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 Engineering

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

MASTER

In **Control Engineering**

Option: **Control**

Title:

Interval-Valued PCA-Based Approach For Fault Detection in dynamic systems

Presented by:

Sami Oussama Hellati

Abdellah Anis Kouari

Supervisor:

Pr. Abdelmalek Kouadri

Mr. Abdelhalim Louifi (Co-supervisor).

Registration Number: …./2022

Dedications

Praise and thanks to Almighty Allah and peace be upon the messenger of God Muhamed El-Moustafa.

This work is dedicated to:

My beloved,precious and valuable mother and father. I would like to thank you so much for your support and sacrifices and may God make your destiny a paradise of Eden. A great dedication also to my family and all my friends, you are all like brothers rather than friends to me.

Oussama

My dear parents, brothers and sisters.

Anis

Acknowledgements

First and first, we want to give praise and thanks to Almighty Allah, the Most Gracious and Most Merciful. We would like to thank our supervisor Pr. Kouadri Abdelmalek, and our co-supervisor Mr. Louifi Abdelhalim, for their valuable suggestions, support, and guidance.

Also, we are sincerely grateful to the IEEE student branch university of Boumerdes and to the Institute of Electrical and Electronics Engineering Teachers, Workers, and Colleagues, for their great help and support during the last five years.

Abstract

Fault detection and diagnosis is an important domain in modern process engineering, where principal component analysis (PCA) is one of its powerful data-driven techniques. The use of PCA in dynamic systems will approximate the dynamic behavior with a static one, which is not convenient. To address this issue, one of the most well-known approaches is the use of time-lag-shifted data; this approach is known as dynamic principal component analysis (DPCA). However, DPCA is still not an optimal solution due to the effect of uncertainties on the model parameters, which will lead to drifts and affect the performance of the model. In this dissertation, a new approach is proposed to overcome this issue by including uncertainties in the modeling phase, which will ensure a safe interval for the data to fluctuate. This approach is called interval-valued dynamic principal component analysis (IV-DPCA). To test the performance of IV-DPCA, real data obtained from a cement manufacturing plant were used to build and test the PCA, DPCA, and IV-DPCA models, then the three models were compared to each other in terms of false alarm rate (FAR), missed alarms rate (MDR), and detection time delay (DTD).

Keywords: Fault Detection; Principal Component Analysis (PCA); Dynamic Principal Component Analysis (DPCA); Interval-Valued Dynamic Principal Component Analysis (IV-DPCA).

Contents

List of Figures

List of Tables

Nomenclature

COV	Covariance
CPV	Cumulative Percent Variance
CTD	Clearing Time Delay
CV	Cross Validation
DPCA	Dynamic Principal Component Analysis
DTD	Detection Time Delay
$\mathcal{F}$	Fisher distribution
FAR	False Alarms Rate
FDD	Fault Detection and Diagnosis
ICA	Independent Component Analysis
IV-DPCA	Interval-Valued Dynamic Principal Component Analysis
IPCA	Interval Principal Components Analysis
IVD	Interval Valued Data
MDR	Missed Detection Rate
PA	Parallel Analysis
PC	Principal Component
PCA	Principal Component Analysis
PLS	Partial Least Squares
QTA	Qualitative Trend Analysis
SPE	Square Prediction Error
SVD	Single Valued Decomposition

Introduction

The world is witnessing a great evolution in the industrial sector and the use of control systems has become a necessity to increase efficiency and product quality. Therefore, faults and errors can not be allowed if productivity is to be maximized as faults usually result in impaired performance, malfunctions, or failures which can lead to partial or complete system breakdown. To avoid this, faults need to be recognized before they cause damages. Therefore, fault detection has become a top priority in modern process automation. The purpose of fault detection and diagnosis is to ensure the reliability of the system by triggering an alarm indicating the existance of at least one fault.

Many fault detection and diagnosis methods have been developed over the years. Each method captures a subset of the diagnostic characteristics which may be more suitable for a certain class of issues than other ones. Model-based methods require the creation of a correct mathematical model to represent the system, while data-driven methods rely on data derived from the processes. Since industrial processes have become more complicated, a precise mathematical model can not be extracted. Therefore, the attention is switched to data-driven methods. One of the best approaches used of data-driven techniques is the principal component analysis (PCA).

The main goal of the PCA technique is to minimize the dimension of a data set with many connected variables while preserving as much variance as possible. On the other hand, the application of PCA to dynamic systems leads to approximate the dynamic behavior. The use of Ku's suggested time-lag-shifted data method is one of the most well-known approaches to overcome this problem [1].

Another challenge when using this approach is that the data can not be described accurately due to the fact that uncertainties and measurement errors exists in the sensors used to obtain it.

In this dissertation, a newly developed technique named Interval-Valued Dynamic PCA (IV-DPCA) has been proposed as a solution to this challenge by extending the single-valued DPCA technique to interval-valued one. The IV-DPCA is a repeated DPCA application on data obtained under the same system conditions where the model is built based on the extracted interval eigenvalues, their corresponding interval eigenvectors, and the interval fault indicators thresholds. Afterwards, various monitoring performances of the proposed technique were established and compared to those of conventional PCA and DPCA methods.

This dissertation is organized as follows:

- In Chapter 1, fault detection and diagnosis terminologies, methodologies, and classifications are defined.
- In Chapter 2, the mathematical foundation of PCA, DPCA, and IV-DPCA is discussed and their applications in a simulated example are given with the obtained results.
- In Chapter 3, PCA, DPCA, IV-DPCA techniques were applied in a real-life system (rotary

cement kiln) with a discussion of the results obtained.

Chapter 1

Fault Detection and Diagnosis

1.1 Introduction

The creation of new equipment and technologies facilities data gathering from modern processes. Engineers tried to use this information to improve plant operations and avoid costly mistakes. There are abnormal circumstances that could lead to a partial or complete breakdown in productivity and control of the factory.

Fault detection and diagnosis (FDD) techniques are categorized into two broad groups: model-based FDD and data-driven FDD. Because modern industrial systems are multivariate processes, significant amounts of data are gathered and utilized to offer operational status information. This characteristic provides precedence for data-driven approaches and an advantage when monitoring systems [2][3].

1.2 Terminology

to gain a better understanding of the field of fault detection and diagnosis, there are a few principles that need to be comprehended.

- **Fault**: A fault is an unpermitted deviation of at least one characteristic property of the system from the acceptable, usual standard condition. It is considered an abnormal condition that may cause a reduction in or loss of the capability of a functional unit to perform a required function [4].

- **Failure**: It is a permanent interruption of a systems ability to perform a required function under specified operating conditions. It is the result of one or more faults, and it usually arises after the beginning of operation or by increasingly stressing the system [4].

- **Malfunction**: It is defined as an intermittent irregularity in the fulfillment of a systems desired function [5]. It is a temporary interruption of a systems function.

- **Fault detection**: The determination of the presence of one or more faults in the system, as well as the time when they occurred [5][6].

- **Fault diagnosis**: The determination of the kind, size, location and time of detection of a fault. It follows fault detection and includes fault isolation and identification.

- **Fault isolation**: Determination of the type and location of a fault [5].

- **Fault identification**: Determination of the size and time variant behavior of a fault [5].
- **Residual**: The deviation between a measured and predicted value [7].
- **Symptoms**: A change in an observable quantity from normal behavior [5].
- **Reliability**: It is the probability that a component or system will perform a required function for a given time when used under stated operating conditions.
- **Safety**: Ability of a system not to cause danger to persons or equipment or the environment [4]. It is concerned with the dangerous effects of faults, failures and malfunctions.
- **Availability**: It is the probability that a system or equipment will operate satisfactorily and effectively for any period of time [4].
- **Integrity**: It is the ability of a system to detect faults in its own operation and to inform a human operator [4].

1.3 Fault models

Faults can be classified into different classes depending on some given criteria:

1. Depending on where the fault occurs in the system, there are three types of defects:
   - **Actuator fault**: It is a type of failure affecting the system inputs, usually due to abnormal operation, excessive loading or material aging. Its consequences may vary from higher energy to total loss of control [6].
   - **Sensor fault**: They represent sensor readings that are incorrect. Cable or lead breakage, broken connectors, and damaged cable insulation can all cause this. Sensors are a system output interface to the outside world, and they provide information about the system's activity and internal states. As a result, sensor faults may degrade the performance of systems that rely on data integrity to make decisions [8].
   - **Plant components fault**: They are faults that are not deemed actuator or sensor faults. They occur as a result of component structural damage, and they can impact the dynamical behavior of the system. The mathematical description or modeling of these flaws can be difficult at times, and much experimentation may be required before a model can be built.

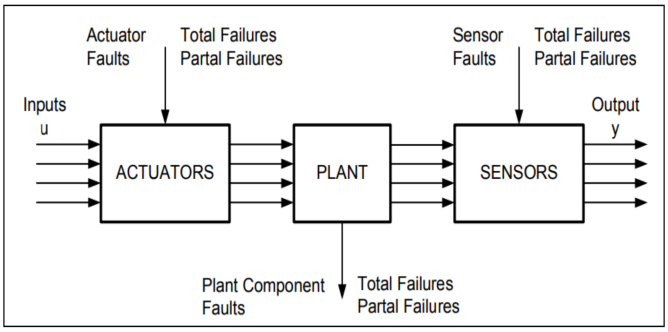


Figure 1.1: Fault types according to their place of occurrence.

2. Fault can also be categorized based on their time dependency:

   - **Abrupt or stepwise faults**: They occur instantaneously, often as a result of a hardware problem. Typical examples of physical defects that result in an abrupt fault are a stuck valve, a broken fan belt and a burnt electric motor.
   - **Incipient or drift-wise faults**: They represent slow in time parametric changes, often as a result of aging.
   - **Intermittent faults**: They are faults that appear and disappear repeatedly.

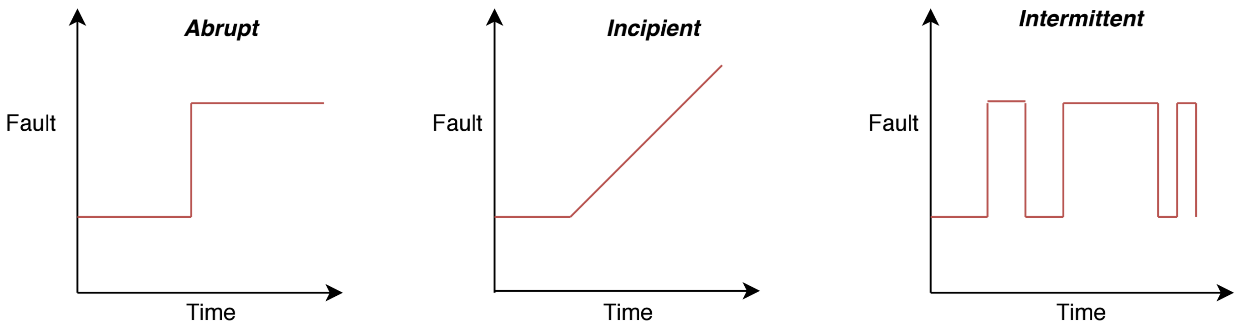


Figure 1.2: Time characteristics of fault

3. Depending on how the fault is added, it might be categorized as follows:

   - **Additive fault**: They commonly show as sensor offsets and can be quantitatively represented as:

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

   - **Multiplicative faults**: Several plant characteristics have been altered. These flaws represent the damage and malfunction of plant equipment in the following mathematical form:

$$Y(t) = (\alpha + \Delta\alpha(t))\, u(t) \tag{1.2}$$

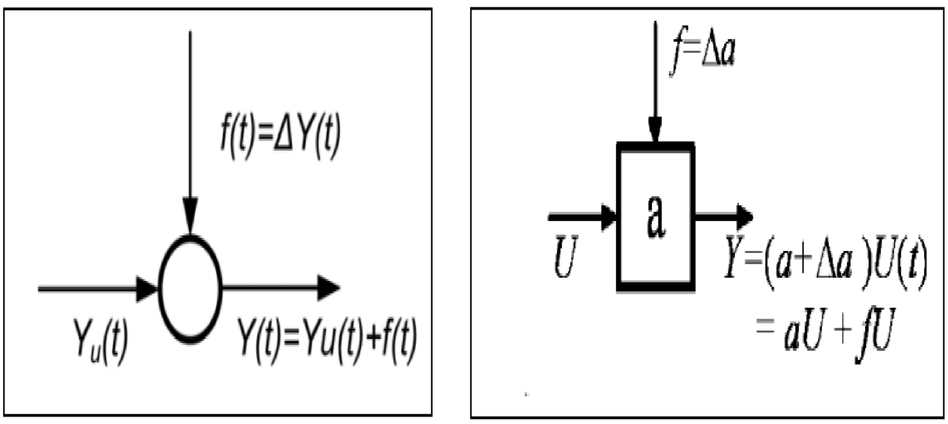


Figure 1.3: Fault types according to their mathematical form

1.4 Desirable characteristics of a fault diagnostic system

It is useful to identify a set of desirable features that a diagnostic system should contain as efficient methodology, and it is utilized to compare the different diagnostic systems. In FDD, there are several criteria that are considered:

- **Quick detection and diagnosis**: When a malfunction occurs, the diagnostic system should respond fast. High frequency effects will be detectable by a system intended to identify failure fast. As a result, the system is sensitive to noise and can cause false alarms in regular operation [9].
- **Isolability**: Isolability refers to the system diagnostic ability to discriminate between distinct types of failures. The majority of classifiers deal with redundant data in various forms. Classifier design has a finite amount of freedom. As a result, a classifier with a high degree of isolability will generally fail to reject the uncertainty of the model and vice versa [9].
- **Robustness**: A robust diagnostic system performance should be insensitive to the effects of various noise and modeling uncertainties [9].
- **Novelty identifiability**: A fault detection and diagnosis system should be able to determine whether a process is operating normally or abnormally, and if an abnormal situation develops, whether the cause is known or unknown.
- **Adaptability**: It is challenging to have systems that can be expanded. This would allow processes to adapt in response to changes in external inputs, structural alterations, and environmental circumstances. As a result, the diagnostic system should be flexible [9].
- **Explanation facility**: A diagnostic system should be able to identify where an issue began and how it spread throughout the system [9].
- **Storage and computational requirements**: This requirement is particularly important for the rapid deployment of diagnostic classifiers in real time. The systems should then be well-balanced in terms of high storage capacities and low computational complexity [9].
- **Modelling requirements**: The quantity of modeling necessary to construct a diagnostic classifier is a significant consideration. Modeling effort should be kept to a bare minimum for quick and easy deployment of real-time diagnostic classifiers [9].

Before making a definitive diagnosis, it is vital to understand the many transformations that process measurements go through. The following are the four most important locations where the diagnosing process can occur at any time:

- **Measurement space**: It is a space that contains all the FDD inputs systems, with no prior problem knowledge linking these measurements. They can be stated mathematically as: $x = [x_1, x_2, \ldots, x_n]$, where n denotes the number of measurements.

- **Feature space**: It is here that the measurements are processed in order to obtain meaningful process behavior information. This can be accomplished by employing prior knowledge of the situation. This space can be thought of as a collection of points $y = [y_1, y_2, \ldots, y_i]$, with y_i denoting the i^{th} feature retrieved from the previous data.

- **Decision**: The mapping from the feature space to the decision space is usually designated to meet some objective function. This transformation is achieved by either using a discriminant function or in some cases using simple threshold functions. The decision space is a set of points $d = [d_1, d_2, \ldots, d_k]$, where k is the number of decision variables, produced through appropriate feature space transformations [9] .

- **Class space**:It is the ultimate interpretation of the diagnostic system that the user receives. Threshold functions, template matching, and symbolic reasoning can all be used to convert the decision space to the class space. This space is made up of integers $c = [c_1, c_2, \ldots, c_m]$, with m denoting the number of failure classes and indicating which one a given measurement pattern belongs to, including the normal region [9].

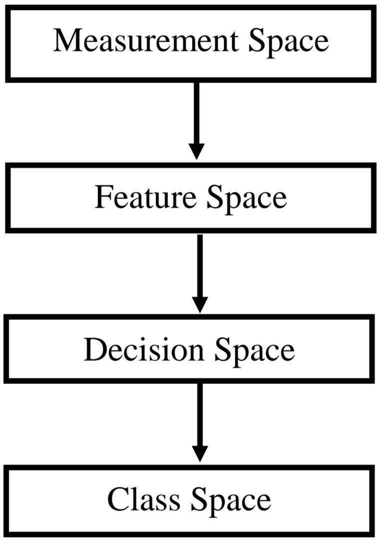


Figure 1.4: Transformation of measurement in a diagnostic system

1.5 Classification of fault diagnosis techniques

FDD methods can be classified into model-based methods and data-driven methods. The most important aspects in a diagnosis classifier are the type of knowledge and the type of diagnostic search strategy. The set of failures and the relationship between the observations (symptoms) and the failures are the basic information required for fault identification, or it could be gathered from previous experience with the procedure.

Prior knowledge is used in model-based approaches to identify disparities between model simulation results and actual operation measurement, the models are created using some physical knowledge about the process that is required to comprehend it. In data-driven approaches, Data can be translated and supplied as prior knowledge to a diagnostic system in a variety of ways, if-then expressions are widely used to summarize expert knowledge.

Both of model-based and data-driven approaches can be divided into Quantitative models and Qualitative models, Qualitative models use the qualitative relationships to detect and diagnose faults instead of quantitative mathematical equations [8][10].

The diagram shown in Figure 1.5 depicts the various FDD approaches:

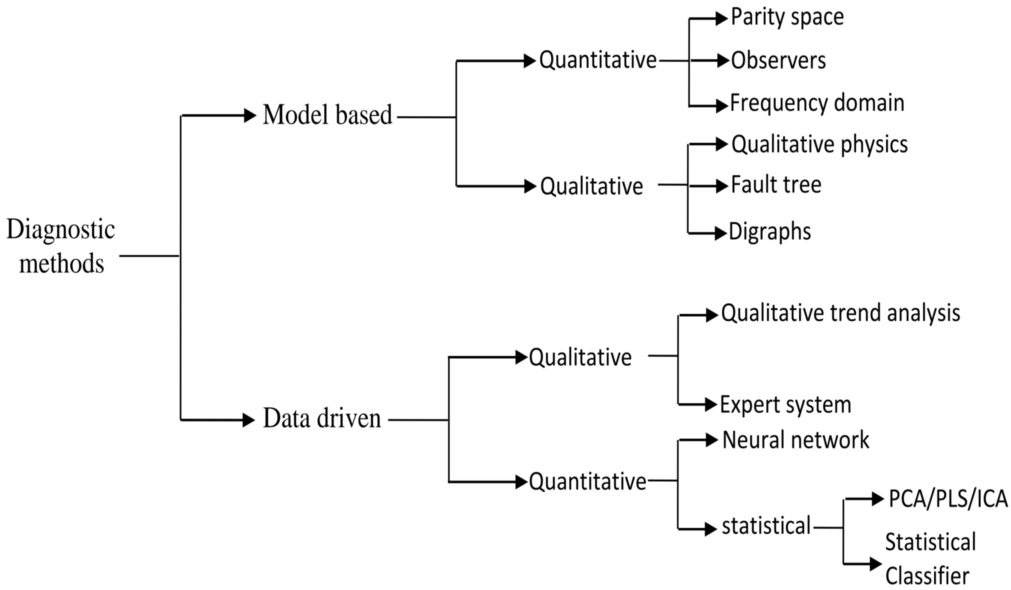


Figure 1.5: Fault diagnosis techniques

1.5.1 Model-based fault detection techniques

Model-based methods are based on analytical redundancy, which is achieved through functional dependence among the process variables and is provided by a site of algebraic equations or temporal relationships among the states and inputs/outputs of the system. Its main goal is to check the actual system behavior against the system model consistency. This approach assumes that the structure and parameters of the model are precisely known [10][11].

Quantitative model-based method

They are categorized according to the use of general input-output and state space models to generate residuals:

- **Observers**: It works by estimating the system outputs based on the measurements [8][10].
- **Parity space**: This method compares process behavior to a nominal process model. The goal is to ensure that the mathematical equations of the system are consistent by using actual data [8][10].
- **Frequency Domain**: Residuals are formed in the frequency domains by factoring the monitoring the transfer function of the system [8][10].

Qualitative model-based method

Declarative information is required for this technique, such as variable sign, variable tendencies (growing, decreasing, or constant), order, and/or relative magnitude [10].They are categorized into the following categories, depending on several types of qualitative knowledge utilized in defect diagnosis:

- **Qualitative physics approaches**: This method is based on a detailed understanding of all physical relationships and properties of a system. These links are then used to build and solve mathematical equations [12].
- **Fault tree approach**: It employs fault progression through a dynamic system that can be represented by fault trees, event trees, or causal networks. The association-based approach relies on established rules to characterize the relationships between problems and faulty system observations [8][13].
- **Digraphs (Causal model approaches)**: A digraph is a graph in which the nodes are connected by directed arcs. From the cause nodes, the arcs lead to the impact nodes. This approach has been the most widely used form of causal knowledge for process fault diagnosis [8][10]. It provides a very efficient way of representing qualitative models graphically.

1.5.2 Data-driven fault detection technique

This method necessitates declarative data such as variable sign, variable tendencies (increasing, decreasing, or constant), order, and/or relative magnitude. It gives information on the state of a system when the physical process is unknown or incomplete [14].

quantitative data-driven method

- **Qualitative trend analysis**: It can be used to explain many critical events that occur in the process, diagnose malfunctions, and predict future states, as it often provides valuable information that improves thinking about process behavior [14].
- **Expert systems**: An expert system is a highly specialized system that solves issues in a certain field of knowledge [15]. The main components in an expert system development include[14]:
  - 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.

qualitative data-driven method

Statistical and non-statistical approaches are two types of methods to extract quantitative data. They approach the diagnostic problem as a pattern recognition problem, with data points being classified into pre-determined groups [16][17].

1. **Multivariate statistical approaches**: They classify data based on prior knowledge of class distributions. These methods are based on the ability to compress data and reduce its dimensionality, allowing critical information to be preserved and analyzed more easily than the original large data set, and they can also handle noise and correlation to extract actual information effectively [14][18].
   - **Principal component analysis (PCA)**: It converts a set of observations of possibly correlated variables into a set of values of uncorrelated variables termed principal components via an orthonormal transformation.
   - **Partial least squares (PLS)**: is a dimensionality reduction technique that aims to maximize the covariance between the predictor (independent) matrix X and the predicted (dependent) matrix Y for each component of the reduced space [19].
   - **Independent component analysis (ICA):** It is a method for uncovering hidden factors that exist beneath a set of random variables, measurements, or signals. It tries to decompose a multivariate signal into non-Gaussian signals that are independent [20].
2. **Statistical classifier approach**: Fault diagnosis is basically a classification problem, and so can be modeled using a traditional statistical pattern recognition framework [14][21][22].
3. **Neural network approach**: Fault detection has been successfully implemented using neural networks. This method does not necessitate a specific understanding of process structure. In general, it can be divided into two categories:
   - The architecture of the network such as sinusoidal, radial basis, and so on.
   - Learning strategy, such as supervised and unsupervised learning [10][14].

A comparison of numerous diagnostic procedures may be seen in Table 1.1:

Table 1.1: Comparison between various diagnostic techniques

	Observer	Diagraphs	Expert system	QTA	PCA	Neural network
Quick detection and diagnosis	✓	?	✓	✓	✓	✓
Isolability	✓	x	✓	✓	✓	✓
Robustness	✓	✓	x	✓	✓	✓
Novelty idenfiability	?	✓	x	?	✓	✓
Classsification error	x	x	x	x	x	x
Abaptability	x	✓	✓	?	x	x
Explanation facility	x	✓	✓	✓	x	x
Modelling requirement	?	✓	✓	✓	✓	✓
Storage and computation	✓	?	✓	✓	✓	✓
Multiple fault identifiability	✓	✓	x	x	x	x

1.6 Conclusion

In this chapter, we looked at some terminology connected to fault detection and diagnosis, as well as the numerous sorts of problems that can be detected in a system and the various methods used to identify and diagnose these faults, before concluding with a brief comparison of some methods. In the next chapter, we will concentrate more on the principal component analysis PCA technique and its dynamic versions, DPCA and IV-DPCA.

Chapter 2

Principal Component Analysis and its Dynamic versions

2.1 Introduction

The manufacturing of sensors has increased enormously on both macroscopic and microscopic scales, they have been inserted in almost every corner of our daily lives. In comparison to model-based approaches for fault detection and diagnosis, the advancement of electronics in general and sensors in particular (which allows for the redundancy of measurement data) has given data-driven methods an advantage because it is a straightforward nonparametric method that helps to extract useful information from complicated data sets.

Principal component analysis (PCA) has been termed one of the most utilized, researched, and powerful techniques among data-driven methodologies [23]. Since PCA is considered the most used technique in FDD, different approaches and methods were developed using it such as its dynamic version DPCA and the developed IV-DPCA.

2.2 Principal Component Analysis (PCA)

PCA is one of the qualitative data driven methods. It converts a set of observations of possibly correlated variables into a set of values of uncorrelated and ordered variables termed principal components via an orthonormal transformation [24].

Principal component analysis is one of the oldest and best known techniques of multivariate analysis. It was first introduced by Pearson (1901), and developed independently by Hotelling (1933) [25].

2.2.1 General description

Principal component analysis main goal is to eliminate correlations between connected variables in the original data set and lower its dimensionality, while retaining as much as possible of the variation present in the data set [26].

The following example depicts PCA using two-dimensional data X_1 and X_2. The best-fit line (or axis in this case) corresponds to the first principal component PC_1 that passes through many points; in PCA, the more correlated the original data is, the better this line will explain the actual values of the observed measurements; this line will best explain all the observations with the least amount of residual error. To put it another way, the line follows the maximum variance of the projections.

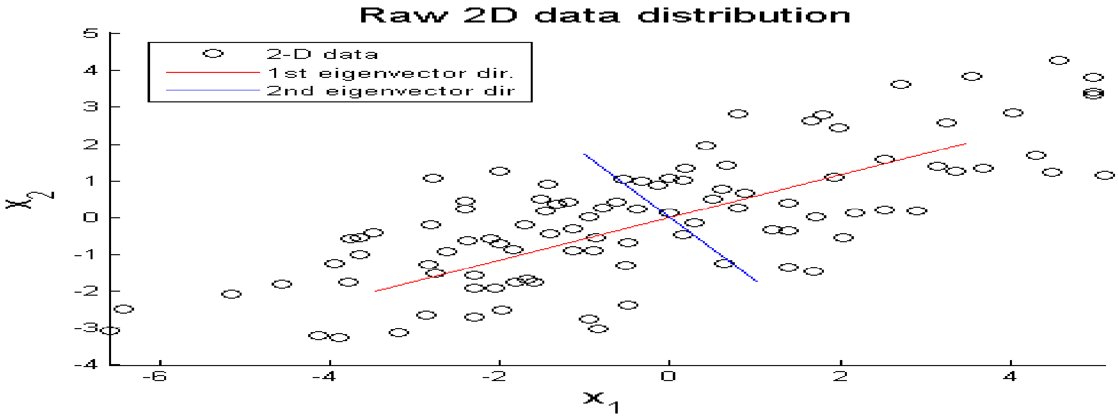


Figure 2.1: Illustration of PCA in 2-D

2.2.2 Mathematical description

Naturally, reducing the number of variables in a data collection reduces accuracy; the answer to dimensional reduction is to trade a little accuracy for simplicity. PCA goal is to reduce a data set number of variables while maintaining as much information as possible.

Let X be our original data matrix with X_i representing the i^{th} column describing the i^{th} random variable.

$$X = [X_1, X_2, \dots, X_m] \tag{2.1}$$

It has the dimension $n \times m$ ($X \in \mathbb{R}^{n\times m}$), where n denotes the number of observations and m denotes the number of random variables that make up the data (number of sensors).

First, our original data should be normalized (standardized), so that all variables have an equal impact on the analysis. To get the normalized data X_n, we need to subtract the mean μ of each random variable from the variable column and divide each value of that column by the associated random variable standard deviation σ.

$$X_n = [\tfrac{X_1-\mu_1}{\sigma_1}, \tfrac{X_2-\mu_2}{\sigma_2}, \dots, \tfrac{X_n-\mu_m}{\sigma_m}] \tag{2.2}$$

The corresponding covariance matrix is computed in the second step [27]. The goal of this step is to figure out how the variables in the original data set varying from the mean in relation to one another.

$$Cov(X_n) = \frac{1}{n-1} X_n^T X_n \tag{2.3}$$

A symmetric square matrix of dimension $m \times m$ is used as the covariance matrix. It is also known as the variance-covariance matrix because the diagonal entries indicate each variable variance, while the remaining entries show how variables fluctuate in relation to one another, which is called the covariance term.

$$Cov(X_n) = \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1}\sigma_{X_2} & \dots & \sigma_{X_1}\sigma_{X_m} \\ \sigma_{X_2}\sigma_{X_1} & (\sigma_{X_2})^2 & \dots & \sigma_{X_2}\sigma_{X_m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_m}\sigma_{X_1} & \sigma_{X_m}\sigma_{X_2} & \dots & (\sigma_{X_m})^2 \end{bmatrix} \tag{2.4}$$

Next we move to the calculation of the covariance matrix eigenvalues and eigenvectors (SVD), which provide us with important information about our data. The eigenvectors that arise are of unit length (or orthonormal) which is crucial for PCA. These eigenvectors are new variables that will be employed in the process of modifying the basis in order to create a more straightforward and accurate data representation.

$$Cov(X_n) = P^T \Lambda P \tag{2.5}$$

P is a square $m \times m$ matrix representing the loading matrix, which contains all the principal components in a column-wise format, and Λ is a diagonal $m \times m$ matrix of ordered eigenvalues, from highest to lowest ($\lambda_1 > \lambda_2 > \cdots > \lambda_m$).

$$P = \begin{bmatrix} P_1 & P_2 & \dots & P_m \end{bmatrix} \tag{2.6}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_m \end{bmatrix} \tag{2.7}$$

There are equivalent principal components $P_1, P_2, \dots, P_m$, with P_1 being more significant than P_2 and so on. P regroups orthonormal eigenvectors in the sense that:

$$PP^T = I \tag{2.8}$$

so,

$$P^T = P^{-1} \tag{2.9}$$

Following the creation of the new basis matrix P, the original data can be transformed into a new representation known as the score matrix T, which is calculated as follows:

$$T = X_n P \tag{2.10}$$

In general,the first few principal components are sufficient to represent the data, since the majority of the initial variation is compressed into them due to the presence of distinct linear correlations among original variables. Instead of considering all PCs, a feature vector is created by picking the first **a** PCs and eliminating the other **m-a** PCs that are of lesser importance.

The goal is to reorient the data from the original axes to the ones indicated by the primary components using this created feature vector (hence the name Principal Components Analysis). The altered variables space $\hat{T}$ will be reduced in dimension as a result of this. where $\hat{T}$ is:

$$\hat{T} = X_n P_a \tag{2.11}$$

$\hat{T}$ is a $n \times a$ matrix and X_n is a $n \times m$ matrix.

$$P_a = [P_1 \quad P_2 \quad \dots \quad P_a] \text{ with } a < m \tag{2.12}$$

Reconstructing the data as a final step is critical, especially when employing the PCA for data reduction and producing the error matrix E that will be used in the squared prediction error (SPE or Q) analysis later on.

Because no information has been lost, the reconstructed data will be identical to the original only when all PCs are used.

$$X_n = TP^T \tag{2.13}$$

Using the feature vector P_a reduces the dimension of the transformed variables' matrix, resulting in a little loss of information and leading the reconstructed data to differ slightly from the original. The relations are expressed as follows:

$$Cov(X_n) = \begin{bmatrix} P_a^T & P_{m-a}^T \end{bmatrix} \begin{bmatrix} \Lambda_a & 0 \\ 0 & \Lambda_{m-a} \end{bmatrix} \begin{bmatrix} P_a \\ P_{m-a} \end{bmatrix} \tag{2.14}$$

$$\hat{X} = \hat{T} P_a^T = X_n P_a P_a^T = X_n C_a \tag{2.15}$$

$$\tilde{T} = X_n P_{m-a} \tag{2.16}$$

$$\tilde{X} = \tilde{T} P_{m-a}^T \tag{2.17}$$

The error is expressed as follows:

$$E = X - \hat{X} = X(I - C_a) \tag{2.18}$$

The question that rises now is how to select the right number of retained component **a** when we just want to maintain a few and keep the majority of the initial variation, that is what the next subsection is going to be about.

2.2.3 Model dimension selection

Many procedures have been proposed for determining the number of components to be retained in a PCA model. Among these methods there are [28][29], cross validation criteria (CV) [30], parallel analysis [31] and cumulative percent variance (CPV) [32], which is used in this study to determine the optimal number of maintained principal components. The following formula is used to calculate the cumulative percent variance:

$$CPV = \frac{\sum_{i=1}^{a} \lambda_i}{\sum_{j=1}^{m} \lambda_j} \times 100\%. \tag{2.19}$$

The percentage of the created model, where, i is the number of maintained PCs with their sum of variances greater than a specific percentage of the total variance [33].

2.2.4 Fault detection Indices

The PCA model of the process is created based on normal operating process data and then used to check new measurement data for fault detection. The residuals are then put to a statistical test to see if they are significant, based on the discrepancies between the fresh measurement data and their predictions to the created model. To reflect the variability in principal component subspace and the residual subspace, Hotelling's T^2 statistic and the squared prediction error (SPE) or Q statistic are commonly utilized [34]. For monitoring the principal and residual space simultaneously, a third monitoring index ϕ statistic is introduced, which is a combination of the T^2 and Q indices, weighted by their control limits [35].

Hotelling T^2 statistic

The variability in the principal component's subspace is measured by the Hotelling's T^2 statistic. It is determined by the initial eigenvalues that capture the most variation of data. We can calculate it as follows:

$$T^2(k) = \hat{t}(k)^T \Lambda_a^{-1} \hat{t}(k) = \sum_{i=1}^{a} \frac{t_i^2(k)}{\lambda_i} \tag{2.20}$$

If the corresponding T^2 is less than or equal to the threshold value T_{th} that corresponds to a specific confidence level $(1-\alpha) \times 100\%$, the process is considered to be in a healthy state [36]. The threshold T_{th} is determined as follows:

$$T_{th}^2 = \frac{a(n^2-1)}{n(n-a)} \times \mathcal{F}_{a,n-a,\alpha} \tag{2.21}$$

Where $\mathcal{F}_{a,n-a,\alpha}$ is an $\mathcal{F}$-distribution with degrees of freedom a,$n-a$ and a significance level α. The number of PCs vectors kept in the PCA model is a, and the number of samples used to develop the model is n [36].

Square prediction error SPE or Q-statistic

The squared prediction error SPE or Q of the residuals for a new observation can be used to discover new occurrences. The sum of the squares of the residuals is used to calculate the Q statistic. In another way, the Q statistic provides a measure of the amount of variance that the PCA model does not capture. It has the following equation:

$$Q = diag(EE^T) \tag{2.22}$$

where E is :

$$E = X_n - \hat{X_n} = X_n - X_n P_a P_a^T = X_n(I - C_a) \tag{2.23}$$

$I_{m \times m}$ is the identity matrix and $C_a = P_a P_a^T$.

If the corresponding Q is less than or equal to the threshold value Q_{th} that corresponds to a specific confidence level $(1-\alpha) \times 100\%$, the process is deemed to be in a healthy (normal) state. If the value of the Q statistic, on the other hand, exceeds the threshold value, the system is considered to be faulty (abnormal) [37] [38].

The threshold Q_{th} is calculated as follows:

$$Q_{th} = \theta_1 \left[\frac{h_a c_a \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{2.24}$$

$$\text{Where } \theta_i = \sum_{j=a+1}^{m} \lambda_i^j \text{ and } h_0 = \frac{2\theta_1 \theta 3}{3\theta_2^2} \text{ for } i = 1,2,3$$

and c_a is $(1-\alpha) \times 100\%$ percentile for a standard normal distribution.

ϕ statistic

Yue and Qin [35] were the first to propose the composite index ϕ statistic. The index is a hybrid of T^2 and Q statistics that provides information on variability over the entire measurement space, it used to monitor both the principal and residual subspaces at the same time.

$$\phi = \frac{T^2}{T^2_{th}} + \frac{Q}{Q_{th}} \tag{2.25}$$

In a similar way, the ϕ statistic threshold can be determined by [36]:

$$\phi_{th} = g\chi^2_{h,\alpha} \tag{2.26}$$

g and h are given as follow [34][36]:

$$g = \frac{\text{trace}(Cov \times w)^2}{\text{trace}(Cov \times w)} \tag{2.27}$$

$$h = \frac{[\text{trace}(Cov \times w)]^2}{\text{trace}(Cov \times w)^2} \tag{2.28}$$

2.2.5 Monitoring performance

The statistical methods are evaluated using the following performance indices: fault detection time delay (DTD) is time required to indicate the fault after it occurs. False alarms rate (FAR) and missed detection rate (MDR).

The goal of a reliable monitoring strategy is to achieve the fastest detection, as well as the lowest FAR and MDR.

The following are the equations of them:

$$DTD = \text{time of detection - time of occurrence} \tag{2.29}$$

$$FAR = \frac{\text{number of faults under the healthy state}}{\text{number of samples under the healthy state}} \times 100\% \tag{2.30}$$

$$MDR = \frac{\text{number of healthy samples under the faulty state}}{\text{number of the samples under the faulty state}} \times 100\% \tag{2.31}$$

For the sake of comparison between different techniques and determining the best among them , a cost function J for each fault index is evaluated:

$$\begin{cases} J_{T^2} = & \frac{1}{3}\left(FAR_{T^2} + MDR_{T^2} + DTD'_{T^2}\right) \\ J_Q = & \frac{1}{3}\left(FAR_Q + MDR_Q + DTD'_Q\right) \\ J_\phi = & \frac{1}{3}\left(FAR_\phi + MDR_\phi + DTD'_\phi\right) \end{cases} \tag{2.32}$$

For the reason that each of the aforementioned cost functions will be minimized in the unit box, DTD needs to be normalized ($DTD' \in [0, 1]$)

2.2.6 Limitations of PCA

PCA has several advantages when it comes to dimension reduction, since it uses simple mathematical computation.

However, the trade-off between dimensional reduction and information loss comes at a cost. The loss of information is an unavoidable feature of PCA. Furthermore, classical PCA has linear and static characteristics, which is not very convenient to deal with non-linear systems or dynamic ones or both.

This limitations' problem have motivated various researchers to develop several approaches. One of those approaches is kernel PCA, which extends linear PCA to deal with nonlinearities [39][40][41]. Another approach is dynamic PCA, which will be focused in the next section.

2.3 Dynamic Principal Component Analysis (DPCA)

2.3.1 General Description

When PCA is applied directly to the data matrix X, a linear static model is created. When the data contains dynamic information, PCA will reveal a linear static approximation rather than the exact relationships between the variables [42]. A static PCA model can be used to detect and isolate disturbances in a dynamic system, as it is previously explained. However, because the data contradict the condition of time independence, the statistical foundation of this approach is lost because auto-correlation and maybe cross-correlation will exist between the altered variables (scores) and could produce misleading results. A solution for this problem is the use of dynamic PCA or DPCA with appending lagged data [1][43][44].

Dynamic PCA modeled on time lagged data is more useful for detecting disturbances fast in dynamic systems [1] since the noise and scores subspaces of the built model will not contradict the condition of time independence [43][44][45].

2.3.2 Mathematical approach

The DPCA approach is based on how to extract the linear relations before using PCA to dynamic systems. The non-trivial solution of the following equation is used to identify linear relationships between variables:

$$Xb = 0 \tag{2.33}$$

There are $m - r$ linear relations where the rank of X is r. Right singular vectors associated with zero singular values which are obtained from SVD are the answer to equation (2.33) [46].

Because of the presence of noise, zero eigenvalues or singular values are uncommon in practice. Random noise in the data causes variance, which causes the zero singular values to be converted to non-zero but small values. The right singular vectors associated with the small singular values will be the best solution to equation (2.33).

Since the current values of variables in a dynamic system are influenced by their previous values, at least there is another one linear relationships between $X(k)$ and $X(k-1)$ that must be found. The connection will be the noise subspace of the following equation if all dynamic relationships are first order systems [1].

$$[X(k)X(k-1)]\, b = 0 \tag{2.34}$$

In General:

$$X_A(l)b = 0 \tag{2.35}$$

Where:

$$X_A(l) = [X(k)X(k-1)\dots X(k-l)] = \begin{bmatrix} X^T(1) & X^T(0) & \dots & X^T(1-l) \\ X^T(2) & X^T(1) & \dots & X^T(2-l) \\ \vdots & \vdots & \ddots & \vdots \\ X^T(n) & X^T(n-1) & \dots & X^T(n-l) \end{bmatrix} \tag{2.36}$$

The data matrix is made up of time shifted duplicate vectors, which is effectively the same as the original PCA technique. Both static and dynamic relations should emerge in the noise subspace with modest singular values if the number l is chosen correctly.

2.3.3 Design procedure

By using dynamic PCA to uncover static and dynamic relationships from data, one must first establish the required number of time lags l and the main components or linear relations [47]. The guidelines for choosing the appropriate number of components and system order are as follows:

1. Set $l = 0$
2. Form data matrix $X = [X(k)X(k-1)\dots X(k-l)]$
3. Perform SVD at each level l and determine a the number of PCs retained.
4. Deduce r the number of relations such that $r = (\text{number of variables} - a)$
5. calculate r_n the number of new relations as follows:

$$r_n(i+1) = r(i+1) - \left(r(i) + \sum_{j=0}^{i} r_n(j) \right) \tag{2.37}$$

6. The procedure ends when $r_n \leq 0$ and the number of time-lag is the one corresponding to the previous level.
7. Apply PCA to the resultant augmented matrix and build the FDD model.

2.4 The proposed Interval-Valued Dynamic Principal Component Analysis (IV-DPCA)

In virtually all circumstances, the true value can not be measured, and the data obtained in a dynamic process are merely an approximation provided by the sensors, making it imperfect to build an FDD model due to measurement errors or experimental conditions[48]. In this thesis, a new technique called Interval-Valued Dynamic Principal Component Analysis (IV-DPCA) is suggested to deal with this bottleneck.

Before going any further, it is preferable to check the basic interval arithmetic given in Appendix A.

Figure 2.2 illustrates how the PCA for interval-valued data works, where the angle between the minimum eigenvector $\underline{P}$ and the maximum eigenvector $\overline{P}$ represents the fluctuation given to the data.

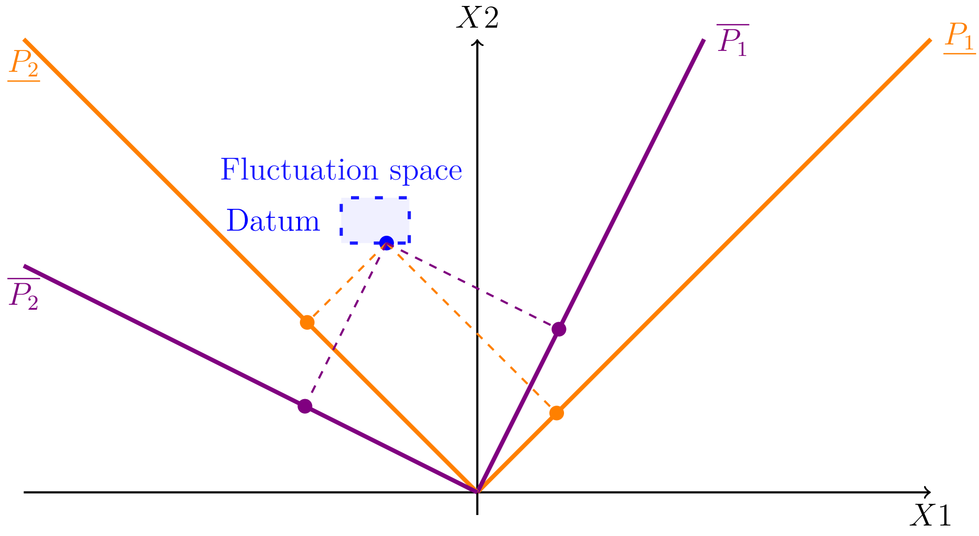


Figure 2.2: Illustration of IV-PCA in 2-D

2.4.1 Generating Interval-Valued Data

In order to generate IVD, several methods can be applied such as the measurement error, mean and standard deviation and repeated experiment method.

Interval-valued data using error of measurement

Let $X \in \mathbb{R}^{n\times m}$ be the data matrix containing n samples of m process variables, with $x_j(k)$ denoting the j^{th} variable. In order to create interval-valued data from single-valued data, we assume that its variation is limited and can be represented by an interval of the form $\left[\underline{x_j}(k), \overline{x_j}(k)\right]$, where $\underline{x_j}$ and $\overline{x_j}$ represent the lower and upper limits respectively for the j^{th} interval-valued variable with respect to the k^{th} observation. We define $\delta x_j(k)$ as the error in measurement:

$$\delta x_j(k) = x_j^m(k) - x_j^r(k) \tag{2.38}$$

Where:
$x_j^m(k)$: The measured value.
$x_j^r(k)$: The real value.

The new global interval matrix $[X]$ can be represented as follows:

$$[X] = \begin{pmatrix} \left[\underline{x_1}(1), \overline{x_1}(1)\right] & \cdots & \left[\underline{x_m}(1), \overline{x_m}(1)\right] \\ \vdots & \ddots & \vdots \\ \left[\underline{x_1}(n), \overline{x_1}(n)\right] & \cdots & \left[\underline{x_m}(n), \overline{x_m}(n)\right] \end{pmatrix} \tag{2.39}$$

With:

$$\begin{cases} \underline{x_j}(k) = x_j^m(k) - \delta x_j(k) \\ \overline{x_j}(k) = x_j^m(k) + \delta x_j(k) \end{cases} \tag{2.40}$$

Interval-valued data using the mean and standard deviation

To create the IVD, we begin with a data matrix $X \in \mathbb{R}^{n\times m}$ that comprises n samples of m process variables, with $x_j(k)$ being the j^{th} variable. The original data is then multiplied by a random uniformly distributed data matrix δx, with a mean of 0.1 and a variance of 1.

$$X_r = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} + \begin{pmatrix} \delta x_{11} & \cdots & \delta x_{1m} \\ \vdots & \ddots & \vdots \\ \delta x_{n1} & \cdots & \delta x_{nm} \end{pmatrix} \tag{2.41}$$

The generated data matrix is then normalised using the original data X mean and variance. The goal of data normalisation is to convert the values of numeric columns in the dataset to a common scale while preserving disparities in value ranges.

$$X_n = \frac{X_r - \text{ mean } (X)}{STD(X)} \tag{2.42}$$

- The mean of an $n \times m$ matrix X is the matrix $M^{1\times m}$ where:

$$M_{1j} = \frac{\sum_{i=1}^{n} x_{ij}}{n} \tag{2.43}$$

- The standard deviation of a matrix $X^{n\times m}$ is defined by a matrix $S^{1\times m}$ where:

$$S_{1j} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} |X_{ij} - M_{1j}|^2} \tag{2.44}$$

Finally, the interval-valued data is obtained from the normalized data matrix and have the following form:

$$X_f = \begin{pmatrix} \left[\underline{x_{1,1}}, \overline{x_{1,1}}\right] & \cdots & \left[\underline{x_{1,m}}, \overline{x_{1,m}}\right] \\ \vdots & \ddots & \vdots \\ \left[\underline{x_{n,1}}, \overline{x_{n,1}}\right] & \cdots & \left[\underline{x}_{n,m}, \overline{x_{n,m}}\right] \end{pmatrix} \tag{2.45}$$

Where: $\underline{x_{i,j}} = \min\left(X_{i,j}(k)\right); \overline{x_{i,j}} = \max\left(X_{i,j}(k)\right)$.

2.4.2 Dimensionality Reduction

By recreating the covariance matrix, PCA seeks to minimise the dimensions of a huge dataset. And the number of main components influences each step of PCA-based fault identification as well as its performance. Various approaches are used to determine the total number of main components, Some of them have already been discussed before (subsection 2.2.3).

These methods were created for single-valued PCA analysis as well as they can also be used for IPCA or IV-DPCA but interval-valued data must be converted to single-valued data before this can be done. There are various strategies for doing so, including the norm and mid-point.

Data Transformation Using Norm Method

The eigenvalues calculated from interval-valued data are in interval form.

$$\lambda = \begin{pmatrix} \underline{\lambda_1} & \overline{\lambda_1} \\ \underline{\lambda_2} & \overline{\lambda_2} \\ \vdots & \vdots \\ \underline{\lambda_m} & \overline{\lambda_m} \end{pmatrix} \tag{2.46}$$

By determining the norm of each row of the matrix above, the interval-valued eigenvalues are turned into single-valued eigenvalues. The eigenvalues matrix λ_{norm} that results is of the form:

$$\lambda_{\text{norm}} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix} ; \text{where } \lambda_i = \sqrt{\frac{\underline{\lambda_i}^2 + \overline{\lambda_i}^2 + \underline{\lambda_i}\overline{\lambda_i}}{3}} \tag{2.47}$$

Data Transformation Using the Mid-point

This method employs the mid-points of the eigenvalues to generate single-valued ones which may then be used to compute the number of retained components using one of the methods previously outlined. The resulting single-valued eigenvalues matrix λ_{mid} is of the form:

$$\lambda_{\text{mid}} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix}, \text{ where } \lambda_i = \frac{\underline{\lambda_i} + \overline{\lambda_i}}{2} \tag{2.48}$$

2.4.3 Fault indicators

Various statistical techniques are employed to quantify the variation in distinct subspaces in order to determine the presence of abnormal behaviour in the system. IV-DPCA uses the same indicators as the traditional PCA: T^2, Q and ϕ, with the inclusion of interval-valued data management approaches.

Interval Hotelling T^2 statistic ($[T^2]$)

The interval T^2 statistic ($[T^2]$) is similar to the traditional PCA. The first a components, which collect the majority of the initial variance, determine the variation in the principle subspace. A lower and upper bound exist for the $[T^2]$.

$$\left[T^2\right] = \left[\underline{T}^2, \overline{T}^2\right] \tag{2.49}$$

The interval principle scoring matrix $[T_a]$ must be determined before the upper and lower bounds can be calculated.

$$[T_a] = \left[\left[\underline{t_1}, \overline{t_1}\right], \left[\underline{t_2}, \overline{t_2}\right], \cdots, \left[\underline{t_a}, \overline{t_a}\right]\right] \tag{2.50}$$

The lower limit is represented by the $\underline{t_i}$ index, while the upper bound is represented by the $\overline{t_i}$ index, with t_i being the $i^{th} n \times 1$ vector. The T^2 interval is then calculated as follows:

$$\begin{cases} \underline{T}^2(k) = \underline{t}^T(k)\Lambda_a^{-1}\underline{t}(k) \\ \overline{T}^2(k) = \overline{t}^T(k)\Lambda_a{}^{-1}\overline{t}(k) \end{cases} \tag{2.51}$$

The threshold computation for the Interval statistic is done for both $\underline{T}^2$ and $\overline{T}^2$ boundaries, resulting in $\underline{T_{th}}^2$ and $\overline{T_{th}}^2$ respectively.

The system is considered to be faulty when both $\underline{T}^2$ and $\overline{T}^2$ exceeds simultaneously their corresponding thresholds.

Interval Square predict error statistic ($[Q]$)

The variance among the residual subspace is calculated using Q, which is similar to the conventional single-valued Q. The primary distinction is that the $[Q]$ generates an interval with a lower bound $\underline{Q}$ and a upper bound $\overline{Q}$. that correspond to the lower and upper residuals respectively, which constitute the interval error (residual) matrix $[E]$.

$$[Q] = [\underline{Q}, \overline{Q}] \tag{2.52}$$

$$[E] = [\underline{E}, \overline{E}] \tag{2.53}$$

The following equations can be used to calculate the interval Q bounds:

$$\begin{cases} \underline{Q} = \text{diag}\left(\underline{E}\underline{E}^T\right) \\ \overline{Q} = \text{diag}\left(\overline{E}\overline{E}^T\right) \end{cases} \tag{2.54}$$

Both $\underline{Q}$ and $\overline{Q}$ are $n \times 1$ column vectors, with each entity representing the SPE value for that observation; and both $\underline{E}$ and $\overline{E}$, which represent the residual lower and upper matrices respectively, are of dimension $n \times m$. The residuals interval $[E]$ is calculated as follows, following the same principle as mentioned before (subsection 2.2.4).

$$[E] = X \times ([I] - [C_a]) = X \times \left([I] - [P_a] \times [P_a]^T\right) \tag{2.55}$$

$[I]$ represents the $m \times m$ interval identity matrix and $[P_a]$ represents the interval principle components matrix.

When dealing with the Interval statistic, the lower and upper $[Q]$ bounds are determined, yielding lower and upper thresholds values $\underline{Q_{th}}$ and $\overline{Q_{th}}$ that can be computed for both $\underline{Q}$ and $\overline{Q}$ with a given confidence level as mentioned before (subsection 2.2.4).

The system is considered to be faulty when both $\underline{Q}$ and $\overline{Q}$ exceeds simultaneously their corresponding thresholds.

Interval ϕ statistic ($[\phi]$)

As previously stated in subsection 2.2.4, The Interval $[\phi]$ is reliant on both $[T^2]$ and $[Q]$ for interval data, as stated by the following equation:

$$[\phi] = \frac{[T^2]}{[T^2_{th}]} + \frac{[Q]}{[Q_{th}]} \tag{2.56}$$

With $[T^2_{th}]$ and $[Q_{th}]$ represents the interval thresholds for $[T^2], [Q]$ respectively.

The system is considered under healthy state if at least one of $[\phi]$ indicator is less than its corresponding threshold.

2.4.4 Interval-Valued Dynamic PCA Procedure

The main principles of Interval Valued PCA has been tackled so far. Based on those principles the proposed IV-DPCA is build.

The flowchart represented in Figure 2.3 and the giving IV-DPCA algorithm summarize the whole procedure.

IV-DPCA proposed algorithm.

Output:

- $[Imean, Istd]$.
- $[\underline{\lambda_a}, \overline{\lambda_a}]$ & their corresponding $[\underline{P_a}, \overline{P_a}]$
- The thresholds $[\underline{T^2{}_{th}}, \overline{T^2{}_{th}}]$, $[\underline{Q_{th}}, \overline{Q_{th}}]$ and $[\underline{\phi_{th}}, \overline{\phi_{th}}]$.

Input:

Load the data $X^* \in \mathbb{R}^{n\times m}$.
Reshape X^* into $x \in \mathbb{R}^{\frac{n}{j}\times m\times j}$.
For $i = 1 : j$ **do**
$x_i = x(\frac{n}{j}, m, i)$
Compute l using the design procedure mentioned in section 2.3.3 and fix it to its smallest value.
Form the augmented matrix $x_{aug,i} = [x_i(0)x_i(1)\cdots x_i(l)]$
Compute the mean μ and the standard deviation σ of $x_{aug,i}$ and store their values.
Normalize $x_{aug,i}$ such that $x_{n,i} = \frac{x_{aug,i}-\mu}{\sigma}$
compute the covariance matrix $\Sigma_i = cov(x_{n,i})$
Perform SVD on the covariance matrix $P\Lambda P^T = \text{SVD}(\Sigma_i)$
Store the eigen-pairs (Λ_i, P_i)
End for
Find $[\underline{\lambda}, \overline{\lambda}]$ and their corresponding $[\underline{P}, \overline{P}]$
Use one of the dimension reduction techniques mentioned in section 2.4.2.
Compute the number of the retained components using
$CPV(a) = 100 \times \frac{\sum_{i=1}^{a}\lambda_i}{\sum_{i=1}^{m}\lambda_i}\%$
Compute $[\underline{T}^2, \overline{T}^2]$, $[\underline{Q}, \overline{Q}]$ and their thresholds $[\underline{T^2_{th}}, \overline{T^2_{th}}]$ and $[\underline{Q_{th}}, \overline{Q_{th}}]$ respectively.
Compute $[\underline{\phi}, \overline{\phi}]$ and their corresponding thresholds $[\underline{\phi_{th}}, \overline{\phi_{th}}]$
Compute [Imean,Istd] such that $Imean = Average(\mu)$ and $Istd = Average(\sigma)$
Return:

- $[\underline{T^2_{th}}, \overline{T^2_{th}}]$, $[\underline{Q_{th}}, \overline{Q_{th}}]$ znd $[\underline{\phi_{th}}, \overline{\phi_{th}}]$
- $[\underline{\lambda_a}, \overline{\lambda_a}]$ and $[\underline{P_a}, \overline{P_a}]$
- $[Imean, Istd]$

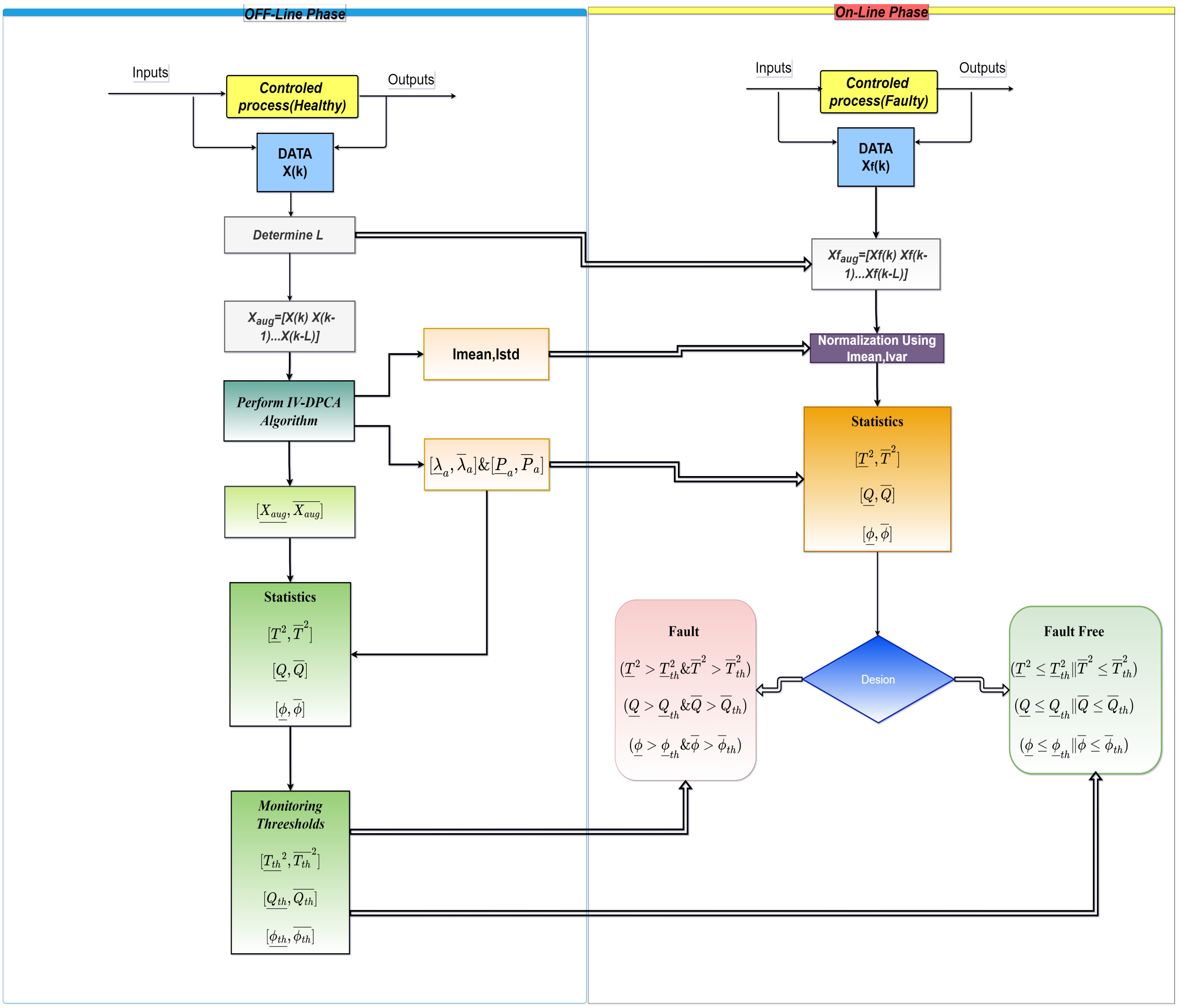


Figure 2.3: IV-DPCA Flowchart

2.5 Simulation Example

In order to put what has been tackled so far into practice, let us consider the following example:

$$z(k) = \begin{bmatrix} 0.118 & -0.191 \\ 0.847 & 0.264 \end{bmatrix} z(k-1) + \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix} u(k-1)$$

$$y(k) = z(k) + v(k)$$

Where u is the correlated input:

$$u(k) = \begin{bmatrix} 0.811 & -0.226 \\ 0.477 & 0.415 \end{bmatrix} u(k-1) + \begin{bmatrix} 0.193 & 0.689 \\ -0.320 & -0.749 \end{bmatrix} w(k-1)$$

- The input w is a random noise with $\mu_w = 0$ and $\sigma_w = 1$.
- $v(k)$ random noise with $\mu_v = 0$ and $\sigma_v = 0.1$.

The data matrix X with 1000 samples consists of:

$$X = [y^T u^T]$$

PCA application

First step is to calculate Cov the covariance matrix of the data matrix X using equation (2.3) .

$$Cov = \begin{bmatrix} 1 & -0.096 & 0.124 & 0.541 \\ -0.096 & 1 & 0.770 & 0.388 \\ 0.124 & 0.770 & 1 & 0.084 \\ 0.541 & 0.388 & 0.084 & 1 \end{bmatrix}$$

Second step is to compute the eigenpairs of the covariance matrix Cov:

$$\Lambda = \begin{bmatrix} 1.956 & 0 & 0 & 0 \\ 0 & 1.395 & 0 & 0 \\ 0 & 0 & 0.602 & 0 \\ 0 & 0 & 0 & 0.047 \end{bmatrix} \qquad P = \begin{bmatrix} 0.270 & 0.690 & 0.559 & -0.374 \\ 0.622 & -0.351 & -0.292 & -0.636 \\ 0.576 & -0.357 & 0.514 & 0.525 \\ 0.456 & 0.523 & -0.582 & 0.425 \end{bmatrix}$$

Remark: Those eigenvalues does not show any dynamic relation since PCA gives a statistical approximation relation rather than a dynamic one.

In this example, CPV is used to select a the number of retained PCs. In order to have CPV$= 95\%$ we need to select $a = 3$.

Both score and residual sub spaces are calculated by using equations (2.11) and (2.18) respectively. Those sub spaces are used then to build the fault indicators shown in figure (2.4).

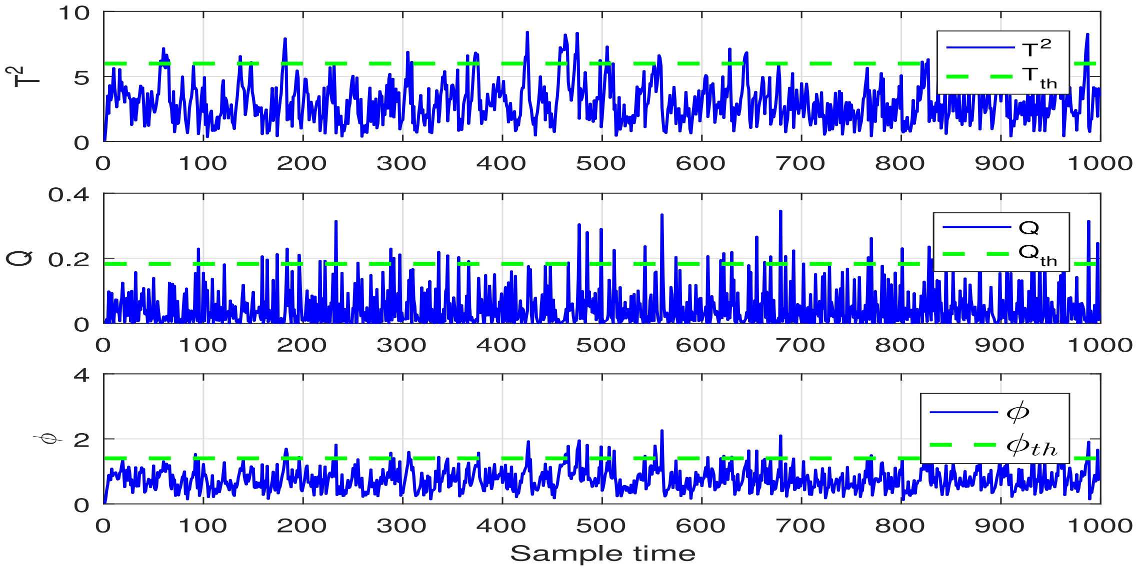


Figure 2.4: Fault detection indicator under healthy operating conditions of the system via PCA technique

Dynamic PCA application

As mentioned before, Dynamic PCA use time lagged data. To detemine l the number of time lags needed for this example, the design procedure mentioned in section 2.3.3 yields to

$l = 1$ since $r_n = 0$ if 2 time lags are used as shown in Table 2.1:

Table 2.1: Number of time-lags determination

Number of time lag l	Number of variables	Number of retrained PCs a	Number of relation r	r_n
0	4	3	1	1
1	8	4	4	2
2	12	5	7	**0**

Therefore, the new data matrix to be used in the analysis is:

$$X = [y(k)^T u(k)^T y(k-1)^T u(k-1)^T]$$

The resultant eigenpaires indicate the existence of a dynamic relation since the 6^{th},the 7^{th} and the 8^{th} eigenvalues are very close to zero.

$$\Lambda = \begin{bmatrix} 3.711 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.590 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.925 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.643 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.108 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.021 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.003 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.177 & 0.411 & -0.645 & -0.221 & -0.522 & 0.067 & 0.239 & 0 \\ 0.431 & -0.0803 & 0.476 & 0.281 & -0.526 & 0.467 & 0.083 & 0 \\ 0.486 & -0.208 & -0.0490 & -0.0717 & 0.224 & -0.199 & 0.345 & -0.710 \\ 0.217 & 0.523 & 0.198 & 0.332 & -0.143 & -0.608 & -0.360 & -0.101 \\ 0.311 & 0.470 & -0.141 & 0.134 & 0.566 & 0.531 & -0.207 & 0 \\ 0.350 & -0.411 & -0.242 & -0.281 & -0.127 & 0.011 & -0.745 & 0 \\ 0.354 & -0.312 & -0.338 & 0.520 & 0.147 & -0.222 & 0.223 & 0.521 \\ 0.395 & 0.146 & 0.350 & -0.624 & 0.148 & -0.190 & 0.199 & 0.462 \end{bmatrix}$$

The next step is to calculate the score and residual sub spaces and then to derive the fault indicators shown in Figure 2.5.

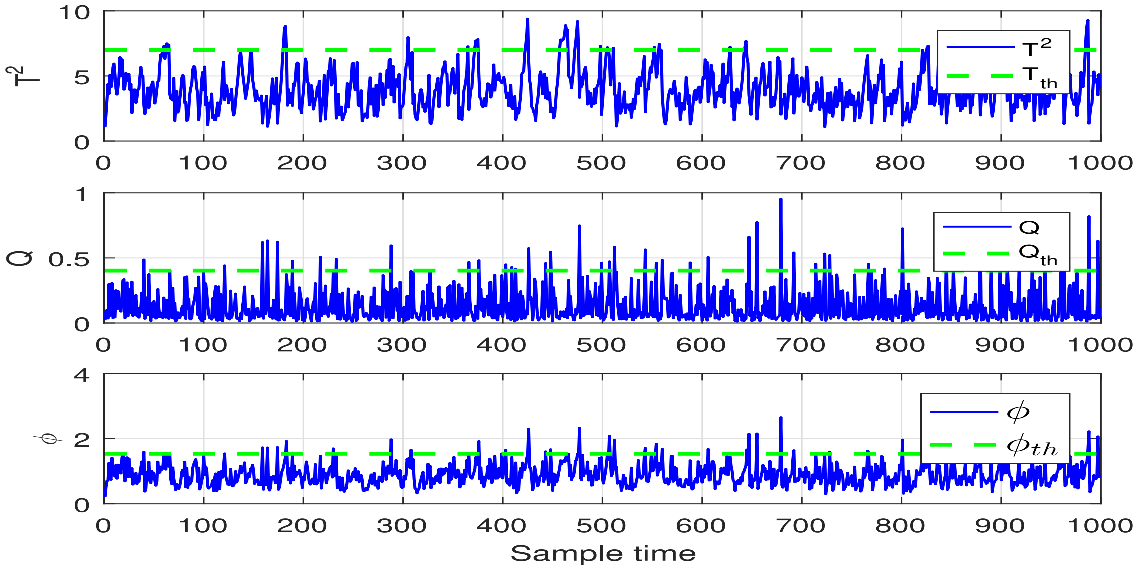


Figure 2.5: Fault detection indicator under healthy operating conditions of the system via DPCA technique

Interval-Valued Dynamic PCA

In order to generate interval valued data, two bounded uniformly distributed uncertainties Un_1 and Un_2 are added to the state variable z as follows:

$$z(k) = \begin{bmatrix} 0.118 + Un_1 & -0.191 \\ 0.847 & 0.264 + Un_2 \end{bmatrix} z(k-1) + \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix} u(k-1)$$

Then we construct our model based on equations and terminologies mentioned before in section 2.4, thus:

$$(\underline{\lambda} \quad \overline{\lambda}) = \begin{pmatrix} [3.659 & 3.765] \\ [2.574 & 2.602] \\ [0.905 & 0.947] \\ [0.635 & 0.650] \\ [0.096 & 0.132] \\ [0.019 & 0.021] \\ [0.002 & 0.003] \\ [0.000 & 0.000] \end{pmatrix}$$

Using the mid-point data transformation technique we get:

$$\lambda_{mid} = \begin{pmatrix} 3.712 \\ 2.588 \\ 0.926 \\ 0.642 \\ 0.109 \\ 0.020 \\ 0.003 \\ 0.000 \end{pmatrix}$$

Applying CPV technique indicates that $a = 4$ is sufficient to get 95% of the total variance. The next step is the calculation of the interval scores and residuals sub spaces using Appendix A, and then to build the interval fault indicators shown in Figure 2.6.

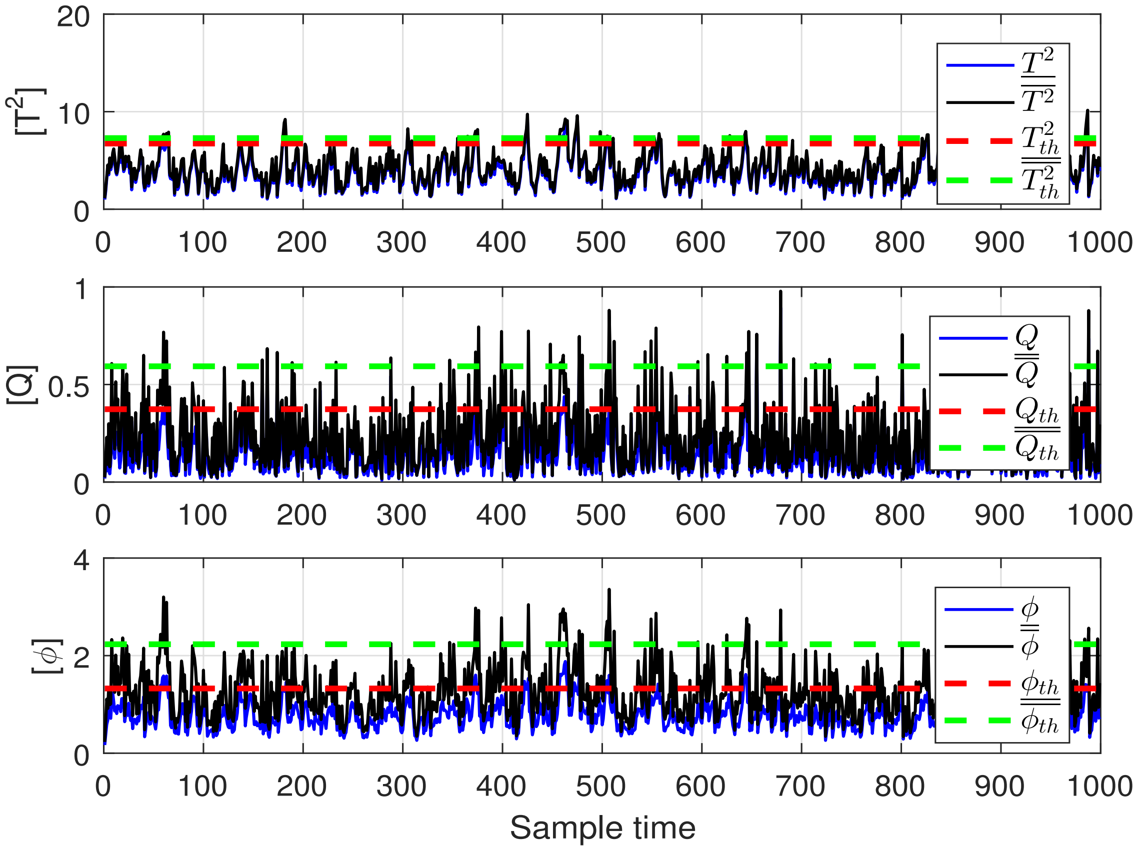


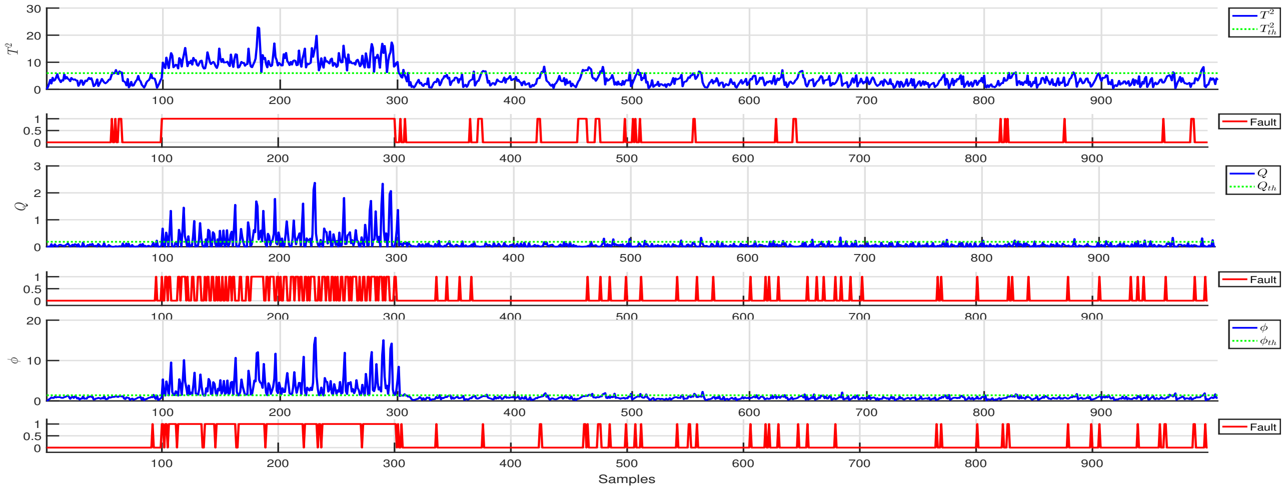
Figure 2.6: Fault detection indicator under healthy operating conditions of the system via IV-DPCA technique

Introducing faults to the three techniques

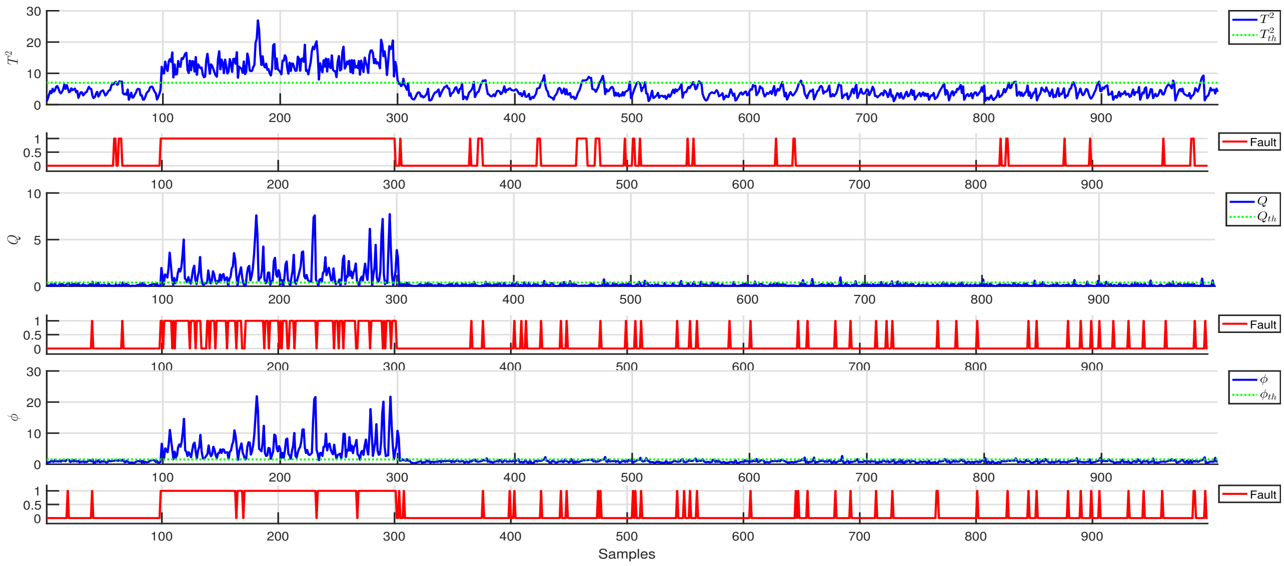
In order to test the performance of the the three techniques, three actuator faults are introduced to the system at sample time 100 until 300:

- Abrupt additive fault of 20%.
- Drift additive fault with slope $k = 0.05$.
- Random additive fault with $\mu = 1$ and $\sigma = 1$

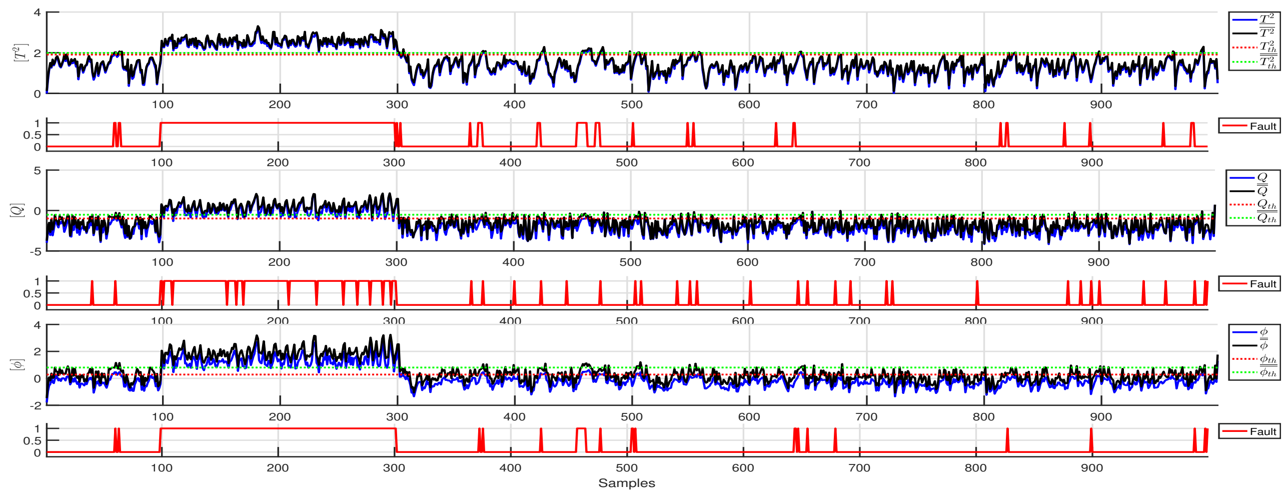
The following figures displays the outcomes:



(a)



(b)



(c)

Figure 2.7: Fault detection indicator before and after the occurrence of an abrupt fault via (a) PCA, (b) DPCA and (c) IV-DPCA

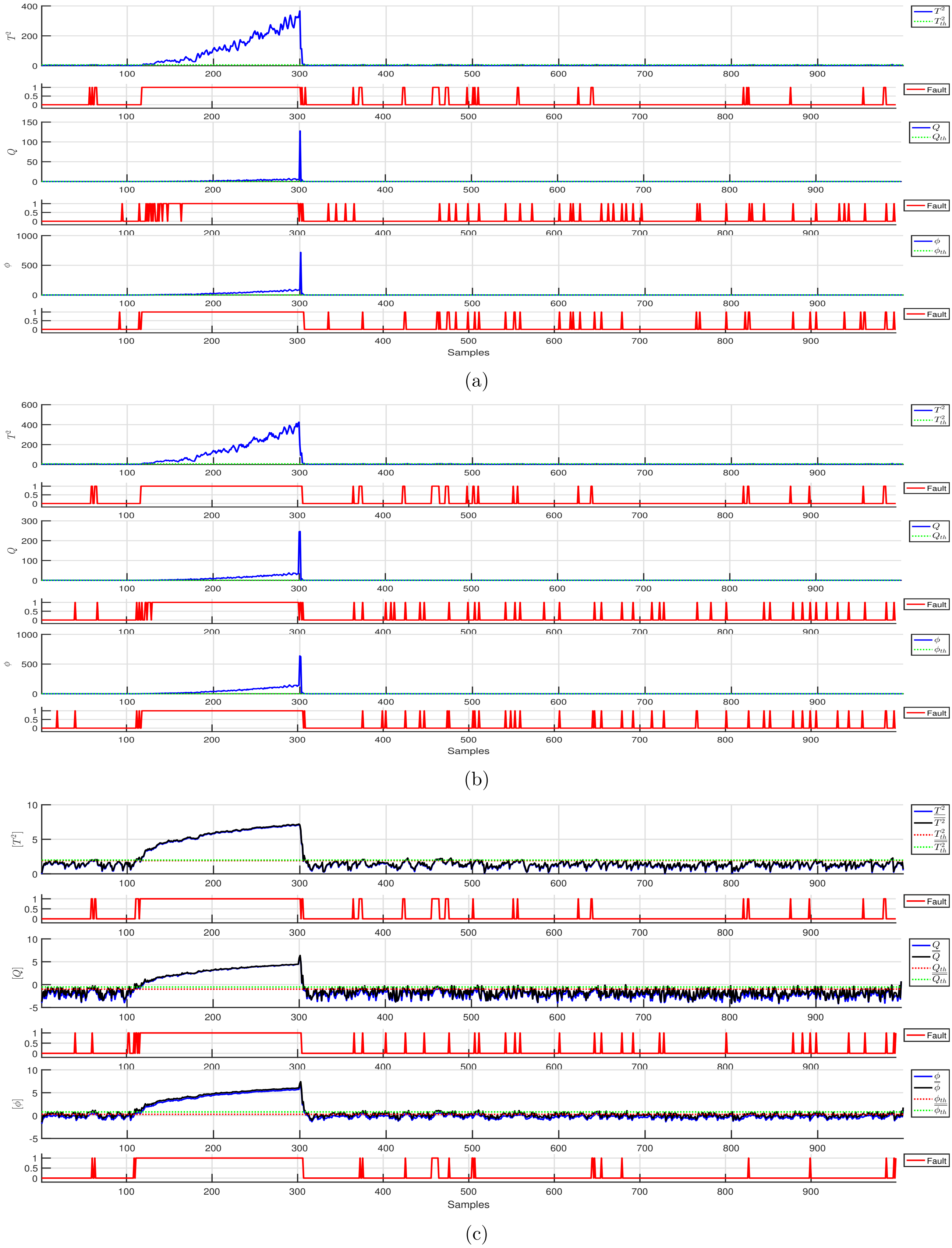


Figure 2.8: Fault detection indicator before and after the occurrence of a drift fault via (a) PCA, (b) DPCA and (c) IV-DPCA

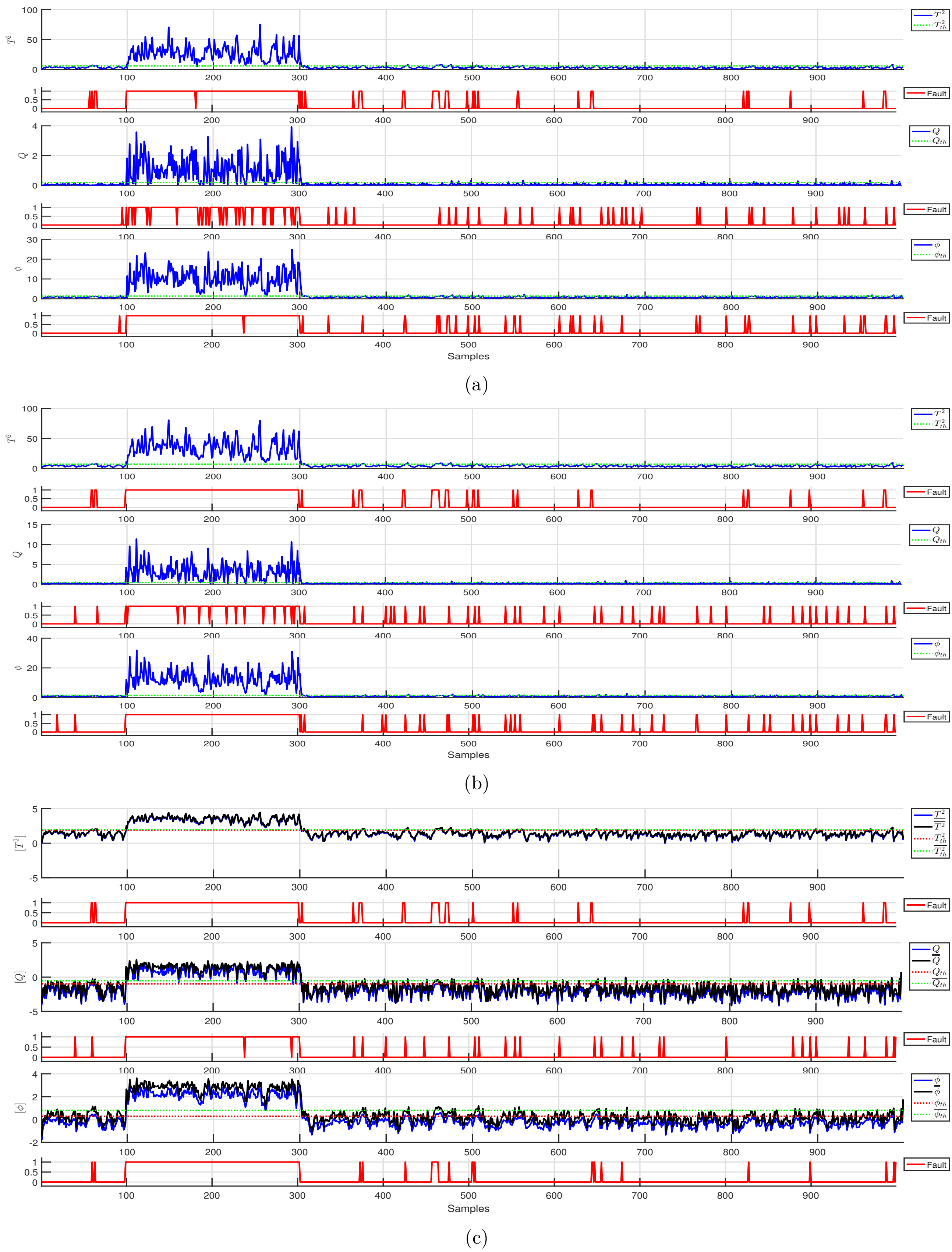


Figure 2.9: Fault detection indicator before and after the occurrence of a random fault via (a) PCA, (b) DPCA and (c) IV-DPCA

The results are giving in Table 2.2:

Table 2.2: Monitoring performances evaluation of fault detection indices via PCA, DPCA and IV-DPCA

Type of fault	The method used	T^2			Q			ϕ		
		FAR(%)	MDR(%)	DTD(s)	FAR(%)	MDR(%)	DTD(s)	FAR(%)	MDR(%)	DTD(s)
Abrupt	PCA	5.63	**0.00**	**0**	4.88	45.27	**0**	5.63	6.46	**0**
	DPCA	5.76	**0.00**	**0**	5.38	16.92	**0**	5.63	1.99	**0**
	IV-DPCA	**4.30**	**0.00**	**0**	**3.20**	**5.97**	**0**	**2.90**	**0.00**	**0**
Drift	PCA	6.01	8.96	18	5.26	16.42	15	6.13	8.46	15
	DPCA	6.13	8.46	17	5.63	10.45	12	6.01	7.96	12
	IV-DPCA	**4.50**	**5.97**	**11**	**3.40**	**4.98**	**2**	**3.30**	**4.98**	**9**
Random	PCA	5.88	0.50	**0**	5.01	15.92	**0**	5.63	0.50	**0**
	DPCA	5.88	**0.00**	**0**	5.63	6.46	**0**	5.76	**0.00**	**0**
	IV-DPCA	**4.30**	**0.00**	**0**	**3.30**	**1**	**0**	**3.00**	**0.00**	**0**

Now, the cost function J is calculated to determine the best method as well as the best fault indicator . the result are in Table 2.3.

Table 2.3: Cost function values

Type of fault	The method used	J_{T^2}	J_Q	J_ϕ
Abrupt	PCA	1.88	16.72	4.03
	DPCA	1.92	7.43	2.54
	IV-DPCA	**1.43**	**3.06**	**0.97**
Drift	PCA	7.98	9.71	7.35
	DPCA	7.68	7.35	6.65
	IV-DPCA	**5.31**	**3.13**	**4.25**
Multiple	PCA	2.13	6.98	2.04
	DPCA	1.96	4.03	1.92
	IV-DPCA	**1.43**	**1.43**	**1.00**
Average	PCA	3.99	11.14	4.47
	DPCA	3.85	6.27	3.70
	IV-DPCA	**2.72**	**2.54**	**2.07**

The results are very interesting concerning the Q indicator using IV-DPCA technique since it optimizes its cost by 77.2%. This results gives an optimistic guess that the proposed IV-DPCA method will perform well in real life application.

2.6 Conclusion

In this chapter, different PCA methods were discussed and the difference between them was highlighted. The major result is that applying static PCA directly on dynamic systems will lead to inaccurate evaluation indices while the use of DPCA will lead to more accurate ones, and IV-DPCA will lead to even more accurate ones.

Chapter 3

Application for cement rotary kiln data

3.1 Introduction

The results presented in the previous chapter bring in mind the following question:"which method among PCA, DPCA and IV-DPCA will acheive the best fault detection results?" To answer this question, PCA, DPCA and IV-DPCA were applied to cement rotary kiln data and the obtained results will be discussed and compared in this chapter.

3.2 PCA-based fault detection approach

At the beginning, a PCA model is constructed based on the first 500 observations of the training data set described in Table B.1 in Appendix B, 18 retained principle components to obtain 95% of the total variance of the data. lastly, a fixed threshold for the monitoring of 95% confidence level. After that, T^2,Q and ϕ fault indicators were used to monitor the real process fault and the simulated sensor fault described in Appendix B.

3.3 Number of time-lags determination

Before building DPCA and IV-DPCA models, the determination of the number of time lags is a must. Therefore, the design procedure mentioned in section 2.3.3 was applied and the results are summarized in Table 3.1 and $l = 1$ is selected based on it.

Table 3.1: Number of time-lags determination

Number of time lag l	Number of variables	Number of retrained PCs a	Number of relation r	r_n
0	44	18	26	26
1	88	30	58	6
2	132	42	90	**0**

3.4 DPCA-based fault detection approach

Based on the results obtained in Table 3.1, the DPCA model is constructed from the first 500 samples of training data set using one time lagged data with the same confidence level and 30 retained principle components.

3.5 IV-DPCA-based fault detection approach

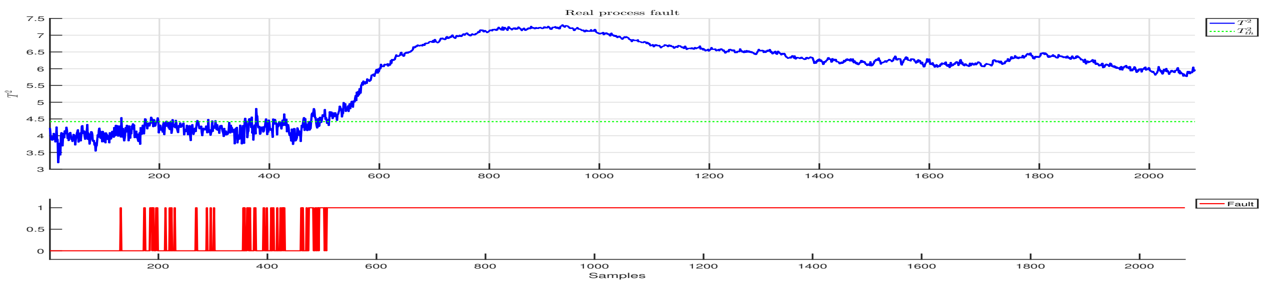
In this case, the interval-valued data were obtained from 22 repeated experiments of 500 observations of the training data to quantify the uncertainties and errors of the measurement.

The IV-DPCA model is then built using one time lagged data with the same confidence level as PCA and DPCA models.

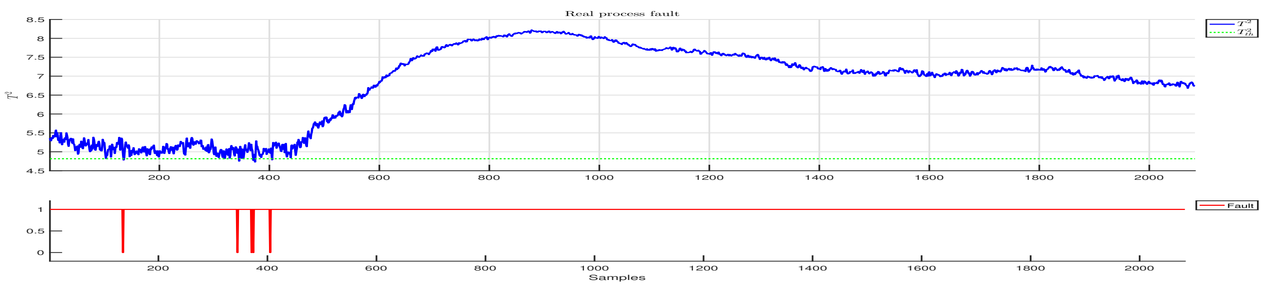
T^2, Q and ϕ fault indicators were calculated and used to monitor the same aforementioned faults.

3.6 Results and discussion

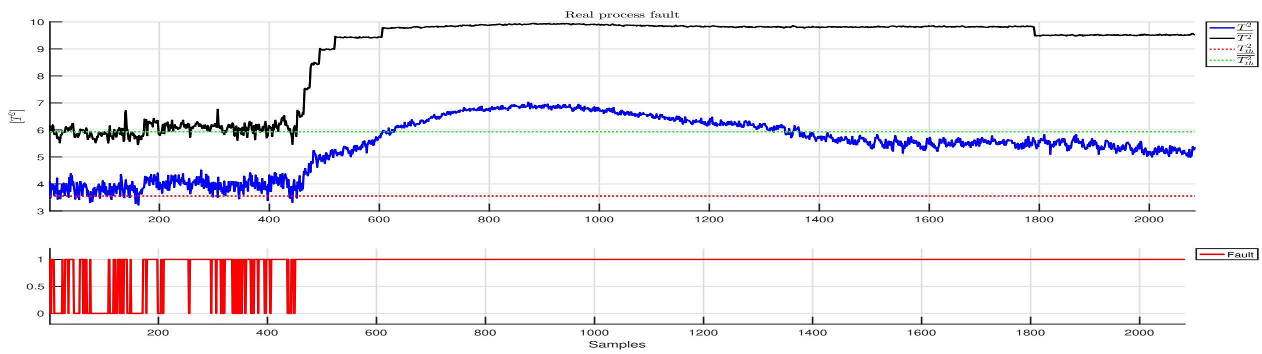
The three statistical indicators ($[T^2]$, $[Q]$, and $[\phi]$) have been generated in order to compare the three methods (PCA, DPCA, and IV-DPCA) that have been presented. Monitoring results are shown in the graphs below, and evaluation indices are given in Tables 3.2 and Tables 3.3.



(a)

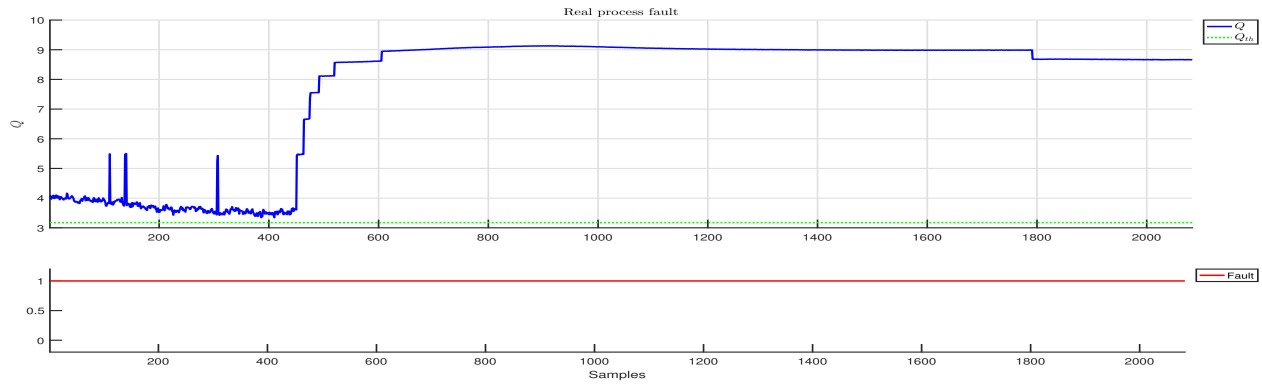


(b)

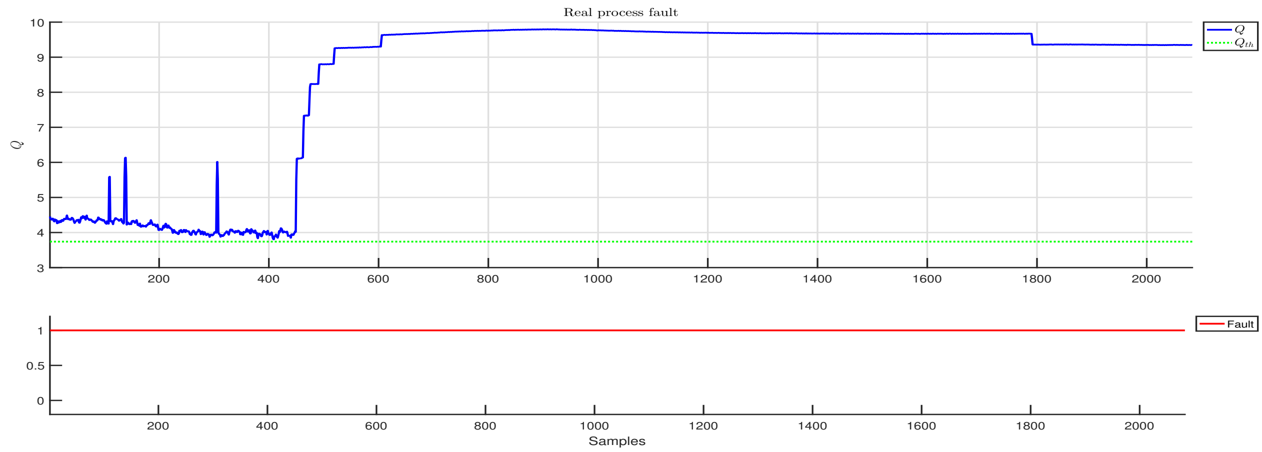


(c)

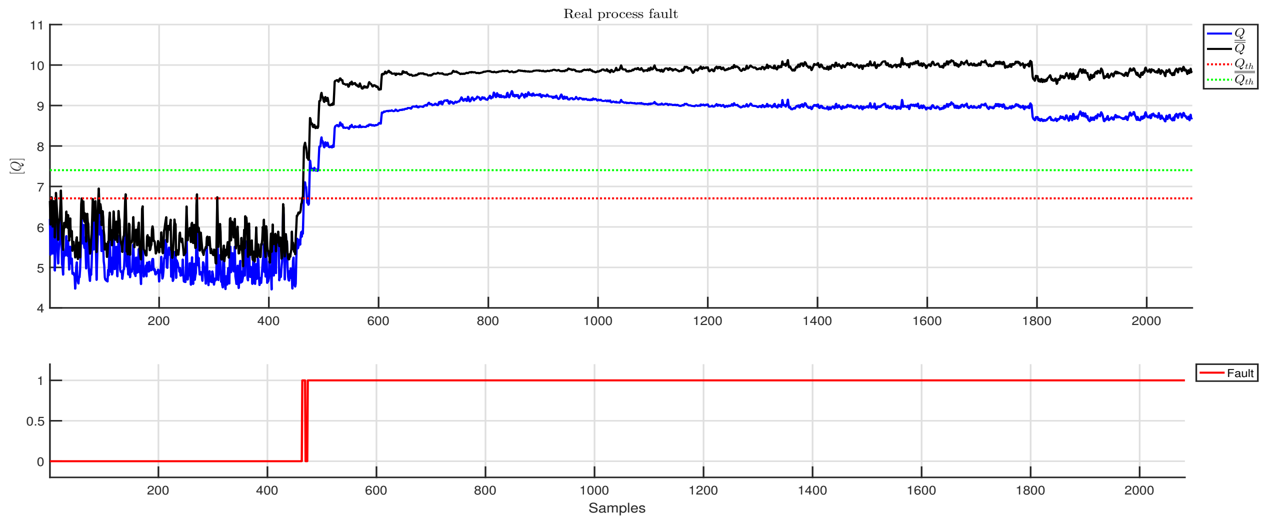
Figure 3.1: T^2 fault indicator monitoring before and after the occurrence of real process fault via (a) PCA, (b)DPCA and (c) IV-DPCA



(a)

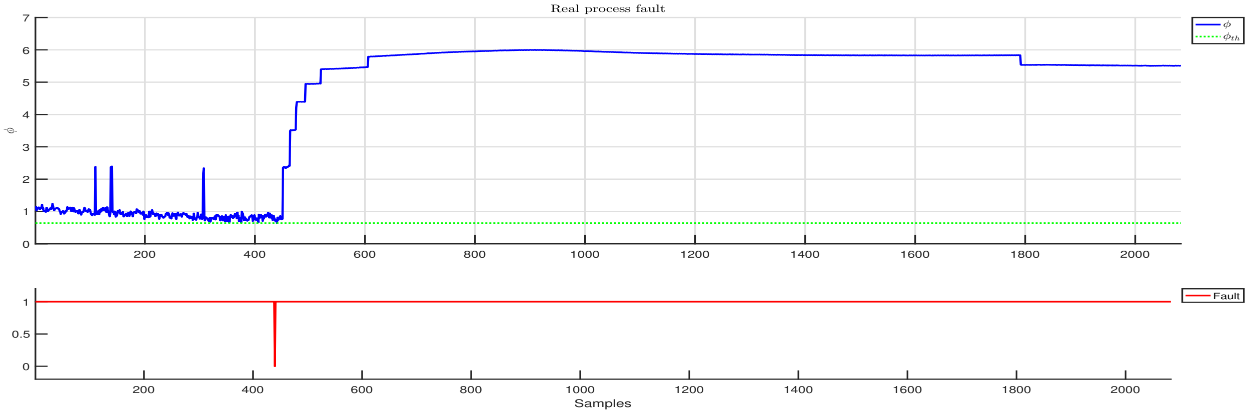


(b)

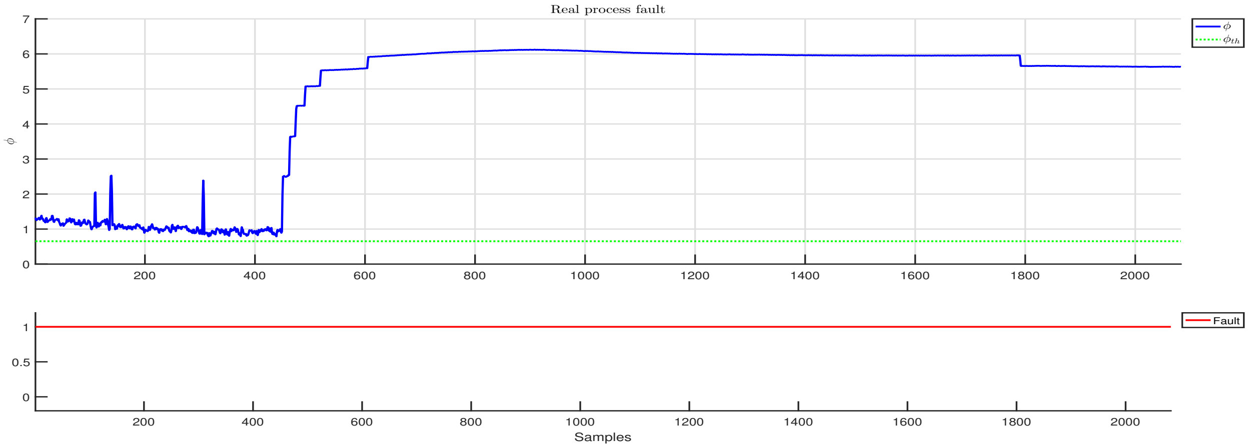


(c)

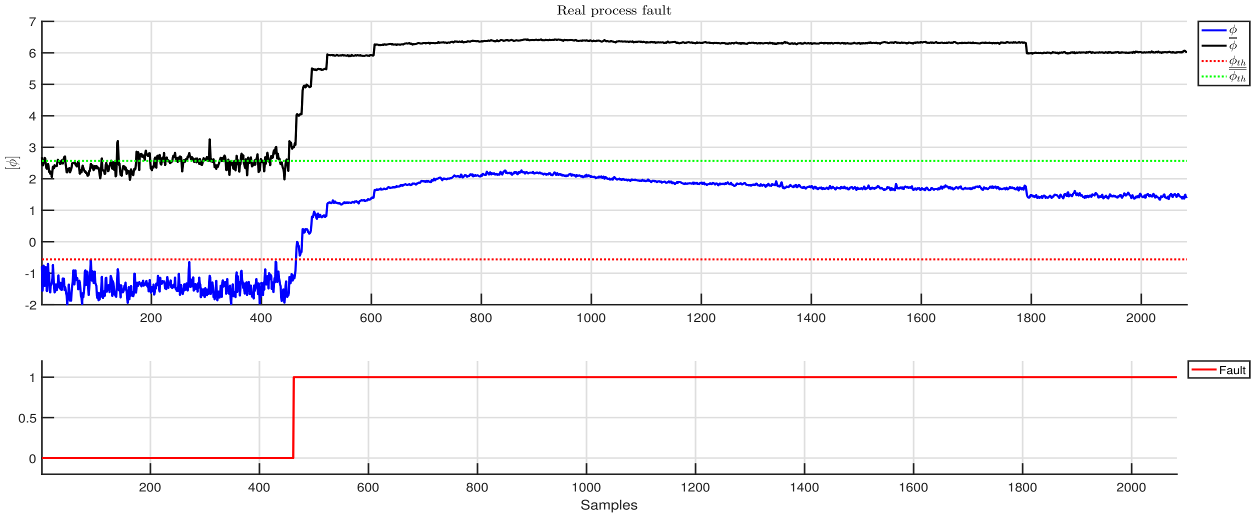
Figure 3.2: Q fault indicator monitoring before and after the occurrence of real process fault via (a) PCA, (b)DPCA and (c) IV-DPCA



(a)



(b)



(c)

Figure 3.3: ϕ fault indicator monitoring before and after the occurrence of real process fault via (a) PCA, (b)DPCA and (c) IV-DPCA

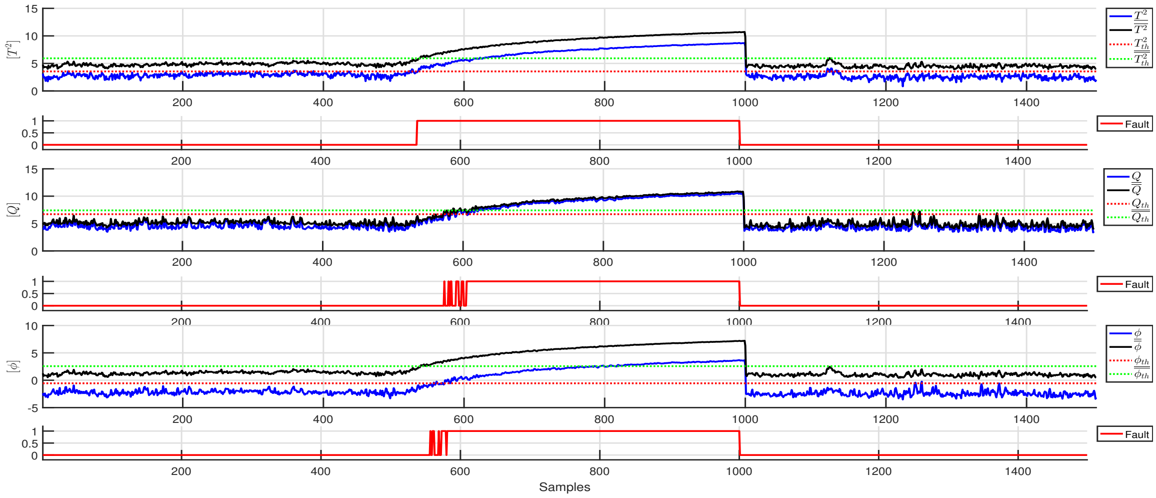


Figure 3.4: Fault detection indicators before and after the occurrence of a fault (3) via IV-DPCA

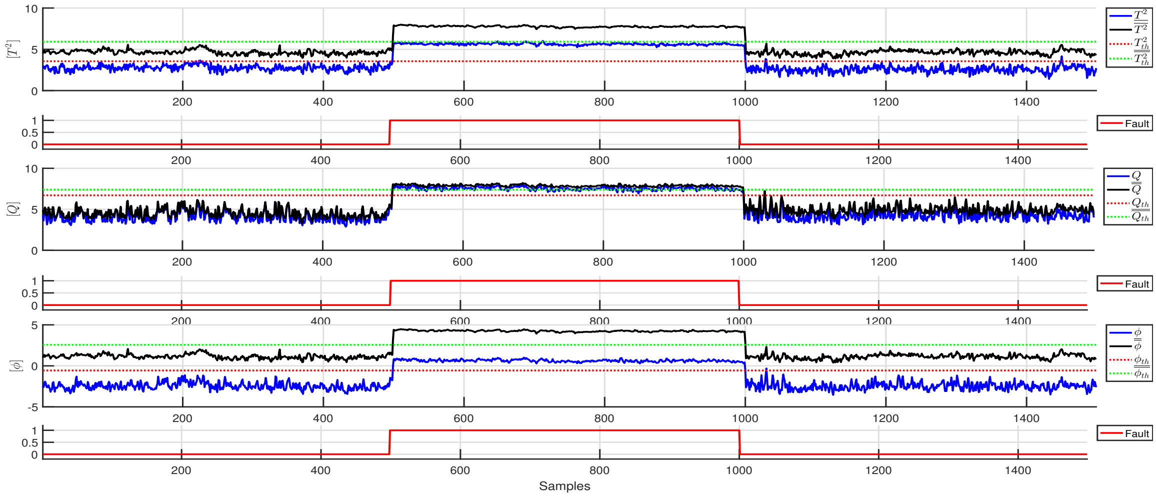


Figure 3.5: Fault detection indicators before and after the occurrence of a fault (5) via IV-DPCA

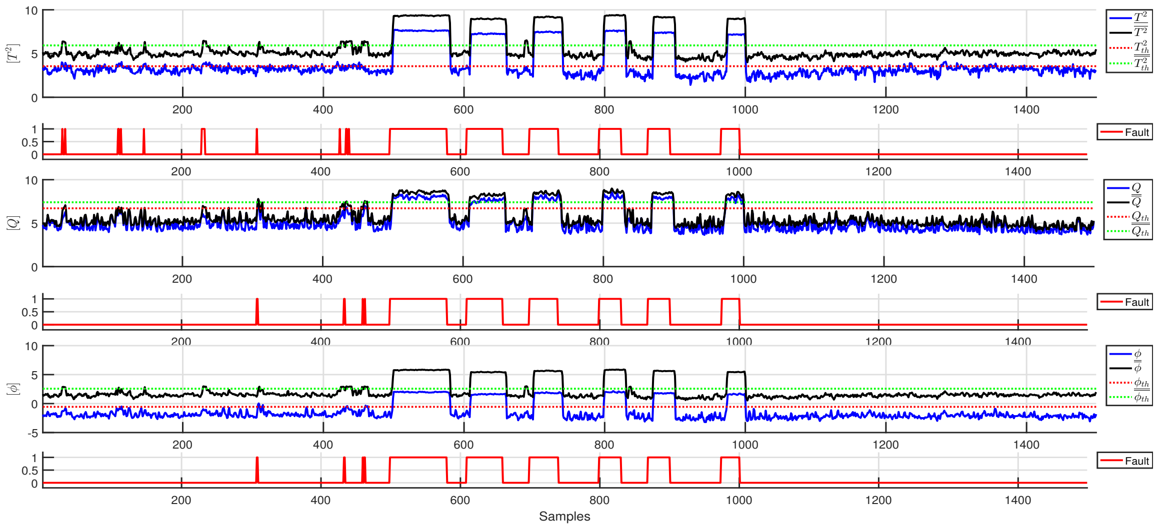


Figure 3.6: Fault detection indicators before and after the occurrence of a fault (7) via IV-DPCA

Table 3.2 lists FAR under the healthy operation of a cement plant. It has been evaluated in which the process was ensured to be in a healthy operating state, for training, testing, and faulty data sets. Its contributed by the T^2, Q and ϕ statistics using the three PCA methods tackled before. The bold values show the recommended greatest performance method.

Table 3.2: FAR contributed by different fault detection indicators via PCA, DPCA and IV-DPCA

Fault indicator	The method used	Training	Testing	Real fault	Fault(1)	Fault(2)	Fault(3)	Fault(4)	Fault(5)	Fault(6)	Fault(7)
	PCA	5.01	1.25	**9.33**	2.50	1.70	1.50	1.00	0.90	2.40	**0.80**
T^2	DPCA	5.01	2.58	98.66	2.80	1.40	1.60	0.30	2.40	15.62	5.80
	IV-DPCA	**0.00**	**1.11**	63.11	**0.10**	**0.8**	**0.00**	**0.00**	**0.10**	**0.50**	1.70
	PCA	5.01	18.46	99.77	3.20	23.72	43.54	**0.00**	0.20	17.32	99.30
Q	DPCA	5.01	21.05	99.78	4.20	26.73	47.35	**0.00**	1.80	7.51	93.59
	IV-DPCA	**0.00**	**0.01**	**0.00**	**0.10**	**0.10**	**0.00**	0.1	**0.00**	**0.10**	**0.70**
	PCA	5.01	11.35	99.55	2.50	6.71	7.51	**0.00**	0.50	5.81	90.56
ϕ	DPCA	5.01	14.19	99.78	2.60	12.91	18.92	**0.00**	0.90	18.72	89.40
	IV-DPCA	**0.00**	**0.01**	**0.00**	**0.10**	**0.20**	**0.00**	**0.00**	**0.10**	**0.10**	**0.80**

The goal of a reliable monitoring strategy is to achieve the lowest FAR. From Table 3.2, it can be seen that the proposed IV-DPCA approach is the best among the three approaches. Furthermore, the results of PCA and DPCA indicates that the models are out of control. This clearly indicates the superiority of the proposed method in the training set, the testing set, and all faults where a low FAR is obtained for the rotary kiln.

The acquired findings demonstrate the potency of the IV-DPCA for uncertainties, where a significant reduction was noted compared to PCA and DPCA.

The following table lists the findings MDR and DTD for the real fault and the simulated sensor faults using PCA, DPCA, and IV-DPCA.

Table 3.3: $MDR|DTD$ contributed by different fault detection indicators via PCA, DPCA and IV-DPCA

<table>
<tr><th>Fault indicator</th><th>The method used</th><th>Real fault</th><th>Fault(1)</th><th>Fault(2)</th><th>Fault(3)</th><th>Fault(4)</th><th>Fault(5)</th><th>Fault(6)</th><th>Fault(7)</th></tr>
<tr><td></td><td>PCA</td><td>1.84 | 12</td><td>1.40 | 0</td><td>96.81 | 36</td><td>7.19 | 28</td><td>36.93 | 11</td><td>0.00 | 0</td><td>16.77 | 61</td><td>0.00 | 0</td></tr>
<tr><td>T^2</td><td>DPCA</td><td>0.00 | 0</td><td>0.00 | 0</td><td>96.00 | 36</td><td>5.99 | 27</td><td>4.59 | 9</td><td>0.00 | 0</td><td>14.17 | 48</td><td>0.00 | 0</td></tr>
<tr><td></td><td>IV-DPCA</td><td>0.06 | 1</td><td>0.00 | 0</td><td>78.44 | 32</td><td>7.58 | 38</td><td>31.54 | 105</td><td>0.00 | 0</td><td>22.16 | 105</td><td>0.00 | 0</td></tr>
<tr><td></td><td>PCA</td><td>0.00 | 0</td><td>3.19 | 0</td><td>0.00 | 0</td><td>38.52 | 98</td><td>3.39 | 17</td><td>0.00 | 0</td><td>11.38 | 55</td><td>0.00 | 0</td></tr>
<tr><td>Q</td><td>DPCA</td><td>0.00 | 0</td><td>0.00 | 0</td><td>0.00 | 0</td><td>26.15 | 112</td><td>2.79 | 8</td><td>0.00 | 0</td><td>10.99 | 54</td><td>0.00 | 0</td></tr>
<tr><td></td><td>IV-DPCA</td><td>1.10 | 14</td><td>0.79 | 0</td><td>0.00 | 0</td><td>19.56 | 77</td><td>32.13 | 115</td><td>0.20 | 0</td><td>68.46 | 176</td><td>0.00 | 0</td></tr>
<tr><td></td><td>PCA</td><td>0.00 | 0</td><td>1.40 | 0</td><td>0.00 | 0</td><td>7.58 | 28</td><td>2.99 | 11</td><td>0.00 | 0</td><td>11.58 | 55</td><td>0.00 | 0</td></tr>
<tr><td>ϕ</td><td>DPCA</td><td>0.00 | 0</td><td>0.00 | 0</td><td>0.00 | 0</td><td>6.19 | 27</td><td>1.80 | 9</td><td>0.00 | 0</td><td>11.18 | 54</td><td>0.00 | 0</td></tr>
<tr><td></td><td>IV-DPCA</td><td>0.80 | 13</td><td>0.00 | 0</td><td>0.00 | 0</td><td>13.77 | 0</td><td>25.95 | 114</td><td>0.00 | 0</td><td>42.91 | 288</td><td>0.00 | 0</td></tr>
</table>

The outcome of Table 3.3 demonstrates once again how effective IV-DPCA is at identifying simulated sensor defects. The detection time of the real fault, fault(2), fault(3), fault(4), and fault(6) is delayed less in Q than T^2 and ϕ.

Similarly, the amounts of missed alarms are acceptable, except for the detection of: fault(2) in T^2,fault(6) in Q using the monitoring indices.

For a broad overview of the performances, The cost function J is utilised to deduce which on of the three techniques is the best approach in terms of all evaluation indices as well as fault indicators.

Table 3.4: Cost function values

fault indicators	Methods	Real fault	Fault(1)	Fault(2)	Fault(3)	Fault(4)	Fault(5)	Fault(6)	Fault(7)	Average cost J
T^2	PCA	**3.97**	1.30	35.23	4.76	13.38	0.30	**10.45**	**0.27**	**8.70**
	DPCA	32.89	0.93	34.86	**4.33**	**2.23**	0.80	13.12	1.93	11.38
	IV-DPCA	21.07	**0.03**	**28.54**	5.05	17.50	**0.03**	14.54	0.57	10.91
Q	PCA	33.26	2.13	7.91	33.87	2.26	**0.07**	13.23	33.10	15.73
	DPCA	33.26	1.40	8.91	31.95	**1.52**	0.60	**9.76**	31.20	14.82
	IV-DPCA	**0.65**	**0.30**	**0.03**	**11.64**	18.39	**0.07**	35.89	**0.23**	**8.40**
ϕ	PCA	33.18	1.30	2.24	6.89	1.73	0.17	**9.46**	30.19	10.65
	DPCA	33.26	0.87	4.30	10.17	**1.20**	0.30	13.56	29.80	11.68
	IV-DPCA	**0.53**	**0.03**	**0.07**	**4.59**	16.23	**0.03**	33.50	**0.27**	**6.90**

From Table 3.4, we can deduce that using ϕ fault indicator via IV-DPCA is the optimal performance. It is true that the technique fails when it comes to fault (4) and fault (6) but from an overall point of view, IV-DPCA monitoring performs is good since it has reduced the average cost of Q indicator by 46.59% and 43.32% compared to PCA and DPCA respectively.

In addition,the proposed technique results in a huge improvement when it comes to Real fault monitoring using Q and ϕ indicators compared to PCA and DPCA which is very promising.

3.7 Conclusion

In this part, the three methods (PCA, DPCA and IV-DPCA) were applied on real data obtained from a Rotary KILN Cement Plant. The techniques were used then to monitor real and simulated fault. Afterwards, they were compared in terms of evaluation indices. In general, using IV-DPCA for fault detection gives strong performance since it has the best cost J defined by the best indices FAR, MDR and DTD. This is due to the use of both dynamic behavior and interval-valued data techniques in the modeling phase.

Conclusion

Many practical process monitoring systems, including FDD techniques, have been exploited and implemented for various industrial processes; these techniques come as a result of the significant attention that fault detection and diagnosis approaches have received from both industrial and academic fields. However, because of their distinct qualities, FDD approaches still face several challenges when applied to actual industrial processes (e.g., multivariate, correlation, non-linearity...).

The classification of fault detection techniques is presented in Chapter 1 of this dissertation; fault diagnosis methods were classified into two main categories: Model-based and data-driven, along with a very brief summary of each class. Information about a process ,that is typically easy to get, is a quantitative data-driven diagnosis method, PCA is one of them.

PCA is a reliable method that has been used for a long time in many technical domains and has seen a number of advancements. In this study, the application of PCA was expanded to include dynamic process features using the well-known method of augmenting the data matrix $X(k)$ with time-shifted copies $X(k-l)$, and to take measurement errors into consideration.

The capacity of the proposed interval-valued dynamic principal component analysis method is to consider measurement uncertainties by treating them as regular process variation. In other words, because the model is interval-based, any information that falls within the interval is seen as a normal variation of the process, however information that falls outside the interval is regarded as a fault.

To compare the models and statistics of PCA, DPCA, and IV-DPCA, the three approaches were evaluated in a simulated example before applying them in real application. The proposed IV-DPCA results demonstrate significant improvement in fault detection performance which is very promising.

Appendix A

Basic interval arithmetic

Basic interval arithmetic is defined as follows:
Given two elementary intervals $[a, b]$ and $[c, d]$, wher $a \leq b$ and $c \leq d$:

- Sum of intervals:
$$[a, b] + [c, d] = [a + c, b + d] \tag{A.1}$$

- Subtraction of intervals:
$$[a, b] - [c, d] = [a - d, b - c] \tag{A.2}$$

- Product of intervals:
$$[\mathrm{a, b}] * [\mathrm{c, d}] = [\min(\mathrm{ac, ad, bc, bd}), \max(\mathrm{ac, ad, bc, bd})] \tag{A.3}$$

- Division of intervals:
$$\frac{[\mathrm{a, b}]}{[\mathrm{c, d}]} = \left[\min\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right), \max\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right)\right] \tag{A.4}$$

- Mean interval: Let us define the mean interval Imean[x] as:
$$\mathrm{Imean[x]} = \frac{1}{N} \sum_i I[x]_i \tag{A.5}$$

  where $I[x]_i \subset R \forall i \in \{1, \ldots, N\}$

- Variance for interval-valued data: Given a set of N real bounded intervals $\mathrm{I[x]}_i$, ($\mathrm{i} \in [1, \ldots, \mathrm{N}]$), we denote with Imean[x] the mean interval and with σ^2 the scalar variance, defined as:
$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} d^2 \left(\mathrm{I[x]}_i, \mathrm{Imean[x]}\right) \tag{A.6}$$

  The variance definition can also be written according to the following formula:
$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \left(|x_i^c - \bar{x}^c| + |x_i^r - \bar{x}^r|\right)^2 \tag{A.7}$$

  Where $x_i^c = \frac{1}{2}\left(\bar{x}_i + \underline{x}_i\right)$ and $x_i^r = \frac{1}{2}\left(\bar{x}_i - \underline{x}_i\right)$

- The interval valued dataset is defined as: Where the lower and upper bound matrices are respectively defined as:

$$\underline{X} = \begin{bmatrix} \underline{x}_{1,1} & \underline{x}_{1,2} & \cdots & \underline{x}_{1,p} \\ \underline{x}_{2,1} & \underline{x}_{2,2} & \cdots & \underline{x}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{x}_{n,1} & \underline{x}_{n,2} & \cdots & \underline{x}_{n,p} \end{bmatrix}, \quad \overline{X} = \begin{bmatrix} \bar{x}_{1,1} & \bar{x}_{1,2} & \cdots & \bar{x}_{1,p} \\ \bar{x}_{2,1} & \bar{x}_{2,2} & \cdots & \bar{x}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{x}_{n,1} & \bar{x}_{n,2} & \cdots & \bar{x}_{n,p} \end{bmatrix}. \tag{A.8}$$

Appendix B

Description of the cement rotary kiln

Following material flow direction from the top of the pre-heater tower downwards to the cooler, the cement plant is divided into two portions (a) and (b). The rotary kiln is the beating heart of each cement factory. Its job is to heat things to a temperature that allows chemical reactions to occur. The proposed method was tested on the rotary kiln of Ain El Kebira cement plant in eastern Algeria (first production line). It is around 80 metres long and 5.4 metres wide, with a 3 degree incline. Two $560kW$ asynchronous motors rotate the kiln at a maximum speed of $2.14rpm$.

(a) pre-heater tower at the right along with rotary kiln laying in horizontal to the left.

(b) cooler system with its heat ex-changer and to the left and kiln appears to the right

Figure B.1: Ain El Kebira cement plant

The of cement material manufacture can be summarised as follows:

- The first phase involves feeding raw materials into a series of four cyclones positioned on four floors at the top of two parallel towers. Using the hot counter stream gas and a supplementary fuel burner located in the tower bottom, this phase primarily facilitates material drying and dehydration.
- The pre-heated mixture enters the kiln in the second phase, due to its rotation, the mixture progressively travels downstream to the kiln's lower end. Using the main burner and hot secondary gas returning from the cooler, the material is further heated in this step. A liquid phase develops at high temperatures (more than $1450C^0$), speeding up complex chemical reactions before entering the vitrification phase, in which materials solidify anew but in a new structural form known as clinker.

- Cement is made by milling clinker with additives. The cooler which comprises of several fans, static and mobile grates, is used to cool the kiln product down to a temperature less than $100C^0$ in the third phase.

The rotary kiln process is intrinsically challenging model due to its complex dynamic, multivariable nature, nonlinear reaction kinetics, significant time delays, and variable raw material input characters. According to the authors, there no mathematical model that sufficiently captures the behaviour of the process. Additionally, the product quality of industrial rotary kilns is often assessed following the cooling of the clinker, which negatively restricts online supervision.

The installation's simplified schematic is shown in the following figure:

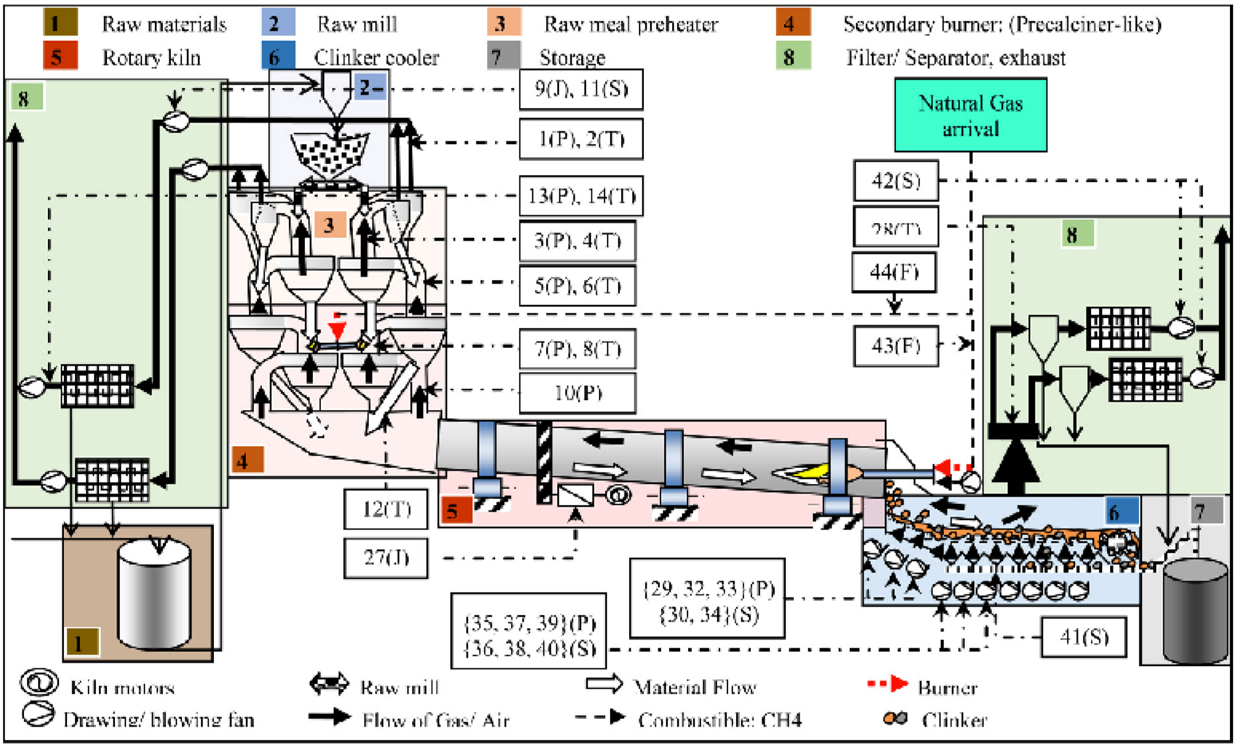


Figure B.2: Overview of the manufacturing process in the cement plant, depicting main unit operations (1-8) including the cement rotary kiln.

The applications in this work were made on a data collected on the cement plant of Ain El-Kbira, Setif, Algeria. from January 23^th, 2014 at $23:39:33PM$ to January 24^th, 2014 at $04:30:00AM$ with acquisition rate of one second. Process now operates in manual centralised mode, where every decision is made based on human experience. A continuous measurement lasting for 4 hours, 50 minutes, and 28 seconds makes up the data set that was utilised (177428 seconds). When the system is surely in a healthy state, the first 15300 seconds are captured. 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.

Table B.1 summarises the various data sets that will be used to build, test, evaluate, and compare the suggested fault detection scheme's performance.

Table B.1: Data sets used for the application

Data sets	Size	Sampling intervals [s]	Description
Training set	500 samples from $X_0 \in \mathbb{R}^{11000\times44}$	1	Normal operation data,to construct FD scheme
Testing set	$X_0 \in \mathbb{R}^{11000\times44}$	1	Normal operation data,to test FD scheme
Process fault	$X_0 \in \mathbb{R}^{2084\times44}$	1	Normal/Faulty operation, process fault
Sensor fault (7 sets)	$X_0 \in \mathbb{R}^{1500\times44}$	1	Sensor fault simulations

- Training data are gathered during normal operation at a sampling rate of one sample every 20 seconds. This information is used to build the suggested fault detection technique, which includes creating a PCA model and calculating thresholds.
- Data for testing is taken from the plant during normal operation with a 1 second sampling interval. In terms of false alarms, this data set is utilised to compare the proposed defect detection technique to the fixed threshold scheme in terms of accuracy and noise rejection.
- Faulty process data: this collection is divided into two sections; the first corresponds to normal operation, while the second refers to abnormal operation. After 7 minutes, the defect appears and progresses slowly. We employ it.
- Sensor faults: this collection includes seven simulated sensor errors that can occur during the rotary kiln operation. It encompasses single and many faults that are abrupt, random, intermittent, and gradual drift additive faults. The original data is acquired during normal rotary kiln operation, and then faults are introduced from the 500th to 1000th sample.

  The different faults are:

Table B.2: Simulated sensors faults.

Fault	Faulty variables	Fault magnitude	Description of the fault
Fault(1)	16	0.5%	Additive random fault, with mean 0, and variance 0.05
Fault(2)	44	-2%	Abrupt additive fault, bias b = 0.02
Fault(3)	30	+2%	Additive fault: Linear drift from 0% to +2%; slope Ks = 4×10^{-5}
Fault(4)	34	-2%	Additive fault: Linear drift from 0% to -2%; slope Ks = -4×10^{-5}
Fault(5)	12,18,43	[+,-,+]2%	Abrupt additive fault 2% (multiple)
Fault(6)	4,6,8,14,24	[+,+,+,-,-]2%	Additive fault: Linear drift from 0% to 2% (multiple); slope Ks = 4×10^{-5}
Fault(7)	11	+4.5%,-5.5%	Additive fault: Intermittent fault, changing intervals and amplitudes

1. variable 16 represents the speed of the induced draught fan installed in the top of tower II for drawing gas from preheater cyclones to filters. It has a mean of 0 and a variance of 0.05. This issue simulates a lack of speed sensor accuracy. It could also signify a fan vibration that causes a minor oscillation in speed without changing the average value.
2. An abrupt bias -2 in natural gas feed to the secondary burner, which can occur for a variety of reasons, including a partial blockage in the natural gas tubes, natural gas leakage, or simply a bias error in sensor readings. As a result, it has a negative impact on the process's performance and safety.
3. fault3 is a positive drift in the speed of the cooling unit's first blowing fan, which increased linearly from 0 to 2 in 500[s].

4. Like fault3, fault4 is a negative drift in the speed of the third blowing fan.This flaw could slow down the cooling of the clinker, resulting in a poor-quality product.

5. Multiple abrupt faults of magnitude 2, occurring simultaneously at different sensors in positive and negative directions, are represented by 5. fault5.

6. fault6 also represents a series of 2 percent abrupt faults that occur simultaneously at different sensors in both positive and negative directions.

7. This actuator/sensor fault has a changing interval (non-periodic) with a variable amplitude from 4.5 percent to 5.5 percent and is a positive intermittent fault in the speed of the exhauster fan of tower I. The actuator may be responsible for these fluctuations in speed due to failures in the drive's semiconductors or vibrations on the fan shaft.

   The speed sensor could possibly be to blame (tachometer). This fault occurs at the following intervals: 500-580, 610-660, 700-740, 800-830, 870-900, and 975-1000, with amplitudes of 5.5 percent, 4.5 percent, 5.5 percent, 5 percent, and 4.5 percent, respectively.

References

[1] Wenfu Ku, Robert H Storer, and Christos Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1):179–196, 1995.

[2] Okba Taouali, Ines Jaffel, Hajer Lahdhiri, Mohamed Faouzi Harkat, and Hassani Messaoud. New fault detection method based on reduced kernel principal component analysis (rkpca). *The International Journal of Advanced Manufacturing Technology*, 85(5):1547–1552, 2016.

[3] Seongmin Heo and Jay H Lee. Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, 51(18):470–475, 2018.

[4] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2005.

[5] Arthur Dexter, Jouko Pakanen, et al. *Demonstrating automated fault detection and diagnosis methods in real buildings*. Technical Research Centre of Finland (VTT), 2001.

[6] Vamshi Krishna Kandula. Fault detection in process control plants using principal component analysis. 2011.

[7] Silvio Simani, Cesare Fantuzzi, and Ronald Jon Patton. Model-based fault diagnosis techniques. In *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, pages 19–60. Springer, 2003.

[8] Alkan ALKAYA and İ Yegingil. Novel data driven-based fault detection for electromechanical and process control systems. *Çukurova University of Natural and applied sciences*, 2012.

[9] Isabel Sartori. Gerenciamento de eventos anormais de uma unidade de processamento de gás natural através de sistemas de detecção e diagnóstico de falhas. 2011.

[10] Dubravko Miljković. Fault detection methods: A literature survey. In *2011 Proceedings of the 34th international convention MIPRO*, pages 750–755. IEEE, 2011.

[11] Srinivas Katipamula and Michael R Brambley. Methods for fault detection, diagnostics, and prognostics for building systemsa review, part i. *Hvac&R Research*, 11(1):3–25, 2005.

[12] Qingsong Yang. *Model-based and data driven fault diagnosis methods with applications to process monitoring*. Case Western Reserve University, 2004.

[13] Emma Eileen Hurdle, LM Bartlett, and JD Andrews. System fault diagnostics using fault tree analysis. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 221(1):43–55, 2007.

[14] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N Kavuri, and Kewen Yin. A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & chemical engineering*, 27(3):327–346, 2003.

[15] Yumi Iwasaki and Herbert A Simon. Causality in device behavior. *Artificial intelligence*, 29(1):3–32, 1986.

[16] James V Kresta, John F MacGregor, and Thomas E Marlin. Multivariate statistical monitoring of process operating performance. *The Canadian journal of chemical engineering*, 69(1):35–47, 1991.

[17] John F MacGregor and Theodora Kourti. Statistical process control of multivariate processes. *Control engineering practice*, 3(3):403–414, 1995.

[18] Barry M Wise and Neal B Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

[19] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[20] Zhiqiang Ge and Zhihuan Song. Process monitoring based on independent component analysis- principal component analysis (ica- pca) and similarity factors. *Industrial & Engineering Chemistry Research*, 46(7):2054–2063, 2007.

[21] Hasan Ocak. *Fault detection, diagnosis and prognosis of rolling element bearings: frequency domain methods and hidden markov modeling*. Case Western Reserve University, 2004.

[22] Raghunathan Rengaswamy and Venkat Venkatasubramanian. A fast training neural network and its updation for incipient fault detection and diagnosis. *Computers & Chemical Engineering*, 24(2-7):431–437, 2000.

[23] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[24] Thippa Reddy Gadekallu, Praveen Kumar Reddy, Lakshman Kuruva, Kaluri Rajesh, Dharmendra Singh Rajput, Gautam Srivastava, et al. Analysis of dimensionality reduction techniques on big data, ieee access, ieee. march16, 2020 (volume: 8): 54776–54788, 2020.

[25] Ian T Jolliffe. Principal component analysis. *Technometrics*, 45(3):276, 2003.

[26] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[27] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI communications*, 30(2):169–190, 2017.

[28] David M Himes, Robert H Storer, and Christos Georgakis. Determination of the number of principal components for disturbance detection and isolation. In *Proceedings of 1994 American Control Conference-ACC'94*, volume 2, pages 1279–1283. IEEE, 1994.

[29] Henry F Kaiser. The application of electronic computers to factor analysis. *Educational and psychological measurement*, 20(1):141–151, 1960.

[30] Svante Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405, 1978.

[31] John L Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

[32] JE Jackson and A Edward. Users guide to principal components. john willey sons. *Inc., New York*, 40, 1991.

[33] F Bencheikh, MF Harkat, A Kouadri, and A Bensmail. New reduced kernel pca for fault detection and diagnosis in cement rotary kiln. *Chemometrics and Intelligent Laboratory Systems*, 204:104091, 2020.

[34] S Joe Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(8-9):480–502, 2003.

[35] H Henry Yue and S Joe Qin. Reconstruction-based fault identification using a combined index. *Industrial & engineering chemistry research*, 40(20):4403–4414, 2001.

[36] Lamiaa M Elshenawy. Fault detection and isolation indices for large-scale systems. In *2019 14th International Conference on Computer Engineering and Systems (ICCES)*, pages 269–274. IEEE, 2019.

[37] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. *Advances in neural information processing systems*, 11, 1998.

[38] Sami Romdhani, Shaogang Gong, Alexandra Psarrou, et al. A multi-view nonlinear active shape model using kernel pca. In *BMVC*, volume 10, pages 483–492. Citeseer, 1999.

[39] Jong-Min Lee, ChangKyoo Yoo, Sang Wook Choi, Peter A Vanrolleghem, and In-Beum Lee. Nonlinear process monitoring using kernel principal component analysis. *Chemical engineering science*, 59(1):223–234, 2004.

[40] Hajer Lahdhiri, Ilyes Elaissi, Okba Taouali, Mohamed Faouzi Harakat, and Hassani Messaoud. Nonlinear process monitoring based on new reduced rank-kpca method. *Stochastic Environmental Research and Risk Assessment*, 32(6):1833–1848, 2018.

[41] Carlos F Alcala and S Joe Qin. Reconstruction-based contribution for process monitoring with kernel principal component analysis. *Industrial & Engineering Chemistry Research*, 49(17):7849–7857, 2010.

[42] Weihua Li, H Henry Yue, Sergio Valle-Cervantes, and S Joe Qin. Recursive pca for adaptive process monitoring. *Journal of process control*, 10(5):471–486, 2000.

[43] L Ljung. System identification, theory for the user, information and system science series. *Englewood Cliffs*, 1987.

[44] George C Tiao and George EP Box. Modeling multiple time series with applications. *journal of the American Statistical Association*, 76(376):802–816, 1981.

[45] Gilbert W Stewart. An updating algorithm for subspace tracking. Technical report, 1998.

[46] Matrix Computations. by gene h. golub and charles f. van loan, 1989.

[47] Gilbert W Stewart. Determining rank in the presence of error. In *Linear algebra for large scale and real-time applications*, pages 275–291. Springer, 1993.

[48] Dražen Slišković, Ratko Grbić, and Željko Hocenski. Multivariate statistical process monitoring. *Tehnički vjesnik*, 19(1):33–41, 2012.

[49] Paul Nomikos and John F MacGregor. Multivariate spc charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995.