:

1. Calculate the eigenvalues of the covariance matrix: $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$.
2. Generate k-sets of random data with the same size as X ($Y_i \in \mathbb{R}^{n \times m},\ i = 1,2, \dots, k$).
3. Compute the eigenvalues Y_i and store them in $\Xi \in \mathbb{R}^{k \times m}$ (row-wise). With ξ_{ij} is the j^{th} pseudo-eigenvalue for the i^{th} experiment.

$$\Xi = \begin{bmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1m} \\ \xi_{21} & \xi_{22} & \cdots & \xi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{k1} & \xi_{k2} & \cdots & \xi_{km} \end{bmatrix} \tag{2.25}$$

   Do either:
   - Compute the average eigenvalue: $\hat{\Xi} = [\overline{\xi_1} \quad \overline{\xi_2} \quad \cdots \quad \overline{\xi_m}]$.
   - Compute the α-percentile for each column: $\hat{\Xi} = [\xi_{1\alpha} \quad \xi_{1\alpha} \quad \cdots \quad \xi_{1\alpha}]$. With α is usually selected between 0.95 and 0.99 serves to fix the tendency of the method to overestimate the number of components.
4. Retain the components for which: $\hat{\Xi}_i > \lambda_i$.

The study carried in [35] concludes that PA is the best of the studied methods, it was found to be able to retain the exact number of components in about 92% of the cases with 66% of the estimation errors are overestimations. Furthermore, PA was found to be insensitive to the distributional form of the data [45,46]. In [44], Glorfeld stated that "one would find few reasons to choose another method over Parallel Analysis".

2.3.7 Cross-Validation (CV)

Cross validation is a statistical method used for the purpose of comparing learning algorithms. CV works based on the principle of dividing the available set of data into two parts, *training* and *testing* sets. The training set is used to estimate a model while the testing set is used to validate that model. Like in MAP, a *criterion of goodness-of-fit* (GFC) have to be selected in order to evaluate the model since the validation step for a PCA model will consist of selecting the number of components to retain out of m possible values. Basically, a PCA model have to be constructed for $i = 1,2, \dots, m$ and the one that best satisfies the selected GFC is selected. Cross-validation is discussed in Appendix B.

2.3.8 Bootstrap method

Bootstrap can be used in a similar way to CV, thereby, taking a training and testing sets. Nevertheless, the Bootstrap is a method used to assess the model performance. Basically, this method is based on randomly drawing n samples from the data set with replacement in order to form the bootstrap sample. The bootstrap sample is then used to train the model, which have to be tested on the unselected instances by the drawing process. Due to the random sampling, the bootstrap method exhibits a large variability. Finally, more discussion concerning the bootstrap method is available in appendix B.

2.4 Main Draw-Backs of PCA

When one comes to applying PCA on pre-acquired data set in order to generate a model, the determination of the exact number of components to retain might be the greatest problem. As it was discussed earlier, overestimating the number of components means that we have

contaminated the extracted information by the addition of a noisy dimensions with approximately no useful information, this cases an excessive amount of false alarms. In the other side, underestimating the number of components means a lack of information since one or more significant dimensions are neglected. If a fault that largely affects the neglected dimensions occurs, it will be either totally contained in those dimensions, and therefore undetectable, or it cannot be detected unless it grows large enough to affect the other dimensions. Clearly, this imposes, in the best case, a *delay* in the detection of the fault, which can be disastrous for the system.

Principal Component Analysis assumes *linear relationships* between the variables, this assumption might be misleading. The different measured variables may exhibit any kind of nonlinear relationships (quadratic, cubic, …etc.), the existence of such a relationship poses the problem of extracting useless and uninformative dimensions. Moreover, the fact that PCA is a non-parametric method serves as a strength and as a weakness simultaneously. The absence of any parameters makes the process of adjusting unwanted results via tweaking some parameters useless.

The third problem existing with PCA rises when monitoring the system. It was recommended that the data is normalized before applying the PCA model. However, when new data instance is received, using the means and the variances from the testing set might be inappropriate. For instance, randomly picked parts of the data usually have different means and variances i.e. the means and the variances are varying with time. A data with fixed means and variances whatever the selected time interval is known as *stationary data*. Stationarity of a data set is quite impossible to be attained in real processes. Furthermore, PCA as presented earlier does not take into consideration the existence of relationships between observations at different time points, or the so called "*autocorrelation*" within the time-series. Autocorrelation often rises when the measurements within one time-series (measurement type) are not independent, this often occurs due to the dynamic behavior of the process and the sensors.

The PCA approach in its simple form discussed earlier is called static PCA because the fitted model remains static as new observations are obtained. In the next sections, different proposed approaches of PCA proposed to cope with the previously stated problems are presented. Nevertheless, no method existing in literature that has the ability to deal with non-linearity, non-stationarity and autocorrelation together.

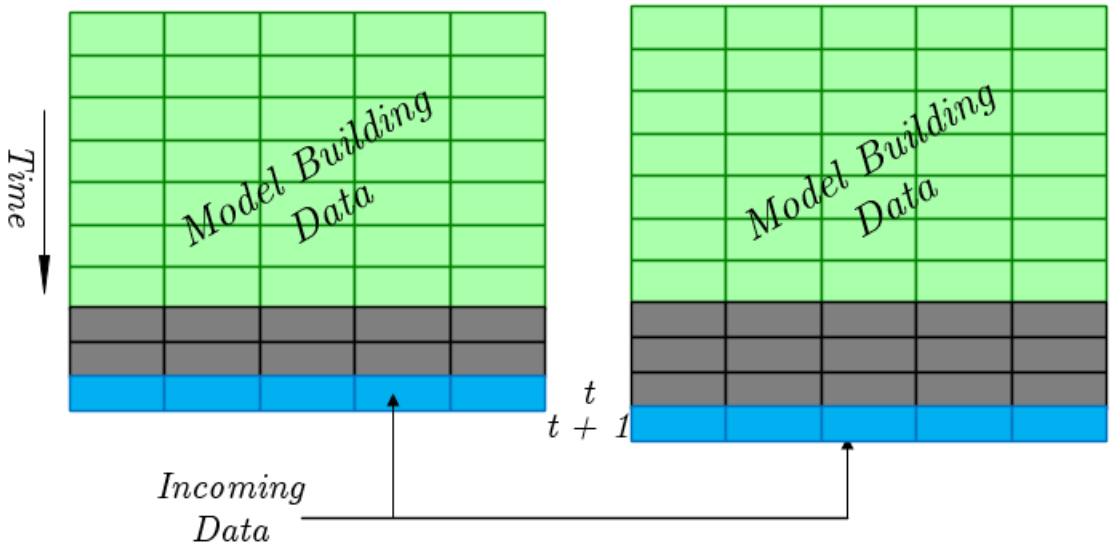

Figure 2. 3. Schematic representation for SPCA at times t (left) and t+1 (right).

2.5 Dynamic PCA Approach (DPCA)

As discussed earlier, PCA shows a poor performance when the measurements have some kind of persistency or auto-correlation. In the case of dynamic systems, auto-correlation among observations is inevitable since it points to the existence of some dynamic features in the system. Therefore, to include the dynamic effects in the PCA based monitoring, a Dynamic extension to the conventional PCA is often used [47]. SPCA was extended for data coming from a *dynamic system* taking into account series correlation of variables instead of only parallel correlation [48]. Dynamic PCA, as presented in [49], attempts to model the auto-correlation structure present in the data, through a *time lag shift* method. This method consists of including time lagged replicates of the variables under analysis, in order to capture simultaneously the static relationships and the *dynamical structure*, through the application of standard PCA [50]. To implement this method, it is possible to incorporate the description of variable autocorrelation into the standard PCA framework, by introducing time-shifted replicates as additional variables in the original data set $X \in \mathbb{R}^{n \times m}$. It is possible to model the relationships between variables (correlation) and between observations (auto-correlation and cross-correlation, depending on whether the variables involved are the same or not) by the extended matrix $\tilde{X} \in \mathbb{R}^{(n-l)\times(m.l)}$. The inclusion of time-shifted variables can be represented according to equation (2.26):

$$\tilde{X} = [\underbrace{x_1(0) \dots x_m(0)}_{X(0)} \quad \underbrace{x_1(1) \dots x_m(1)}_{X(1)} \quad \dots \quad \underbrace{x_1(l) \dots x_m(l)}_{X(l)}] \tag{2.26}$$

Where $x_i(j)$ represents the i^{th} variable (in column format) shifted j times into the past (i.e., with j lags), $x_i(j)[k] = x_i(0)[k+j]$ (the indices inside square brackets are the entry identifiers of the column vectors $x_i(j)$ and $x_i(0)$, respectively). $X(j) \in \mathbb{R}^{(n-l)\times m}$ is the submatrix containing all the original variables shifted j times. $\tilde{X}$ is the resulting extended matrix (with l lags). *Figure 2.4.* provides a summary for the DPCA approach where SPCA is applied on the lagged data matrix.

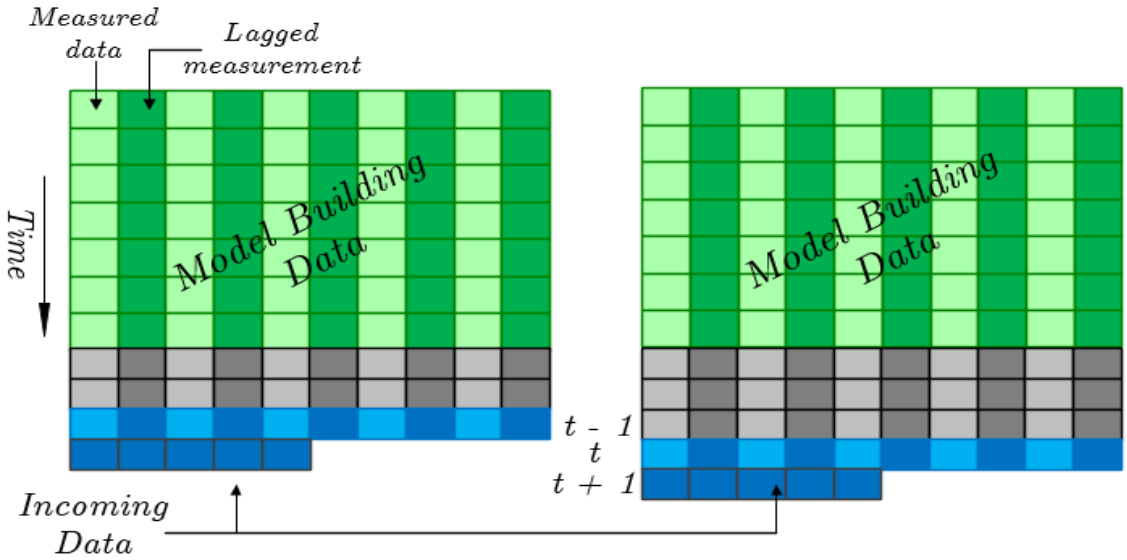

Figure 2. 4. Schematic representation for DPCA with one lag at time t (left) and t+1 (right).

2.5.1 Selecting the lag structure in DPCA

The most important and the crucial point in the implementation of the DPCA method is the selection of the number of lags to be used (the number of shifted versions for each variable to include in the DPCA model). Several methods to select the number of lags are presented in [50] and [25]. The different available algorithms can be divided into two major approaches:

1. Selecting a global number of lags, where all the components are delayed by the same amount.
2. Selecting a specific number of lags for each measurement type.

The simplest method in the first approach follows the same reasoning used when constructing the autocorrelation functions, i.e., using $l = \frac{n}{4}$. For large data sets, this will impose a high delay and will cause a large increase in the computational time. Another reasonable method consists of applying no lags and examining the *Autocorrelation functions* (ACF) of the Scores. If a significant autocorrelation is observed, additional lag has to be added and the process is repeated. A method presented in [49] starts from one lags and takes a lag into consideration only if it adds an important linear relationship, ones an added lag does not provide an additional important linear relationship, the algorithm comes to an end. A method proposed in [50] and claimed to be efficient assumes that there are m *linear dynamical relationships* to be identified, whose order is not known a priori, but is at least a first order *auto-regressive* process or *Markovian*. The algorithm consists of sequentially introducing an additional set of time-shift replicates for all original variables (which corresponds to the consideration of one more lag in the extended matrix), after which the singular values are computed for the corresponding covariance matrix. This procedure is repeated until a pre-defined upper limit on the number of lags, l_{max}, (this limit is usually a number high enough for allowing the description of all the dynamical features present in the data, but it can be adjusted during the analysis if it is concluded that it was underestimated initially). In each stage, l (which also coincides with the number of lags introduced in the extended matrix), the following quantities are computed from the singular values:

1. *Key Singular Value* (KSV): The key singular value in the l^{th} stage $(l \geq 1)$, $KSV(l)$, is defined as the $(m.l + 1)^{th}$ singular value, after sorting the set of singular values according to the decreasing order of their magnitude.
2. *Key Singular Value Ratio* (KSVR): is defined by the following expression:

$$KSVR = \frac{KSV(l)}{KSV(l-1)} \tag{2.27}$$

The maximum number of lags to be considered in the extended matrix for implementing DPCA should obey the following two criteria:

1. Have a small KSV
2. Have a low value for KSVR.

In order to find the number of lags that best satisfies the two conditions, the following procedure was proposed where the objective is to get the minimum of the *Distance to Optimum* (DTO) function ϕ defined as:

$$\phi = \sqrt{KSV_N^2(l) + KSVR_N^2(l)} \tag{2.28}$$

Where:

$$KSV_N(l) = \frac{KSV(l) - \min(KSV)}{\max(KSV) - \min(KSV)} \tag{2.29}$$

And

$$KSVR_N(l) = \frac{KSVR(l) - \min(KSVR)}{\max(KSVR) - \min(KSVR)} \tag{2.30}$$

DPCA can be seen as a combination of the ability of SPCA to deal with high dimensional data and the ability of the *Autoregressive Integrated Moving Average models* (ARIMA) to cope with the autocorrelation. In a study carried in [51], it was found that the scores of DPCA model will inevitably exhibit some autocorrelation. And it was shown that the presence of auto-correlated score variables will definitely leads to an increased rate of false alarms when the Hotelling's T^2-statistic is used. Furthermore, it was suggested that the Q-statistic will not be affected by the autocorrelation existing in the scores. And as a solution, an *Autoregressive Moving-Average* (ARMA) filtering was suggested [25]. The use of a wavelet filtering was proposed in [52] as a way to reduce the rate of false alarms via separating the signal of the sensors from the contaminating noise.

2.6 Recursive PCA (RPCA) and Moving Window PCA (MWPCA)

SPCA and DPCA control charts are both unable to cope with the non-stationarity present in most of the industrial data. If one of these models is applied to data coming from a nonstationary process, then issues can arise where the mean and/or covariance structure of the model become misspecified because they are estimated using observations from a time period with little similarity to the one being monitored. The idea behind the Recursive and the Moving-window approaches is to limit the effect of the *nonstationarity* through limiting the effect of *old data* [25]. For both approaches, when a new data instant is measured, it is evaluated according to the existing model. If the fault indicator for this new data exceeds the control limits, i.e. a fault or an outlier, the model is kept unchanged. However, when the new data is in control, the new data is added to the data set and a new model is evaluated. When the incoming data is judged to be healthy and added to the *model-construction data* set (MCDS), the two approaches becomes different:

1. In RPCA, all historical data are stored in the model-construction data set. Except for the new data, the MCDS is down-weighted using a *forgetting factor* $0 < \eta < 1$, and the model is updated. Nevertheless, carrying the method in this way results in high computational time an increase of need for storage capability in order to keep all the old data. In practice, updating is not performed using the full data matrix, but rather a weighting is performed to update only the mean and the covariance matrix. A proposed method to deal with the computational time is to reduce the number of the updates for the model [53]. *Figure 2.5* summarizes the Recursive PCA approach.

   The forgetting factor is selected Typically in the range $0.9 < \eta < 0.999$ because forgetting occurs exponentially, but lower values may be necessary for highly nonstationary processes [25].

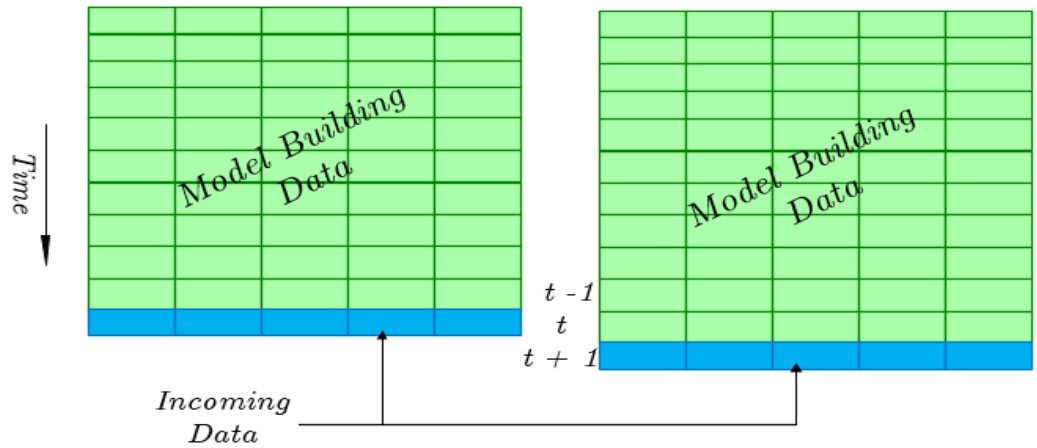

Figure 2. 5. Schematic representation for RPCA at time t (left) and t+1 (right).

2. Unlike RPCA, MWPCA updates the PCA model with each incoming (healthy) data instant while restricting the observations used in the estimations to those that fall within a specified *window* of time. With each new observation, the MCDS has the same size. This is achieved by excluding the oldest observation in the MCDS and including the new observation. At two successive time instances, the MCDS will have the form in equations (2.31) and (2.32) respectively (with "w" is the length of the window).

$$X_t = [x_{t-w+1}, x_{t-w+2}, \dots, x_t]^T \tag{2.31}$$

$$X_{t+1} = [x_{t-w+2}, x_{t-w+3}, \dots, x_{t+1}]^T \tag{2.32}$$

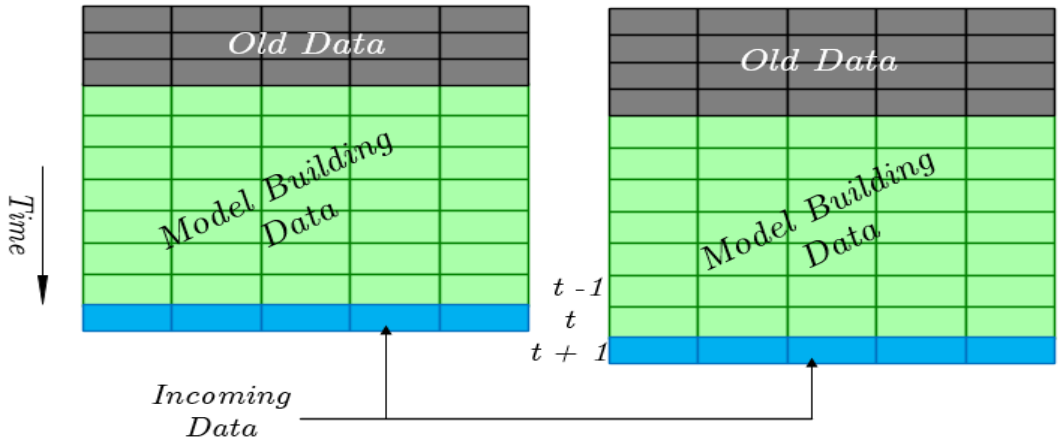

Figure 2. 6. Schematic representation for MWPCA at time t (left) and t+1 (right).

The main challenge in MWPCA is the selection of the size of the window. A rough estimation based on the convergence of the χ^2-distribution to the F-distribution for the T^2-based monitoring in [54]; lead to the result: the size of the window needed to correctly estimate the T^2-statistic has to be at least ten times the number of variables. A summary for the MWPCA approach is illustrated in *Figure 2.6.*

Furthermore, to reduce the effect of the nonstationarity on the whole monitoring process, the calibration data set used to generate the model at the first instant of the monitoring process

has to be carefully selected, i.e., the calibration data set has to be stationary. To optimize the selection, a methodology presented in [55] was to start by applying RPCA at the early monitoring period since no specific number of observations is needed. Once the required number of observations is reached, MWPCA is used to reduce the computational time and optimize the amount of stored data.

2.7 Robust PCA (ROBPCA)

Robust PCA is proposed as a modification for the conventional PCA method. Conventional PCA suffers from *grossly corrupted observations*. A number of different approaches exist for Robust PCA, including an idealized version of Robust PCA, which aims to recover a low-rank matrix L_0 from highly corrupted measurements.

$$M = L_0 + S_0 \tag{2.33}$$

This decomposition results in Low-rank and sparse matrices. Unlike the small noise term E in conventional PCA, the entries in S_0 can have arbitrarily large magnitude, and their support is assumed to be sparse but unknown [56].

2.8 Kernel PCA

As discussed earlier, PCA is linear technique that is used to compress, filter and extract information from a high-dimensional data set based on the existing linear relationships between the different measurement types. However, in practice, industrial data have an excessive amount of *nonlinear* relationships among the different variables. The method named Kernel Principal Component Analysis has been presented in [57] based on the so-called *Support Vector Machines* (SVM). *Figure 2.7* from [58] serves as a summary for the KPCA process.

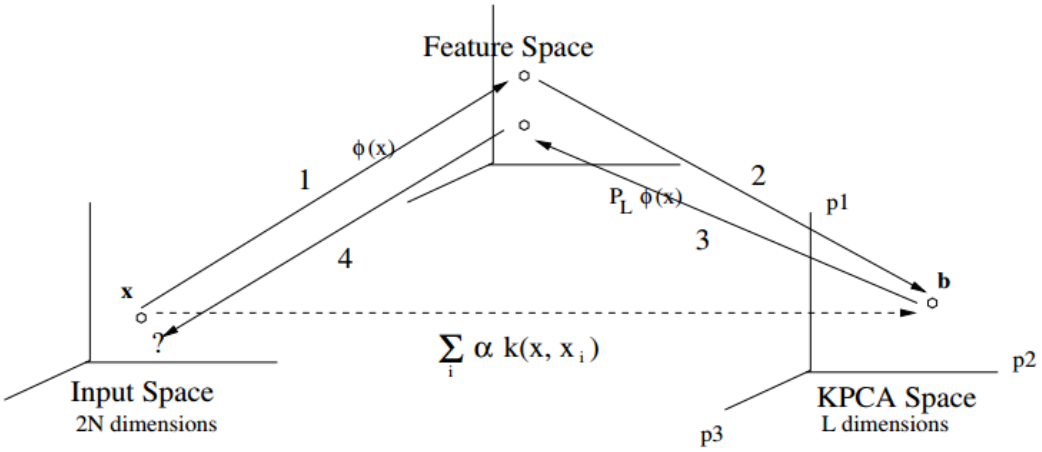

Figure 2. 7. KPCA process Summery.

Since conventional PCA cannot be applied effectively unless the linearity is assumed, the observations can always be mapped to higher dimensional space, called *Feature Space* ($\mathcal{F}$), where the new observations are varying linearly. KPCA utilizes SVM to find a proper nonlinear mapping from the input space to the features space through a simple kernel function. After that, linear PCA is applied on the data in order to move it from the feature space to the KPCA space [58].

Conceptually, KPCA is carried in the following steps:

1. A kernel function $\Phi(X)$ is used to map the data from the input space to the features space.

2. Linear PCA is performed in the feature space giving a lower dimensional KPCA space based representation.
3. To reconstruct the original data, the Kernel space representation has to be projected into the feature space and then the feature space representation is projected into the input space.

Nevertheless, the procedure described above is time consuming. Therefore, none of the previous steps has to be carried out. The computational procedure is carried out directly from the input space to the KPCA space using a set of *kernel functions* $\sum_i \alpha . k(x, xi)$ [58,59,60,61].

2.9 PCA-Based Fault Detection

In order to monitor a multivariate process, control charts have been developed based on PCA models. Having a PCA model based on historical data collected in the process's healthy state (only chance cause variation exist), future behavior can be referenced against this in-control model. New multivariate observations can be projected onto the plane defined by the PCA loading. Whatever the PCA approach used to generate the scores and the loadings, an online monitoring can be provided using the *Hotelling's* T^2 *statistic* and the Q*-statistic*, this last is usually referred to as the *Squared prediction error (SPE) statistic.* At each sampling instance, a new data points are acquired and saved in the row vector x_i and this incoming data is used to generate a fault indicator based on the PCA fitted model. For constant thresholding, a learning set is used to train the model, this training consists of generating relatively large number of *control points* and assessing their empirical distribution in order to determine a constant control limits. The T^2 and the Q statistics are defined as:

$$T_t^2 = (x_t - \mu) . P_a . \Sigma_a . P_a^T . (x_t - \mu)^T \tag{2.34}$$

$$Q_i = (x_t - \mu) . (I_m - P_a . P_a^T) . (x_t - \mu)^T = \|x_t - \mu\|^2 \tag{2.35}$$

Where Σ_a is the diagonal matrix of the largest a singular values of the covariance/correlation matrix, I_m is the $m \times m$ identity matrix, and P_a is the matrix of the principal loadings. The Q statistic is the quadratic orthogonal distance to the Principal Components space [25]. Both statistics are based on the generated symptoms which are always positive and usually close to zero. Therefore, the use of these statistics leads to a control chart where only an upper control limit is needed. In a process monitoring, the use of Hotelling's T^2-statistic based on the first a components only is not sufficient. Since the number of components can be only estimated, this method will only detect whether or not the variation in the measured properties through the first a PCs in the process is greater than what can be explained by the pre-defined chance cause only. If a new type of special which was not present in the data used to develop the PCA model appears, then new PCs will appear and the monitoring process for the new observations will be misleading [62]. This problem is solved by the Q-statistic where the prediction error is highly affected by the variations in the last $m - a$ components.

If we assume no autocorrelation and multivariate normality for the scores, the upper control limits can be calculated as follow:

1. For the T^2-based monitoring as:

$$UCL_{T^2}(\alpha) = \frac{a(n^2 - 1)}{n(n - 1)} . F_\alpha(a, n - a) \tag{2.36}$$

With $F_\alpha(a, n-a)$ is the $1-\alpha$ percentile of the F-distribution with a and $n-a$ degrees of freedom and n is the number of observations in the training set. If the number of observations is sufficiently large, the UCL can be found from the *chi-squared distribution* with a degrees of freedom using (2.37).

$$UCL_{T^2} = \chi_\alpha(a) \tag{2.37}$$

A similar and alternative test to the Hotelling's T^2 can be found in [63].

2. Using the Q-statistic, based on the *sum of prediction errors*, the following formula is used [64]:

$$UCL_Q(\alpha) = \theta_1 . \left(1 + z_\alpha . \frac{\sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2}{\theta_1^2} . h_0(1-h_0) \right)^{1/h_0} \tag{2.38}$$

Where:

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{2.39}$$

$$\theta_i = \sum_{j=a+1}^{m} \lambda_j^i \tag{2.40}$$

And λ_j is the j^{th} largest eigenvalue of the covariance/correlation matrix. And z_α represents the $1-\alpha$ percentile of the *standard normal distribution*. There exist other methods to compute the threshold for the Q-statistic, like the one presented in [65]. Nevertheless, the method presented by equation (2.38) is faster to compute and hence, more suitable for real time monitoring.

Usually, if the normality of the scores is not guaranteed, it is preferred to define the UCL of either statistic using the *empirical probability density function*.

In literature, hybrid models are also used. The output of principal component analysis which is a lower dimensional and uncorrelated data set could be used as an input for a second stage of neural networks or fuzzy logic. This combination was found to presents a good prediction accuracy and the reduced dimensionality of the input space results in decreased neural network complexity and training time. Most of the time, the use of hybrid methods results in better performance compared with the use of only one method [66,67].

2.10 PCA-Based Fault Identification:

Right after a fault is detected in a process, the determination of the *root causes* of the fault is a critical task. Even though the ability of PCA-based method to isolate faults is not very good due to the linear transformation that takes place on the data, we still can get some preliminary insight on the possible sources of the detected abnormal behavior. The most popular approach for PCA-based fault diagnosis is the contribution plots, this approach is simple and requires no prior knowledge. If a fault is detected at a time instant t_f, the fault indicator crosses the control limits and grows larger. The fault indicator signal can be decomposed to its basic components where we calculate the proportion with which each sensor is contributing to the total fault indicator at this instant. If a variable is found to have a large contribution, it has to be investigated. Diagnosis

based on contribution plots (CPBD) can be carried using the Q or the T^2 statistics. For the Q-based diagnosis, the SPE have to be broken down to the form:

$$Q_t = \sum_{i=1}^{m} Q_t^i \tag{2.41}$$

Where Q_t is the value of the Q-statistic at the time instance t and Q_t^i is the contribution of the i^{th} variable on the fault indicator. Q_t^i is the i^{th} element of Q_{t_Cont}

$$Q_{t\_Cont} = (x_t - \mu).(I_m - P_a.P_a^T) \tag{2.42}$$

The T^2-based contribution plot can be generated using the formula:

$$T_{t\_cont}^2 = (x_t - \mu).P_a.\Sigma_a.P_a^T \tag{2.43}$$

The contribution plots may not explicitly identify the cause of an abnormal condition, but they determine the variables that are not consistent with the normal operating conditions. Many other methods exist in the literature, and many of them was proven to be very acceptable in locating the sources of faults; these methods include: The reconstruction approach that consists of finding the true fault from a set of candidate faults, Fisher discriminant analysis (FDA) and support vector machines (SVM) [62,5,68,69,20].

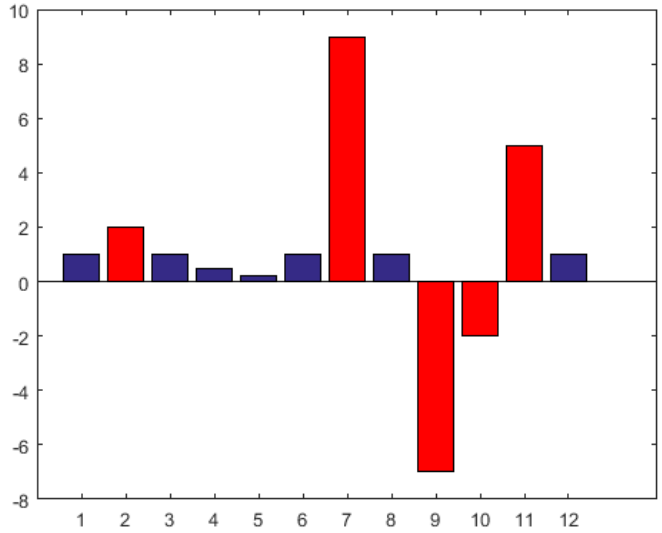

Figure 2. 8. Typical Contribution plot at a given time t, Red bars indicated possible sources of the fault.

2.11 Conclusion

In this chapter, PCA as a multivariate statistical approach for detecting faults in complex processes is presented. The mathematical basis of PCA models was established as well as the selection of the model parameters, such as the number of PCs and the loadings. Furthermore, different PCA approaches proposed to cope with the problems of autocorrelation, nonstationarity, and linearity were presented. The foundations of PCA-based fault detection and fault identification using the well-known Hotelling's T^2 and Q-statistics was presented.

Chapter 3:

Application of PCA in fault detection

3.1 Introduction

As discussed in the previous chapter, PCA is one of the most used methods in the field of Fault Detection. From the large variety of PCA approaches, Static models, i.e., models that are not updated with time, exhibits a poor efficiency when a new state is created or a parameter is changed in the system. Nevertheless, static models are still highly used due to the less complexity, the fast execution time of the algorithms, and due to the fact that in most industrial processes, significant changes in parameters are not highly occurring. This chapter is dedicated to the investigation of Static and Dynamic PCA approaches on an industrial data set (see Appendix C).

3.2 Static Principal Component Analysis (SPCA)

For the application of SPCA, a part of the data set, which was ensured to be healthy, is used to create the static model:

$$H = T.P_a + \mathrm{E} \tag{3.1}$$

And the model was used for the purpose of:

1. Data monitoring.
2. Abnormal behaviors detection.

3.2.1 Determination of the number of retained components

Several methods were applied on the data set in order to pick the best and the most descriptive number of components to successfully separate the informative parts of the signal from the noise. *Table 3.1*. provides a summary for most of the used methods. The results given by the different methods are widely spread, this phenomenon can be explained on the scope of the used data. Starting from *k1* and *PA*, the restriction of the retained components to those with singular values strictly greater than one causes a great loss in the retained variance, which is supposed to be a measure for how much the dimension is informative. Furthermore, KSS restricts the limits to values greater than one resulting in even higher loss of information. The interpretation of Scree plot is highly subjective while the Log-Eigenvalues plot were non-informative which makes both of them unreliable. For BS, the method works on the basis of the random breaking under the assumption that the data to be tested against is random. In our data set, a high persistency was observed in the measurements of different sensors, which violates the assumption of randomness. MAP resulted in the most inferior amount of retained variance where more than half of the variance carried by the dimensions is considered as noise. In the other side, *J7* and cross-validation are apparently showing a good behavior.

Table 3. 1. Summary of the different methods used to select the number of retained components in SPCA

Method	*Number of PCs*	*% of variance*
K1	14	72.97
J7	21	85.10
Scree-plot	10	65.12
	22	86.35
CPV	18	80.35
	21	85.10
	26	90.94
	34	96.80
Bootstrapped K1	14	72.97
Bootstrapped J7	22	85.10
MAP	4	43.67
Broken-Stick	7	57.71
Parallel Analysis	11	67.11
Resubstitution CV	33	96.28
Two-folds CV	28	92.83
LOOCV	33	96.28

k-fold cross-validation shows an acceptable robustness in picking the number of components associated with the minimum MPRESS. *Figure 3.1.* shows the MPRESS for LOOCV (left) and the number of retained components for each fold (right). From the left graph, MPRESS reaches a minimum of 43.7 at $a = 33$ components. The right graph shows that the number of retained components by KFCV is not highly varying with respect to the number of folds. Based on these results, the number of retained components needed to construct the SPCA model has to be looked between 21 and 42.

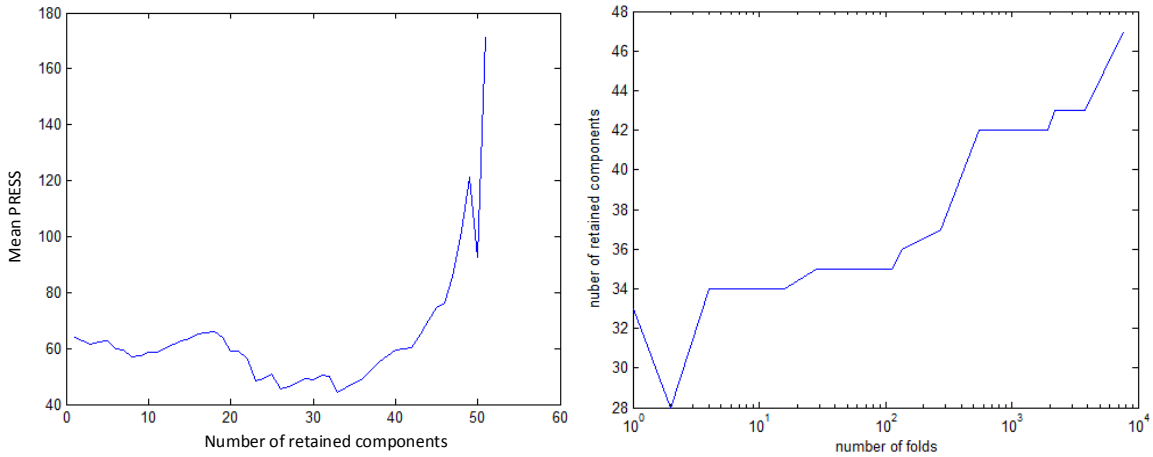

Figure 3. 1. LOOCV PRESS vs. # of retained components (left) and the number of retained components vs. the number of folds in KFCV (right).

Finally, in order to formalize the decision about the number of retained components, the number of component is varied from 1 to 52 and the rate of false alarms based on each model is recorded.

A training set of 11000 sample is used to construct the model while a set of 4000 (healty) observations is used as testing set to evaluate the false alarms rate.

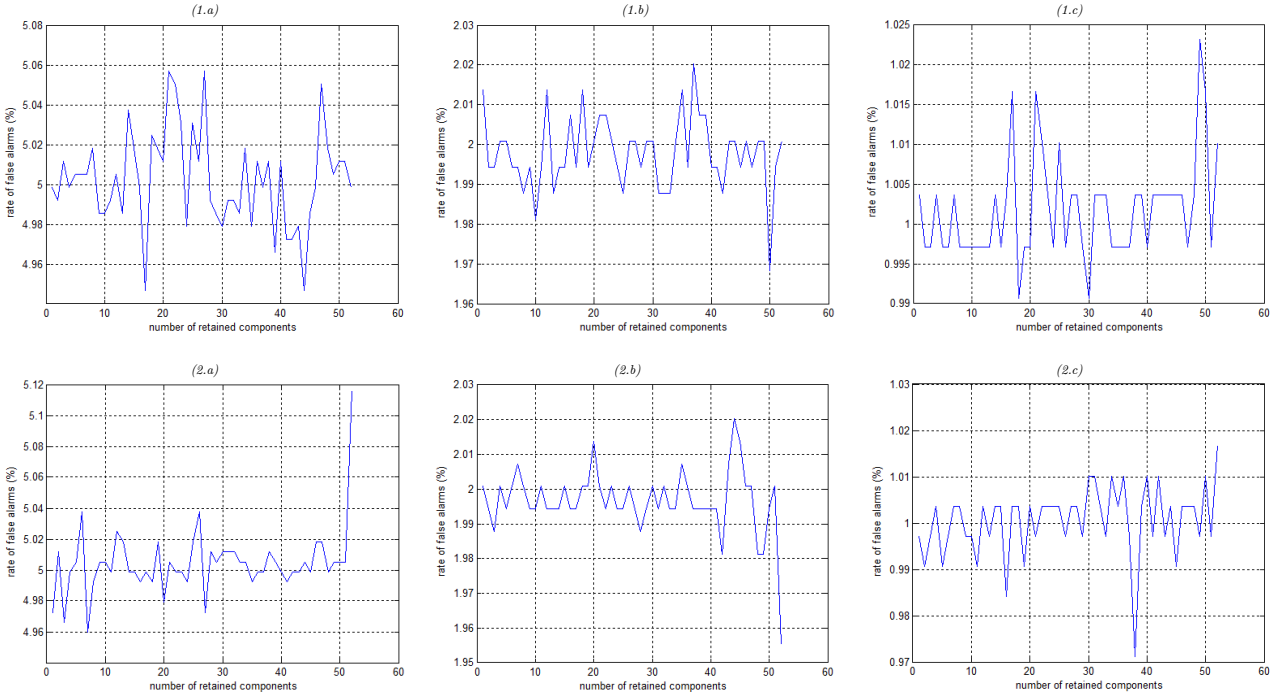

Figure 3. 2. The rate of false alarms vs. the number of retained components using the empirical distribution based on SPCA, for: (1.) T-square statistic. (2.) Q-statistic. At: (a) 95% (b) 98% (c) 99%. Confidence levels.

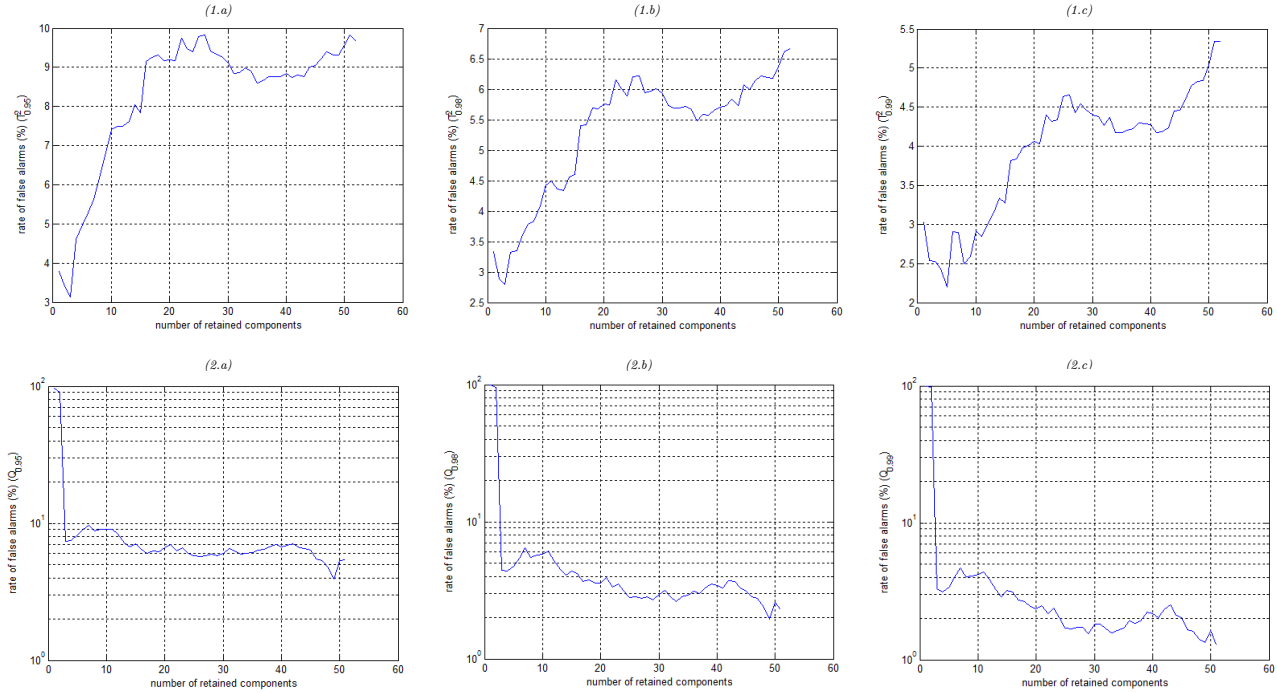

Figure 3. 3. The rate of false alarms vs. the number of retained components based on SPCA, using: (1.) T-square statistic (using equation (2.36)). (2.) Q-statistic (using equation (2.38)). At: (a) 95% (b) 98% (c) 99%. Confidence levels.

The rate of false alarms versus the number of retained components is shown in *Figure 3.2.* and *Figure 3.3.* from these figures, we can notice that in the range of 21 to 42 components, the rate of false alarms is reaching its local minimum at 33 components. Therefore, a number of components $a = 33$ that corresponds to a total retained variance of 96.28% is used whenever a SPCA is appearing in this context. Based on the healthy segment of the data, the loadings and the singular values are constructed for the SPCA model.

3.2.2 Development of UCL

Usually, the upper control limits for the Hotelling's T^2 and the Q statistics are calculated from equations (2.36) and (2.38). This calculation is carried based on the assumptions that the scores are multivariate normal. This latter assumption makes the UCL calculated based on standard normal distribution for the SPE and using F-distribution for the Hotelling's T^2 discussable. If the scores are not normal, then evaluating the UCLs using the mathematical formulae will lead either to an excessive rate of false alarms or a misdetection. The Empirical distribution for Hotelling's T^2 and Q-statistics are shown in *Figure 3.4.*

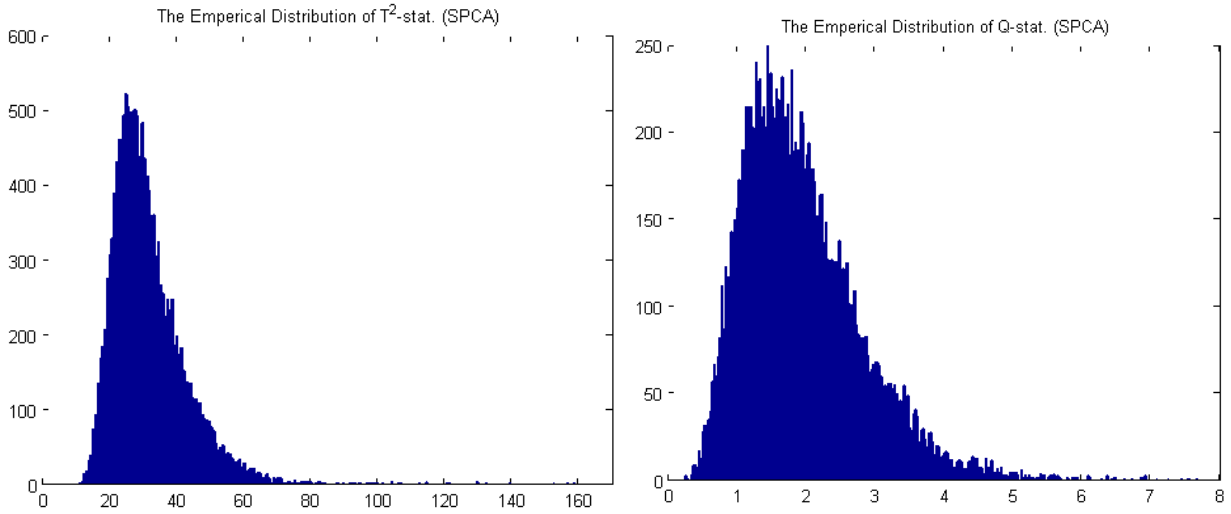

Figure 3. 4. Empirical Distributions for Hotelling's T^2 and SPE for SPCA.

In order to test the normality of the scores, a test of hypothesis has to be carried.

The Upper control limits calculated based on the empirical distribution and equations (2.36) and (2.38) are listed in *Table3.2.* it can be noticed that the upper control limits resulting from the Empirical distributions are larger than those from the mathematical formulae, this can be explained from two different aspects:

1. The deviation of the scores from multivariate normal distribution causes the threshold found based on the Fisher and normal distributions to be less accurate since the process being described is not following the intended distributions.
2. The presence of outliers with large magnitudes in the monitored statistics may cause the empirically estimated threshold to be larger with no considerable improvement in terms of decreasing the rate of false alarms.

Table 3. 2. The upper control limits at different confidence levels using SPCA.

	Hotelling's T^2		Q-*Statistic*	
Confidence level	Empirical	Using Fisher distribution	Empirical	Using Normal distribution
95%	53.2990	47.4284	3.5695	3.4514
98%	64.6310	51.7815	4.2201	4.0063
99%	88.5681	54.8218	4.6782	4.4147

Based on the fixed thresholds from *Table 3.2.*, the resulting false alarms rate when the SPCA model is fitted on the whole healthy part of the data set is summarized in *Table 3.3.*

Table 3. 3. The rate of false alarms for the different SPCA thresholds.

	Hotelling's T^2		Q-*Statistic*	
Confidence level	Empirical	Using Fisher distribution	Empirical	Using Normal distribution
95%	5.5313 %	8.9412 %	5.0201 %	6.0523 %
98%	2.3312 %	5.6732 %	2.0832 %	2.6078 %
99%	1.2578 %	4.3333 %	1.0376 %	1.5686 %

From this table, it is noticeable that the empirically developed thresholds are resulting in less false alarms. The difference becomes clearer in the case of the T^2-statistic, where the difference in the thresholds and the rate of false alarms is considerable. For the Q-statisic, the thresholds are closer and the rate of false alarms is highly acceptable in both cases. As we aim to reduce the rate of false alarms without deteriorating the sensitivity, the empirical threshold will be adopted throughout this work.

Figure 3.5. shows the Q and T^2 statistics for a fault free process. The thresholds shown at different significance levels are those calculated based on the empirical distribution of the two statistics.

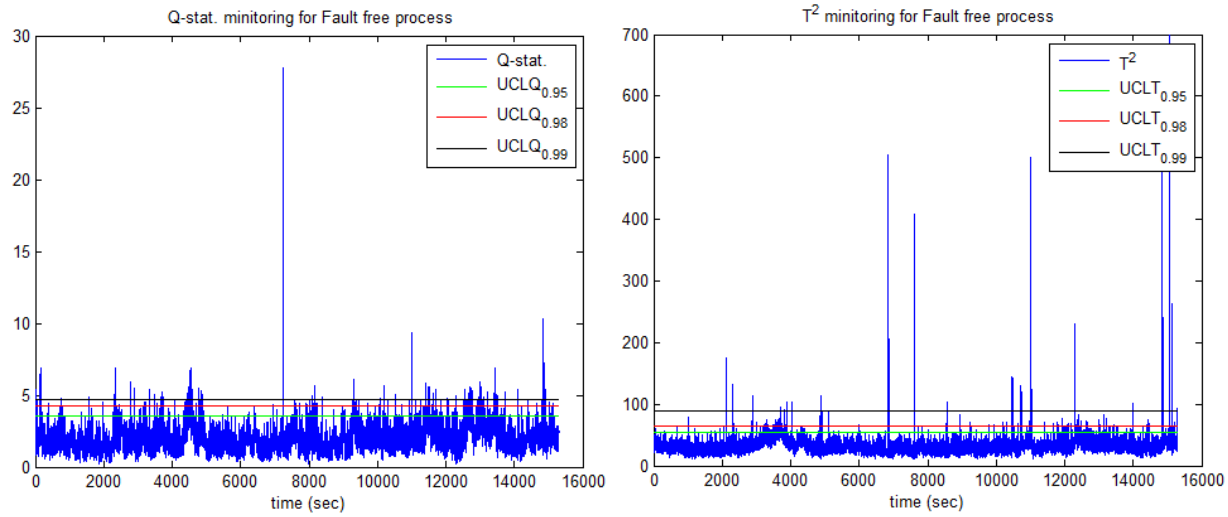

Figure 3. 5. Fault free Process monitoring based on SPCA.

In the monitoring statistics, a considerable event appears around the second 4000. This uprising for the statistics comes from some unmonitored factors that affects the reading of the sensors such as the vibrations within the process. It is also noticeable that the Q-statistic is not highly affected by this uprising.

In order to decide about the presence of any kind of abnormal behavior in the system, a decision rule has to be made in order to tell if the exceedance is due to an outliers or a fault. For simple processes, the Western Electric rules (WER) may serve as a good decision rule. WER conclude that the process is out of control if either:

1. One point plots out of the three sigma control limits.
2. Two out of three consecutive points plot beyond the 2-sigma limit.
3. Four out of five consecutive points plot at or beyond the 1-sigma limit.
4. Eight consecutive points plot on one side of the central line.

These rules can be inspired by assessing the maximum number of consecutive outliers for each statistic under the healthy case. *Table 3.4.* shows the maximum and the average runs detected in the healthy part of the data set. In order to avoid false indications, a decision rule for SPCA-based fault detection is to declare the presence of the fault if the number of successive rums beyond any of the upper control limits exceeds the maximum number of healthy runs. This decision rule, however, imposes a minimum delay in the detection of the fault equals to the maximum number of healthy consecutive runs (in time unit). *Table 3.4.* imposes an insignificant time delay for the case of the Q-statistic based monitoring compared with the delay for the T^2-based monitoring. Deciding whether the delay of 47 seconds is acceptable or not depends on two factors; whether a small amount of false indications is acceptable or not and whether this delay is significant or not compared to the possible improvement in the detection time of the fault. However, even though this way of deciding about faults is able to reduce false indications, it works poorly with intermittent faults once the on time of the fault is less than the indicated maximum run. Furthermore, due to the high concentration of false alarms at the end of the healthy operation time, false indications is unavoidable.

Table 3. 4. The mean and maximum runs at different confidence levels using SPCA and Empirical UCL.

	Hotelling's T^2		Q-*Statistic*	
Confidence level	Mean	Max	Mean	Max
95%	2.2187	47	1.6630	17
98%	2.8585	45	1.4712	8
99%	6.2500	33	1.3784	6

3.2.3 Sensitivity of SPCA for Artificial Faults

After establishing the model and setting a decision rule, the sensitivity of the method has to be tested. To assess the sensitivity, artificial abrupt faults was created at known points of time with specific magnitudes in order to simulate the presence of sensor faults. The targeted sensors are selected such that none of them is within any control loop. In a given measurement type x_i where $X = [x_1, x_2, \dots, x_m]$, a step fault between moment t_0 and t_1 with magnitude of $p\%$ is a deviation of the form:

$$\bar{x}_i(t_0{:}\,t_1) = x_i(t_0{:}\,t_1).\left(1+\frac{p}{100}\right) \tag{3.44}$$

Depending on the contribution of the selected sensor to the used statistic, it can be seen from *Table 3.5.* that the magnitude of the detectable fault present in individual sensors differs from one sensor to another and between one statistic to the other. Some of the open loop sensors are investigated and they confirm the detectability of simple faults by the magnitude of 10% in the worst case.

Table 3. 5. the minimum detectable step deviation (in %) for some Sensors with "A" means the beginning of an abnormal behavior and "C" the appearance of clearly distinguishable fault with the delay in the detection is shown in seconds.

Confidence level		Hotelling's T^2				Q-*Statistic*			
		S# 05	S# 06	S# 21	S# 28	S# 05	S# 06	S# 21	S# 28
95%	A	5.50% (537s)	3% (542s)	4.20% (729s)	0.95% (249s)	6.4% (728s)	0.3% (474s)	6.0% (724s)	1.2% (551s)
	C	5.75% (060s)	3.3% (047s)	4.30% (047s)	1.10% (047s)	7.9% (060s)	0.43% (017s)	8.5% (031s)	1.7% (058s)
98%	A	6.20% (591s)	3.6% (325s)	4.80% (285s)	1.10% (247)	6.5% (798s)	0.3% (568s)	6.0% (724s)	1.2% (554)
	C	7.11% (045s)	3.9% (045s)	4.90% (045s)	1.20% (045s)	8.65% (030s)	0.45% (008s)	9.5% (022s)	1.75% (033s)
99%	A	7.60% (289s)	4.2% (687s)	5.65% (547s)	1.33% (190s)	06% (903s)	0.3% (821s)	6.0% (731s)	1.2% (556)
	C	8.35% (033s)	4.6% (033s)	5.70% (033s)	1.37% (033s)	9.3% (022)	0.47% (006s)	10% (020s)	1.8% (031s)

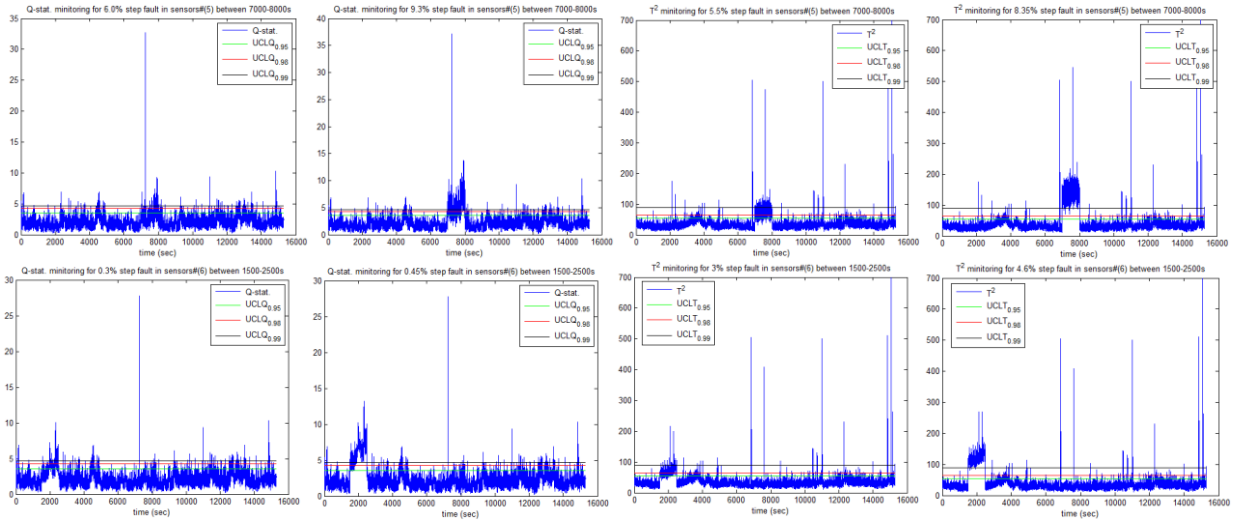

Figure 3. 6. Different appearances of artificial faults in the sensors 5-331_01/PE (upper) and 6-331_01/TE (lower) at different fault magnitudes.

Figure 3.6. shows different simple faults affecting the healthy data for some selected sensors. The upper graphs are collected for the pressure sensor 5-331_01/PE while the other are for the

temperature sensor 6-331_01/TE. For each row, the first two graphs stand for the Q-statistic monitoring with the first showing the beginning of the fault and the second shows the clear appearance of the fault. Similarly, the last two are for the T^2 -based monitoring.

Usually, industrial faults affect multiple sensors which increases the sensitivity. Furthermore, it is clear that the larger the deviation, the smaller the delay in the detection. At large deviations, the detection time tends to the limits of the decision rule.

3.2.4 Real Process Fault Monitoring

The process data consists of healthy and faulty part; no information are available about the real occurrence time of the fault. The previously developed SPCA scheme is applied on the whole data set, including all the healthy section, the intermediate section, and the faulty section.

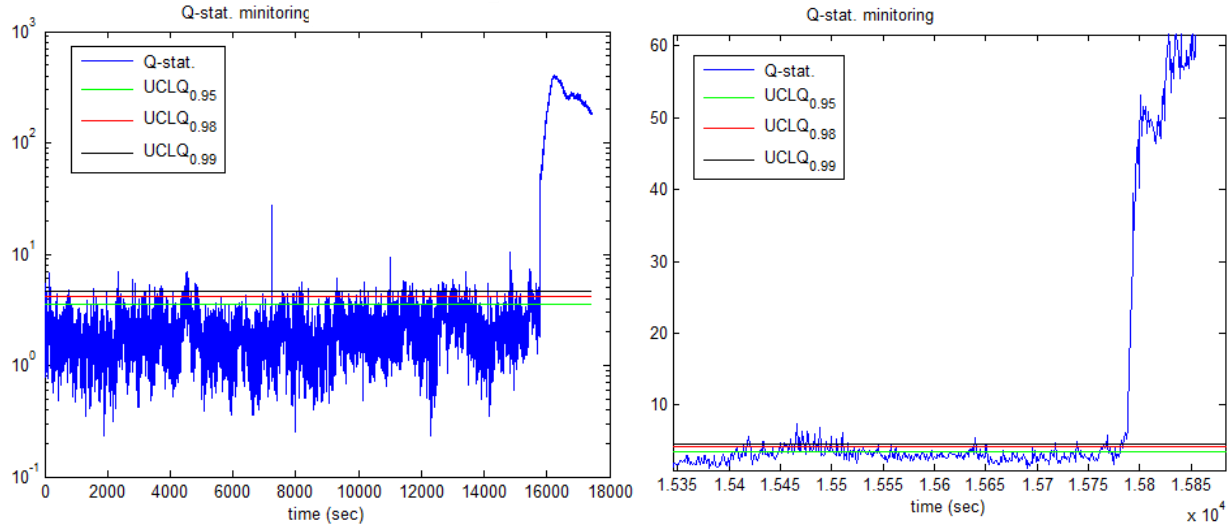

Figure 3. 7. SPCA control chart using SPE. The whole data set in semi-log plot (left), zoom into the fault appearance time with linear plot (right).

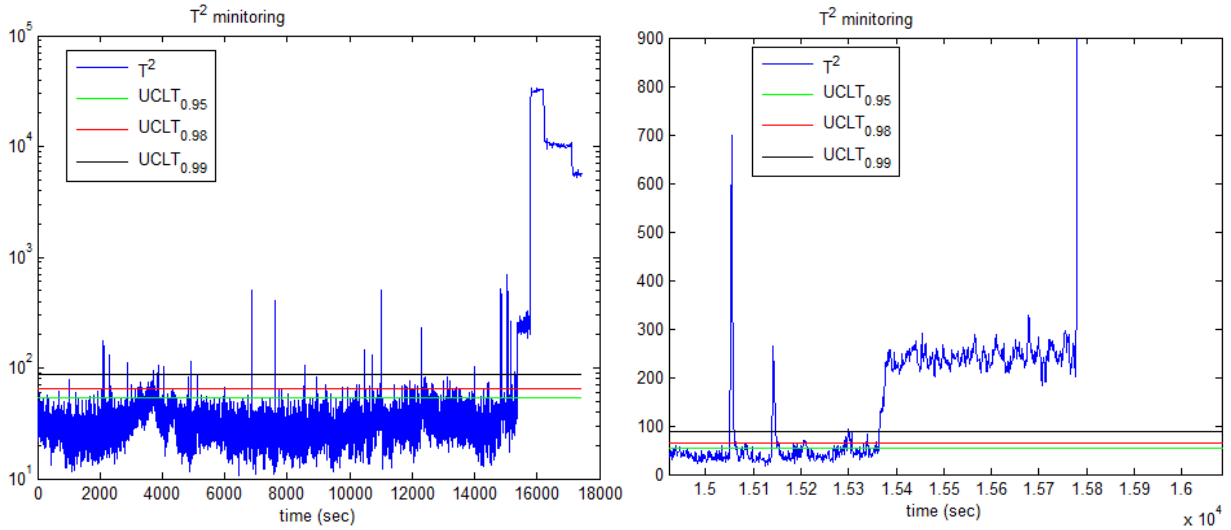

Figure 3. 8. SPCA control chart using Hotelling's T^2. The whole data set in semi-log plot (left), zoom into the fault appearance time with linear plot (right).

Figure 3.7. and *Figure 3.8.* shows the SPCA control charts for SPE and Hotelling's T^2-statistics. The left graph is plotted in semi-log scale due to the large increase of the fault while the right graph is plotted in linear scale and zoomed to the fault appearance time.

The time when the SPCA algorithm at the different confidence levels with the selected decision rules indicates the presence of the fault in the process is shown in *Table 3.6.*

Table 3. 6. Appearance and detection times of the fault based on SPCA.

	Confidence Level	95%	98%	99%
Hotelling's T^2	Appearance Time	15364 (sec)	15364 (sec)	15365 (sec)
	Detection Time	15411 (sec) (03:56:23 AM)	15409 (sec) (03:56:21 AM)	15398 (sec) (03:56:10 AM)
SPE	Appearance Time	15781 (sec)	15784 (sec)	15785 (sec)
	Detection Time	15798 (sec) (04:02:50 AM)	15792 (sec) (04:02:44 AM)	15791 (sec) (04:02:43 AM)

Due to the nature of this fault, the delay in detection reaches its minimum. It is also noticeable that the indication of the fault starts in the Hotelling's T^2-statistic 387 seconds (6' 27") earlier than the SPE. Furthermore, from *Figure 3.8.* one can notice that the Hotelling's T^2 rises suddenly at the second 15783, which is, approximately, the same moment the fault rises in the Q-statistic. This behavior can be explained based on the nature of each statistic. First of all, the T^2-statistic carries the monitoring on the principal components space, i.e., it measures the deviation of the incoming data from the fitted model. The Q-statistic resembles the remaining $m - a$ components, i.e., it provides a measure for the residuals. Therefore, the Q-statistic is more suitable for indicating the presence of the fault since the Hotelling's T^2-statistic can deviate from normal behavior as a result of any changes in the parameters, which might not be a faulty situation. Nevertheless, in *Figure 3.8.*, crossing the T^2-statistic the limits between the moments 15364 and 15783 can be seen as some kind of deviation in the system parameters that precedes the failure of the system detected about 6 minutes and half later.

3.3 Conclusion

In this chapter, an application of SPCA for fault detection in multivariate industrial data set was achieved. The main goals behind this chapter was to investigate the capability of SCPA to detect the presence of abnormal behaviors in an online acquired data. Through this chapter, a static PCA model construction was achieved and the model was used to achieve the monitoring. The sensitivity of SPCA-based fault detection was investigated in terms of step deviation tests. The application of SPCA shows an acceptable detectability and an acceptable behavior. The observed problems with this approach were the high amount of false alarms and the misdetection of intermittent faults with small on duration.

Chapter 4:

CFAR-based process monitoring

4.1 Introduction

A monitoring method based on a Constant False Alarms Rate is proposed in order to eliminate false detections and delays between the rising of the fault and the decision about its existence without deteriorating the detection time. The proposed methodology is tested on SPCA from Chapter three and the results are compared in order to judge the efficiency of the CFAR based monitoring scheme.

4.2 Proposed Methodology

The idea behind CFAR-based monitoring is to develop a threshold that limits the rate of false alarms in each window of time to a preselected value γ, usually $5\%, 2\%$ or 1%. Merely, the monitoring process has to be carried out on a *sliding window* with a fixed size. Once the threshold is validated, the incoming data is subjected to this window –along with the selected threshold. Furthermore, a decision can be made about the state of the process based on the false alarms rate per window. Once FARW exceeds the preselected limit, the presence of fault is indicated.

For the purpose of reducing FAR, a piecewise constant threshold (PWCT) basically developed to limit the rate of false alarms at each time instant of the selected window to a value of $\gamma\%$. After setting the window length to w, the following steps are used to develop the PWCT:

1. Select a descriptive portion of the healthy data set and fit a pre-selected PCA model to this data set.
2. Compute the T^2 and Q- statistics for each window.
3. Based on the window frame indices, at each index t_i, develop an UCL at a confidence level of $1-\gamma/100$ for the random variable formed out of the selected statistic values at the window index t_i.

Connecting the different points representing the upper control limits at each index results in a PWCT that limits the FAR at each index to a value of $\gamma\%$. Subjecting the T^2 and Q- statistics to this threshold limits the rate of false alarms per window to be near $\gamma\%$.

Figure 4.1. illustrates the procedure, the lines labelled *"signal i"* represents a selected fault indicator (T^2, Q, …etc.) for three successive windows. At window index t_i, the values of the fault indicators are treated as a random variable, the empirical distribution of the random variable is assessed to determine the $\frac{\gamma}{100}$ percentile (the red dots). Then the threshold (red line) is constructed. Within the scope of this work, the used threshold is fixed and no adaptation is considered.

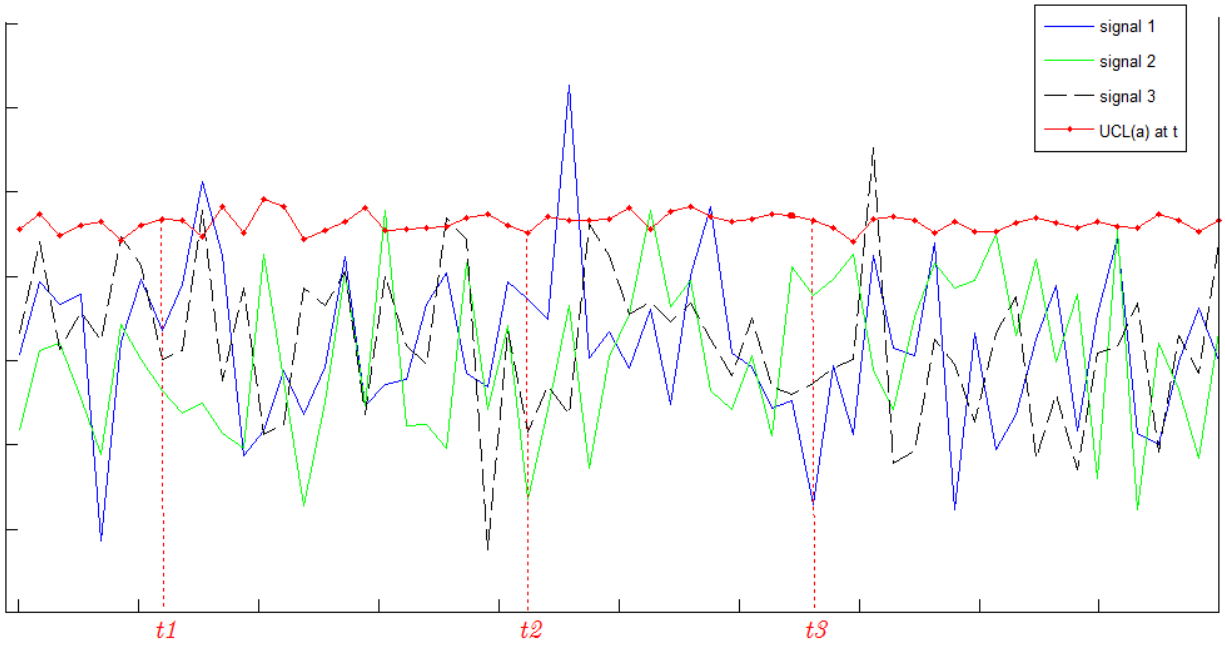

Figure 4. 1. Illustration for the process of building the CFAR threshold.

4.2.1 Selection of the window size

Carrying out a monitoring based on a sliding window poses a major question about the selection of the size of this window. Based on the previously stated description, it can be seen that if a detectable fault occurs between t_0 and t_1, when the monitoring is carried using a window of length w, the maximum delay in the detection of the fault would be:

$$\Delta t_{max} = w.\frac{\gamma}{100}\ (in\ time\ unit) \tag{4.1}$$

Where γ is the constant false alarms rate.
It is noticeable that the delay in detection is proportional to the window size. The larger the window, the longer the delay.

Furthermore, due to the effect of the non-stationarity, a small sized window may result in an excessive rate of false alarms per window, which pushes the threshold to higher values. This last will cause a tremendous deterioration in the sensitivity of the method. A tradeoff between the sensitivity and the delay-in-detection (DID) can be achieved by properly selecting the size of the sliding window.

Based on the available data set, windows with different sizes have been slide over the data set, for each window size, the means and the variances in each window are recorded and treated as random variables. This step results in two $(n-w)\times m$ matrices one contains the mean and the other contains the variance for each window size w_i. For each window size, and for each sensor, the standard deviations of the mean and the variance are shown in *Figure 4.2* and *Figure 4.3.* respectively.

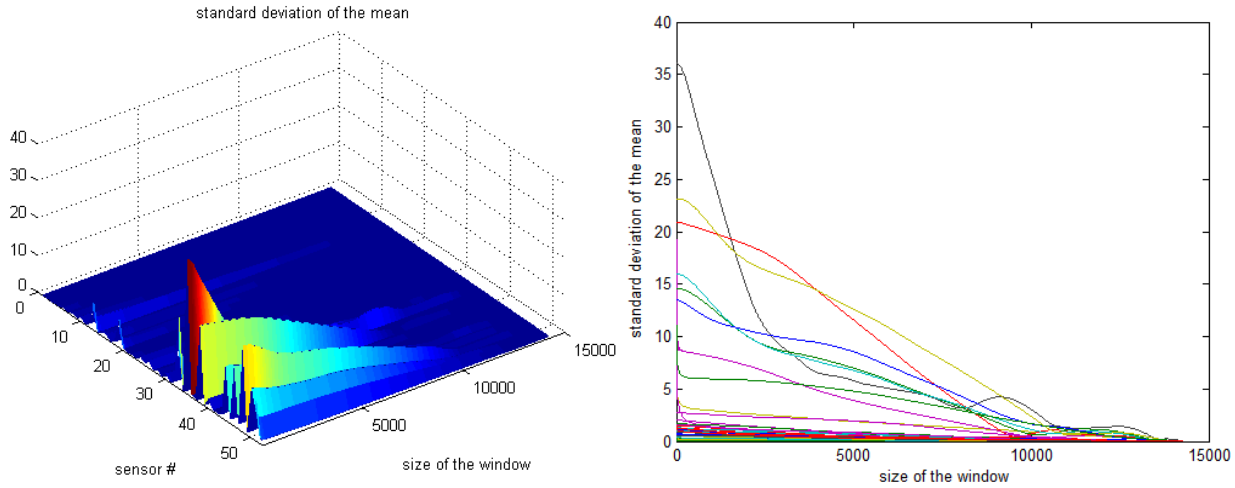

Figure 4. 2. Standard deviation of the mean of the healthy data vs. the window size. a 3D plot (left) and 2D plot where each line represents one sensor (right).

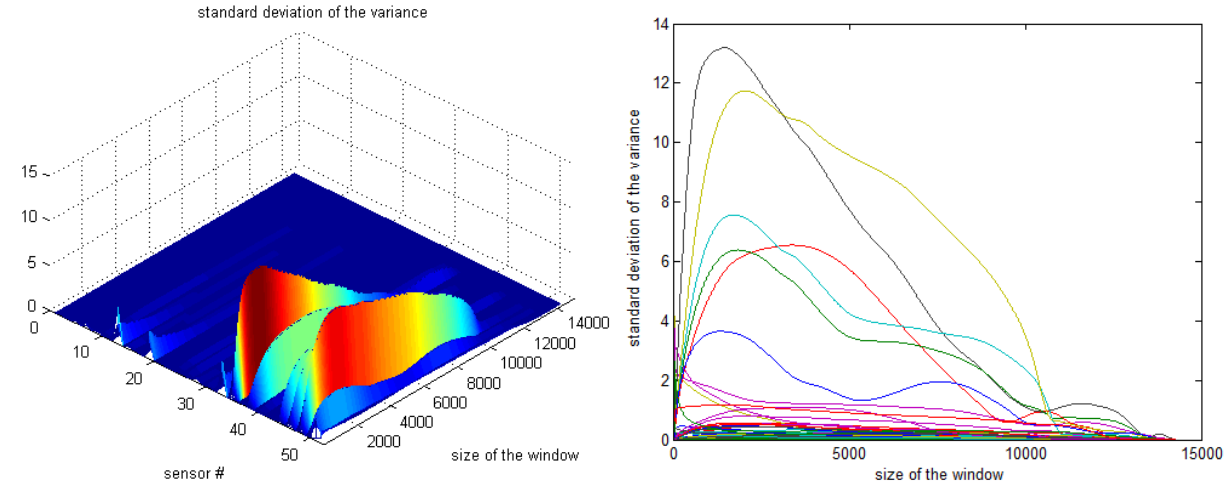

Figure 4. 3. Standard deviation of the variance of the healthy data vs. the window size. a 3D plot (left) and 2D plot where each line represents one sensor (right).

The range of the maximum variability for the means is between $w = 1$ and $w = 3140$ which is not an advisable range. Furthermore, the variances exhibit a maximum of variance between $w = 1300$ and $w = 3400$. Thus, a reasonable choice for the window size that would be used for the purpose of testing the CFAR monitoring scheme through this work is $w = 5000$.

4.3 Development of UCL

Using a window of size $w = 5000$ along with the SPCA model developed in *Chapter 3*, and by applying the procedure described earlier, the PWCTs shown in *Figure 4.4.* (for T^2 (left) and Q- statistics) are developed for different confidence limits. It can be seen that these thresholds are, in average, close to those of *Table 3.2.* In general, these thresholds are not exhibiting a very large variability except for the threshold of the Hotelling's T^2-statistic in the 99% case.

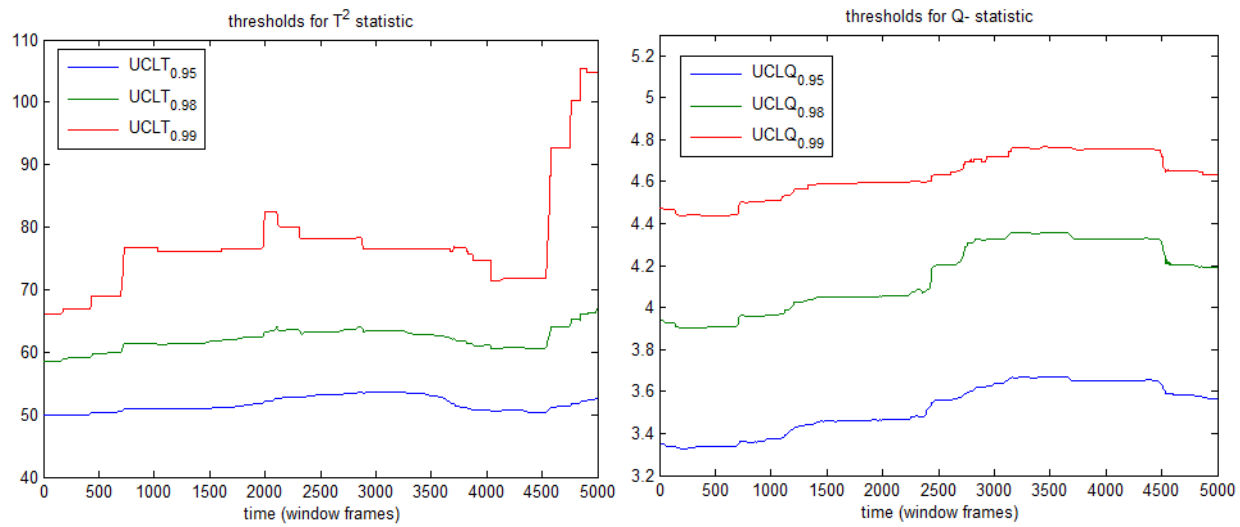

Figure 4. 4. PWCTs for T^2-statistic (left) and Q-statistic (right) at different confidence levels.

Subjected to these thresholds, the rate of false alarms per window was evaluated while monitoring the healthy data. The results are recorded in *Figure 4.5.*

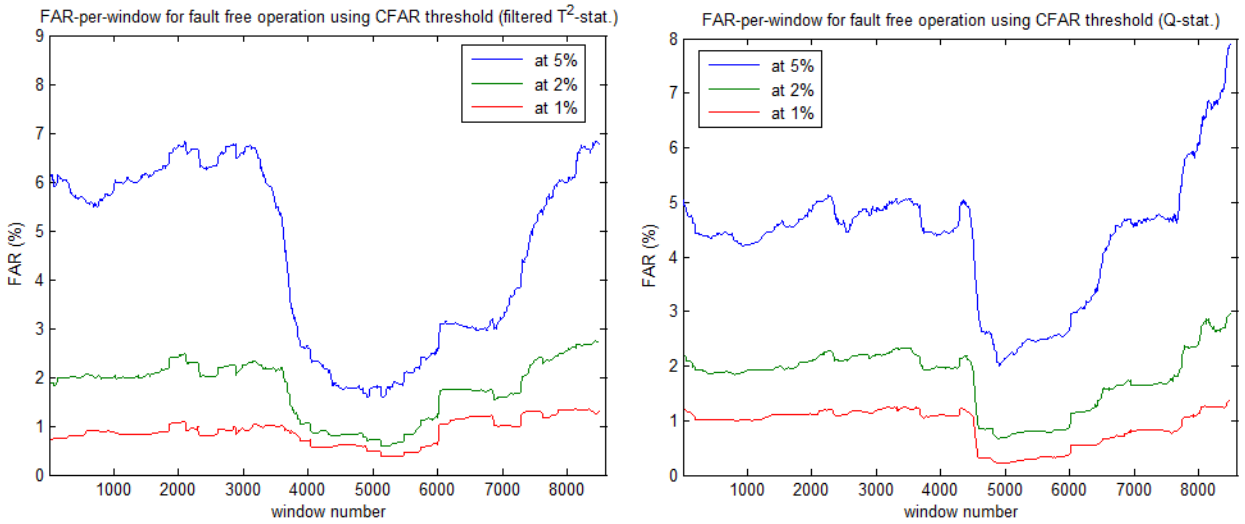

Figure 4. 5. FARW using the PWCT for different intended false alarms rates (or confidence levels).

From these figures, we notice that the rate of false alarms per window is, in general, within the expectations. By assessing the empirical distributions of the FARs from *Figure 4.5.*, the following upper control limits can be derived at different confidence levels (*Table 4.1*).
Using these thresholds provides no false detections when tested on the whole healthy data set. Nevertheless, for CFAR monitoring, a large FAR threshold is not desirable. Mainly, the expected values for the false alarms UCLs are 5%, 2% and 1% respectively. The observed increase in the UCLs when the whole healthy data set is cast to the PWCT is quite expected. Thus, to keep the large CFAR threshold from deteriorating the detection time, filtering the Hotelling's T^2 and the SPE statistics might be a reasonable choice.

Table 4. 1. UCL for the rate of false alarms (%) with the maximum DID (in seconds) imposed by each one at w=5000.

	T^2			Q		
FAR (%)	$ci = 95\%$	$ci = 98\%$	$ci = 99\%$	$ci = 95\%$	$ci = 98\%$	$ci = 99\%$
5	6.86 (343s)	8.50 (425s)	8.80 (440s)	8.30 (415s)	8.40 (420s)	8.46 (423s)
2	2.98 (149s)	4.04 (202s)	4.18 (209s)	3.10 (155s)	3.36 (168s)	3.44 (172s)
1	1.36 (068s)	2.24 (112s)	2.40 (120s)	1.48 (074s)	1.78 (089s)	1.78 (089s)

4.4 CFAR monitoring for filtered statistics

For this part, a simple filtering is tested for the sake of decreasing the rate of false alarms per window without highly affecting the signal. For instance, the use of median filtering is proposed due to the simplicity and the acceptable SNR. The median filter is a nonlinear statistical digital filtering technique. The main idea of the median filter is to run through the data replacing each entry with the median of the neighboring entries within preselected sliding window. A simple 1-D filtering is applied on the two statistics. Furthermore, the size of the filter window does highly affect the Signal-to-Noise Ratio (SNR) and the Mean Squared Error (MSE) of the filter. Finally, it is worthy to note that in some applications, the odd and even window sizes may have different effects due to the nature of the median as a statistic. In order to determine an appropriate size for the filter, the SNR and MSE for the T^2 and Q-statistics as function of the window size are respectively shown in *Figure 4.6.*

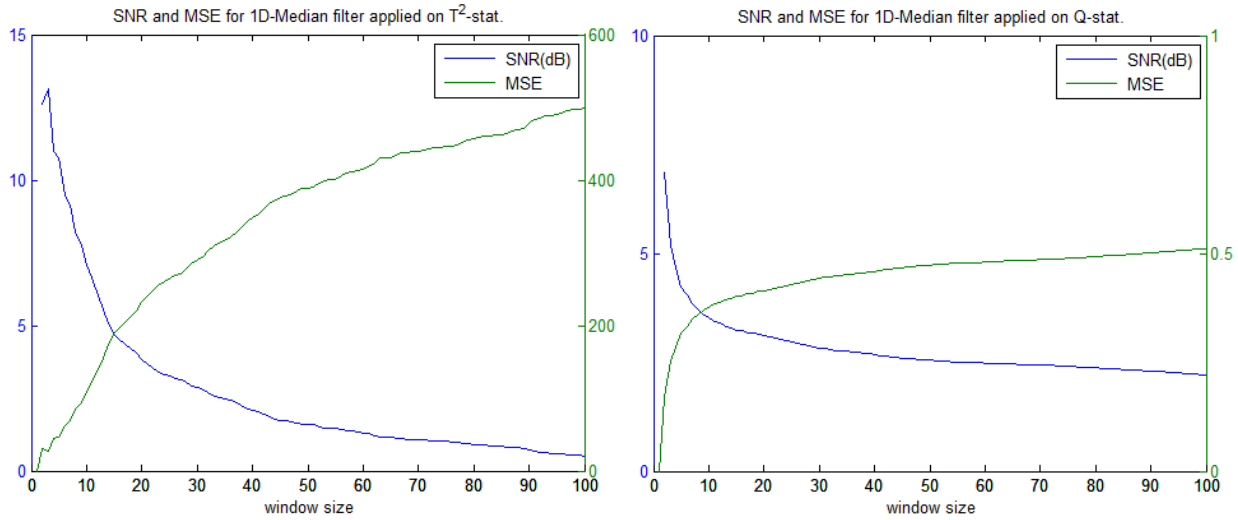

Figure 4. 6. MSE and SNR vs. window size using Median filter. For T^2 (left) and Q (right).

First of all, for the T^2 statistic, a value of $w_m = 9$ provides $SNR = 7.091dB$ and $MSE = 97.4$ which seems to be adequate as a compromise between the MSE and SNR. For the Q-statistic, $w_m = 4$ seems to be reasonable. For this window, we have recorded $SNR = 4.671dB$ and $MSEQ = 0.2861$. these two SNRs are acceptable.

Filtering the Hotelling's T^2 and the SPE statistics with the pre-specified median filters results in the next graphs, where it can be seen that the fault indicators are still preserving their shapes compare to those of *Figure 3.5.*

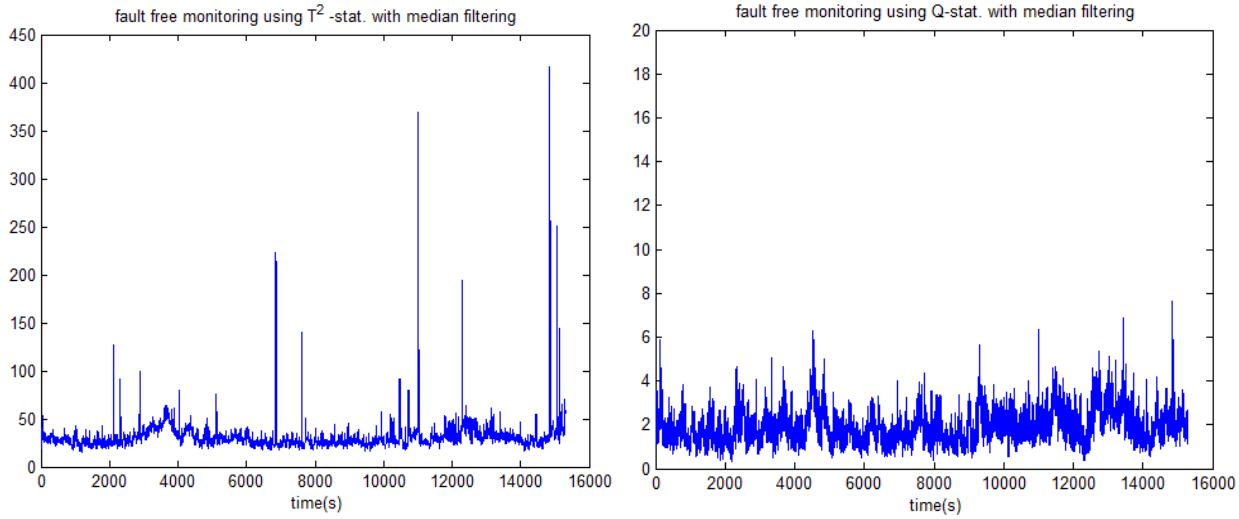

Figure 4. 7. Filtered $T^2(w_m = 9)$ and $Q(w_m = 4)$ -statistics of SPCA monitoring.

Monitoring these statistics with the PWCTs developed earlier results in the graphs shown in *Figure 4.8.* where the false alarms rate per window is considerably smaller than the values observed in *Figure 4.5.* For the moment, the following steps are applied:

1. Calculate the T^2 and SPE.
2. Apply a median filter for the incoming data along with the previous ones within the filter window range and update the values.
3. Cast the whole window (5000 sample) to the Piecewise constant thresholds of *Figure 4.4.* and compute the rate of false alarms in the current window.
4. If the rate of false alarms exceeds the constant thresholds (5%, 2% and 1%), the presence of a fault is indicated.

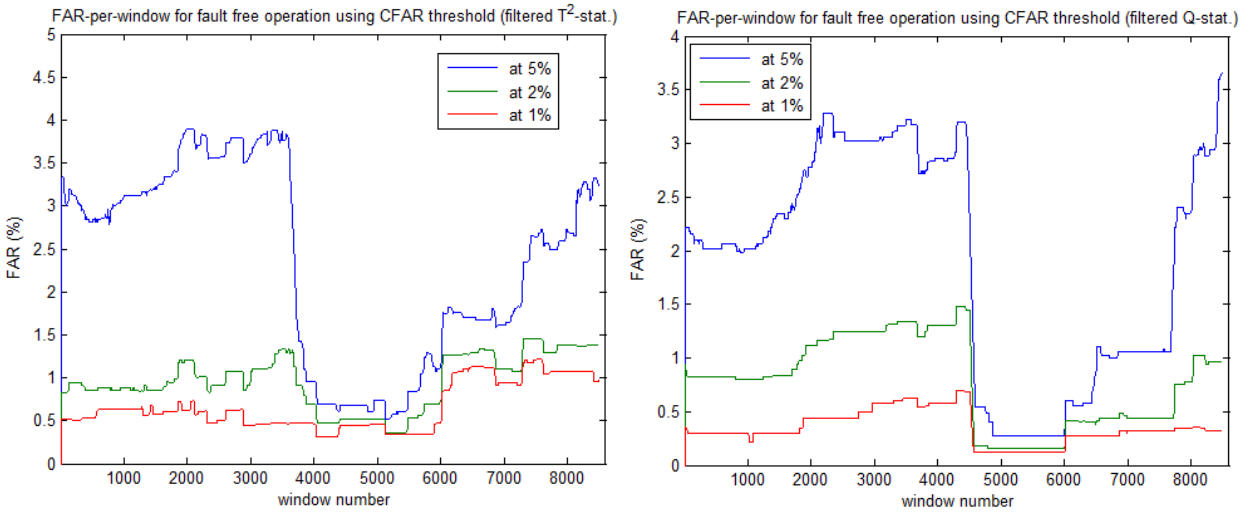

Figure 4. 8. FARW using the PWCT on filtered statistics for different intended false alarms rates (or confidence levels).

Finally, this approach was tested on the whole data set in order to assess its ability to detect the presence of real process fault. *Figure 4.9.* and *Figure 4.10.* shows the monitoring based on the T^2-statistic and Q-statistic respectively; for the three different false alarm rate limits. The graphs are more systematic and provide a clear monitoring with no false detections.

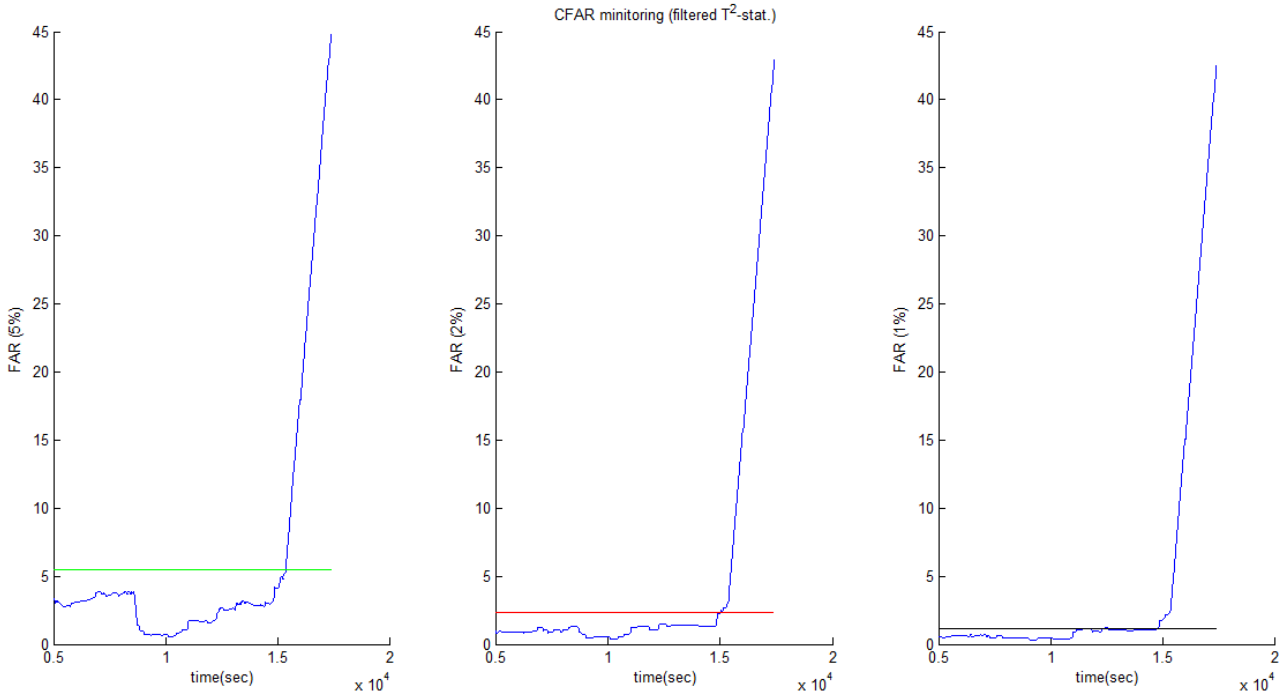

Figure 4. 9. CFAR based monitoring for filtered T^2-statistic.

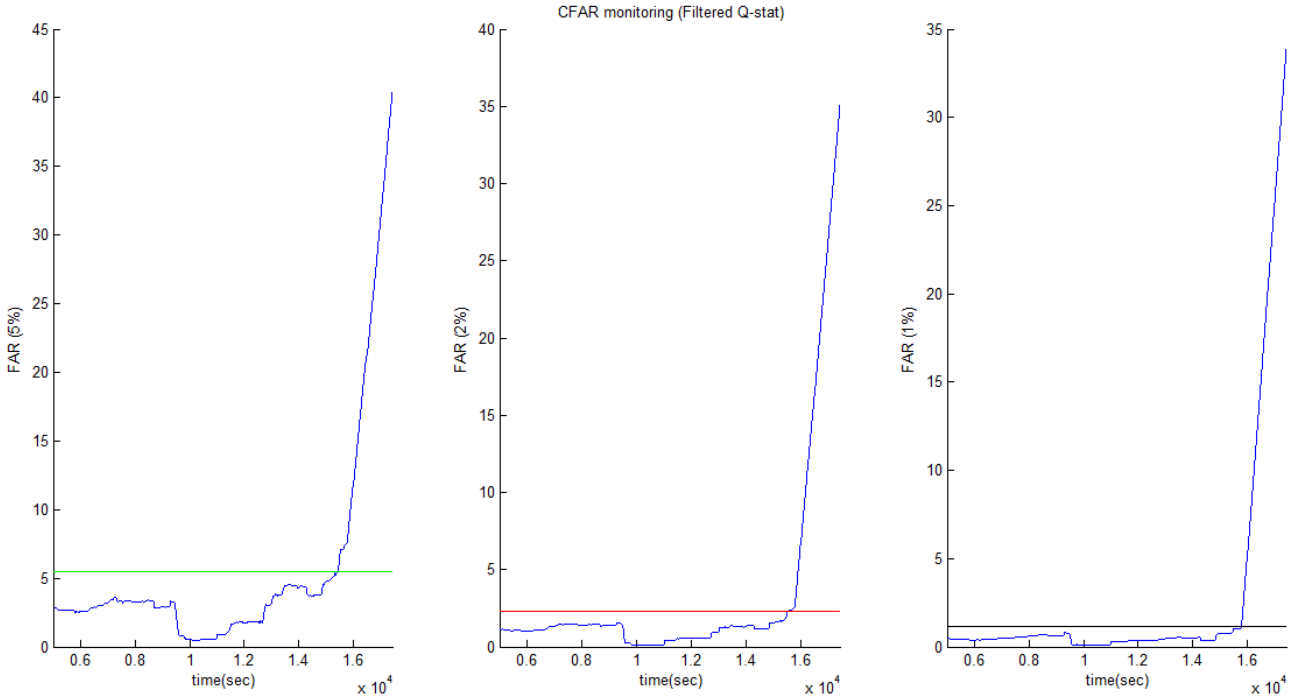

Figure 4. 10. CFAR based monitoring for filtered Q-statistic.

Table 4.2. provides a summary for the detection time of the fault. It can be noticed that the fault was detected a bit earlier than the case of simple SPCA, this is due to the facts:

1. The decision rule in the CFAR monitoring has no delay time, i.e., once the rate of false alarms exceeds the limits, a fault is declared.
2. Just before the appearance of the fault, a strangely high amount of false alarms appeared in the T^2-statistic. These false alarms are counted when the fault was declared by the T^2-statistic. Disputing the fact that a small amount of old false alarms within the window are helpful in terms of decreasing the fault appearance delay; a large amount can be misleading.

Table 4. 2. Detection time of the fault using CFAR monitoring (with median filter).

	FAR limits	5%	2%	1%
T^2	Detection Time	15376 (sec) (03:55:48 AM)	15273 (sec) (03:54:05 AM)	15140 (sec) (03:51:52 AM)
SPE	Detection Time	15451 (sec) (03:57:02 AM)	15686 (sec) (04:00:58 AM)	15790 (sec) (04:02:42 AM)

4.5 CFAR monitoring using forgetting factors

Merely, a large window size is not advisable, since the monitoring based on the RFA per window in its simple form suffers from the persistency of the faults, i.e., once a detectable fault arises, it will impose an increase in the rate of false alarms in the current window. However, when the fault is removed from the system, say at time instant t_1, the fault will continue to impose excessive amount of false alarms until a certain time instant between $t_0 + w$ and $t_1 + w$. In this case, the fault will not disappear and if another fault occurs before about w windows passes from the previous fault, it may be missed and thought of as a remaining of the previous fault. Two simple solutions may be used to cope with this problem in order to increase the efficiency of the CFAR monitoring scheme. The first consists of clearing the window after the occurrence of the fault. The second method consists of introducing a forgetting factor in order to decrease the effect of the old data in the window on the decision. Usually, the selection of the forgetting factor $0 < \eta < 1$ depends on the size of the window and the amount of required suppression for the old values. For $w = 5000$, a forgetting factor 0.999 causes the oldest value to be multiplied by 0.00672. In addition to reducing the persistency, the use of forgetting factor is able to decrease the FARW, which replaces the functionality of the filter.

Figure 4.11. shows the weighting envelope for $\eta = 0.9998$. where the weighting factor at each instant is calculated using the expression:

$$\eta_i = \eta^{w-i}, \qquad i = 1,2,\dots,w. \tag{4.2}$$

This envelope can be seen as an attenuation curve where the statistics at every window has to be multiplied with; before the window content is subjected to the PWCTs and the FAR is evaluated. The old values of the statistics are more attenuated than the new ones, which makes the outliers (or faults) dies out at the far end of the window. If the fault is still in progress, the incoming faulty signals will keep the FAR above the limits. If the fault is removed, the fault will disappear from the control chart faster and less persistency is obtained.

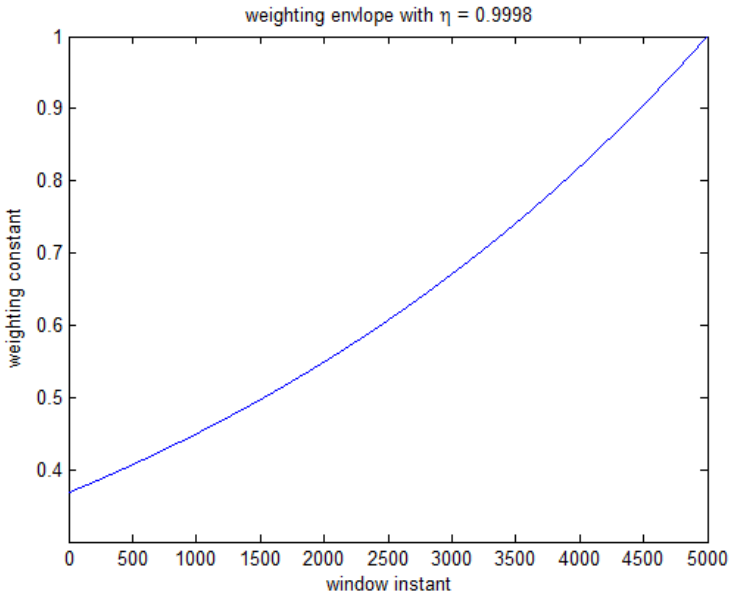

Figure 4. 11. The weighting envelope for $\eta = 0.9998$.

Figure 4.12. and *Figure 4.13.* shows the CFAR monitoring of the whole data set when the previous forgetting envelope is used on the Hotelling's T^2 and the Q- statistics. For the healthy part of the data set, no outlier was detected. Furthermore, the rate of false alarms was reduced without any need for filtering the T^2 and Q-statistics.

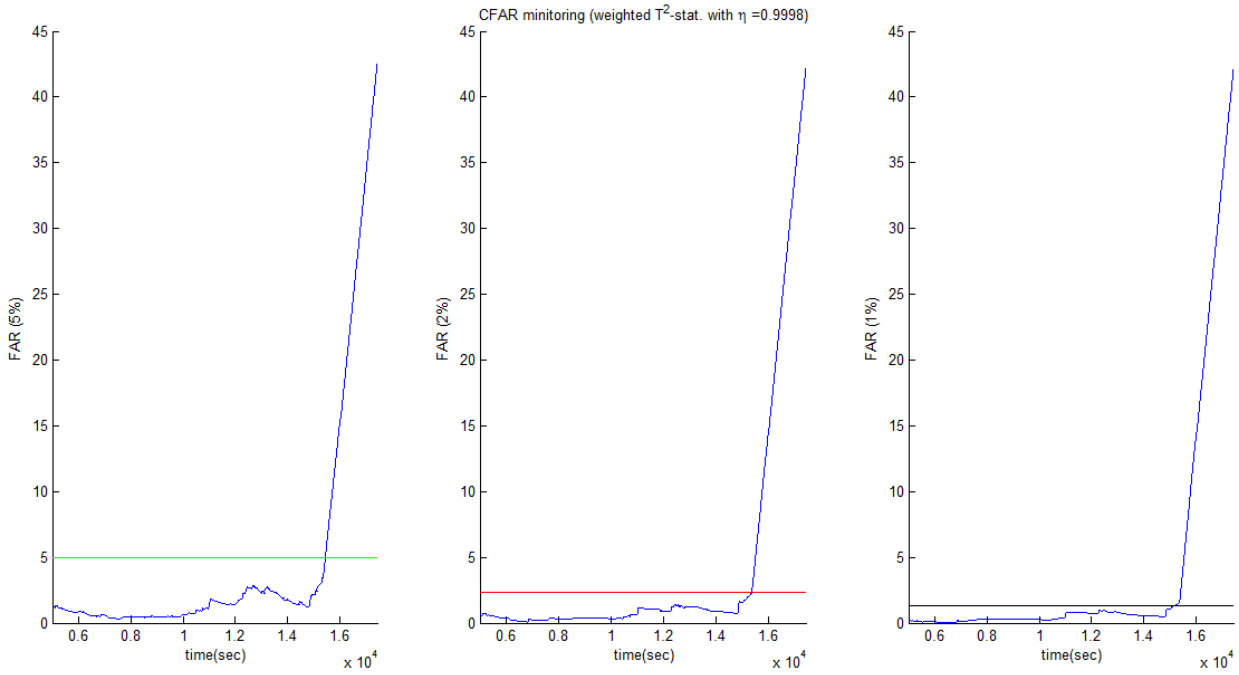

Figure 4. 12. CFAR based monitoring for weighted T^2-statistic.

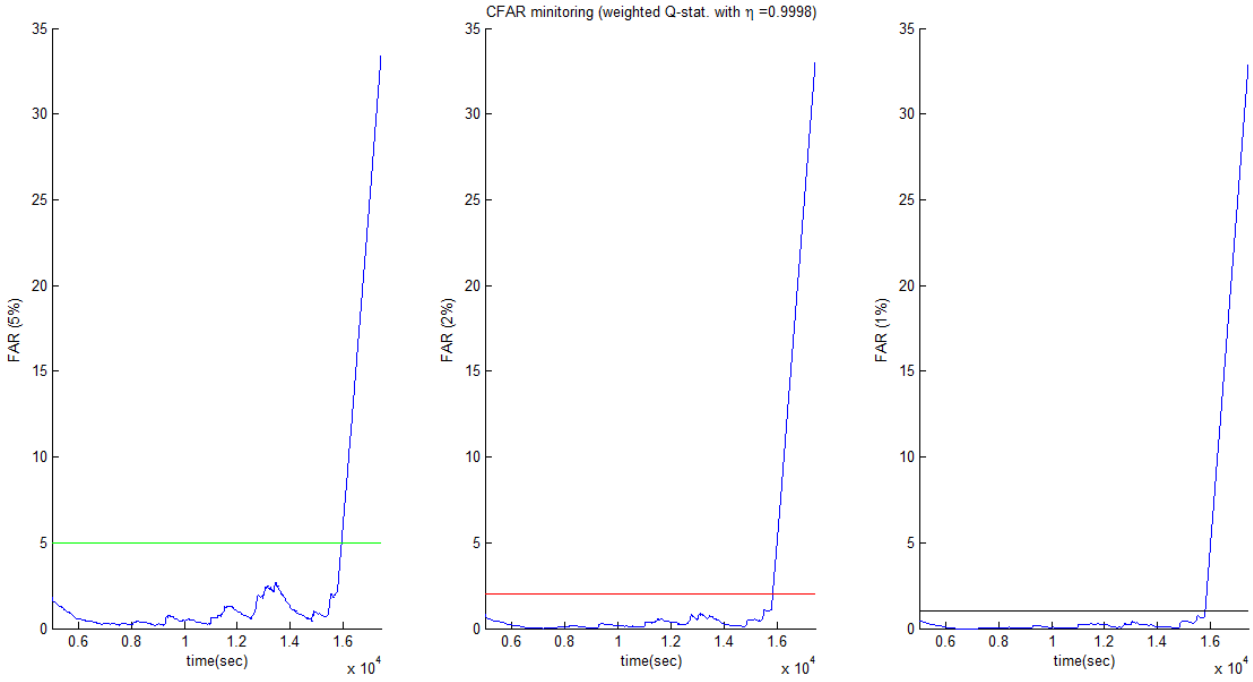

Figure 4. 13. CFAR based monitoring for weighted Q-statistic.

From *Table 4.3.*, it can be noticed that the problem of detecting the fault before its appearance in the different statistics is reduced. The detection time provided is extremely close to the appearance time of the statistics and, basically, it remains far away from the maximum expected delay of $w.\frac{\gamma}{100}$ (seconds) as a result of the high magnitude of the fault.

Table 4. 3. Detection time of the fault using CFAR monitoring (with $\eta = 0.9998$).

	FAR limits	5%	2%	1%
T^2	Detection Time	15441 (sec) (03:56:53 AM)	15367 (sec) (03:55:39 AM)	15293 (sec) (03:54:25 AM)
SPE	Detection Time	15940 (sec) (04:05:12 AM)	15828 (sec) (04:03:20 AM)	15802 (sec) (04:02:54 AM)

4.6 Sensitivity of CFAR-based monitoring

In this part, the sensitivity of the two CFAR-based approaches discussed earlier is tested under artificial faults. Basically, the test is carried on the same sensors as in *Table 3.5.* in order to provide a mean of comparison. First, the test is carried using step faults with appropriate magnitudes. Then the sensitivity of CFAR-based monitoring to detect the presence of intermittent faults is assessed.

4.6.1 Test 1

A step fault was simulated in the temperature sensor 28-341_04/TE with a magnitude of 1.3% between time instances $t_0 = 7000s$ and $t_1 = 8000s$. From *Table 3.5.*, this deviation is clearly detectable by the T^2-statistic at 95% and 98%.

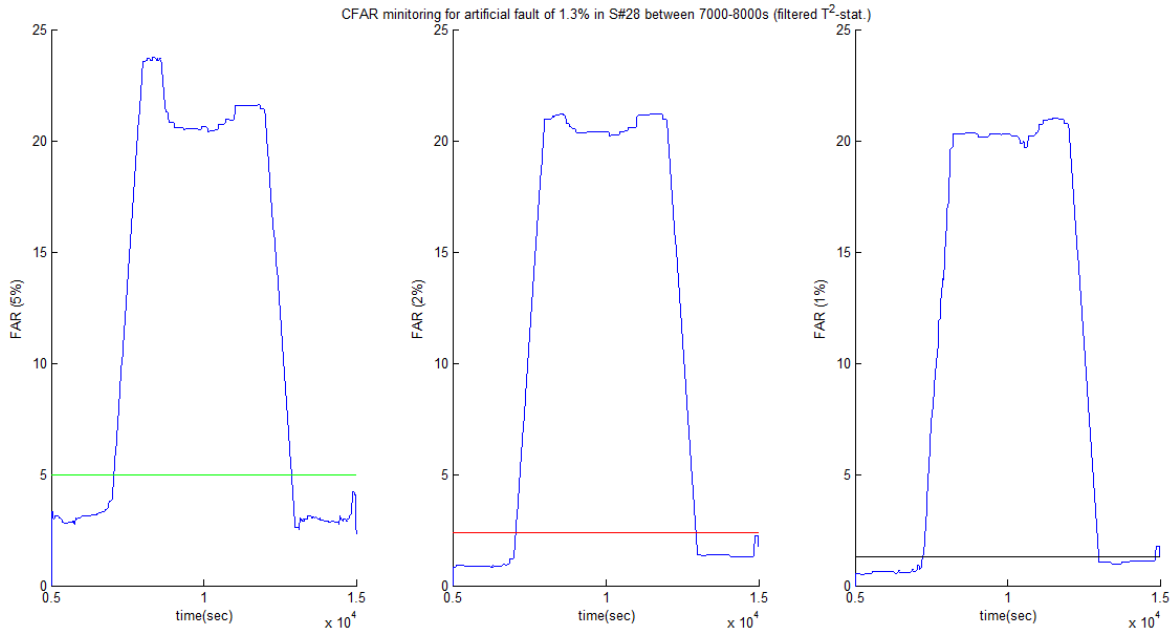

Figure 4. 14. Simulation of step deviation of 1.3% in S#28 in the time interval [7000~8000s] using CFAR with filtered T^2- statistic.

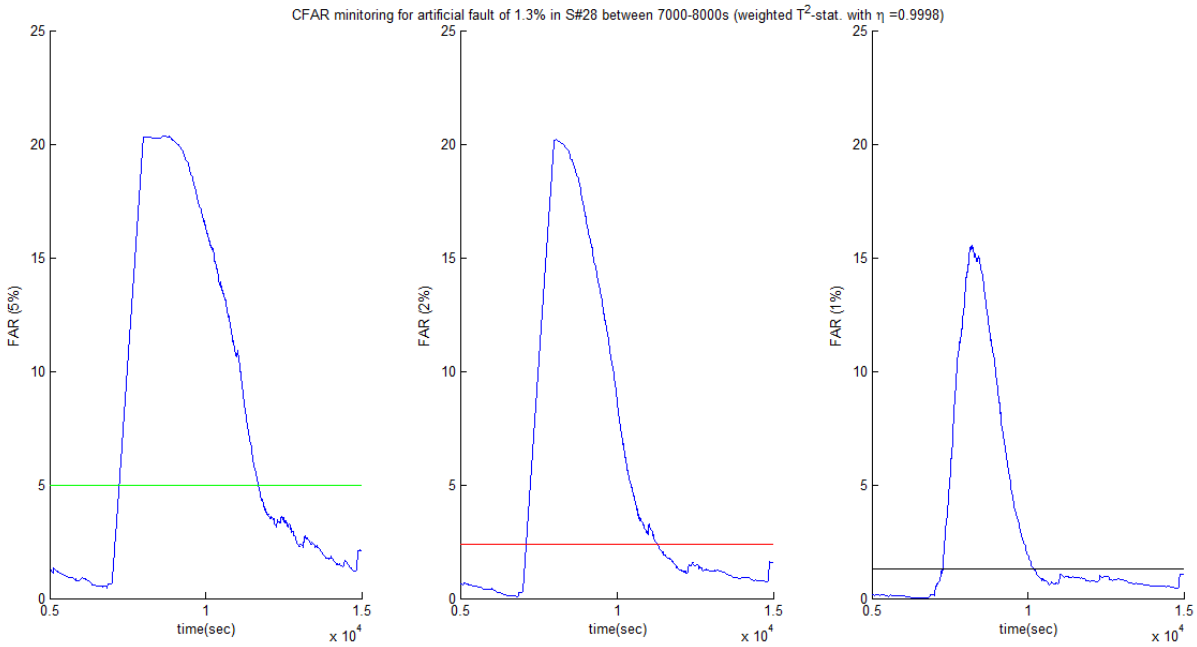

Figure 4. 15. Simulation of step deviation of 1.3% in S#28 in the time interval [7000~8000s] using CFAR with weighted T^2- statistics.

The problem of the persistency of the fault is clearly appearing in the previous two figures. Even with a fault ending at $t_1 = 8000s$, the fault dies out at about 5000 seconds later. From *Figure 4.15.*, the effect of the forgetting factor appears as a decay starting at $t = 9013s$ in the 5% monitoring and at $t = 8004s$ in all of the three cases and causes the fault to die out faster as it can be seen from *Table 4.4.*

Table 4. 4. Detection and Dying times of 1.3% step deviation in S#28 using CFAR monitoring.

		Using Median Filter			*Using Forgetting factor*		
	FAR limits	5%	2%	1%	5%	2%	1%
T^2	Detection Time	7055	7041	7186	7231	7088	7214
	Dying Time	12880	12970	12990	11690	11280	10180

4.6.2 Test 2

A step deviation of 0.45% is simulated in the temperature sensor 6-331_01/TE between time instances $t_0 = 7000s$ and $t_1 = 8000s$. Through *Figure 4.16*, *Figure 4.17* and *Table 4.5* along with the results of Test 1, three conclusions can be drawn:

1. CFAR-based monitoring is able to provide a full monitoring with no false indications.
2. The detection time of the anomalies is not deteriorated.
3. The persistency of the faults causes a considerable drawback.

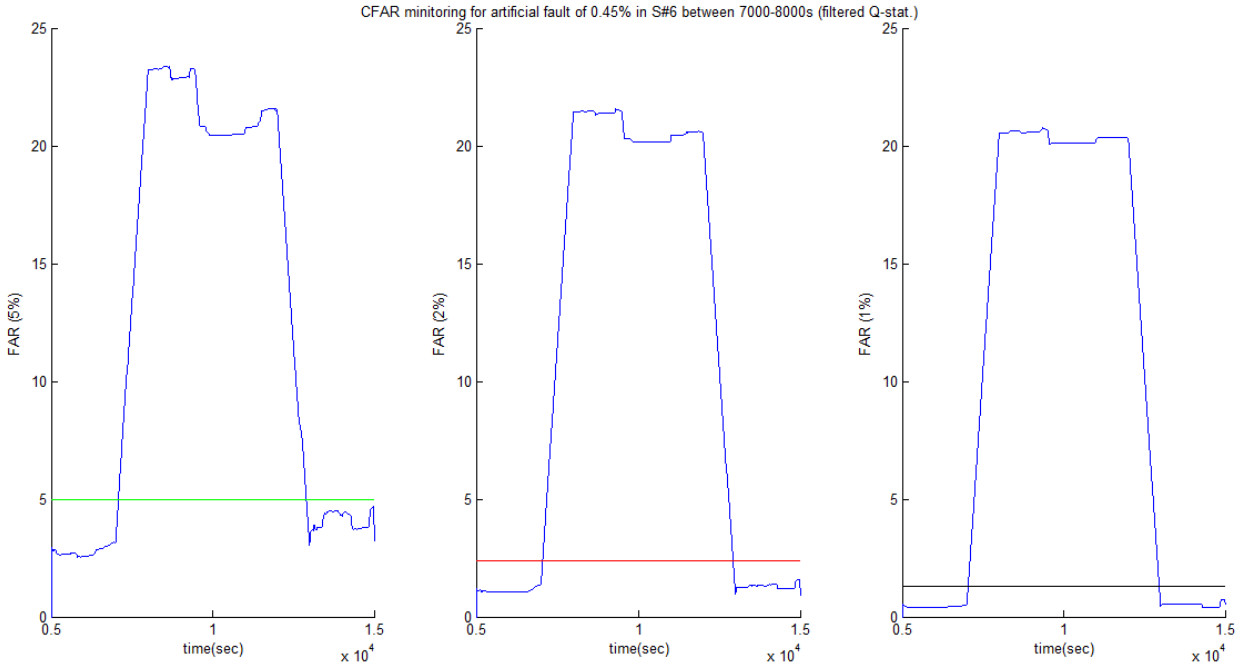

Figure 4. 16. Simulation of step deviation of 0.45% in S#06 in the time interval [7000~8000s] using CFAR with filtered Q- statistics.

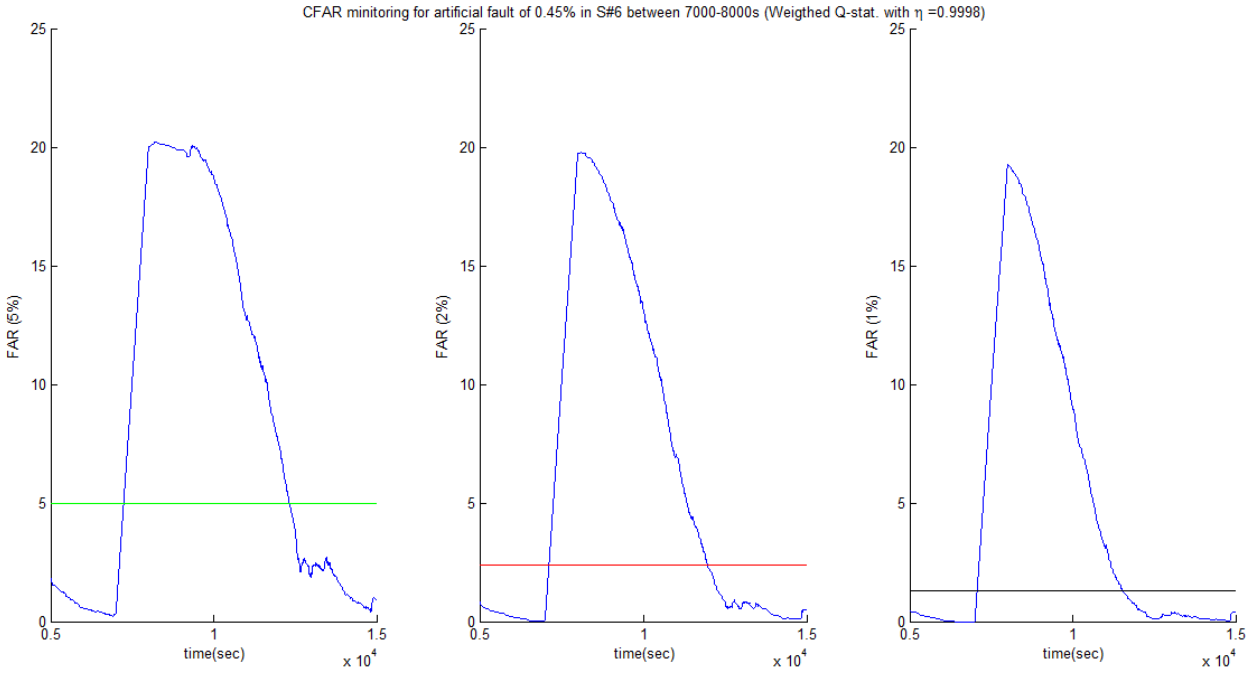

Figure 4. 17. Simulation of step deviation of 0.45% in S#06 in the time interval [7000~8000s] using CFAR with weighted Q- statistics.

Again, it can be noticed that the weighted CFAR monitoring is performing better in terms of reducing the persistency in the detection. While CFAR based on filtered signals is better in terms of the detection time, with no significant differences.

Table 4. 5. Detection and Dying times of 0.45% step deviation in S#06 using CFAR monitoring.

		Using Median Filter			*Using Forgetting factor*		
	FAR limits	5%	2%	1%	5%	2%	1%
SPE	Detection Time	7082	7034	7027	7239	7099	7050
	Dying Time	12900	12950	12970	12320	11970	11530

4.6.3 Test 3

Intermittent faults are faults with a time nature that makes them hard to detect. In most cases, they are interpreted as outliers for a sufficiently long time before they are indicated as faults. For instance, based on the decision rule used in SPCA, intermittent faults with deviation durations less than the maximum rums indicated in *Table 3.4.* will be totally missed. In practice, intermittent faults may rise as a result of bad wirings and connections, in this case, the deviation time is significantly, which increases the difficulty of detection. However, the idea of monitoring based on the false alarms rate makes the nature of the fault insignificant, i.e., whatever the time behavior of the fault, the detection is guaranteed once the magnitude of the fault is sufficiently large and a sufficient number of anomalies appeared.

The form of the intermittent fault used to test the detection is a regular on-off deviation with the on time equals 40 seconds and the off time equals 10 seconds.

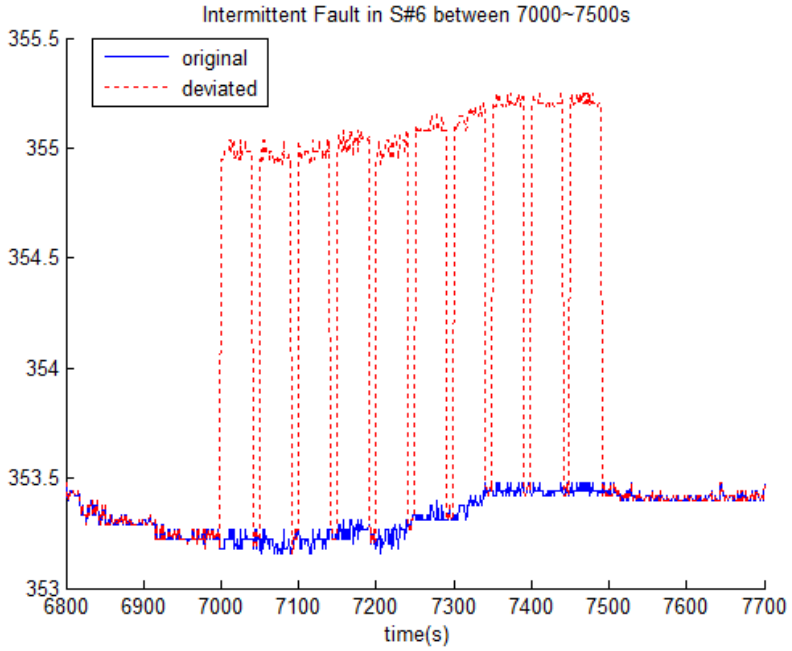

Figure 4. 18. Simulation of Intermittent deviation in S#06 in the time interval.

Testing the previous intermission applied on the temperature sensor 6-331_01/TE under a CFAR monitoring with weighted statistics gives the form of *Figure 4.19.*, the intermittent fault is successfully detected with an acceptable delay.

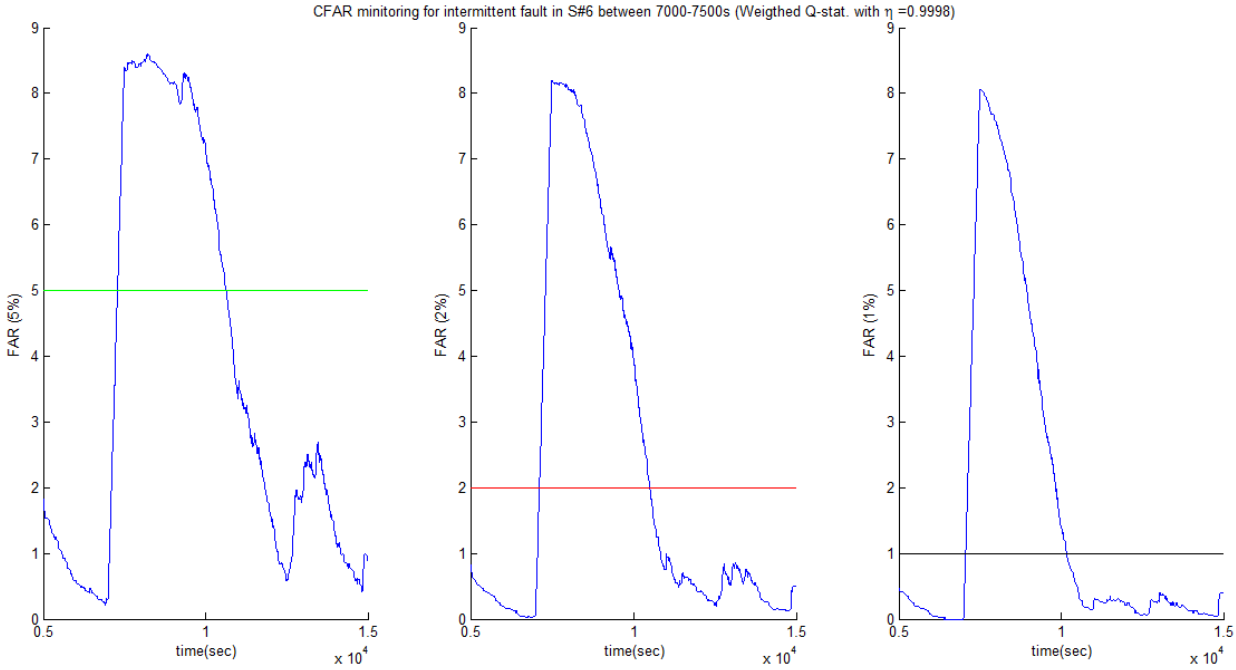

Figure 4. 19. Simulation of intermittent deviation in S#06 in the time interval [7000~7500s] using CFAR with weighted Q- statistics

Table 4. 6. Detection and Dying times of intermittent deviation in S#06 between [7000~7500] using CFAR monitoring.

		Using Forgetting factor		
	FAR limits	5%	2%	1%
SPE	Detection Time	7282	7117	7059
	Dying Time	10620	10500	10160

Table 4.6. shows that the detection time of the intermittent fault ranges from about 1 minute to a maximum of 4 minutes and 42 seconds, depending on the on and off times.

4.7 Elimination of Fault persistency

The problem of persistency can be solved after reducing the rate of false alarms using weighting envelope. A solution as simple as resetting the window after the fault is removed is able to solve the fault persistency problem. Testing this approach with weighting envelope and window resetting after removing the fault is tested for:

1. Single fault created at $t = 6000$.
2. Intermittent fault created at $t = 8000$.
3. Multiple fault at $t = 13000$.

With each lasts for 1000 seconds. The following figure summarizes the final results. Where the fault persistency is totally removed.

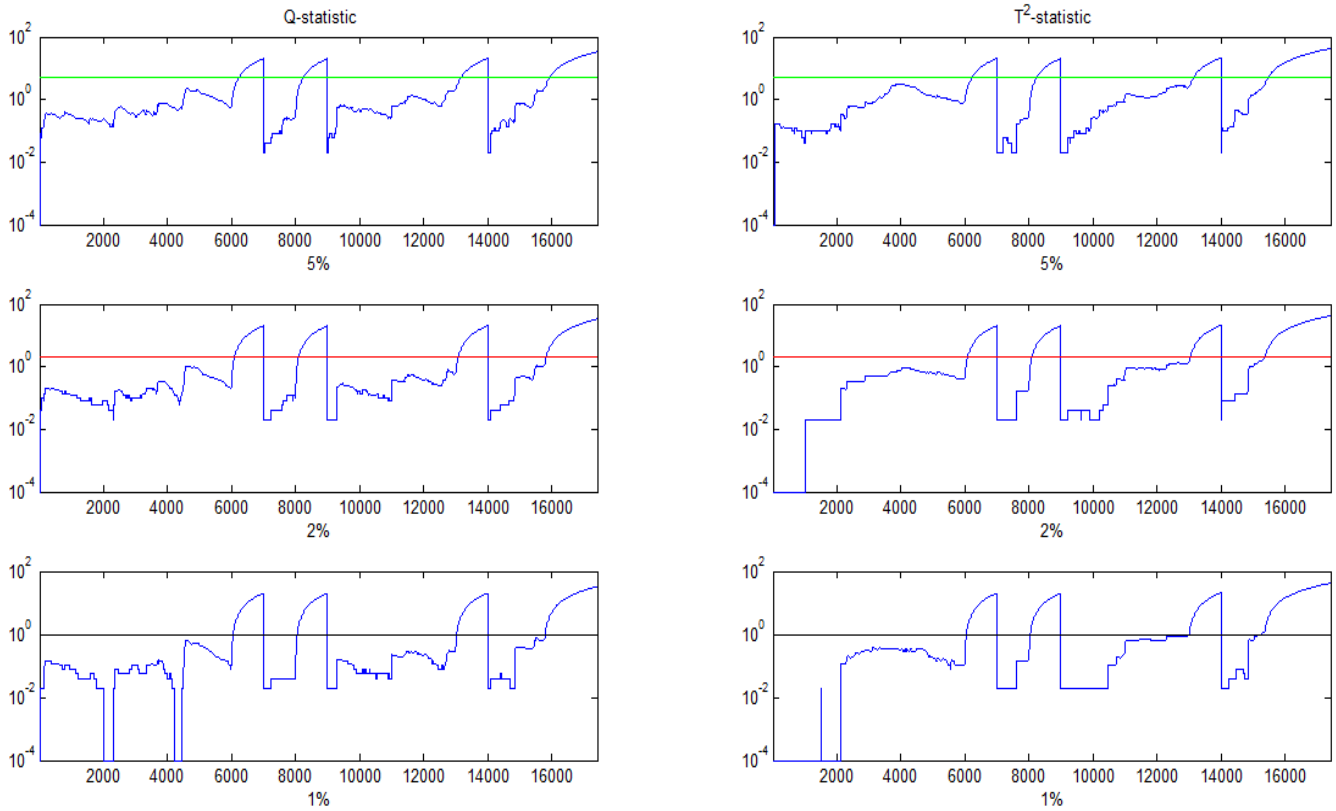

Figure 4. 20. Final CFAR monitoring for Q-statistic (left) and T²-statistic (right), for different FAR limits.

4.8 Results

In this chapter, CFAR-based approach for fault monitoring was developed and tested on SPCA. The main findings can be resumed in the following points:

1. CFAR-based monitoring was able to reduce false indications to zero, without imposing a highly deteriorating the detection time of the fault and the sensitivity compared to the origin detection method (SPCA in this case).
2. This monitoring scheme suffers from the persistency of the faults. Resetting the monitoring window after the fault is removed was able to totally eliminate this problem.
3. CFAR-based monitoring was able to detect the presence of intermittent faults with an acceptable delay. If a fault of reasonably large magnitude is present, the 1% monitoring will indicate its presence in less than 60 seconds from its appearance time.
4. The use of fixed PCA model and a fixed threshold imposes many problems when the data set is not fully descriptive or when some changes affect the process parameters. Then it is desirable to develop an adaptive threshold and model.

4.9 Conclusion

In this part of the work, a monitoring method based on constant rate of false alarms was proposed and investigated. The foundation of the methodology was set and its application was carried. As it was presented, CFAR-based monitoring works on the basis of a preselected PCA scheme in order to provide a monitoring as efficient as the conventional PCA-based methods, in terms of the fault detection time, and eliminating the problem of false detections. In this chapter, CFAR-based monitoring approach was tested on SPCA model. The results given by the proposed methodology has proven their capability to illuminate false detections without posing significant delays.

Conclusion

Fault detection and diagnosis is becoming a highly important tool for process monitoring because of the increasing complexity of industrial systems and the need for high performance, safe, and reliable dynamic processes. The early detection of the presence of faults is critical in providing high quality products, reducing the percentage of defectives, and preventing equipment damages and economical losses. Data driven approaches for fault detection were extensively studied over the last few decades due to their simplicity compared to the model based methods. As it was declared in [86], no accurate mathematical model for cement rotary kiln and similar systems exists, this makes the use of data driven methods less costly and more efficient.

In this report, principle component analysis (PCA) was used as a fault detection approach to provide a monitoring for a cement rotary kiln system. Static PCA was tested as a primary detection method resulting in a good detection time with acceptable delays and sensitivity. However, control charts used to provide process monitoring are suffering from the presence of false detections and outliers. A methodology for process monitoring was presented in this work; based on constant false alarms rate (CFAR), the presence of the fault can be declared once the rate of false alarms exceeds its upper control limit. The CFAR-based monitoring scheme was successfully tested on the basis of SPCA. The proposed methodology was able to provide a monitoring with zero false detections without deteriorating the detection time of the fault. The execution time of the CFAR-based monitoring was about 0.75 seconds in average, which makes the proposed methodology suitable for providing online monitoring in our application where the data is collected each second.

As discussed earlier, the proposed methodology suffers from the persistency of the faults. However, a proposed weighting envelope was able to reduce that persistency. Therefore, as a future work, the development of an adaptive weighting envelope to illuminate the fault persistency is advisable in order to increase the efficiency of the proposed CFAR-based monitoring technique. Furthermore, testing the efficiency of the proposed methodology with other PCA approaches is advisable in order to test its ability to provide a good monitoring. An adaptive thresholding specially designed to keep the rate of false alarms under a selected value for the healthy operation is to be investigated. Finally, the work presented in this report dealt only with the first step of the Fault Detection and Diagnosis process, fault identification and fault diagnosis are to be investigated.

Appendix A:

Linear Algebra and Singular Value Decomposition

This section is dedicated to clarify the algebraic terms and theorems used within the report body. However, a basic knowledge of algebra is still required from the reader.

A.1 Hermitian matrices

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be *self-adjoint* or *Hermitian* if $A^{\mathcal{H}} = A$, where: $A^{\mathcal{H}} = (A^*)^T$ with " $*$ " denotes the complex-conjugate and " T " denotes the matrix transpose.
The set of Hermitian matrices of order *n* is denoted by $\mathcal{H}_n$. A *Real* matrix $A \in \mathbb{R}^{n\times n}$ is Hermitian if and only if $A^T = A$.

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be *symmetric* if $A^T = A$. The set of real symmetric matrices of order *n* is denoted by $\mathcal{S}_n$. Therefore, it is clear that $\mathcal{S}_n \subset \mathcal{H}_n$.

A.2 Inner Product Spaces

Let $\mathcal{V}$ be a vector space over the field $\mathbb{F}$, where $\mathbb{F}$ is either $\mathbb{R}$ or $\mathbb{C}$. An inner product on $\mathcal{V}$ is a function $\langle\cdot,\cdot\rangle$: $\mathcal{V} \times \mathcal{V} \rightarrow \mathbb{F}$ such that for all $u, v, w \in \mathcal{V}$ and $\alpha, \beta \in \mathbb{F}$, the following hold:

1. $\langle v, v\rangle \geq 0$ and $\langle v, v\rangle = 0$ if and only if $v = 0$.
2. $\langle \alpha u + \beta v, w\rangle = \alpha\langle u, w\rangle + \beta\langle v, w\rangle$.
3. $\langle u, v\rangle = \langle v, u\rangle^*$.

We define A real (or complex) *inner product space* as a vector space $\mathcal{V}$ over $\mathbb{R}$ (or $\mathbb{C}$), along with an inner product defined on it [70].
Based on the previous statements and under *standard inner product* (dot product), two fundamental spaces are defined as:

1. The *n-dimensional Euclidean Space:* is the inner product space $\mathbb{R}^n$ under the *standard inner product*, defined by:

$$\langle u, v\rangle = u^T . v = \sum_{i=1}^{n} u_i . v_i \tag{A.1}$$

2. The *n-dimensional Unitary Space:* $\mathbb{C}^n$ is also an inner product space under the *dot product*, defined by:

$$\langle u, v\rangle = v^{\mathcal{H}} . u = \sum_{i=1}^{n} u_i . v_i^* \tag{A.2}$$

The Euclidean norm of a vector u is defined as the inner product of u with itself and it is denoted as:

$$\|u\|_2^2 = \|u\|^2 = \langle v, v \rangle = \sum_{i=1}^{n} {u_i}^2 \tag{A.3}$$

A.3 Change of Basis

Let $\mathcal{V}$ being a vector space over $\mathbb{F}$ in which one choses the basis $\mathcal{B} = \{e_1, e_2, \dots, e_n\}$, and let P be an $n \times n$ invertile matrix. The set of vectors: $\Psi = \{v_1, v_2, \dots, v_n\}$ such that:

$$v_i = \sum_{j=1}^{n} e_j . p_{ji} \quad ; \quad i, j = 1,2, \dots, n \tag{A.4}$$

Is another basis for $\mathcal{V}$. And P is said to be the *transformation* or *change-of-basis matrix*.

If $\mathcal{L}: \mathcal{V} \rightarrow \mathcal{W}$ is a linear mapping. For different selection of basis for the vector spaces $\mathcal{V}$ and $\mathcal{W}$, infinite number of linear mapping matrices can be formed. Let $\mathcal{B}, \widehat{\mathcal{B}}$ be bases of $\mathcal{V}$ and let $\Psi, \widehat{\Psi}$ be bases of $\mathcal{W}$. Let us denote by P, Q the change-of-basis matrices of $\mathcal{B} \rightarrow \widehat{\mathcal{B}}$ and $\Psi \rightarrow \widehat{\Psi}$ respectively. Furthermore, let A, M be the linear mapping matrices associated with the basis $\{\mathcal{B}, \Psi\}$ and $\{\widehat{\mathcal{B}}, \widehat{\Psi}\}$ respectively. Then, it can be shown that:

$$AP = QM \quad or \quad M = Q^{-1}AP$$

it is customary to say that M and A are *equivalent* [71].

In cases where $\mathcal{V} = \mathcal{W}$, a trivial selection of basis $\mathcal{B} = \Psi$ and $\widehat{\mathcal{B}} = \widehat{\Psi}$ leads to $Q = P$, and therefore:

$$M = P^{-1}AP \tag{A.5}$$

This special case of equivalence is often called *Similarity*, and we say that A and M are *Similar* or *Conjugate* matrices. The rank of a matrix is preserved under any equivalence transformation.

A.4 Eigenvalues and Eigenvectors

An *eigenvector* of a square matrix $A \in \mathbb{F}^{n \times n}$ is a nonzero vector that satisfies the equation

$$AX = \lambda X \tag{A.6}$$

where the scalar λ is a scalar called an *eigenvalue*, and $X \in \mathbb{F}^n$ is the associated eigenvector. Eigenvalues and eigenvectors are also known as, respectively, *characteristic roots* and *characteristic vectors* [72]. The term Eigen-pair is used to address the pair (λ, X). An eigenvalue can be either simple or repeated. The spectrum of A, $\sigma(A)$, is the multiset of all eigenvalues of A, with eigenvalue λ appearing $m(\lambda)$ times (*algebraic multiplicity*) in $\sigma(A)$. The *geometric multiplicity*, $q(\lambda)$, of an eigenvalue λ is the number of linearly independent eigenvectors associated with the eigenvalue λ.

1. An eigenvalue λ is *simple* if $m(\lambda) = 1$.
2. An eigenvalue λ is *semi-simple* if $m(\lambda) = q(\lambda)$.

One of the simplest forms that a matrix A can be transformed into, using equivalence or similarity transformation, is the diagonal form. Based on the eigenvalues of the matrix A, the following results have been proved:

1. Let $\lambda_1, \lambda_2, \dots, \lambda_r$ be distinct eigenvalues of A, with $r \leq n$. If $A \in \mathbb{C}^{n \times n}$, then A is *diagonalizable* if and only if $m(\lambda_i) = q(\lambda_i)$ for $i = 1, 2, \dots, r$. If $A \in \mathbb{R}^{n \times n}$,, then A is

diagonalizable by a nonsingular matrix $M \in \mathbb{R}^{n\times n}$ if and only if all the eigenvalues of A are real and $m(\lambda_i) = q(\lambda_i)$ for $i = 1, 2, \dots, r$.

2. A is diagonalizable if and only if A has n linearly independent eigenvectors.
3. If A has n distinct eigenvalues, then A is diagonalizable.

More properties and results are presented in [70].

A.5 Orthogonality

In an inner product space $\mathcal{V}$, two vectors $x, y \in \mathcal{V}$ are said to be *orthogonal* if and only if $\langle x, y\rangle = 0$, and this is denoted by writing $x \perp y$. This notion can be extended to multiple vectors and matrices.

1. $\mathcal{B} = \{u_1, u_2, \dots, u_n\}$ is called an *orthonormal set* if and only if:

$$\langle u_i, v_j\rangle = \begin{cases} 1 & , \quad i = j \\ 0 & , \quad i \neq j \end{cases} \tag{A.7}$$

   - Every orthonormal set is linearly independent, which implies that every orthonormal set of n vectors from any n-dimensional space $\mathcal{V}$ is an orthonormal basis for $\mathcal{V}$.
   - Orthogonality is used as a measure of linear correlation between different vectors of data.
2. A unitary matrix is defined to be a complex matrix $A \in \mathbb{C}^{n\times n}$ whose columns (or rows) constitute an orthonormal basis for $\mathbb{C}^n$
3. An orthogonal matrix is defined to be a real matrix $A \in \mathbb{R}^{n\times n}$ whose columns (or rows) constitute an orthonormal basis for $\mathbb{R}^n$.[73]

The following statements are equivalent to saying that a real matrix $A \in \mathbb{R}^{n\times n}$ is orthonormal.

1. A has orthonormal columns.
2. A has orthonormal rows.
3. $A^{-1} = A^T$.
4. $\|A.x\|_2 = \|x\|_2$ for every $x \in \mathbb{R}^n$.

An *orthogonal projection* on the direction of the vector $u \in \mathbb{C}^n$ with $\|u\|_2 = 1$, is characterized by the *Elementary Projection Matrix*:

$$Q = I_n - u.u^{\mathcal{H}} \in \mathbb{C}^{n\times n} \tag{A.8}$$

With $v_u = (I_n - Q).v$ is the component of v in the direction of u by orthogonal projection and $\overline{v_u} = Q.v$ defines the projection of v onto the orthogonal complement of u, I_n is the $n \times n$ identity matrix. And it can be shown that Q is an orthonormal matrix.

- The orthogonal complement of u, denoted $u^{\perp}$, is the space of all vectors orthogonal to u.

A.6 Singular Value Decomposition (SVD)

A *singular value decomposition* of a matrix $A \in \mathbb{C}^{n\times m}$ with $rank(A) = r$ and $m \leq n$ is a factorization:

$$A = U.\Sigma.V^{\mathcal{H}} \tag{A.9}$$

With $\Sigma = \text{diag}(s_1, s_2, \dots, s_p) \in \mathbb{R}^{\text{n}\times\text{m}}$, $p = min\{m, n\}$ and $s_1 \geq s_2 \geq \cdots \geq s_p \geq 0$, and where we have $U \in \mathbb{C}^{n\times n}$ and $V \in \mathbb{C}^{m\times m}$ are both unitary.

1. The diagonal entries of Σ are called the *singular values* of A.
2. The columns of U are called *left singular vectors* of A.
3. The columns of V are called *right singular vectors* of A.

Some basic facts and properties of SVD are listed here:

1. The first r singular values are nontrivial while the last $p-r$ are zeros.
2. Every $A \in \mathbb{C}^{n\times m}$ has a singular value decomposition. If $A \in \mathbb{R}^{n\times m}$, then U and V may be taken to be real.
3. The singular values of a matrix are *unique*. [70]
4. If $A = U.\Sigma.V^{\mathcal{H}}$ is the SVD of A, then the following relations can be proved:

$$A.v_i = s_i.u_i\,, \qquad A^{\mathcal{H}}.u_i = s_i.u_i\,, \qquad u_i^{\mathcal{H}}.A.v_i = s_i\ ,\ \ i = 1,2,\dots,p \tag{A.10}$$

5. The nonzero singular values of A are the square roots of the nonzero *eigenvalues* of $A^{\mathcal{H}}.A$ or $A.A^{\mathcal{H}}$. The columns of V are *eigenvectors* of $A^{\mathcal{H}}.A$ and The columns of the unitary matrix U are *eigenvectors* of $A.A^{\mathcal{H}}$.[74]
6. If $A \in \mathbb{C}^{n\times n}$ is *Hermitian* with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$, then the singular values of A are $|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|$.

There exist many algorithms in the literature for computing the singular value decomposition matrices for a given rectangular matrix. The computation of SVD is basically built on the computation of the eigenvalues and eigenvectors, which can be performed either via standard numerical algorithms or via the use of more sophisticated algorithms, depending on the structure of the matrix. The computation of the SVD of the matrix A is, at least in theory, equivalent to the computation of eigenvalues and eigenvectors for the matrix $A^{\mathcal{H}}.A$ [75]. A considerable number of ways that was used to derive the SVD are presented in [76]. Singular value decomposition has found a wide range of applications in modern science, such as Image Compression, Network analysis and Data mining (Face and Handwriting recognition, Principle Component Analysis).

Appendix B:

Cross-Validation and Bootstrapping

In practice, selecting an appropriate model is a critical step in almost any process. It is quite clear that selecting a model that under fits a data set will make the selected model blindly loosing important information. In the other side, overfitting a data set is also undesirable due to the increasing amount of noise accumulated by the high-order models, this fact makes the selection of the model's order or dimension an ambiguous task. Furthermore, it is a desirable characteristic of any model to be able to work properly when the incoming data experiences some gaps. For PCA models, the selection of the model dimensionality (the number of principal components) is critical for the sensitivity of the model. Therefore, many methods have been developed through the last decades to deal with such problems, this section is dedicated to deal with two of them, the well-known Cross-Validation and the Bootstrapping algorithms.

B.1 Cross-Validation

Cross-Validation is a statistical method used for evaluating and comparing learning algorithms, the adopted philosophy is to divide a set of existing data into two segments: one used to *learn* or *train* a model i.e. to generate a model that fits the *training set*. The other segment of the data is used to *validate* the model generated by the first segment. Typically, the training and validation sets in a CV algorithm must cross-over in successive rounds such that each data point has a chance of participating in the construction of at least one model and a chance of being validated for the other models. The concept of cross-validation was initially proposed in 1951 as a design for assessing the effectiveness of model weights. The application of cross-validation to principal component and factor analysis was investigates the first time at 1978 in [77]. Based on [77], cross-validation statement can be summarized in the following: if we have a data set, and we want to *approximate* these data by applying one of *class* of models to optimize some *Goodness-of-Fit Criterion (GFC).* The first step in *CV* is to formulate the selection of the model from the target *class,* this is done by the determination of a single parameter "*S*", which is put in the heart of the optimization process. Applying this philosophy to Principal component analysis would lead to the following selection which was used throughout the text:

1. *The class:* Principal Component Model.
2. *The optimized parameter "S":* the number of principal components "a".
3. *GFC:* the simple least squares criterion applied to the model residuals where the estimation is dome using the *Prediction Residual Sum of Squares (PRESS).*

There exist a considerable number of cross-validation algorithms, the most popular algorithms are presented in the following few pages:

1. *Resubstitution Validation*: all the available data is used as a learning set, and then the estimated model is validated using the same, whole data set. This algorithm is easy, simple to build and has the smallest time complexity. But, it suffers from overfitting since there is nothing that guarantees the quality of the model when a new data set is used.

2. *Hold-Out Validation:* the natural approach would be to divide the data set into two non-overlapped parts (mutually exclusive sets), training and test parts. During the training (model construction), the testing set is totally held out of consideration while the training set is sent to the *inducer* and the *induced classifier* (model) is tested on that held out set. Clearly, this algorithm will reduce the overfitting and increase the accuracy of the estimation and resulting in less computation time than the Resubstitution Validation. But, the fact that we are using smaller data set to construct the model, where a certain pattern might be dominant, makes the process of *splitting* the data into training and testing sets a hard task since the final results are highly dependent on that data split [78]. Furthermore, the more data instances left in the test set, the higher the bias of the estimate [79].
3. *K-fold Cross-Validation (KFCV):* the proposal of this algorithm follows logically from the splitting problem that arises in the hold-out validation. Basically, to cope with the splitting problem, one is supposed to redo the hold-out validation but this time with the first part is used for testing and the second is used for construction the model, in this case two goodness-of-fit parameters will be recorded and the average will be a reasonable choice. Merely, this approach forms the basis for the *K-fold Cross-Validation algorithm.* In k-fold cross –validation, the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining $k-1$ folds are used for learning [78]. It is a necessary for the accuracy of the method to make sure that every fold is a *good* representative for the whole data set. This algorithm suffers from a high variance in the Goodness-of-Fit Parameter (GFP) and a large time complexity as k increases. The following figure illustrates a 4-fold cross validation process.

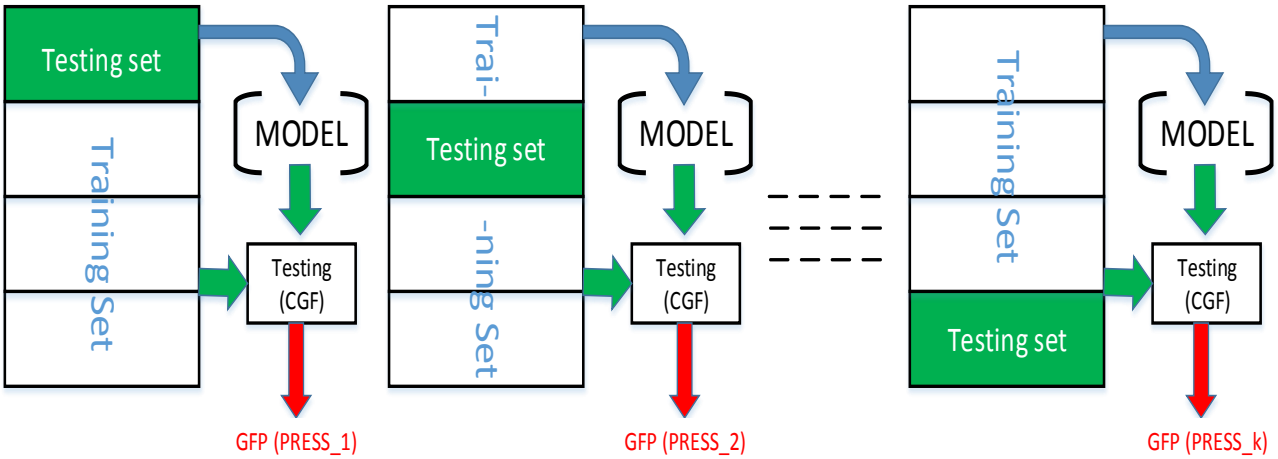

Figure B. 1. K-Fold Cross-Validation Process.

4. *Leave-one-out Cross-Validation (LOOCV):* this algorithm can be seen as a special case of the *KFCV* with $k = n \triangleq the\ number\ of\ observations$. At each instant, all the data, except one, is used to training and the model is tested on the remaining single observation. The accuracy estimate for the *LOOCV* is almost unbiased but the problem of the variance stated earlier is magnified to its maximum. This algorithm is highly used

in practice for cases when the data set contains a relatively small number of observations.

The following flow chart summarized the *KFCV* algorithm used to estimate the number of principal components.

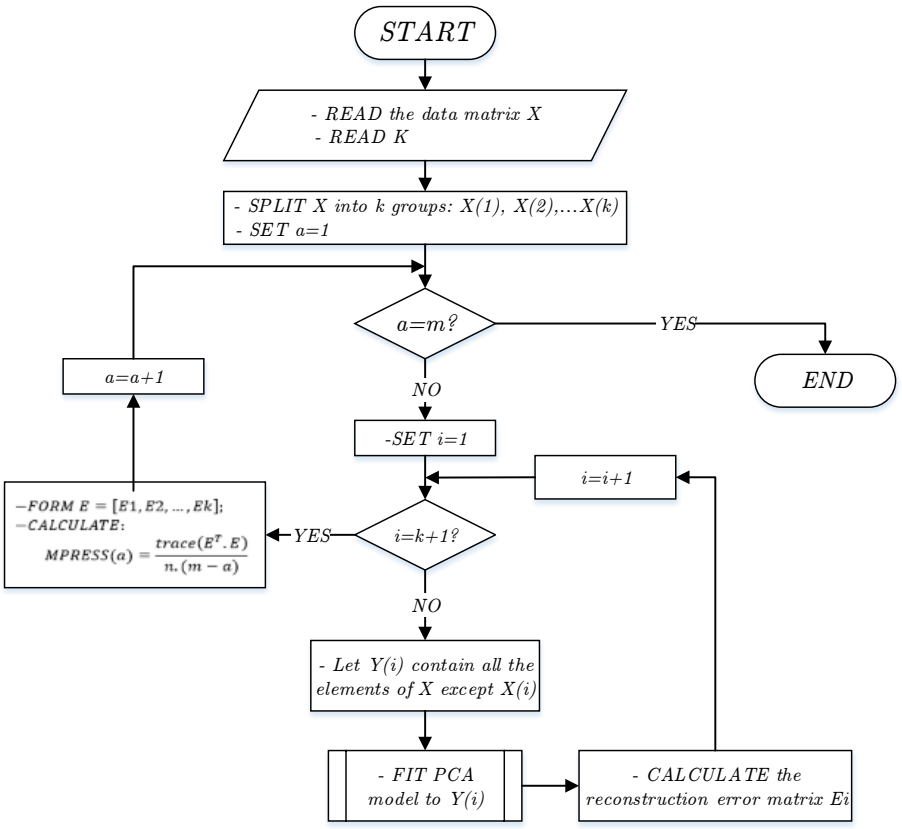

Figure B. 2. K-fold Cross-Validation algorithm for determining the number of principal components in PCA model based on the MPRESS.

the left-out data are used in the computation of the reconstruction error based on:

$$E = X(i) - X(i).P_a.P_a^T \tag{B.1}$$

The residuals from the model of $X(i)$ are highly dependent on $X(i)$, which will result in overfitting. Merely, overfitting means that the more components we add, the smaller the residuals will be. This is not appropriate because the whole idea of cross-validation is to avoid overfitting by estimating the model independently from the data to be modeled. The denominator of the $MPRESS(a)$ is used to correct the overfitting [80]. Nevertheless, the used formula and its validity as an overfitting correction tool is not clear.

5. *Cross-Validation by eigenvectors:* after dividing the data into G sets, a set will be left out each time and a model fitted for the remaining. For the left out sample, each variable is predicted independently. This method gives estimation independent of the predicted elements. The algorithm, as presented in [80], was explored and found to be efficient and offering the a very good choice since it provides a compromise between time complexity, precision and robustness. The notation P_{cj} means the j^{th} column of the

matrix P and $P^{(-j)}$ denote the matrix P with the j^{th} column removed. The algorithm is summarized in the following steps:

a) Divide the data into groups X_i , $i = 1,2, \dots, G$
b) For $a = 1,2, \dots, m$
   i. For $i = 1,2, \dots, G$
      1) Let Y_i contains all the elements of X except X_i
      2) Normalize Y_i and X_i.
      3) Fit PCA model to Y_i; i.e., (T, P).
      4) For $j = 1,2, \dots, m$
         - Estimate the scores:

$$t^{(-j)} = X_i^{(-j)}.\left(P^{(-j)}\right)^T.\left(P^{(-j)}.\left(P^{(-j)}\right)^T\right)^{-1} \quad \text{(B.2)}$$

         - Calculate the reconstruction error:

$$e_{ij}(a) = X_{ij} - t^{(-j)}.P_{cj} \quad \text{(B.3)}$$

   ii. Compute the prediction residual sum of squares:

$$Press(a) = \sum_{i=1}^{G}\sum_{j=1}^{m}\left(e_{ij}(a)\right)^2 \quad \text{(B.4)}$$

Other algorithms were presented to enhance the reliability of the cross-validatory estimation, the most popular methods are the ones proposed in [81] and the *Wold-procedure* in [77] which works based on the NIPALS algorithm, this last algorithm has a good number of successor improvements. A full discussion for a set of six CV algorithms, containing most of the ones discussed earlier, applied to the problem of determining the dimensionality of a PCA model along with a comparison between the six algorithms can be found in [80].

B.2 Bootstrapping

The *Bootstrap* is a statistical method used for performance estimation of classifiers. Based on *sampling with replacement*, a learning set of n instances is selected from the total data set. The fact that the sampling is done with replacement makes all instances *equally-likely* to be selected in each sample. This makes selection operation uniformly distributed i.e. the probability of selecting a certain instance equals to $1/n$, thus any instance has a probability of $1 - \frac{1}{n}$ to *not being selected*. After taking n samples, the probability that a given instance wasn't selected during the whole sampling process is:

$$p_n = \left(1 - \frac{1}{n}\right)^n \quad \text{(B.5)}$$

For significantly large number of time instances, n, we have:

$$\lim_{n\to\infty}\left(1 - \frac{1}{n}\right)^n = e^{-1} \quad \text{(B.6)}$$

This merely means that for large n, about 36.8% of the original data will not be selected in the training set, this remaining $0.368n$ instances will form the test set while the other $0.632n$ instances will form the learning set, hence the name $.632\ Bootstrap$. The assumptions made by

bootstrap are basically the same as that of cross-validation, i.e., stability of the algorithm on the dataset is crucial, again a criteria of Goodness-of-Fit have to be selected in order to evaluate the selected parameter to optimize. If the application is made on a PCA model, then the reasonable selection for the optimized parameter would be the number of retained components and the CGF is usually selected as the PRESS. The .632 bootstrap method is expected to fail completely if the data set if totally random [79,82].

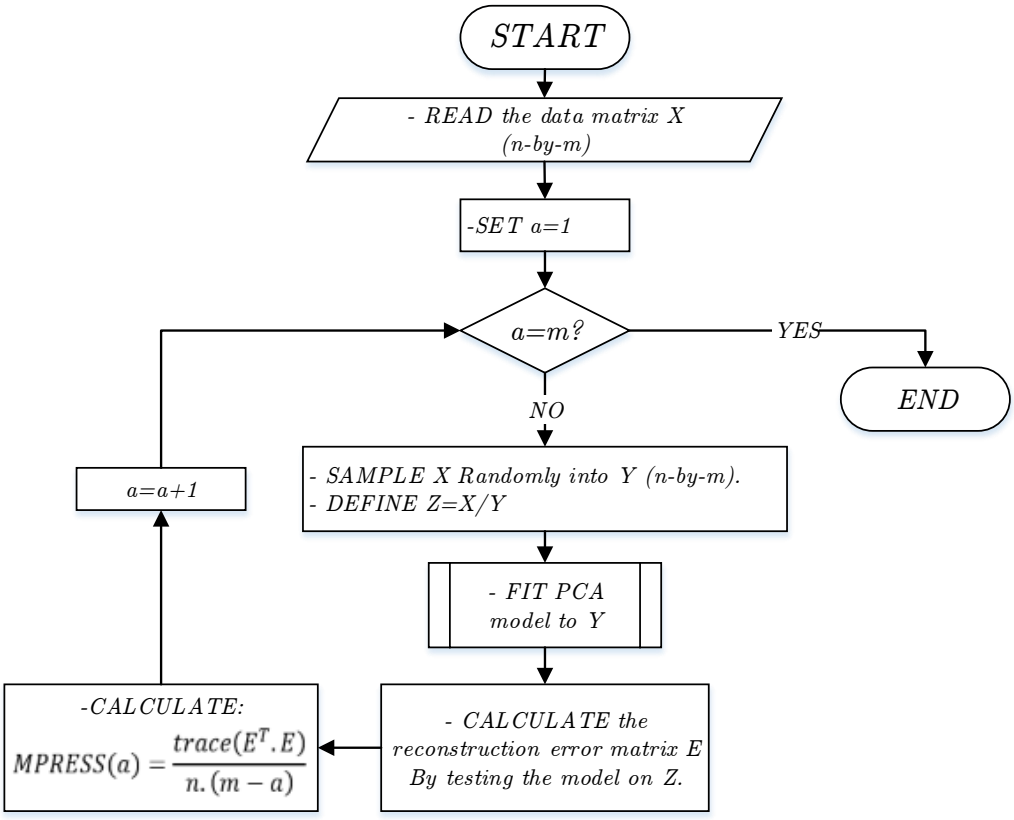

Figure B. 3. ".632 Bootstrap" algorithm for determining the number of principal components in PCA model based on the MPRESS.

Figure A.1. represents the .632 bootstrap algorithm steps applied to the selection of the number of principal components in PCA model using PRESS. The random sampling is generated based on the MATLAB pseudo-random function "rand". Furthermore, there exist other Bootstrap approaches, the most popular ones are the *Bootstrapped k1* method which consists of applying the Guttman-Kaiser criteria on a bootstrapped sample and the *Bootstrapped Eigenvalues/Eigenvectors* method.

In addition to the determination of the PCA model dimensionality, Bootstrap method allows to perform the required stability study of the Principal Component Analysis results [83].

B.3 The variance and Bias of an estimator

For learning algorithms, estimating the accuracy of future predictions and choosing a classifier out of a given set of classifiers is a highly important task. The most basic requirement of any estimation method is the *low bias* and the *low variance*.

1. *Bias:* The bias of a given method in estimating a parameter $\mathcal{S}$ is the difference between the expected and the estimated values of that parameter. An unbiased estimation method is a method with zero bias, i.e.,

$$Bias = E(\mathcal{S}) - \hat{\mathcal{S}} \tag{B.7}$$

2. *Variance:* even if an estimator has a low bias, or even zero, the performance of the estimator may still be surprisingly poor due to the high variance of the estimated parameter. A method with high variance makes the limits of the *confidence interval*, where $\mathcal{S}$ is supposed to be located, wider. This means that any expected value revealed by the method is not accurate.

For instance, k-fold cross-validation is known to have a large bias for small number of folds and a large variance for large number of folds. Due to these facts, a tradeoff between bias and variance is required for any method.

Appendix C:

Description of the Cement Rotary kiln

Cement is a substance used in building, it is obtained by grinding an intermediate product called clinker. The typical process that results in producing the clinker passes through the three steps:

1. Grinding a mixture of limestone and clay or shale to make a fine "rawmix".
2. Heating the rawmix to sintering temperature (up to 1450 °C) in a cement kiln.
3. Grinding the resulting clinker to make cement.

The clinker is a product that comes out from one of the most critical parts in the cement plant, the cement kiln. Cement kilns, in general, are the heart of the cement production process; their capacity usually determines the overall capacity of the cement plant. Cement kiln is mainly used for calcining the cement clinker and it can be divided into dry-producing cement kiln and wet-producing cement kiln. In the wet process, raw meal is supplied at ambient temperature in the form of a slurry with about 40% of water. In modern works, the blended raw material enters the kiln via the pre-heater tower. Here, hot gases from the kiln, and probably the cooled clinker at the far end of the kiln, are used to heat the raw meal. As a result, the raw meal is already hot before it enters the kiln. The dry process is much more thermally efficient than the wet process.

The rotary kiln consists of a steel plate tube lined with firebrick. The tube slopes slightly, between 1 and 4 degrees, and slowly rotates on its axis at between 30 and 250 revolutions per hour. Rawmix is fed in at the upper end, and the rotation of the kiln causes it gradually to move downhill to the other end of the kiln. At the other end, fuel; in the form of gas, oil, or pulverized solid fuel, is blown in through the "burner pipe", producing a large concentric flame in the lower part of the kiln tube. As material moves under the flame, it reaches its peak temperature, before dropping out of the kiln tube into the cooler. Air is drawn first through the cooler and then through the kiln for combustion of the fuel. In the cooler the air is heated by the cooling clinker, so that it may be 400 to 800 °C before it enters the kiln, thus causing intense and rapid combustion of the fuel. *Figure C.1*. from [84], provides a schematic view for the rotary kiln

As the most energy consuming, and the most critical and complex part in the production process, improvements of cement kilns has been the central concern of the cement manufacturing technology, thereby limiting the energy consumption, the green-house gases and improving the efficiency. To increase the kiln efficiency, a system of suspension preheater cyclones plus pre-calciner are usually used. In such complex system, failures and malfunctions have a high probability of occurrence. Therefore, the need of an automatic system for detecting the presence of faults and determining their roots is mandatory.

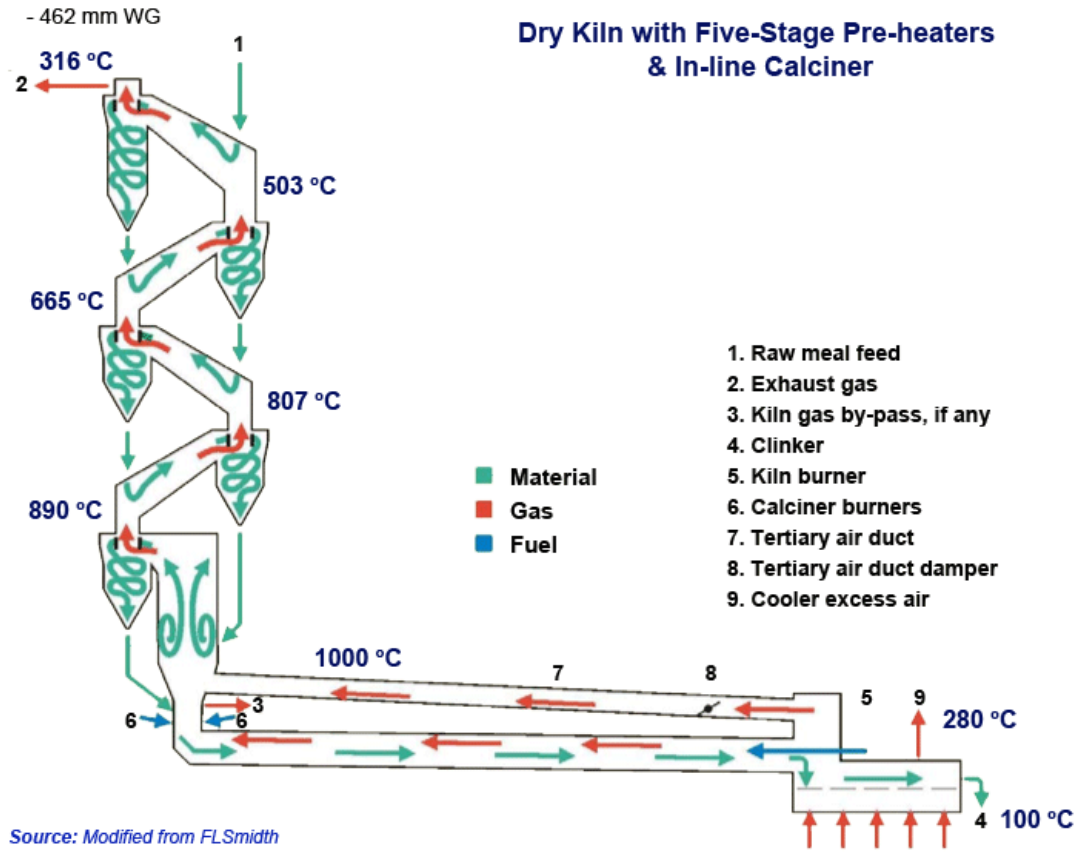

Figure C. 1. Dry Kiln schematic with Five-stage Pre-heater and in-line calciner (used with permission from [84]).

Due to the complex dynamic, multivariable nature, nonlinear reaction kinetics, long time delays and variable raw material feed characteristics, the rotary kiln process is inherently difficult to model. It was declared in [86] that *"to the authors knowledge, there is no mathematical model that adequately describes the process behavior"*. Moreover, the product quality of industrial rotary kilns is usually measured after the clinker has cooled down which adversely limits the online supervision [85, 86].

The applications in this work were made on a data collected on the cement plant of Ain El-Kbira, Setif, Algeria. from January 23rd, 2014 at 23:39:33PM to January 24th, 2014 at 04:30:00AM with acquisition rate of one second. The current control performed in Process is manual centralized mode, where any action is based upon the human experience [13]. The used data set consists of a continuous measurement for 4 hours, 50 minutes and 28 seconds (177428 seconds). The first 15300 seconds are collected when the system is surely in healthy state. The later 2084 seconds are collected in the presence of a fault in the system. The total data set holds information from 52 sensors spread along the process to monitor the different quantities, i.e., temperature, pressure, flow, …etc. Finally, amongst the 52 sensors, the sensors "331-01~07 / PE (or TE)" and "341-01~07 / PE(or TE)" are not used as feedback sensors for any control loop.

List of References

[1] R. Isermann, "*Fault-diagnosis systems–An introduction from fault detection to fault tolerance*", Springer, Heidelberg, 2006.

[2] R. Isermann, "*Model-based fault-detection-status and applications*". Annual Reviews in Control, vol. 29, pp. 71-85, 2005.

[3] P. F. Odgaard, B. Mataji, "*Observer-based fault detection and moisture estimating in coal mills*". Control Engineering Practice, vol. 16 (8), pp. 909–921, 2008.

[4] F. Karami, J. Poshtan, M. Poshtan, "*Detection of broken rotor bars in induction motors using nonlinear Kalman filters*", ISA Transactions, vol. 49 (2), pp. 189–95, 2010.

[5] S. Joe Qin, "*Statistical process monitoring: basics and beyond*", Journal of Chemometrics, vol. 17, pp. 480–502, 2003.

[6] V. Venkatasubramanian, R. Rengaswamy, K. Yin, Surya N. Kavuri. "*A review of Process Fault Detection and Diagnosis part 1: Quantitative model-based methods*", Computers and Chemical Engineering, vol. 27, pp.293-311,2003.

[7] Silvio Simani, "*Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*", Ph.D. thesis, University of Ferrara. Italy, Department of Engineering, 1999.

[8] Vamshi Krishna Kandula, "*Fault Detection in Process Control Plants Using Principal Component Analysis*", Master thesis, Louisiana State University, The Department of Electrical Engineering, 2011.

[9] E. Sobhani-Tehrani, K. Khorasani, "*Fault diagnosis of nonlinear systems using a hybrid approach*", Springer, 2009.

[10] Alkan Alkaya, "*Novel Data Driven-Based Fault Detection for Electromechanical and Process Control Systems*", Ph.D. thesis, Çukurova University, Institute of natural and applied sciences Department of electrical and electronics engineering, 2012

[11] Dubravko Miljkovic, "*Fault Detection Methods: A Literature Survey*", Hrvatska elektroprivreda, Zagreb, Croatia, 2011.

[12] A. Benotmane, H. Djemai, "*PCA-based Approach for Fault Detection in Cement Rotary Kiln*", Ingenieur d'état thesis, University M'Hamed BOUGARA-Boumerdes, Institute of Electrical and Electronic Engineering, Department of Power and Control, 2014.

[13] Srinivas Katipamula, Michael R. Brambley, "*Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems -A Review, Part I*", HVAC&R Research, vol. 11 (1), 2005.

[14] V. Venkatasubramanian, R. Rengaswamy, Surya N. Kavuri, K. Yin, "*A Review of Process Fault Detection and Diagnosis, Process History part III: Process history based methods*", Computers & Chemical Engineering, vol. 27, pp.327-346, 2003.

[15] Masayuki Tamura, Shinsuke Tsujita, "*A Study on the Selection of Model Dimensions and Sensitivity of PCA-Based Fault Detection*", Computers & Chemical Engineering, vol. 31 (9), pp. 1035-1046, 2007.

[16] Prem Krishnan, "*Applications of Multivariate Analysis Techniques for Fault Detection, Diagnosis and Isolation*", National University of Singapore, 2011.

[17] Zhiqiang Ge, Zhihuan Song, "*Process Monitoring Based on Independent Component Analysis-Principal Component Analysis (ICA-PCA) and Similarity Factors*", Ind. Eng. Chem. Res, vol. 46 (7), pp. 2054-2063, 2007.

[18] Wei Lou, Guoying Shi, Jun Zhang, "*Research and Application of ICA Technique in Fault Diagnosis for Equipments*", IEEE International Conference, vol. 4, pp. 310-313, 2009.
[19] Douglas C. Montgomery, George C. Runger, "*Applied Statistics and Probability for Engineers*", 3rd Edition, John Wiley & Sons, Inc., New York, pp. 595-644, 2002.
[20] Kristen Severson, Paphonwit Chaiwatanodom, Richard D. Braatz, "*Perspectives on Process Monitoring of Industrial Systems*", IFAC-PapersOnLine, vol. 48. (21), pp. 931–939, 2015.
[21] Herve Abdi and Lynne J. Williams, "*Principal component analysis*", John Wiley & Sons, Inc. WIREs Comp Stat, vol. 2, pp. 433–459, 2010.
[22] I.T. Jolliffe, "*Principal component analysis*", Springer, New York, 2002.
[23] Stephen So, "*Why is the sample variance a biased estimator?*", Signal Processing Laboratory, Grith School of Engineering, Grith University, Brisbane, QLD, Australia, 4111., 2008.
[24] Dawen Liang, "*Maximum Likelihood Estimator for Variance is Biased: Proof*", Carnegie Mellon University.
[25] Bartdeketelaere, Miahubert, Ericschmitt, "*Overview of PCA-Based Statistical Process-Monitoring Methods for Time-Dependent, High-Dimensional Data*", Journal of Quality Technology, vol.47 (4), pp. 318-335, 2015.
[26] Jonathon Shlens, "*A Tutorial on Principal Component Analysis*", Systems Neurobiology Laboratory, Salk Institute for Biological Studies, 2005.
[27] Aly A. Farag, Shireen Elhabian, "*A tutorial on principle component analysis*", University of Louisville, CVIP Lab, 2009.
[28] 36-350, Data Mining, "*The Truth about Principal Components and Factor Analysis*", September 2009.
[29] Gregoria Mateos-Aparicio, "*Partial Least Squares (PLS) Methods: Origins, Evolution, and Application to Social Sciences*", Communications in Statistics - Theory and Methods, vol. 40 (13), pp. 2305-2317, 2011.
[30] Richard Noonan & Herman Wold, "*NIPALS Path Modelling with Latent Variables*", Scandinavian Journal of Educational Research, vol. 21 (1), pp. 33-61, 1977.
[31] Donald. A. Jackson, "*Stopping rules in principal component analysis: a comparison of heuristical and statistical approaches*", Ecological society of America, vol. 74 (8), pp. 2204-2214, 1993.
[32] I. T. Jolliffe, "*Discarding Variables in a Principal Component Analysis. I: Artificial Data*", Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 21 (2), pp. 160-173, 1972.
[33] Gilbert Saporta, "*A control chart approach to select eigenvalues in Principal Component and Correspondence Analysis*", CNAM, Chaire de statistique Appliquée & CEDRIC.
[34] Raymond B. Cattell, "*The Scree Test for The Number of Factors*", Multivariate Behavioral Research, vol. 1 (2), 245-276, 1966.
[35] William R. Zwick and Wayne F. Velicer, "*Comparison of Five Rules for Determining the Number of Components to Retain*", Psychological Bulletin, Vol. 99 (3), 452-442, 1986.
[36] Rubén Daniel Ledesma and Pedro Valero-Mora, "*Determining the Number of Factors to Retain in EFA: an easy-to use computer program for carrying out Parallel Analysis*", Practical Assessment, Research & Evaluation, vol. 12 (2), 2007.
[37] Sergio Valle, Weihua Li, S. Joe Qin, "*Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods*", industrial & engineering chemistry research, vol. 38, pp. 4389-4401, 1999.

[38] P. LEGENRE, L. LEGENDRE, "*Numerical ecology*", 2nd edition, Elsevier Science B.V, Amsterdam, pp. 409-410, 1998.

[39] Richard Cangelosi, Alain Goriely, "*Component retention in principal component analysis with application to cDNA microarray data*", Biology Direct, vol. 2 (2), 2007.

[40] Pedro R. Peres-Neto, Donald A. Jackson, Keith M. Somers, "*How many principal components? stopping rules for determining the number of non-trivial axes revisited*", Computational Statistics & Data Analysis, Vol 49, pp. 974–997, 2005.

[41] Wayne F. Velicer, "*determining the number of components from the matrix of partial correlations*", PSYCHOMETRIKA, vol. 41 (3), pp. 321-327, 1976.

[42] Brian P. O'Connor, "*SPSS and SAS Programs for Determining the Number of Components Using Parallel Analysis and Velicer's MAP Test*", Behavior Research Methods, Instruments, & Computers, vol. 32, pp. 396-402, 2000.

[43] John L. Horn, "*A rationale and test for the number of factors in factor analysis*", PSYCHOMETRICA, vol. 30 (2), pp. 179-185, 1965.

[44] Louis W. Glorfeld, "*An improvement on horn's parallel analysis methodology for selecting the correct number of factors to retain*", Educational and Psychological Measurements, vol. 55 (3), pp. 377-393, 1995.

[45] Alexis Dinno, "*Exploring the Sensitivity of Horn's Parallel Analysis to the Distributional Form of Random Data*", Multivariate Behavioral Research, vol. 44 (3), pp. 362-388, 2009.

[46] Scott B. Franklin, David J. Gibson, Philip A. Robertson, John T. Pohlmann, James S. Fralish, "*Parallel Analysis: A Method for Determining Significant Principal Components*", Journal of Vegetation Science, vol. 6 (1), pp. 99-106, 1995.

[47] J. Mina, C. Verde, "*A Refinement of Dynamic Principal Component Analysis for Fault Detection*", Instituto de Ingeniería-UNAM, Coyoacan DF 04510, México, 2006.

[48] J. Mina, C. Verde, "*Fault Detection Using Dynamic Principal Component Analysis by Average Estimation*", 2nd International Conference on Electrical and Electronics Engineering, pp.374-377, 2005.

[49] Ku. W, Storer. R, Georgakis. C, "*Disturbance Detection and Isolation by Dynamic Principal Component Analysis*". Chemometrics and Intelligent Laboratory Systems, vol. 30 (1), pp. 179–196, 1995.

[50] Tiago J. Rato, Marco S. Reis, "*Defining the Structure of DPCA Models and its Impact on Process Monitoring and Prediction Activities*", Chemometrics and Intelligent Laboratory Systems, Vol. 125, pp 74-86, 2013.

[51] Kruger. U, Zhou. Y, Irwin. G, "*Improved Principal Component Monitoring of Large-Scale Processes*", Journal of Process Control, vol. 14 (8), pp. 879–888, 2004.

[52] Luo. R, Misra. M, Himmelblau. D, "*Sensor Fault Detection via Multiscale Analysis and Dynamic PCA*". Industrial & Engineering Chemistry Research, vol. 38 (4), pp. 1489–1495, 1999.

[53] He. X, Yang. Y, "*Variable MWPCA for Adaptive Process Monitoring*". Industrial & Engineering Chemistry Research, vol. 47 (2), pp. 419–427, 2008.

[54] Chiang. L, Russell. E, Braatz. R, "*Fault Detection and Diagnosis in Industrial Systems*", Springer-Verlag, London, UK, 2001.

[55] Jyh-Cheng Jeng, "*Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms*", Journal of the Taiwan Institute of Chemical Engineers, vol. 41, pp. 475–481, 2010.

[56] Emmanuel. J. Candés, Xiaodong Li, Yi Ma, John Wright, "*Robust Principal Component Analysis*", Journal of the ACM, vol. 58 (11), 2011.

[57] B. Scholkopf, A. Smola, and K. Muller, "*Nonlinear component analysis as a kernel eigenvalue problem*", Neural Computation, vol. 10 (5), pp. 1299–1319, 1998.
[58] Sami Romdhani, Shaogang Gong, Alexandra Psarrou, "*A Multi-View Nonlinear Active Shape Model Using Kernel PCA*", BMVC, pp. 483-492, 1999.
[59] Zhiqiang Ge, Chunjie Yang, Zhihuan Song, "*Improved kernel PCA-based monitoring approach for nonlinear processes*", Chemical Engineering Science, vol. 64, pp. 2245-2255, 2009.
[60] Majdi Mansouri, Mohamed Nounou, Hazem Nounou, Nazmul Karim, "*Kernel PCA-based GLRT for nonlinear fault detection of chemical processes*", Journal of Loss Prevention in the Process Industries, vol. 40, pp. 334-347, 2016.
[61] Pei-Chann Chang, Jheng-Long Wu, "*A critical feature extraction by kernel PCA in stock trading model*", Soft Comput., vol. 19, pp. 1393–1408, 2015.
[62] J.F. MacGregor, T. Kourtl, "*Statistical process control of multivariate processes*", Control Eng. Practice, vol. 3 (3), pp. 403-414, 1995.
[63] S. Ding, P. Zhang, E. Ding, S. Yin, A. Naik, P. Ding, W. Gui, "*On the Application of PCA Technique to Fault Diagnosis*", Tsinghua Science and Technology, vol. 15 (2), pp. 138-144, 2010.
[64] Jackson. J, Mudholkar. G, "*Control Procedures for Residuals Associated with Principal Component Analysis*". Technometrics, vol. 21 (3), pp. 341–349, 1979
[65] Nomikos. P., MacGregor. J, "*Multivariate SPC Charts for Monitoring Batch Processes*". Technometrics, vol. 37, pp. 41–59, 1995.
[66] Radu Platon & Mouloud Amazouz, "*Application of data mining techniques for industrial process optimization*", CANMET Energy Technology Centre – Varennes, 2007.
[67] Edwards. P. J., et al., "*The application of neural networks to the paper-making industry*", European Symposium on Artificial Neural Networks proceedings, pp. 69-74, 1999.
[68] Hiranmayee Vedam, Venkat Venkatasubramanian, "*PCA-SDG based process monitoring and fault diagnosis*", Control Engineering Practice, vol. 7, pp. 903-917, 1999.
[69] Bo Zhou, Hao Ye, Haifeng Zhang, Mingliang Li, "*Process monitoring of iron-making process in a blast furnace with PCA-based methods*", Control Engineering Practice, vol. 47, pp. 1–14, 2016.
[70] Leslie Hogben, et. al., "*The handbook of linear algebra"*, Chapman & Hall/CRC, Taylor & Francis Group, 2007.
[71] Denis Serre, *"Matrices theory and applications"*, Springer-Verlag Inc., New York, 2002.
[72] Kirk Baker, *"Singular Value Decomposition Tutorial"*, March 29, 2005 (Revised January 14, 2013).
[73] Carl. D. Meyer, *"Matrix Analysis and applied linear algebra"*, 2000.
[74] R. A. Horn and C. R. Johnson. "*Matrix Analysis*", Cambridge University Press, 1985.
[75] Radu. C. Cascaval, *"Eigenvalues, Singular Value Decomposition",* Department of Mathematics, University of Colorado.
[76] G. W. STEWART, *"On the early history of the singular value decomposition",* Society for Industrial and Applied Mathematics, 1993.
[77] Svante Wold, *"Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models",* Research Group for Chemometrics Institute of Chemistry, Umeå University, Sweden, 1978.
[78] Payam Refaeilzadeh, Lei Tang, Huan Liu, "*Encyclopedia of Database systems",* pp. 532-538, Springer Science and Business Media, LLC, USA, 2009..

[79] Ron Kohavi, "*A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection",* International Joint Conference on Artificial Intelligence (IJCAI), 1995.

[80] R. Bro, K. Kjeldahl, A. K. Smilde, H. A. L. Kiers, *"Cross-validation of component models: A critical look at current methods",* Analytic and Bioanalytic Chemistry, vol. 390, pp. 1241-1251, 2008.

[81] H. T. Eastment, W. J. Krzanowski, *"Cross-Validatory Choice of the Number of Components from a Principal Component Analysis",* Technometrics, vol. 24 (1), 1982.

[82] Wise. B. M, Gallagher. N. B, Bro. R, Shaver. J. M., *"PLS_Toolbox for use with MATLAB , Version 3.0, Software"*, Eigenvector Research, Inc., Nov. 2002.

[83] J. J. Daudin, C. Duby, P. Trecourt, *"Stability of Principle Component Analysis Studied by the Bootstrap Method",* Institute national Agronomique Paris-Grignon, 1988.

[84] http:// ietd.iipnetwork.org/content/dry-kilns-multistage-pre-heaters-and-pre-calcination.

[85] M. Jarvensivu, K. Saari, S. L. Jamsa-Jounela, "*Intelligent control system of an industrial lime kiln process*", Control Engineering Practice, vol. 9, pp. 589–606, 2001.

[86] Abdelmalek Kouadri, Abderazak Bensmail, Aissa Kheldoun, Larbi Refoufi, "*An adaptive threshold estimation scheme for abrupt changes detection algorithm in a cement rotary kiln*", Journal of Computational and Applied Mathematics, vol. 259, pp. 835–842, 2014.