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 Engineering**

Title:

Fault Detection and Diagnosis in a Grid-Connected Photovoltaic System.

Presented by:

Abdaoui Djihene

Supervisor:

Pr. Kheldoun Aissa

Registration Number:...../2022

Abstract

Photovoltaic (PV) power generation has been an active research topic in the recent few years. One of the main goals of researchers in grid integrated PV systems is to improve the performance of the system in terms of efficiency, availability and reliability. For this reason, it is crucial to develop efficient methods for PV system's fault detection and diagnosis.

In this report, an automatic fault detection and diagnosis approach is proposed for a grid connected PV system. The main objective is to improve the classification accuracy and reduce the detection time.

This method merges the benefits of machine learning (ML) technique and statistical process monitoring approaches. The analytic methods were first investigated for fault detection due to their quick implementation time. Kernel based independent component analyses KICA techniques was developed to overcome the shortcomings of principle component analysis PCA based fault detection. The support vector machines SVM classifier was built mainly for fault diagnosis and classification, such that feature extraction step is done using both KICA and PCA to optimize the best model. In this work the "one to one" classification SVM algorithm is used.

To validate our method, fault detection and diagnosis of a lab implemented grid-connected PV system was performed. In this experiment, 7 typical PV systems faults were injected. The experiments were carried out for about 15 seconds in each fault scenario, and several measurements were recorded. Data samples were filtered, smoothed then processed through PCA and KICA to set the thresholds for fault detection, then the obtained reduced data sets were used to train the multi-layer SVM classifiers.

Acknowledgement

I thank Allah for his blessings upon us, for giving me the strength and will to accomplish this work.

I would like to thank my supervisor Pr. A.Kheldoun for his advice and assistance throughout this project.

A special thank you for Mr. Zahaf Yacine for his help and guideness.

Dedication

I dedicate this work for my parents, Djalil and Naima. I will forever be grateful to them for all their love and support. To my sisters Rihem, Yasmine and Imene.

I would also like to thank my friends for sharing this experience with me and being by my side throughout these five years.

TABLE OF CONTENTS

List of figures

CHAPTER ONE: Theoretical background

CHAPTER TWO: Proposed Fault Detection method

CHAPTER THREE: Results and discussion

List of Tables

Nomenclature

AC: Alternative Current.

CL: Confidence Level.

CPV: Cumulative Percentage Variance.

DC: Direct Current.

DD: Detection Delay (DD

FAR: False Alarms Rate.

FD: Fault detection.

FDAD: Fault detection and diagnosis.

FDR: Fault Detection Rate.

GCPV: Grid Connected Photovoltaic.

I^2: ICA monitoring statistic.

IC: Independent components.

ICA: Independent component analyses.

KDE: Kernel density estimation.

KICA: kernel independent component analyses.

KNN: K-Nearest Neighborhood.

KPCA: Kernel principle component analyses.

ML: Machine learning.

MPPT: Maximum Power Point tracking.

MSPM: Multivariate statistical process monitoring.

PC: Principal Component.

PCA: Principal Component Analysis.

PLL: Phase Locked Loop.

PMU: Phasor Measurement Unit

PSO: Particle Swarm Optimization.

PV: Photovoltaic.

SPE: Squared Prediction Error.

SVD: Singular Value Decomposition.

SVM: Support vector machines.

T^2: Hotelling's statistic.

VOC: Voltage Oriented Control.

GENERAL INTRODUCTION

The over-use of fossil-based energy sources in the last few decades is becoming a growing problem, especially with the technological advancement. These sources not only are limited in nature, but also has serious environmental impacts.

Exploiting renewable and green energy alternatives have become an ultimate goal for Scientists and engineers. One of the most common renewable energy forms is the photovoltaic or the solar energy. This technology works based on the photovoltaic effect, which is the process of generating an electric current in a photovoltaic cell when it is exposed to sunlight.

Although PV systems may be a promising solution, scientists are aware of the many challenges that this technology faces. This process is heavily based on theoretical research, expensive, and not to mention unpredictable as it depends on the environmental conditions. As a result, system monitoring and fault detection techniques had become a mandatory process in PV systems to ensure maximum efficiency and reliable power generation.

Early detection of faults is essential to maintain the normal functioning of the system, and many methods and techniques are proposed in this topic. These methods are classified into: Model-Based methods, Electrical Signals-Based methods, and Process History-Based methods. Process History-Based methods are adopted by several researchers and widely applied due to the high computers performance, speed and their large memory storage that can handle the complexity of calculations and huge amount of acquired data.

The main aim of this project is to realise a data driven fault detection and diagnosis approach that merges the benefits of analytic methods and machine learning classifiers. The proposed method consider an analytic kernel independent analyses KICA threshold for early detection and a support vector machines SVM multi-classifier for diagnosis. The model is then applied to differentiate between 7 types of faults in a lab implemented grid connected PV system. In order to evaluate the efficiency of the KICA based SVM model, False Alarms Rate (FAR), Fault Detection Rate (FDR), and Detection Delay (DD) are considered in addition to the training and testing accuracy of the SVM classifier.

This project is organized as follows:

- Chapter 1: an introduction to problematic and purpose of this work, a brief introduction of the theoretical background of GCPV systems and the typical faults, an overview of the numerous FD strategies.
- Chapter 2: an elaboration on the PCA-SVM and KICA-SVM based FDAD approach.
- Chapter 3: the training and testing results and corresponding discussions.
- Chapter 4: general conclusion and achievements we made through this work, in addition to future work.

CHAPTER 1: THEORITICAL BACKGROUND

1.1- INTRODUCTION

Photovoltaic systems, in either a stand-alone or a grid connected configuration, are the most promising renewable power plants. However, it is important to acknowledge the great challenges that faces PV power generation. This source although available, faces many problems, which affect the reliability of the systems. Fault detection and diagnosis methods had become a necessity in order to assure an optimal system performance.

This chapter presents a brief description of PV systems and its most common faults, system monitoring steps and an overview of the most common Fault detection and diagnosis methods.

1.2- PV system description

Photovoltaic systems rely on the photovoltaic technology to produce electricity. Meaning, PV panels convert physical energy (sunlight) into electric current. Typical PV systems can be in the form of stand-alone or grid connected systems.

Stand-alone PVS are usually used as utility power alternate [1]. The most common configuration consists of PV array, DC choppers, a storage battery and some type of controller or regulator.

A few examples of such systems are solar streetlights, solar water pumping, and rooftop home solar PV systems. Figure 1.1 shows the schematic view of a stand-alone PV system.

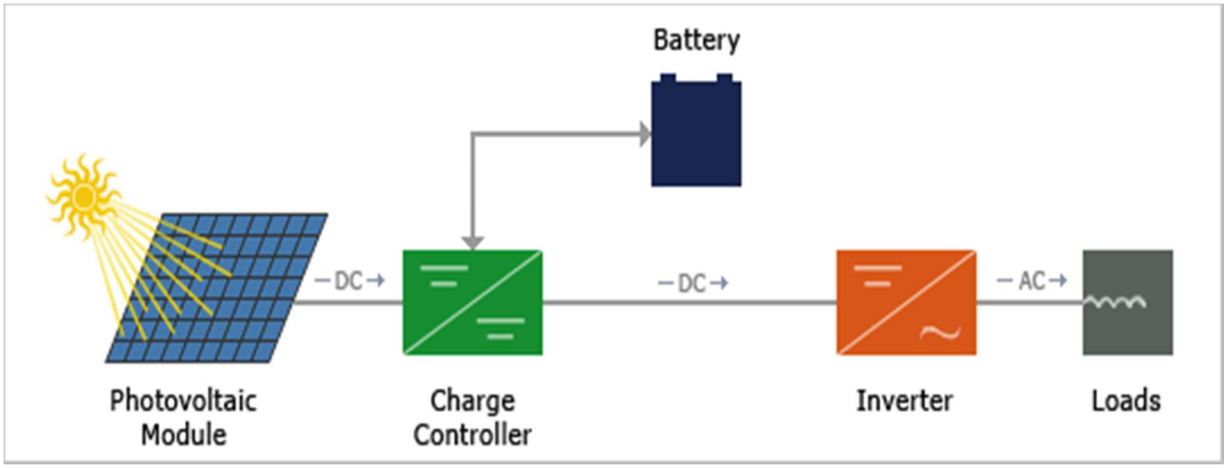


Figure 1.1: Stand-alone PV system typical configuration

Grid-connected PV systems are designed to operate in parallel with the utility grid as shown below. We can find two types: systems that interact with the utility power grid and without containing a backup storage (battery), and systems that interact with the grid while including a battery [1]. The main components for this type of configurations are: PV array, DC-DC converter (although it can be disposable for some systems), DC-AC inverter and controllers. This scheme is demonstrated in figure1.2.

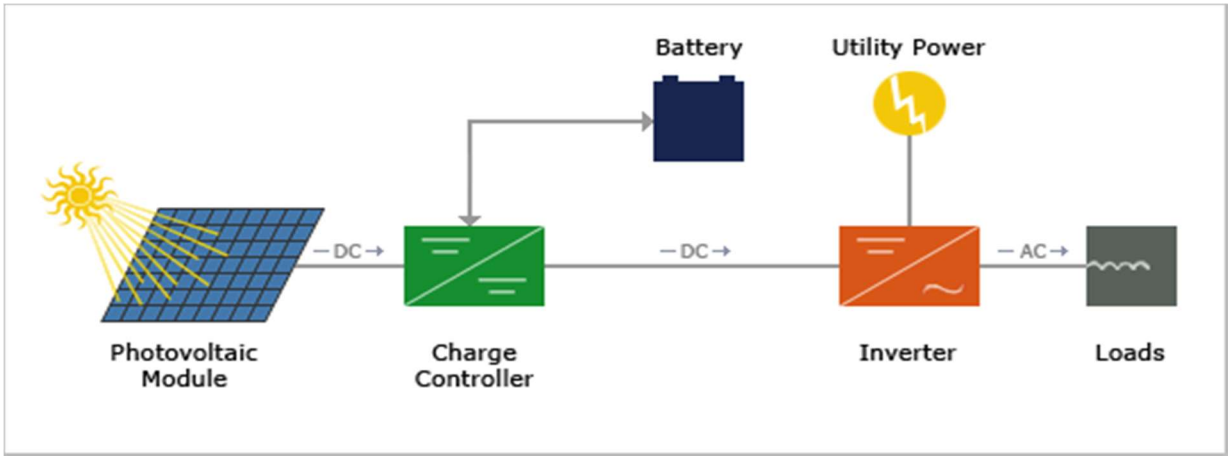


Figure 1.2: Grid-connected PV system typical configuration

1.2.1- DC converter

The DC-DC converter ensures that the PV output voltage is kept at a suitable level that corresponds to the grid demand. There are two main types of converters depending on the direction of voltage change:

- Boost converters: A boost converter is a switching device intended to boost (or increase) the input voltage higher output voltage
- Buck converters: A buck or step-down converter is a switching device intended to buck (or lower) the input voltage to a lower output voltage.

1.2.2- AC inverter

The DC part (PV array + DC-DC converter) is connected to the ac grid via a DC-AC inverter. The inverter is used to step down and modulate the output voltage according to the grid demand. The control of the power flow to the grid is based on the control of active and reactive power.

1.2.3- MPPT control

Due to the low efficiency of PV power generators under the changing environmental conditions, it is desirable to operate the system under maximum power point. The maximization of the power is achieved using a maximum power point tracker MPPT. To evaluate an MPPT algorithm, there are a few aspects to consider, such as tracking speed,

stability, simplicity and cost of implementation, and tracking efficiency. The basic idea of these controllers is to use a specific algorithm to search for peak power point and thus to allow the converter to extract the maximum power available from the PV module [3]. Among these techniques we mention:

- Perturb and observe.
- Incremental conductance.
- Parasitic Capacitance.
- Constant voltage based Peak Power Tracking.
- Constant current based peak power Tracking.

The perturb and observe algorithm is the most popular MPPT algorithm. It operates by constantly measuring the PV array voltage and current then constantly perturbing the voltage by adding a small disturbance. The output power is then observed to determine the next control signal, if the power increases the disturbance will continue, else the perturbation direction will be reversed. Figure 1.3 sums up the P&O algorithm.

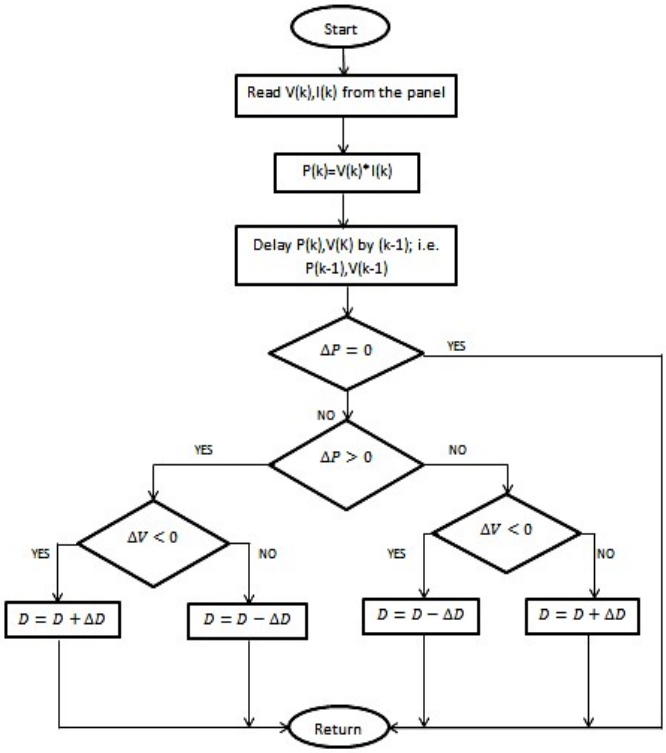


Figure 1.3: perturb and observe algorithm for MPPT [4]

1.3- Effect of Temperature and irradiance on PV arrays

As mentioned previously, the cell parameter are directly affected by any change in the environmental conditions. Both the PV cell current and voltage are functions of irradiance, with coefficients determined by the manufacturer in the datasheet of the PV panel.

Figure 1.4 demonstrates the relationship between temperature change and the PV output power. We clearly notice that the most affected parameter by temperature variation is the PV voltage, Voc values shows that the higher the temperature, the lower he voltage.

On the other hand, Figure 1.5 demonstrates the relationship between change in solar irradiance and the PV power. In this the case, it is mostly the current that is effected as we can see from the short circuit current Isc values. The more the irradiance is increased, the more current will flow.

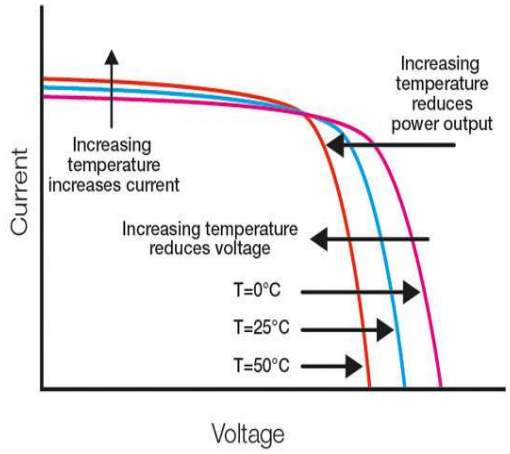


Figure1.4: effect of temperature on PV cell [6]

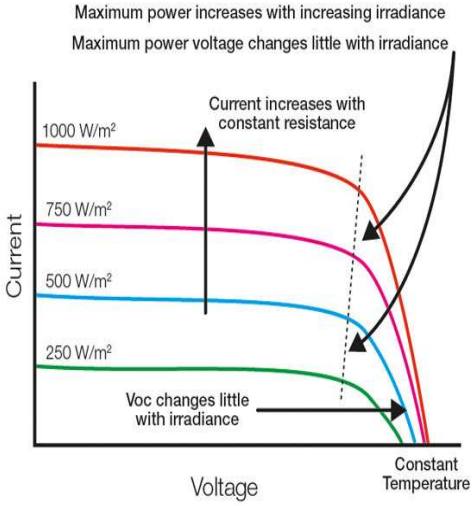


Figure1.5: effect of irradiance effect on PV cell [6]

1.4- PV systems common faults

The PV system usually operates in harsh outdoor conditions and might be subjected to various faults that we can classify into three main sections:

1- Physical faults

These are mainly degradation faults that effects the physical condition of the PV array like internal or external damage, cracks, or aging effects.

2- Environmental faults

These faults include mainly soiling and dust accumulation, bird drops, and temporary or permanent shading.

3- Electrical faults

Includes faults in both:

- AC part of the system: mainly the Inverter and the grid.
- DC part of the system: PV array + DC-DC converter.

These faults include:

- Line-line faults: created by unintentional low impedance current path in a PV array.
- Ground faults, either in PV modules, arrays, or in the whole systems: Ground faults are similar to line-line faults; however, the low impedance path is from current-carrying conductors to ground/earth.

PV system common electrical faults are listed in Figure 1.6.

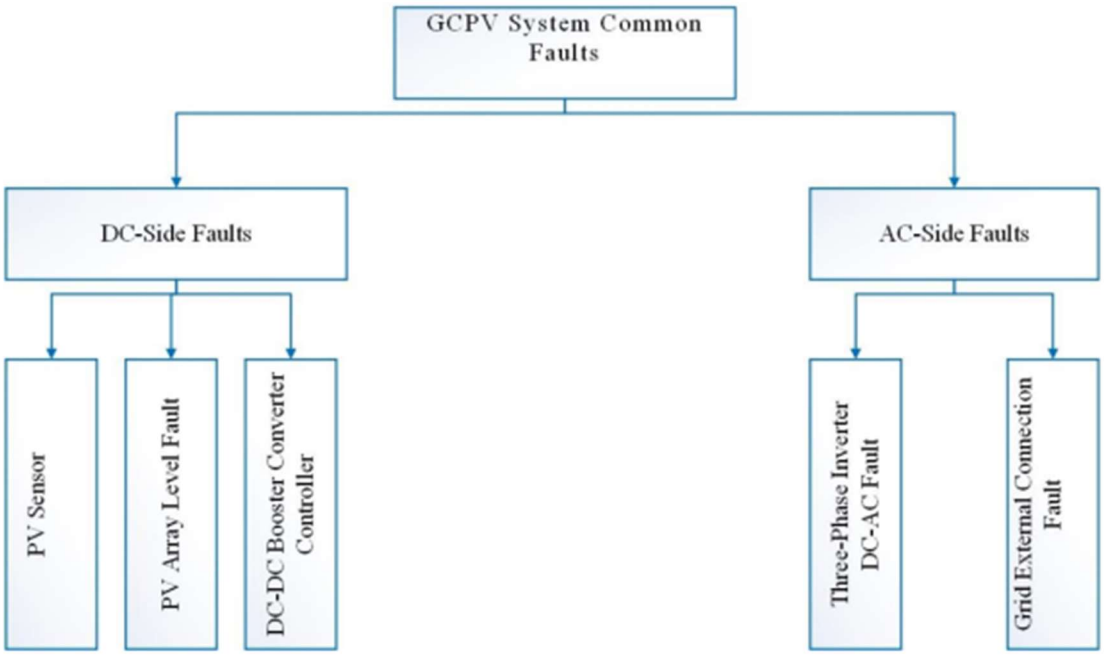


Figure 1.6: GCPV system common electrical faults.

1.5- System monitoring

PV system monitoring is a crucial step to ensure the efficiency, reliability and safety of the system. Monitoring systems collect the PV system data and transmit it to the control station in order to evaluate these criteria and keep track of meeting the grid demands and detect any anomalies.

The general scheme of PV monitoring systems involves 4 main steps: data acquisition, data transmission, data storage, and data analyses.

1.5.1- data acquisition systems

Data acquisition system DAS is used to collect data from different sensors then digitalize this data to send it to the control centre for processing and presentation.

1.5.2- data transmission systems

These systems are used to transfer data from a point to another. The components of any communication system include transmitters for sending the information, a communication channel that carries the data, either wired or wireless communication can be used, and receiver that record and presents the data

1.5.3- data storage

Data storage is mandatory process before performing data analyses. The collected data is usually stored in an SD (secure digital) card. SD cards allow lower electromagnetic interference and prepare an easy solution for data storage while having a good storage capacity [7].

A database system called My Sequel is also used for storing data. The data is stored HTML format, giving better performance and data storage with easy access.

1.5.4- data analyses

Data analyses is performed to evaluate the system's performance. Various system analyses techniques had been developed. Nowadays, standard guidelines are generally used. The guidelines explain energy generation, system losses, and solar sources [8].

1.6- Fault detection in PV systems

The demand of safe and reliable operation of processes in the industry has propelled researchers into the fault detection and diagnosis methods. Various approaches had been developed by researcher. The efficiency of each approach is evaluated according to these criteria:

- Quick detection of the faults.
- The ability to distinguish between the different faults having similar symptoms.
- Robustness of the detection system to noise and uncertainties.
- The reliability of the detection system.

We can categorize fault detection and diagnosis techniques into 3 main classes.

1.6.1- Approaches based on the system's model

This methods heavily rely on the mathematical description of the system. In the model based methods, a priori knowledge about the model is assumed. This knowledge can be classified into qualitative or quantitative. [9]

- **qualitative models**

Qualitative based fault diagnosis methods use the input-output relationship of the process and the system's dynamics to describe the system behaviour in qualitative terms centred around different units in a process [9]. Qualitative fault detection include abstraction hierarchy, fault trees, diagraphs and fuzzy systems [10].

- **Quantitative models**

In quantitative models, the system dynamics is expressed in terms of mathematical functional relationships between its inputs and outputs. Quantitative model-based fault diagnosis is broadly classified into: analytical redundancy, parity space, Kalman filter (KF), parameter estimation and diagnostic observers. [9]

a) Analytical redundancy: model based fault diagnosis require two major steps: inconsistencies (or residuals) generation, and choosing a decision rules for diagnosis. [9].The analytical redundancy schemes for fault diagnosis are basically signal processing techniques using state estimation, parameter estimation, adaptive filtering for residual generation.

b) Parity space: the idea of this approach is to check the consistency (parity) of the input-output relationship of the plant. In theory, under steady-state operating conditions, the residual generated by the parity equations method is zero. However, in reality, the residual are non-zero due to input–output measurement and process noise beside to some modelling errors.

c) Kalman filters: Kalman filter is a recursive algorithm for state estimation which is designed on the basis of the system model in its normal operating mode to achieve minimum estimation error. The prediction error of the KF is then used to form fault detection residual.

d) Parameter estimation: used to estimate the parameter drifts that are not measured directly [9].

e) Diagnostic observers: there is exist different types of diagnostic observers for residual generation. We mention: residual generation using Eigen structure assignment, residual generation using unknown input observer, Residual generation using bilinear observer [9].

1.6.2- Approaches based on the system's electrical signals

These methods detect any faulty behaviour by measuring different electrical signals and comparing them to the healthy mode ones. Many approaches based on this techniques can be used, we mention:

a) Hardware redundancy: It is the classical approach for fault detection and diagnosis as it uses multiple sensors and actuators in order to measure a particular variable. A major drawback for this technique is the reliance on electrical equipment, leading to extra weight and cost.

b) Limit checking: The process variables are measured and compared to known limit for each variable. Firstly, the variables thresholds are established based on the healthy measurements then they are compared with the measured values.
c) Frequency analysis: Most plant variables exhibit a typical frequency spectrum under normal operating conditions, using frequency analysis, any deviation from this can be interpreted as abnormality.

1.6.3- Approaches based on the system's historical data

These methods have shown quite the efficiency in detecting faults within larger and more complex industrial processes. The fundamental principal of process history based approach is to transform acquired data into a priori knowledge about the system, also known as feature extraction. These features will be used to generate a model to be applied in real processes for residual generation.

In general, history-based fault diagnosis methods are broadly classified into Fuzzy Logic, neural networks, clustering, self-organising maps (SOM), analytic methods, experts systems and pattern recognition. [11].

1.7- Multivariate Statistical Process Monitoring

Multivariate statistical process monitoring MSPM is an approach used to extract meaningful information from collected large data in order to obtain the model description of the said process. MSPM methods are applied on data under normal operating conditions of the process to construct some statistics for monitoring the process. The fundamental tasks of MSPM are dimensionality reduction and feature extraction.

Principle component analyses is one of the most widely used techniques in MSPM due to its simplicity and efficiency in detecting faults. It was originally used in chemical process control [12] and has recently been applied in GCPV systems.

PCA is also widely used as a feature extraction tool. This method identify a fewer possible independent features that cause most of the process variation. Due to its application in dimensionality reduction, this approach have seen a great success when combined with DL and ML tools. Hichri at al. in [13] have developed a PCA based ML technique for fault classification aiming to reduce the computational time of the KNN and SVM algorithms by applying PCA for dimensionality reduction of the data. Similarly, PCA-SVM-Based Automated Fault Detection and Diagnosis for Vapor-Compression Refrigeration Systems was investigated in [14]. The results had shown better performance when compared to SVM without PCA.

Along with classical PCA, several PCA based methods have been developed to solve the problems related to the system's behaviour. Kernel techniques for example are becoming really popular when dealing with nonlinear systems, Cho et al. in [16] proposed different methods based on the kernel PCA for fault detection. Bin Shams in [17] and Kallas et al. in [18] recently proposed nonlinear PCA for fault diagnosis and more generally for process monitoring. Yan Liang, and Ming Y. Gong [19] Proposes a novel fault detection method based on SIP-PCA algorithm for non-Gaussian process where the survival information potential (SIP) is used to characterize the non-Gaussian randomness of the process data.

Adaptive Multivariate Statistical Process Control for Monitoring Time-Varying Processes based on a recursive PCA model was discussed in [20]. Also, [15] proposed a real-time fault detection in PV systems under MPPT using PMU and high frequency multi-sensor data through online PCA-KDE-based multivariate KL divergence. This approach was developed to overcome the shortcoming of the classical PCA approach.

1.8- Machine learning in Fault diagnosis

As mentioned in the previous sections, the fundamental idea of history-based fault diagnosis is to generate a model of the process which relates the measured inputs to measured outputs [9]. This process can be quite computationally complex, thus generating such models using machine learning algorithms could be more efficient.

Machine learning algorithms are mainly divided into 2 sections as shown in figure 1.7:

- Supervised learning
- Unsupervised learning

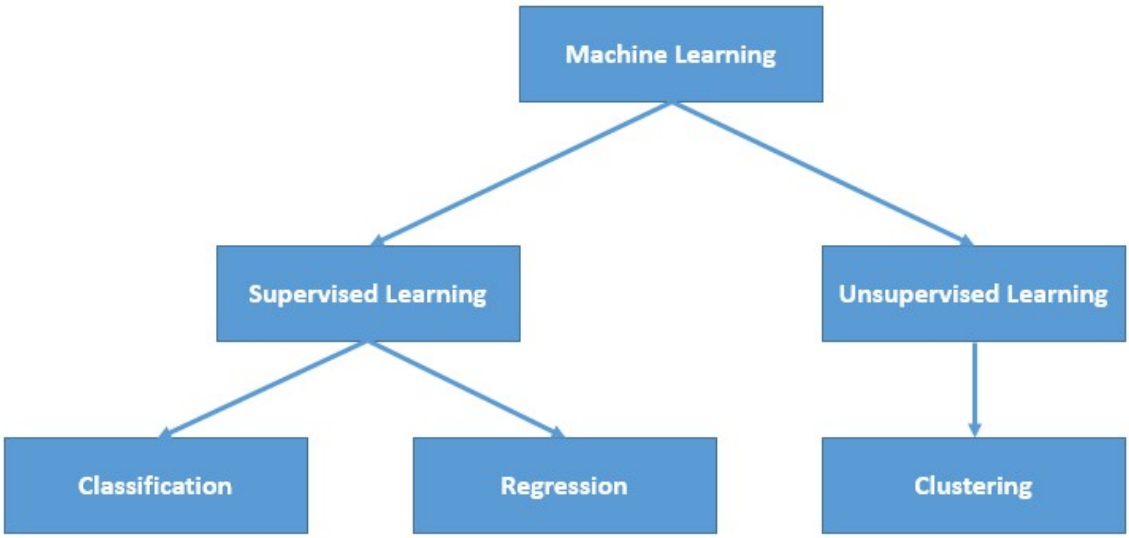


Figure 1.7: Machine learning methods.

8.1.1- Supervised Learning

Supervise ML approaches use labelled datasets for training the algorithm. The training set teach the models to yield the desired output. It includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized. Supervised learning is divided into two main classes: Classification and Regression [21].

a) **Classification:** uses an algorithm to accurately assign test data into specific classes. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labelled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbour, and random forest [21].
b) **Regression:** is used to understand the relationship between dependent and independent variables. Linear regression, logistical regression, and polynomial regression are popular regression algorithms [21].

8.1.2- Unsupervised Learning

This machine learning algorithms are used to analyze and cluster unlabeled data sets. These algorithms search automatically for hidden patterns in data. [21]

Unsupervised learning models are used mainly for clustering, association and dimensionality reduction

a) **Clustering:** is a data mining technique for grouping unlabeled data based on their similarities or differences. This technique is usefull for market segmentation, image compression, etc.

b) **Association:** is another type of unsupervised learning method that uses different rules to find relationships between variables in a given dataset.

8.1.3- Fault diagnosis using ML classification

Machine learning algorithm have become popular in fault detection system due to their reliability, adaptability and robustness. The application of these algorithms have helped in developing a reliable and effective solution for fault diagnosis. These systems do not need prior knowledge on the system mathematical model for their operation and only rely on history datasets. The fault detection in IMs is divided into multiple stages like data acquisition, data processing, feature extraction and implementation of ML algorithms for fault recognition.

We can divide ML-based algorithm for fault classification into: Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Artificial Neural Network (ANN), Decision trees, Bayesian Classifier, random forest and Convolutional Neural Network (CNN).

1.9- Conclusion

This chapter presented an overview of fault detection and diagnosis in PV systems. Starting off by giving a general description of PV systems and common PV system faults and the general scheme for system monitoring. Then, we gave a review on various fault diagnosis approaches. After that, a more specific fault diagnosis techniques were tackled. In (1.7) the multivariate statistical process monitoring techniques were presented, were we mentioned one of the most used one PCA and its application in FDAD. In (1.8) we talked about the use of machine learning in fault diagnosis, while explaining the supervised and unsupervised algorithms and how they differ.

CHAPTER 2: PROPOSED FAULT DETECTION AND DIAGNOSIS METHOD

2.1- Introduction:

Since GCPV systems are quite complex and large, developing a fault detection technique based on the system's historical data seems optimal in this case. As mentioned previously, multivariate statistical monitoring methods are considered to be the most popular due to their effectiveness and rapidity in detecting different faults. This works aims to improve the PCA based fault detection technique by using kernel techniques, and independent component analyses KICA to deal with the non-linear and non-Gaussian characteristics of the system. Moreover, a machine-learning technique is used to classify the detected faults by introducing the PCA and KICA extracted data as features for the proposed model.

2.2- Principle component analyses:

PCA is a multivariate statistical method initially developed by Karl Pearson in 1901 and was later developed by hoteling in 1947. It is mainly used in monitoring processes due to its easy implementation and effectiveness in detecting abnormalities within systems. The basic idea of PCA is to find the principle components representing the main variance possible of a given dataset by performing a linear transformation [22].

Since the initial feature set is always quite large with many interrelated/correlated features, which greatly increases the computation time, using PCA for feature extraction and data compression technique always plays a significant role in solving such time complexity problems.

Advantages of using PCA:

- Removes correlated features effectively and in optimal time: Without PCA, this process is quite complicated and time consuming, especially if the number of features is large.
- Improves machine learning algorithm performance: ML algorithms can take significant training time since they rely on huge datasets. Through PCA dimensionality reduction, the time taken to train the ML model can become significantly lower.
- Reduces over fitting: this is done by removing the unnecessary features in the dataset.
- Improves Visualization: PCA transforms a high dimensional data to low dimensional data so that it can be visualized easily [22].

2.2.1- Dimensionality reduction using PCA:

The main application of PCA is dimensionality reduction of the original correlated data set to an uncorrelated data set that captures most of the information of the original set.

We consider the data matrix X of simulated measured parameters under "healthy" conditions with N samples of m variables.

X=[X1, X2, X3,……..XN] such that X∈ $R^{N\times m}$.

First the data matrix X is normalized to zero mean and unit variance.

$$Xnj = \frac{Xi - \mu j}{\sigma j} \quad (2.1)$$

Where μ and σ are the mean and the standard deviation of the variableX_i.

The covariance matrix is calculated using the training data then decomposed using singular value decomposition (SVD).

$$\Phi=\frac{1}{N-1} Xn^{T}Xn. \quad (2.2)$$

$$\Phi = U^{T} \lambda U \quad (2.3)$$

Where: λ= diag [λ1,λ2,λ3,…..λm] , is the diagonal n× n matrix of n eigenvalues in descending order, and U = [U1,U2,……Um] is the corresponding loading matrix consisting of the eigenvectors of the covariance matrix. Each eigenvalue determine the amount of variance in each component [23].

The transformed data matrix is then given by:

$$X= T U^{T}. \quad (2.4)$$

$$T=X U. \quad (2.5)$$

Where T is the scores matrix.

After retaining *l* principle components, PCA decomposes the principle data set into 2 main subspaces: principle subspace and residual subspace such that:

U= [$\hat{U}$ $\breve{U}$] Where $\hat{U} \in R^{m\times l}$ and $\breve{U} \in R^{m\times(m-l)}$. (2.6)

T= [$\hat{T}$ $\breve{T}$] Where $\hat{T} \in R^{m\times l}$ and $\breve{T} \in R^{m\times(m-l)}$. (2.7)

Λ= [$\hat{\lambda}$ $\breve{\lambda}$] where $\hat{\lambda} \in R^{l\times l}$ and $\breve{\lambda} \in R^{(m-l)\times(m-l)}$. (2.8)

The resulting PCA model is described by the coefficient matrix

$$\hat{C} = \hat{U}\ \hat{U}^{T} \quad (2.9)$$

And the residual PCA model

$$\breve{C} = \breve{U}\ \breve{U}^{T} \quad (2.10)$$

The data matrix X is then expressed as the combination of the retained variations and non-retained variation of X by the projection on the principal space and residual space as follow

$$X=\boldsymbol{X}\hat{U}\ \hat{U}^{T} + \boldsymbol{X}\breve{U}\ \breve{U}^{T} = \boldsymbol{XC} + \boldsymbol{X}(\boldsymbol{I} - \boldsymbol{C}) = \hat{X} + \breve{X} \quad (2.11)$$

We can summarize the feature extraction process of PCA by the flowchart in figure 2.8.

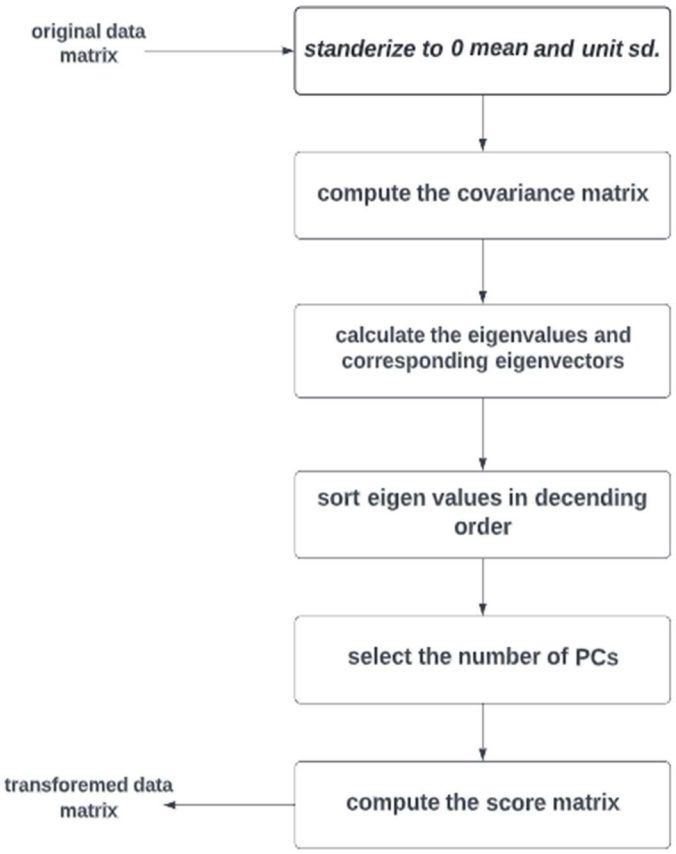


Figure 2.8: PCA feature extraction flowchart.

2.2.2- Choosing the principle components

Choosing the correct number of principle components l to use is quite crucial when applying PCA. For that, many methods were proven efficient to determine just the right number to use.

1. Kaiser-Guttman method (or size of variance): it is used in correlation based PCA, where only components whose eigenvalues that are greater than one (>1) are retained [24].
2. Scree Plot: which is a graphical method of selecting the PCs to be retained by plotting the amplitude of the eigenvalues versus their indices [24]. One way to use Scree Plot in selecting the number of PCs, is looking at the slopes of the lines connecting the points and when these slopes starts to become less steep (at the knee of the graph) , that is the number of PCs that should be retained.
3. Cumulative percent variances CPV is one of the most adopted method in determining the number of PCs proposed by the author of [25]. The main concept of CPV is to retain those l PCs that contribute a specific cumulative percentage of total variation in original data (usually taken from 70 percent to 90 percent), which is calculated using:

$$CPV_l = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \ 100\ \%. \qquad (2.12)$$

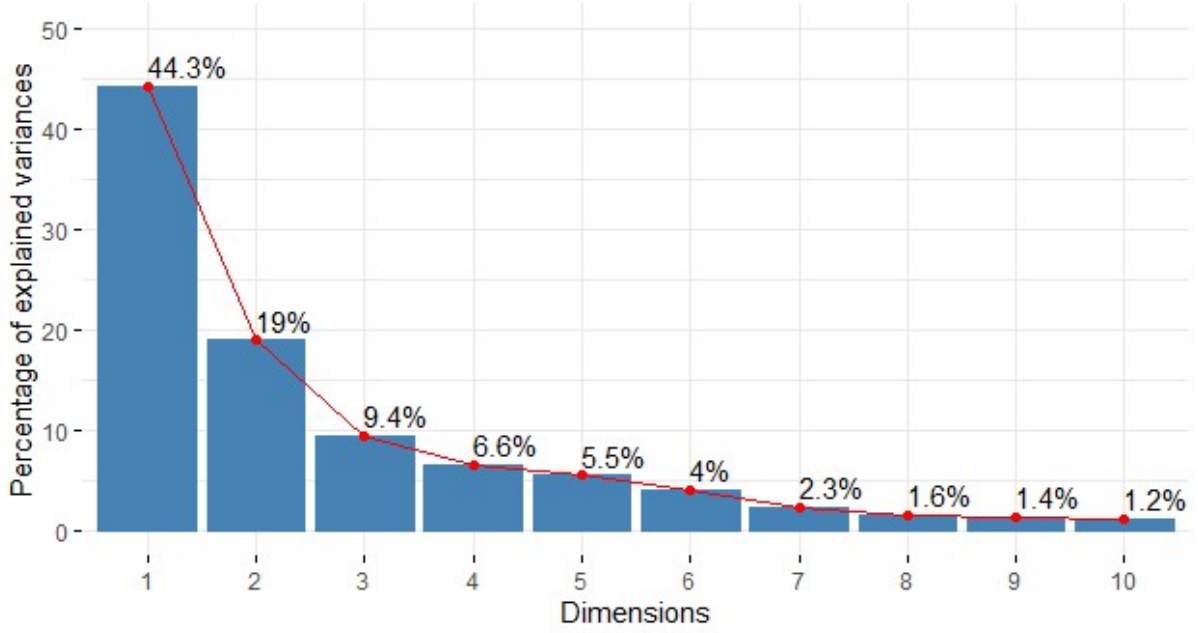


Figure 2.9: CPV for choosing PCs

It is important to mention that selection features using any of these methods is a trade-off. It may appear that the more number of components retained, the accurate the model is, in reality, this may enclose the sensors and the surrounding circuitry and external noises, which will make the model sensitive to noise and uncertainties. In the other hand, selecting lower components may lead to the loss of information, and this will make the model less sensitive to abnormalities. Thus the choice of the right number Pcs is absolutely critical.

2.2.3- Fault detection using PCA:

PCA uses statistical signals as indices for fault detection. Two of the most used are Hotelling's T^2 statistic developed by Hotelling [26], and the square prediction error SPE, developed by Jackson and Mudholkar[27]. T^2 computes the variation within the principle subspace and compares it to that of the healthy model. Whilst SPE deals with the variations within the residual subspace. Both these indices are static thresholds.

Other thresholds based on PCA were developed. Mahalanobis distance based on the T^2, is widely used and forms the global Hotelling's $T2$ test. Raich and Cinar proposed a new combined static based on Mahalanobis distance using both SPE and T^2[28].

In this work, the indices considered are T^2 and SPE

The Hotelling's T^2 statistic is given by:

$$T^2 = x^T \hat{U} \lambda^{-1} \hat{U}^T x. \quad (2.13)$$

For a given confidence level (100.(1-α)%), we define:

$$T\alpha = \frac{(n^2-1)\alpha}{n(n-l)} F\alpha(\alpha, n-\alpha). \quad (2.14)$$

Where Fα is a critical value of Fisher distribution with n and n-α degrees of freedom. It is recommended that α takes the values from 95% to 99%.

Tα defines the system's normal behaviours meaning that any observation above this threshold indicates a faulty behaviour.

The square prediction error SPE index is given by:

$$SPE = \breve{X}^T \breve{X} = \sum_{j=1}^{N} \breve{X}_j^2 \quad (2.15)$$

$$SPE\alpha = \theta 1 \left[\frac{h0c\alpha\sqrt{2\theta 2.}}{\theta 1} + 1 + \frac{\theta 2 h0(h0-1).}{\theta 1^2}\right]^{1/\lambda 0} \quad (2.16)$$

Where:

$$\theta i = \sum_{j=1}^{l} \lambda_j^i \quad (2.17)$$

and

$$h0 = 1 - \frac{2\theta 1\theta 3}{3\theta 2^2} \quad (2.18)$$

cα is the value of the normal distribution with α the significance level.

When a faulty event occurs and ~~it~~ produces a change in the covariance structure of the model, it will be detected by SPE.

2.3- Kernel Principle component analyses:

It is important to note that PCA works based on 2 assumptions: Linearity of the system and its Gaussian distribution. These assumptions are far from reality in most processes. The kernel trick was developed to deal with non-linear processes such that it maps the non-linear data into a high dimensional space driving the system to behave in a somehow a linear way. That is, according to Cover's theorem, the nonlinear data structure in the input space is more likely to be linear after high-dimensional nonlinear mapping [29] into a feature space F. In other words, KPCA performs non-linear principle components analyses.

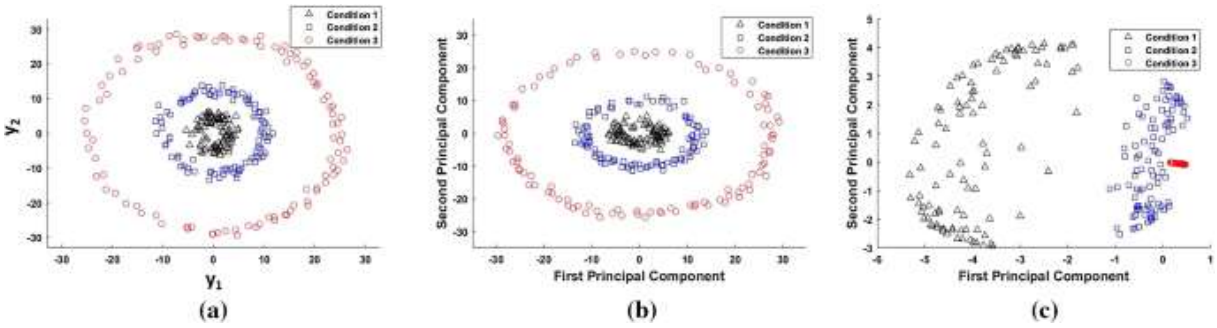


Figure2.10: (a) original dataset, (b) PCA transformation, (c) KPCA transformation.

We assume the mapping into the feature space is expressed by the function Φ(x) where:

$$\Phi = [\Phi(x1), \Phi(x2), \ldots. \Phi(xk)].$$

The covariance matrix is given by:

$$C = \frac{1}{N}\sum_{j=1}^{N} \Phi(Xj)\Phi(Xj)^{\mathrm{T}}. \quad (2.19)$$

Similarly to PCA, we perform singular value decomposition to find $\lambda \geq 0$ eigenvalues and

$V \in F/\{0\}$ eigenvectors satisfying:

$$\lambda V = \bar{C}V \quad (2.20)$$

Where: $\bar{C}$ is the centered covariance matrix.

The V corresponding to the largest eigenvalues represents the principle PCs in F, where the rest Vs represents the residual subspace. $\bar{C}V$ can be expressed then by:

$$\bar{C}V = \left(\frac{1}{N}\sum_{j=1}^{N} \Phi(xj)\Phi(xj)^{\mathrm{T}}\right) V$$

$$= \frac{1}{N}\sum_{j=1}^{N} < \Phi(xj), V > \ \Phi(xj). \quad (2.21)$$

Where $< x.y >$ denotes the dot product between x and y, implying that all solutions V must lie in the span of {Φ(x1), Φ(x2), ……. Φ(xN)}. Hence we consider the equivalent system:

$$\lambda(\Phi(xj) \cdot V) = (\Phi(xj) \cdot \bar{C}V) \text{ for all } j=1,2,\ldots\ldots N \quad (2.22)$$

and there exist coefficients α1, α2, …….αN such that:

$$V = \sum_{j=1}^{N} \alpha . \Phi(xj). \quad (2.23)$$

In general, the mapping Φ(x) may not always be computationally possible although its existence. [30]

To project the input space into the KPCA space, one can avoid calculating the nonlinear mapping and compute instead the dot product given by the kernel function given by [31]:

$$K(x,y) = < \Phi(x), \Phi(y) >. \quad (2.24)$$

The kernel trick is based on the Mercer's theorem, which states that if a kernel function is continuous kernel of a positive integral operator, there exists a mapping into a space where the kernel function acts as a dot product. [30]

There exists several kernel functions, the most used ones are:

- Polynomial Kernel

$$K(x,y) = < x, y >^{\mathrm{d}}. \quad (2.25)$$

- Sigmoid Kernel

$$K(x,y) = \tanh(\beta_0 < x, y > + \beta 1). \quad (2.26)$$

- Radial Basis Kernel

$$K(x,y)= \exp\left(-\frac{\|x-y\|^2}{c}\right). \quad (2.27)$$

- Linear kernel

$$K(x,y) = < x, y >. \quad (2.28)$$

Where d, β_o, β1, and c are kernel parameters to be determined by the user. A specific choice for a kernel function will determine implicitly the mapping Φ(x) [32].

In this work the Radial Basis function is considered. It should be noted that before applying PCA, mean centering of the kernel matrix has to be performed. This can be done using the following expression:

$$\bar{K} = K - 1_N K - K 1_N + 1_N K 1_N. \quad (2.29)$$

Where:

$$1_N = \frac{1}{N}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \in R^{N \times N}. \quad (2.30)$$

We obtain :

$$N\lambda K\alpha = K^2\alpha. \quad (2.31)$$

Where α represent the column vector of [α1, α2, …… αN]. Solving (2.31), implies the following equation:

$$N\lambda\alpha = K\alpha. \quad (2.32)$$

For all non-zero eigenvalues λi. The reader is referred to [31] for more details.

Now, PCA is performed in the obtained F feature space to solve equation (2.32). The dimension of the new space can be found by retaining only l eigenvectors corresponding to l largest eigenvalues. [32]

α1, α2,…..αl are normalized by requiring that the corresponding vectors in F are also normalized. i.e. $< vk. vk >= 1. for\ all\ k = 1,2, \dots l.$

Using:

$$V= \sum_{j=1}^{N} \alpha_i \Phi(xi). \quad (2.33)$$

We obtain

$$1= < \sum_{i=1}^{N} \alpha_i \Phi(xi), \sum_{j=1}^{N} \alpha j \Phi(xj)$$

$$= < \alpha_k , K\alpha_k >$$

$$= \lambda < \alpha_k, \alpha_k >. \quad (2.34)$$

The Pcs P_k are then extracted by projecting $\Phi(xi)$ into the eigenvectors v_k in F where k=1,2,…l.

$$P_k= \sum_{i=1}^{N} \alpha_i < \Phi(xi), \Phi(x) >. \quad (2.35)$$

KPCA is also used for monitoring and Fault detection using T^2 and SPE indicator. T^2 is calculated using kernel function:

$$T^2 = k(x)^T V \lambda^{-1} V^T k(x). \quad (2.36)$$

Similarly to PCA, the control limit is calculated using the F-distribution.

The SPE index is defined in the feature space as follow: [30]

$$SPE = k(x,x) - k^T(x)Ck(x) \text{ and } C = V^T V. \quad (2.37)$$

2.4- INDEPENDENT COMPONENT ANALYSIS ICA

2.4.1- Background

ICA is a statistical approach that has the potential ability for blind source separation (BSS) without the prior information about the mixtures under the source signals that are statistically independent. The concept of independent component analysis is similar to that of PCA and KPCA. Its goal is to find a mapping that transforms the data to a feature space where the data becomes as statistically independent from each other as possible by maximizing some function A that measures "independence".

ICA of a random vector x consists of estimating the following generative model for the data:

$$x = As. \quad (2.38)$$

This model was introduced by Jutten and Hérault in their seminal paper [33], which was probably the earliest explicit formulation of ICA.

The identification of the ICA model has been treated based on the following assumptions:

1- All the independent components s_i, with the possible exception of one component, must be non-Gaussian.
2- The number of observed linear mixtures m must be at least as large as the number of independent components n, i.e., m smaller n.
3- The matrix A must be of full column rank.

Plus the assumption that both x and s are centred. These restrictions implies based on the assumption that the x variables are some random variables [33].

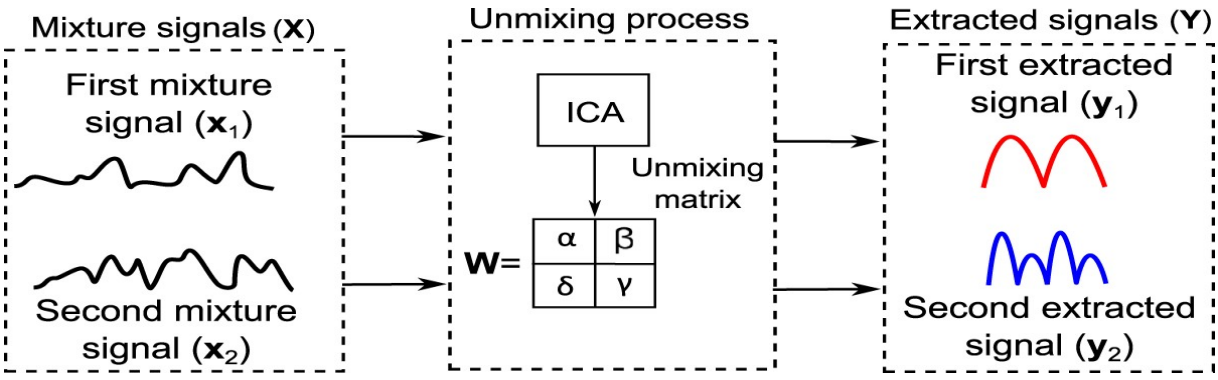


Figure 2.11: ICA process.

The applications of ICA can extend to feature extraction. The use of ICA for feature extraction is motivated by results in neurosciences that suggest that the similar principle of redundancy reduction explains some aspects of the early processing of sensory data by the brain. ICA has also applications in exploratory data analysis in the same way as the closely related method of projection pursuit. [34]

The goal of ICA is then to estimate the mixing matrix W to extract the independent component from the projected data. This matrix can be estimated using three main approaches:

- Maximizing the non-Gaussianity: the non Gaussianity can be measured using function such as the negentropy function or the kurtosis function.
- Minimizing the mutual information.
- Maximum likelihood estimation.

The definition of ICA given above implies no ordering of the independent components as opposed to PCA. It is possible, however, to introduce an order between the independent components using for example the norms of the columns of the mixing matrix A, which gives the contributions of the independent components to the variances of the xi. A second way is to use the non-Gaussianity of the independent components i.e., ordering the independent components according to non-Gaussianity [34].

In this work we consider ICA by maximizing the non-Gaussianity.

2.4.2- measure of non-Gaussianity:

Maximizing the non-Gaussianity is one way of finding the independent components of data vectors. For that purpose, two measurements can be used: kurtosis and negative entropy.

- **Kurtosis:**

The ICs can be obtained by finding the scores, which maximizes kurtosis of extracted signals. The Kurtosis (K) of any probability density function (pdf) is defined as follow:

$$K=E\left[x^4\right]-3\,E\left[x^4\right]^2 \tag{2.39}$$

The kurtosis is simple to calculate, however it should be mentioned that it is sensitive to outliers.

For a whitened data Z, $E\left[Z^2\right]$= 1 since Z has unit variance.

Hence, the Kurtosis will be:

$$K(Z) = E\left[x^4\right]-3. \tag{2.40}$$

Given two source signals s1 and s2, and the matrix $Q=A^TW=A^{-1}W$. Hence,

$$Y=W^TX=W^TAS=QS=q1s1+q2s2. \tag{2.41}$$

Using the kurtosis additively property, we have:

$$K(Y)=K(q1s1)+K(q2s2)=q_1^4K(s1)+q_2^4K(s2) \quad (2.42)$$

A scaling step is performed so that s1, s2, and Y have a unit variance, meaning that Q is constrained to a unit circle in the 2D space. This implies that:

$$E[Y2]=q_1^2E[s1]+ q_2^2E[s2]= q_1^2+q_2^2=1. \quad (2.43)$$

The optimal solutions of the kurtosis in this case are the points when one of Q is zero and the other is nonzero either be 1 or −1 , where each vector in the matrix Q extracts only one source signal since $Q=A^TW=I$.

the ICs are then the ones which maximizes kurtosis of extracted signals $Y=W^TZ$, where Z is the whitened data. Thus, the kurtosis can be expressed as:

$$K(Y)=E[(W^TZ)^4]-3. \quad (2.44)$$

Where the term (E[yi2])2 is set to one because W and Z have a unit lengths.

To find the gradient of the kurtosis K(Y), the following formula is used:

$$\frac{\partial K(W^TZ)}{\partial W} = E\,[Z(W^TZ)^3]. \quad (2.45)$$

It should be highlighted that the weight vector is updated with each iteration such that:

$$Wn = Wo + \eta\, E[Z(Z(Wold^TZ)^3]. \quad (2.46)$$

Considering that η is the step size for the gradient descent [34].

- **Negative entropy:**

Negative entropy is termed negentropy, and it is defined as follows:

$$J(y) = H(y_{gaussian}) - H(y). \quad (2.47)$$

where $H(y_{gaussian})$ is the entropy of a Gaussian random variable whose covariance matrix is equal to the covariance matrix of y. [33]

The entropy of a random variable *Q* which has *N* possible outcomes is

$$H(Q)=-E[\log p_q(q)]= -\frac{1}{N}\sum_t^N \log p_q\,(q^t). \quad (2.48)$$

Where $p_q\,(q^t)$ is the probability of the event q^t=1, 2, …, N. [34]

The negentropy is zero when all variables are Gaussian. In this work, maximizing non-Gaussianity using Kurtosis was considered.

2.5- Kernel Independent component analyses KICA:

Since ICA assumes that the problem is linear, it fails to separate nonlinear mixed source signals. To tackle this problem, a method based on the kernel trick is introduced. KICA (kernel independent component analyses) combines the benefits of Kernel functions in dealing with nonlinearities and the ICA for non-Gaussian problems.

2.5.1- data whitening using KPCA:

Before getting the linear and orthogonal scores that the ICA algorithm requires, the first step of KICA is to whiten the measurement matrix in the kernel space. Data whitening means transforming the original data set into uncorrelated set then rescale each vector to have unit variance. It is important to note that a mandatory centring step should be performed beforehand. The whitening step is usually done using PCA or KPCA [34] [35].

In this work, kernel principle component analyses is used. As discussed previously, this is done using a non-linear mapping Φ(x), where the kernel method will transforms a measurement vector into a higher-dimension feature space to get linear associations [30], then the ICA algorithm can be applied to the higher-dimension linear feature space.

Denoting V = (α1,α2,…….,αk) the kernel eigenvectors matrix and λ = (λ1,λ2, …., λk) the corresponding eigenvalues diagonal matrix, the whitening matrix can be obtained using:

$$P = \sqrt{N}\ \Phi\ V\ \lambda^{-1}. \tag{2.49}$$

Then the mapped training data in the feature space can be whitened as follow:

$$Z_i = P^T \Phi = \sqrt{N}\ \lambda^{-1}\ V^T\ K_i^T. \tag{2.50}$$

Where Ki is the i th row of the Gram matrix K. Whitening the data in the feature space using KPCA is considered to be the initial step of KICA. Choosing fewer whitened scores can reduce the number of ICs effectively but may lose some important information. Therefore, reducing the whitened scores should be done carefully. In this work, a criterion, λi/sum(λ) > 0.0001 [34], is used to select the whitened scores.

2.5.2- Online monitoring and fault detection

For a new data X new to be monitored, its kernel-mapping vector knew should be cantered as follow:

$$K_{new} = Knew - 1new\ K - Knew\ 1_N + 1new\ K\ 1_N. \tag{2.51}$$

The whitened new scores are:

$$Z_{new} = P^T\ \Phi(x_{new}) = \sqrt{N}\ \lambda^{-1}\ V^T\ K_{new}^T. \tag{2.52}$$

To estimate W and obtained the ICs, the ICA algorithm used is kurtosis. After d ICs have been selected, the first index for monitoring can be given by:

$$I^2 = s_d^T\ .\ s_d \tag{2.53}$$

Where d is the number of retained independent components. Here, the ICs with larger kurtosis are selected to be the dominant ones, since larger kurtosis means stronger nonGaussianity [35].

Unlike the linear methods, the SPE of KICA cannot be calculated by the residuals of the original measurement because the non-linear mapping Φ(x) is unknown. However, the residual of mapped data into the kernel space can be used to calculate SPE indirectly:

$$\text{SPE} = \|\Phi(x^{\iota})\ \Phi'(x^{\iota})\| = \|\Phi - \Phi'\|^2 \tag{2.54}$$

Since the distributions of I^2 and SPE are unknown, their control limits can be calculated by using the kernel density estimation KDE [36]. It is defined as non-parametric way to estimate the probability density function of a random variable. The kernel density for a distribution f of a random variable x is given by:

$$\text{D(x)} = \frac{1}{n}\sum_{i=1}^{n} Kh(x - xi) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-xi}{h}). \tag{2.55}$$

When a new online measurement is available, I^2 new and SPE new are computed, and compared to that of the healthy case, if they are found to be beyond their control limits, a fault has occurred.

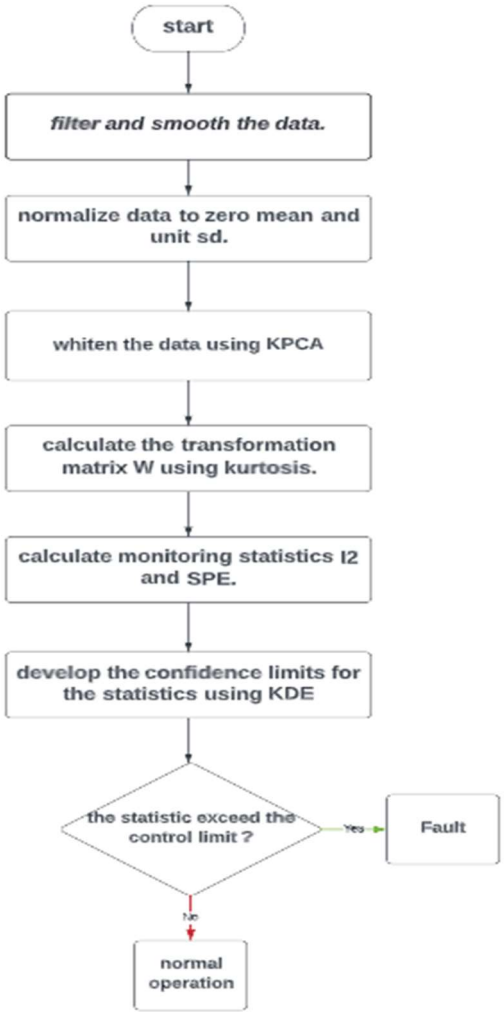


Figure 2.12: KICA flowchart

2.6- Fault diagnosis using support vector machines

2.6.1- SVM theoretical background

SVM or support vector machines, is one of the most popular classification techniques designed to solve binary classification problem.

The main advantage of SVM is the ability to perform both linear and non-linear classification. SVM training algorithm transforms training samples to a higher dimensional space that maximises the gap between the two classes or categories as much as possible. Then, the new samples are mapped into that same space and predicted to belong to one of the two classes based on which side of the gap they fall into. The model created is called a hyperplane.

A good separation of classes is achieved when the hyperplane has the largest distance to the nearest training-data point of any class.

The SVM can avoid the shortcomings of overlearning, under-learning, and biased optimization that easily occur in other intelligent algorithms, for example, the BP neural network, and it has stronger generalization ability compared to the BP neural network [36].

Consider a set of training samples $\{xk, yk\}$ which takes xk ∈ R as inputs, and yk= $\{1, -1\}$ as labels for each class. The svm model is given as:

$$Yk = f(xk) = w^T xk + b. \tag{2.56}$$

A good separation is obtained given that the margin defined by $\frac{1}{\|w\|}$ is maximized. This is equivalent to solving the following optimization problem.

$$\min \frac{1}{2}\|w\| \text{ subject to: } \quad Yk\,(w^T xk + b) - 1 \geq 0\ \forall k. \tag{2.57}$$

Hence, as for linear separable data, the optimal separating hyperplane satisfies the following function:

$$\min Q(w) = \frac{1}{2}\|w\|^2. \tag{2.58}$$

Taking the noise in the data and the misclassification of hyperplane into consideration, we reformulate the described function of the separate hyperplane:

$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, i = i \ldots, m \tag{2.59}$$

where the variable $\xi_i \geq 0$ represents a measure of distance from hyperplane to misclassified points. To find the optimal generalised separating hyperplane, the following optimal problem should be solved:

$$\min_{w,b} \frac{1}{2} \| w \|^2 + C \sum_{k=1}^{m} \xi_i \tag{2.60}$$

s.t: $$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \tag{2.61}$$

$$\xi_i \geq 0, i = i \ldots, m$$

Where the parameter C, a given value, is called error penalty. As for the above-mentioned data inseparable case, in order to simplify the optimal problem, define the Lagrangian to be $\ell(w, b, \xi, \alpha, \gamma) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i [y_i(\langle w, x_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{m} \gamma_i \xi_i$

Where α, γ are the Lagrangian multipliers. We consider the minimization problem as original, primal problem. Consider

$$\min_{w,b,\xi} \theta_p(w) = \min_{w,b,\xi} \max_{\alpha,\gamma} \ell(w, b, \xi, \alpha, \gamma) \tag{2.62}$$

When satisfying the Kuhn-Tucker condition, then the primal problem is transformed to its dual problem, which is

$$\max_{\alpha,\gamma} \theta_d(\alpha, \gamma) = \max_{\alpha,\gamma} \min_{w,b,\xi} \ell(w, b, \xi, \alpha, \gamma) \tag{2.63}$$

The objective is minimizing ℓ in (2.63) by adjusting the value of w, b, ξ. At the optimal point, derivatives of ℓ should be zero.

$$\frac{\partial \ell}{\partial w} = 0, \Longrightarrow \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{2.64}$$

$$\frac{\partial \ell}{\partial b} = 0, \Longrightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{2.65}$$

$$\frac{\partial \ell}{\partial \xi} = 0, \Longrightarrow \alpha_i + \gamma_i = C. \tag{2.66}$$

the dual quadratic optimization problem s then obtained:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{m} \alpha_k \right\} \tag{2.67}$$

And satisfy the constrains:

$$0 \leq \alpha i \leq C, i = 1 \ldots, m, \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{2.68}$$

When solving the dual quadratic optimization problem, the αi will be obtained. Then the form of the hyperplane can be changed to

$$\langle w, x_i \rangle + b = \sum_{i,j=1}^{m} \alpha_i y_i (\langle xi, x_j \rangle + b) . \tag{2.69}$$

The classifier implementing the optimal separating hyperplane has following form:

$$\text{f(x) = sgn } (\sum_{i,j=1}^{m} \alpha_i y_i (\langle xi, x_j \rangle + b)). \tag{2.70}$$

Usually linear classifier is not a suitable solutions, SVM uses the kernel trick for mapping the data into high dimensional feature space to deal with nonlinear behaviour. The classification problem then becomes:

$$f(x) = \text{sgn}\left(\sum_{i,j=1}^{m} \alpha_i y_i(\langle K(xi, x_j)\rangle + b)\right). \quad (2.71)$$

Where $K(xi, x_j)$ is the kernel function. [32]

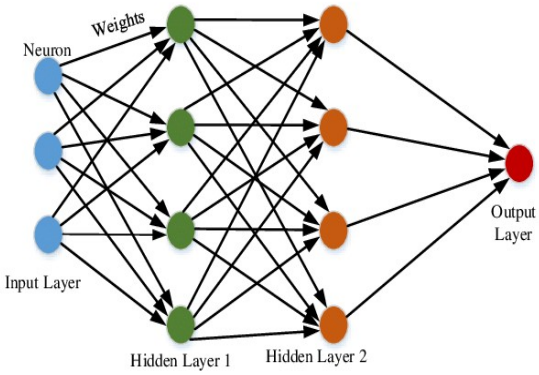


Figure 2.13: SVM architecture

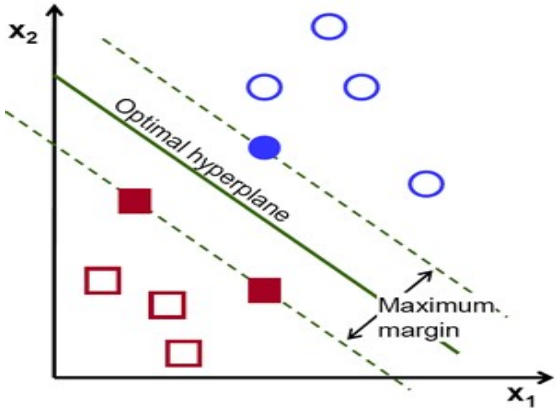


Figure 2.14: SVM data separation principle

2.6.2- SVM Model Selection

Building an SVM based model for fault detection is a relatively simple procedure. Training dataset and testing sets are first split. Usually the training set is composed of 70 to 80% of the total dataset leaving the rest for testing. An SVM training set contains two parts: one is the class label and the other is several features (observed variables). Thus the SVM data classifier will first build a model based on the training set and then use it to predict the target value of the data in testing set where only the observations are known.

The SVM based fault detection and classification algorithm can be summaries in these steps [39].

- Transform data collected from real process to the format that SVM classifier can use. In our model two data transforming methods were used: the PCA and the KICA to determine the optimal classification.
- Try a few Kernel functions to optimize the best one for the model. The optimal parameters for the kernel functions are obtained using cross-validation. This thesis uses Gaussian RBF Kernel function.
- Test the model using the testing data.

Originally SVM was designed for solving two class classification problems. To extend it to perform multi-class classification, several approaches were designed. One approach is to split the multi-class classification dataset into multiple binary classification ones and fit a binary classifier on each. This approach could be performed using either the One-vs-One method or One-vs-All method [39] [40].

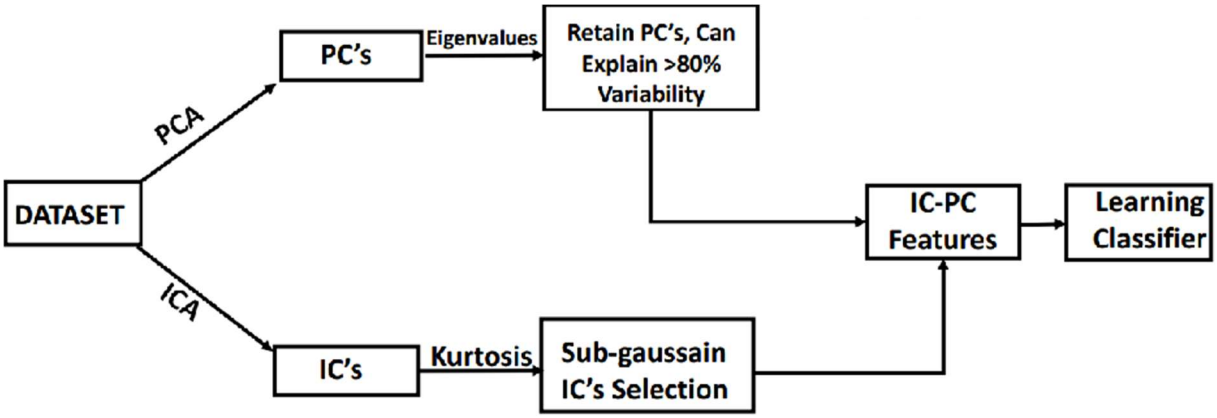


Figure 2.15: Proposed SVM model

2.6.3- Cross-Validation

As discussed previously, when solving a nonlinear classification problem, Kernel functions are used. Selecting the appropriate parameters is a crucial step to ensure that the classifier will accurately predicts unknown data. The optimal parameter searching can be accomplished using cross-validation algorithm [23].

The cross-validation algorithm is a model validation method used to evaluate the accuracy of a predictive model. Cross-validation gives an insight on how the model generalizes to an unknown dataset by testing the model using a defined dataset during the training process. To perform k-fold validation, we divide the dataset randomly into k classes, with equal sizes. Sequentially, one data subsets is used a testing set for a predictive model with the rest of the subset as training set. This process is repeated k times, with every subset acting as a testing set once. The repeating process helps to pick out the parameters which make the overall misclassification rates of all the testing sets minimum.

This thesis uses 10-fold cross-validation algorithm to find out the overall misclassification rate across all testing sets.

2.7- Conclusion

In this chapter, the Principle component analyses technique for dimensionality reduction and process monitoring were elaborated in the aspects of mathematical modelling, principle components selection, and fault detection indicators. Moreover, kernel independent component analyses was described by first introducing each of the kernel trick, KPCA, and ICA techniques, then deriving monitoring statistic equations. Lastly, fault diagnosis based on support vectors machines was discussed by walking through the mathematical foundations of the SVM, the model selection steps and cross validation for parameter estimation.

CHAPTER 3: Results and discussion

3.1- Introduction:

The proposed method aims to detect the injected PV faults with high accuracy, minimum detection time, and low false alarm rate.

The main focus of this approach is to diagnose all the faults, especially the controllers' faults which are hard to detect as the controller is designed to overcome the happening faults. The hybrid analytic-ML technique is designed on 2 stages:

- Developing analytic fault indicators for a quick fault detection.
- Training the machine learning classifier for fault diagnosis.

Before coming to the obtained results, this chapter first provides a description on the lab implemented circuit, the collected data and the injected faults. Both standard PCA and the proposed KICA methods are applied on the collected data, the results are then further injected as features for the proposed SVM classification model. The results of each method are then compared based on the Fault detection rate FDR and the false alarm rate FAR, and the ML model accuracy.

3.2- System description and data acquisition:

To evaluate our approach and ensure its reliability, a good and precise knowledge about the different causes of each possible fault is needed, in order to well label the data of the different faults. In this project, labeled data from an experiment conducted in a research laboratory in Malaysia is used [15]. PV array emulator and grid emulator are used instead of real PV array and grid, in order to be able to inject different faults in these two main parts of the GCPV system with accurate labeling of each fault.

The PV array output is generated through the programmable Chroma 62150H-1000S solar array emulator that allows varying effects of environmental conditions (irradiance and temperature). The programmable AC source Chroma 61,511 is used and set to the three phase mode to match a real grid system network. A DC-DC converter was implemented using a chopper for maintaining the output current of the PV panel emulator to a specific level as much as possible. A DC-AC Inverter is implemented using six IGBTs to provide a three phase signal that will be transmitted to the grid emulator system after rectification and transformation.

The control algorithm was implemented using the DSpace 1104 environment with MATLAB, which is also used for data acquisition. The grid phase synchronization is achieved using Voltage Oriented Control (VOC) technique in combination with Space Vector Pulse Width

Modulation (SVPWM). This is done by controlling the active and reactive power based on the grid-side signals. The output voltage is synchronized with the grid voltage through the Phase Lock Loop (PLL). The AC load in this work is used for protection purposes while injecting real faults. This MPPT controller used in this system is based on Particle Swarm Optimization (PSO) technique. This system is used to generate and collect real faulty data for experimental validation of real-time online FD.

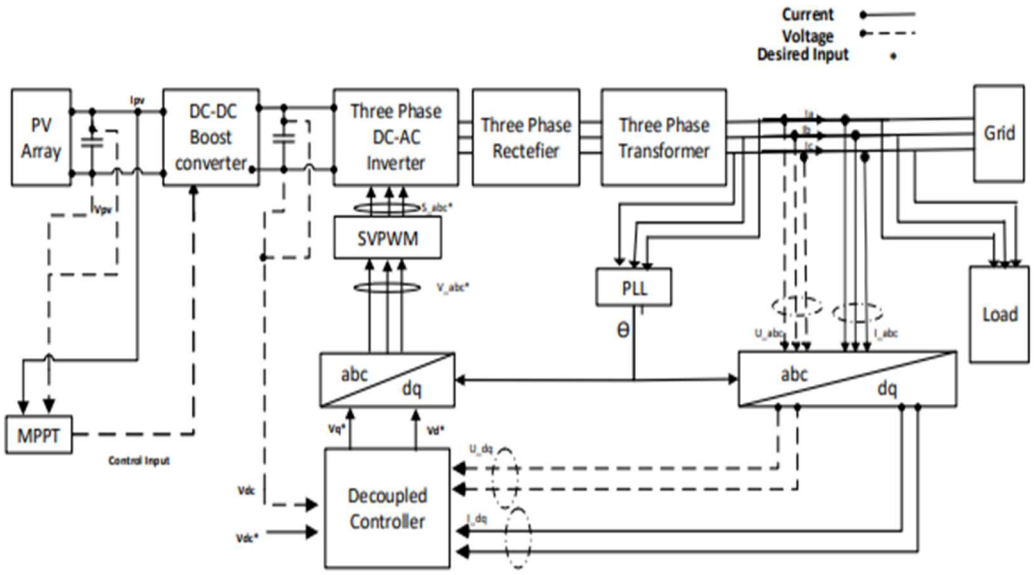


Figure 3.16: circuit description of the GCPV system implemented in the lab [15].

The real time collected measurements are: PV system voltage and current, DC voltage, grid voltage and grid current. Acquired with sampling time of Ts=100 µs. This method is designed and validated based on a small-scale micro-grid application where data are collected from sensor measurements and a virtual PMU is used to extract the positive sequence components from three-phase signal.

Table 3.1: measurement description.

Measurements	*Symbols*	*description*
PV outputs	VPV IPV	The output voltage and current of the PV array emulator.
DC output	Vdc	The output voltage of the DC-DC convertor
Grid three-phase currents	Ia Ib Ic |Iabc|, f_i	The three phase currents of the grid after DC-AC inverter. Rms current and frequency of the grid.
Grid three-phase voltages	va vb vc |Vabc|, f_v	The three phase voltages of the grid after DC-AC inverter. Rms voltage and frequency of the grid.

3.3- Fault description and analysis

This project considers the detection of 7 faults listed in table1. All faults are injected manually in several experiments running from 10 to 15 seconds, where the fault is introduced between the 7th and 8th seconds except for the faults in the controller which were introduced around the 10th second. The sampling time for the data acquisition is 100us [15].

The faults injected can be classified into 2 sections:

- Faults on the DC side of the system.
- Faults on the AC side of the system.

Table 3.2: injected faults description.

Faults	Fault side.	Fault type	Fault description
F2	DC Side	• Feedback Sensor fault	One phase sensor fault 20%
F4		• PV array mismatch	10 to 20% nonhomogeneous partial shading.
F5		• PV array mismatch	15% open circuit faults in the PV array.
F6		• MPPT controller fault.	−20% gain parameter of PI controller in MPPT controller of the boost converter.
F7		• Boost converter controller fault	+20% in time constant parameter of PI controller in MPPT/IPPT controller of the boost converter.
F1	AC Side	• Three phase inverter fault	Damage of one IGBT at a time among the total of 6 IGBTs inside the threephase inverter.
F3		• Grid anomaly (external connection faults)	Intermittent voltage sags

To investigate the exact time for each fault injection, the different PV system' measurements were plotted for each experiment. The obtained plots are shown in Appendix A.

- Inverter Fault F1: this type of fault is quite easy to detect since it affects mostly the AC part of the system. As demonstrated in the appendix, the fault started at ts=8.58s until around tf=13s. Due to their severity, however, these faults must be detected at their early stages within a limited delay time.
- Feedback current sensor fault F2: this fault effect the DC side of the PV system. The fault is injected at ts=8s and lasts the remaining of the experiment.
- Intermittent voltage sags fault F3: similarly to the inverter fault, this fault effect the AC side of the grid. The fault happened around ts=6.5s and lasted for the remaining experiment.
- Array mismatch faults F4 and F5: both shading fault F4 and open circuit array faults F5 are challenging to detect due to the large variability in sensor data at the DC-side;

Fortunately, these faults are of lower severity levels causing mainly power losses as demonstrated by the Ipv and Vpv graphs.
The Fault F4 was introduced at ts=8s and lasted the whole experiment, while F5 was introduced at ts= 8.87s. It is clear that the system shut down around tf=13s.

- Parametric faults F6 and F7 in MPPT/ IPPT Proportional Integral (PI) controller: these faults are classified as DC side faults. Controller fault F7 indicates an increased time-constant parameter whereas F6 is a biased gain in the PI controller which results in a reduced MPPT/ IPPT trajectory tracking performance without affecting the stability of the closed-loop system. This can be clearly seen from the Vdc and Vpv curves, as only a small variation in the voltages is shown. These faults are widely common in practice. Both faults were injected at ts=10.5s and lasted the remaining of the experiment.

3.4- Proposed fault detection and diagnosis method

In this project, fault detection using both PCA-SVM and KICA-SVM is applied. Both methods are compared using fault detection rate indicator and false alarms rate indicator.

The experiment is applied on data under MPPT mode, where ¾ of the data set is used for training the models while the rest is used for validation.

3.4.1- FD using Principle component analyses

Since the measurements are highly noisy and corrupted, a pre-processing stage was performed where a smoothing moving average filter was applied.

such that : $$x_i = \frac{1}{\sum_{i=1}^{w} r}\, r^i y_{j-w+i.} \quad (3.1)$$

Where $y_{i.} = y_{ij}$ for j = 1 to 7 represents the *ith* row vector measurement of all variables. r is a weighting factor that controls the smoothing , and w is the window length.

After filtering and smoothing the original data set, dimensionality reduction using Principle component analyses was performed. The data was first centered to 0 mean and unit variance.

To choose the number of PCs, CPV method was used. As discussed previously, this method observe the explained (or the variance) percentage for each component.

Figure 3 illustrates the number of components versus the CPV of each one. The highest contributing components are retained. In his work, the first 4 components represents over 80% of the total variance.

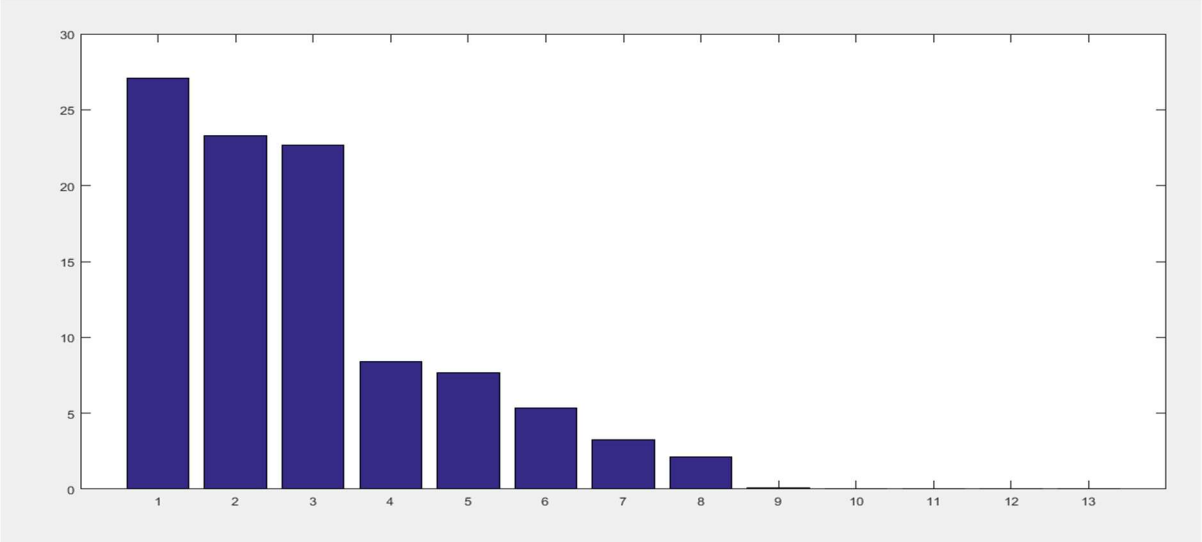


Figure 3.17: Explained variance by each principle component.

To perform PCA fault detection, first PCA is applied on the whole set of healthy data and retains only the first 4 components. The control limits of T^2 and SPE are calculated to set the thresholds. The faulty sets are then projected into the principle subspace. Then, monitoring statistics were calculated so that the systems fault can be detected.

For validations, we chose to illustrate the results applied on 3 sample data, one for each fault type. For the AC part we chose the inverter fault F1, while for the DC part we chose the open circuit fault in the PV array F5. Since detecting controller faults is such a challenge, we focused also on the detection of the MPPT fault F6.

Figures 16 and 17 represent the obtained PCA results applied on the faulty set F1. The threshold was obtained considering 95% tolerance. Similarly Figure 18,19 represent the obtained PCA results applied on the faulty set F5 circuit , and figures 20,21 represent the obtained PCA results applied on the faulty set F6.

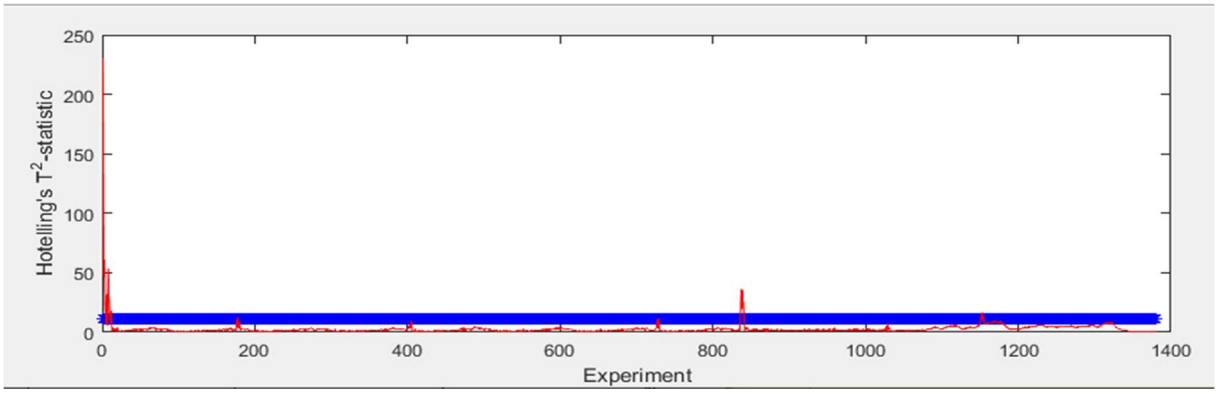


Figure 3.18: T^2 fault indicator for inverter fault F1.

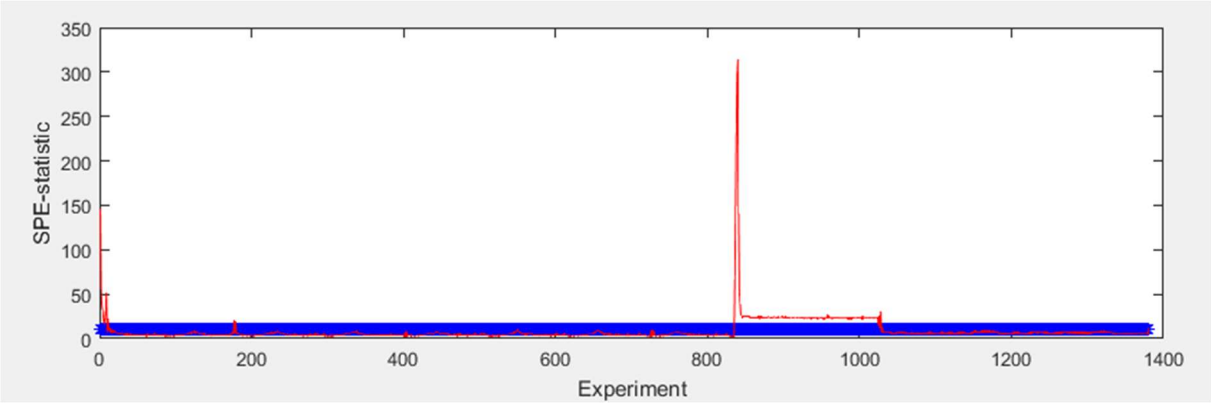


Figure 3.19: SPE fault indicator for inverter Fault F1.

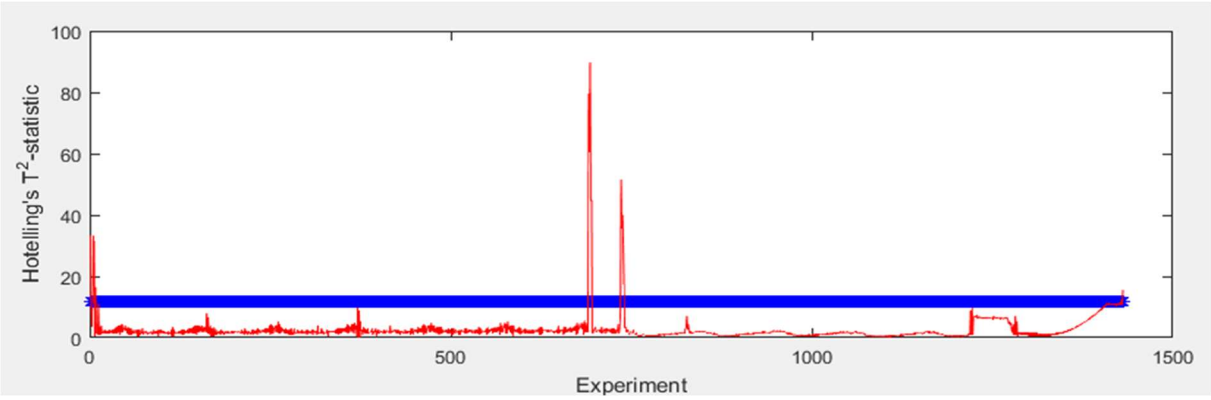


Figure 3.20: T^2 fault indicator for open circuit faults F5.

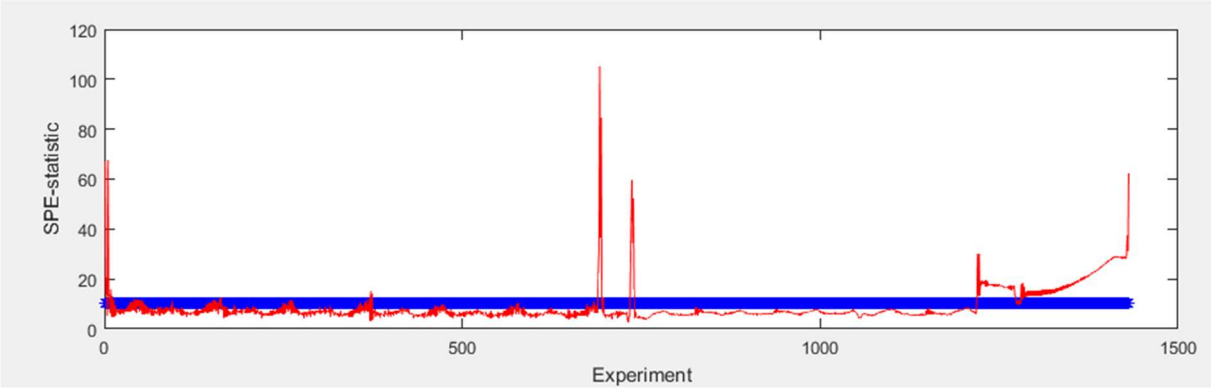


Figure 3.21: SPE fault indicators for open circuit faults F5.

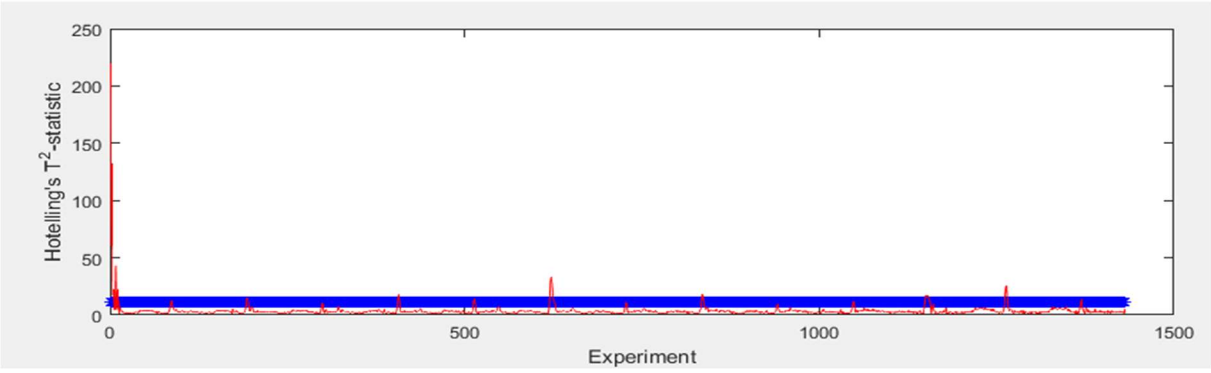


Figure 3.22: T^2 indicator for MPPT controller fault F6.

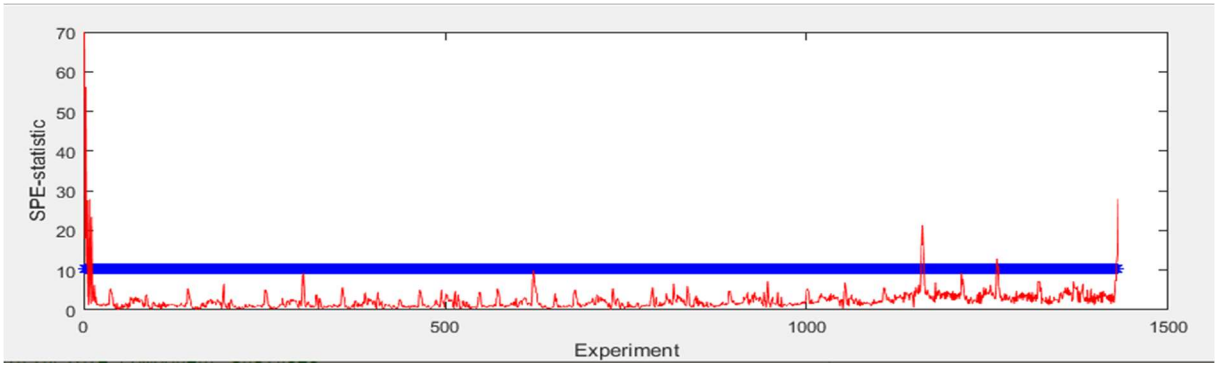


Figure 3.21: SPE fault indicator for MPPT controller fault F6

3.4.2- Discussion

PCA was successful in detection only some of the faults including F1, F2, F3 and F5. However, it can be clearly seen that it fails to show when exactly the fault is cleared. Also, It can be noticed that the rate of false alarms is quite high, and the detection was a little delayed in some cases such as F5 and F4, considering that the faults were introduced between the 7^{th} and 8^{th} second. In the case of F6, both T^2 and SPE could not detect any deviation from the healthy case meaning that PCA is ineffective in detecting controller faults. The same case for F7.

Our guess was that the reasons for these results were the nonlinear nature of real time processes, also the heavy assumption made when using PCA that the process follow a Gaussian distribution. To investigate a solution for this problem, the KICA technique is proposed.

3.4.3- FD using Kernel independent component analyses

As mentioned previously, KICA is implemented in 2 steps: KPCA for mapping the nonlinearities to a higher dimensional linear Feature space, then ICA for mapping the data into a subspace where data is as independent from each other as possible.

For performing the kernel PCA, the Radial Basis function was used. Since there is no theoretically proven way to find the kernel parameter, it was found by trying and testing repeatedly until good detection results were obtained. Eventually, setting c=20 has led to satisfactory detection results.

A step for data whitening was necessary before finding the ICs. In this work, KPCA is used for mapping the data into a high dimension linear subspace and Whitening the signals. Using the kurtosis algorithm, the mixing matrix W is obtained, then the ICs are estimated. The projected data is then used to estimate the monitoring statistics I^2 and SPE.

Similarly to the PCA approach, the model was validated on the faulty data sets. We demonstrated 3 faulty samples: F1 (inverter faults), F5 (open circuit faults) and F6 (MPPT controller faults).

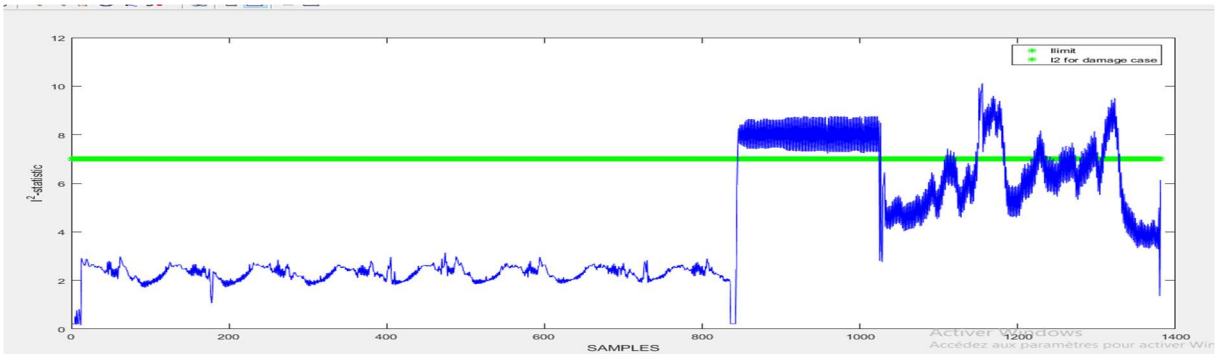


Figure 3.22: I² fault indicator for inverter fault F1.

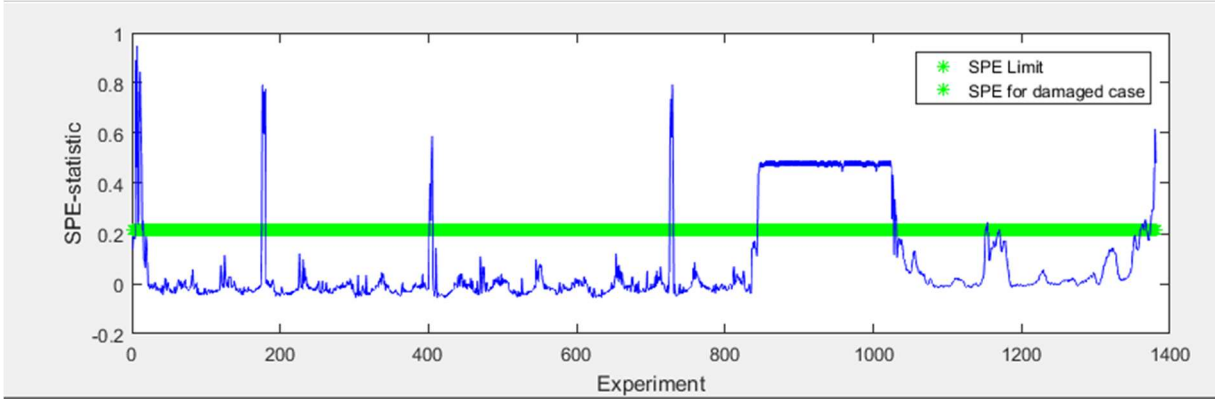


Figure 3.23: SPE fault indicator for inverter fault F1

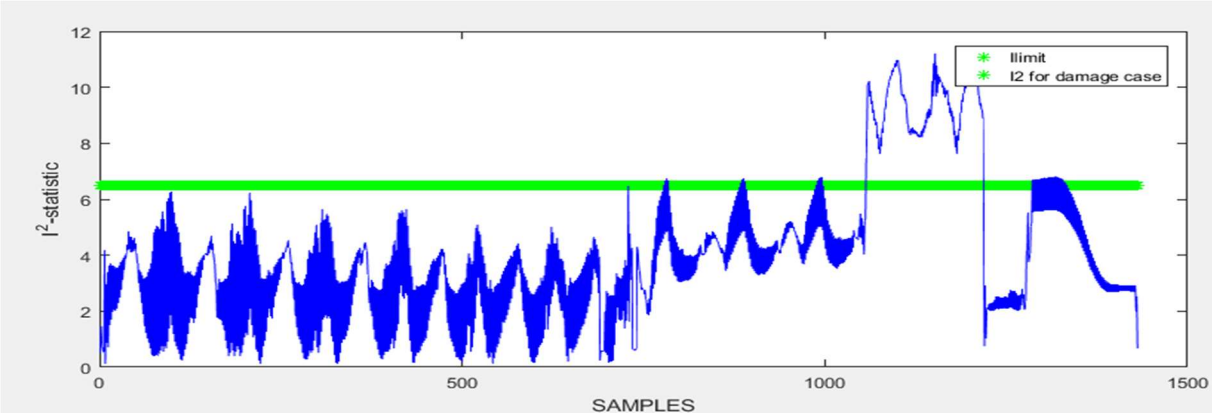


Figure 3.24: I² fault indicator for open circuit faults F5.

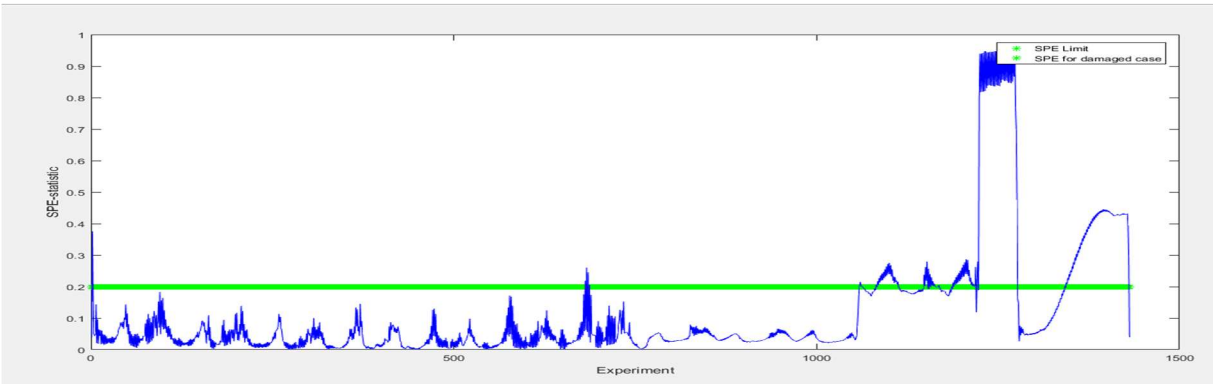


Figure 3.25: SPE fault indicator for 20% open circuit faults F5.

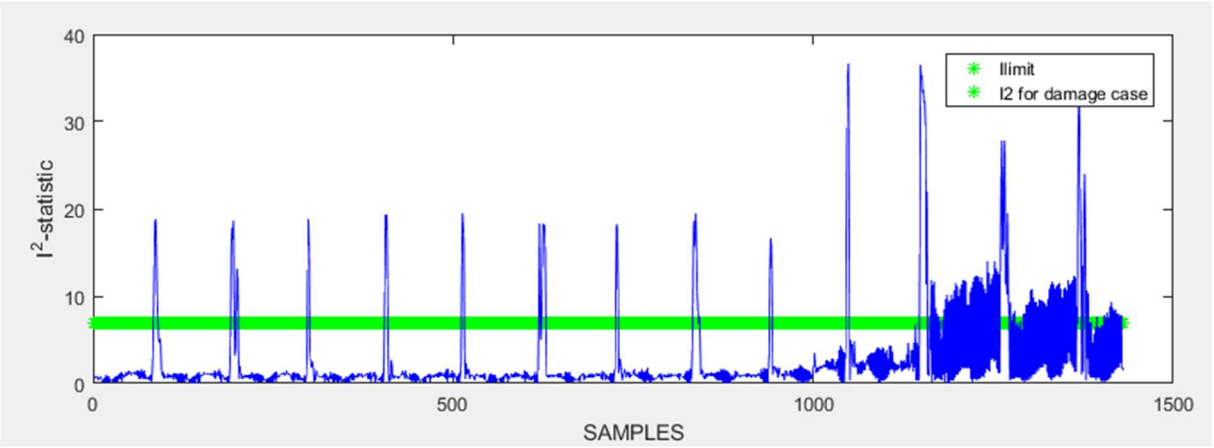


Figure 3.26: I² fault indicator for MPPT controller faults F6.

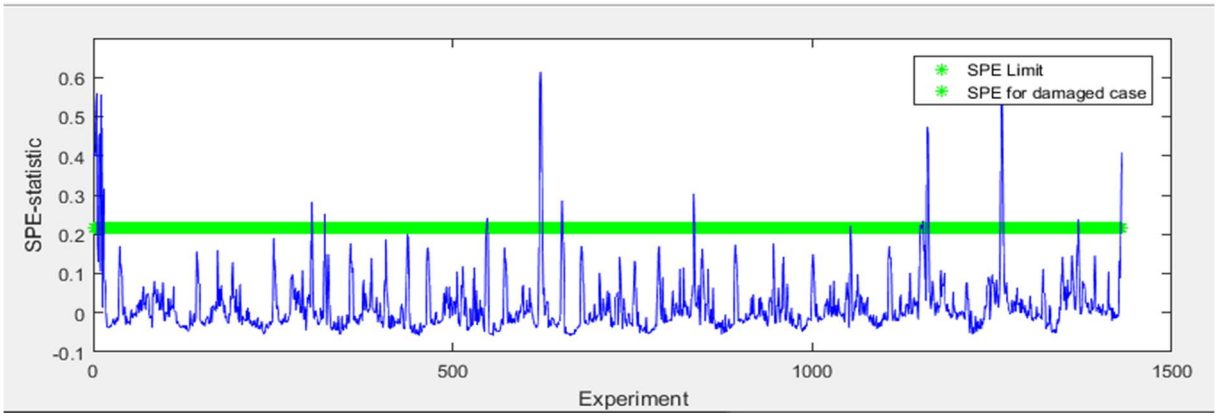


Figure 3.27: SPE fault indicator for MPPT controller faults F6.

3.4.4- Discussion

In this case we can clearly see an improvement in detecting F1 and F5 faults using the I² monitoring index, such that the detection lasted for the whole fault duration. Also, the delay detection time is noticeably small especially in F1, F2,F3 and F5.

As mentioned before, the SPE indices were calculated from the residuals of the projected data in feature linear space when using KPCA. SPE could detect the F1 and F5 faults effectively but we notice a considerable delay time and higher false alarms especially in the case of the controller fault F6.

The MPPT boost controller fault F6 was also successfully detected using KICA on the contrary to the PCA technique.

3.4.5- Method evaluation using FDR and FAR

To evaluate the efficiency of each algorithm, the following indices were used.

- Fault detection rate:

$$\text{FDR} = \frac{n}{N} \times 100. \tag{3.2}$$

Where N is the total number of the faulty samples within the detecting interval and n is the number of correctly detected samples.

- False alarm rate:

$$\text{FAR} = \frac{d}{D} \times 100. \tag{3.3}$$

Where d is the number incorrectly classified samples and D is the total number of normal samples within the interval of detection.

This is done for a confidence level of 95%.

- Detection time delay DD:

$$\text{DD} = \text{D}(Detection) - \text{D}(Occurance). \tag{3.4}$$

The FDR for the PCA model shows poor results for F1 and F4 faults when using the T^2 indicator. While there is no detection for F6 and F7. F5 and F3 had the highest detection rates.

The FDR for the KICA model showed highest results, F1, F3 and F5 were effectively detected with 100% efficiency. Similarly, the controller faults F6 and F7 showed quite the improvement as their FDR exceeded 70%.

The shading faults F4 did not show a high detection rate, but it still improved significantly from the PCA model.

FAR results were somehow low for F3 and F5 in both models with under 2%. Whereas, the highest FIR belongs to the shading fault F4 as it reached almost 8%. The rest of the faults also had a considerable amount of false alarms as their FIRs were between 4 and 7%.

Table 3.3: detected faults using PCA and KICA.

FD methods	Inverter Fault F1	Feedback sensor fault F2	Grid anomaly fault F3	Partial shading fault F4	Open circuit fault F5	MPPT controller fault F6	Boost controller fault F7.
PCA	✓	✓	✓	-	✓	-	-
KICA	✓	✓	✓	✓	✓	✓	✓

Table 3.4: Fault detection rates comparison between the used methods.

Method / faults	PCA			KICA		
	T2	SPE	DD	T2	SPE	DD
F1	37.3%	70%	0.02s	100%	84.26%	0s
F2	70.5%	80.5%	0.05s	74.5%	70%	0.02s
F3	100%	75.44%	0s	100%	80.1%	0s
F4	33.33%	33.78%	1s	71.6%	42.07%	1.5s
F5	84.37%	80.69%	0s	100%	97.81%	0s
F6	0%	0%	undetermined	71%	67.01%	0.7s
F7	0%	0%	undetermined	74.7%	55.8%	0.8s

Table 5: False alarms rates in the used methods within confidence level of 95%.

Method / faults	PCA		KICA	
	T2	SPE	I2	SPE
F1	7.20%	1.15%	3.04%	0.86%
F2	7.11%	3.58%	4.36%	6.82%
F3	0 %	1.25%	0%	1.2%
F4	1.350%	3.46%	7.8%	6.07%
F5	2.77%	3.07%	0%	0.7%
F6	100%	100%	3.6%	4.60%
F7	100%	100%	3.56%	6.48%

Although these techniques are relatively easy to implement and take little detection time, the problem of the high false detection rates is always present. Moreover, the issue regarding the kernel parameters estimation is uncertain and time consuming.

3.5- FD using SVM

To classify the different types of faults and improve the PCA and KICA techniques, an additional approach based on Machine learning tools was also implement. ML classifiers are well known for solving complex problems characterized by nonlinearity and high dimension. We still can benefit from the advantages of both PCA and KICA while solving the problems regarding these techniques. In this work, Multi class classification based on Support vector machines was used.

3.5.1- The one vs. one method

As mentioned in the previous chapter, to build a multiclass SVM classifier, one approach is to split the multi-class classification dataset into multiple binary classification ones and fit a binary classifier on each. For this purpose, the one-vs-one method is used. Based on conducted experiments, compared to the one-vs-all approach, the one-vs-one is more efficient to differentiate between the classes. The SVM in this case will build a binary classifier between every 2 classes. In total, we will have k*(k-1)/2 two class classifiers. To validate this model, 80% of the PCA reduced data was used for training the classifier leaving 20% for testing. The features selected were the 4 PCs, plus T2 and SPE for the PCA-SVM model and the 4 ICs for the KICA-SVM model.

The software package we used was the OSU SVM Classifier Matlab Toolbox, which is based on the software LIBSVM . On each dataset, we trained multi-class OVO SVM. The chosen kernel was the RBF kernel. The regularizing parameters C and σ were determined via 10 fold cross validation on the training set.

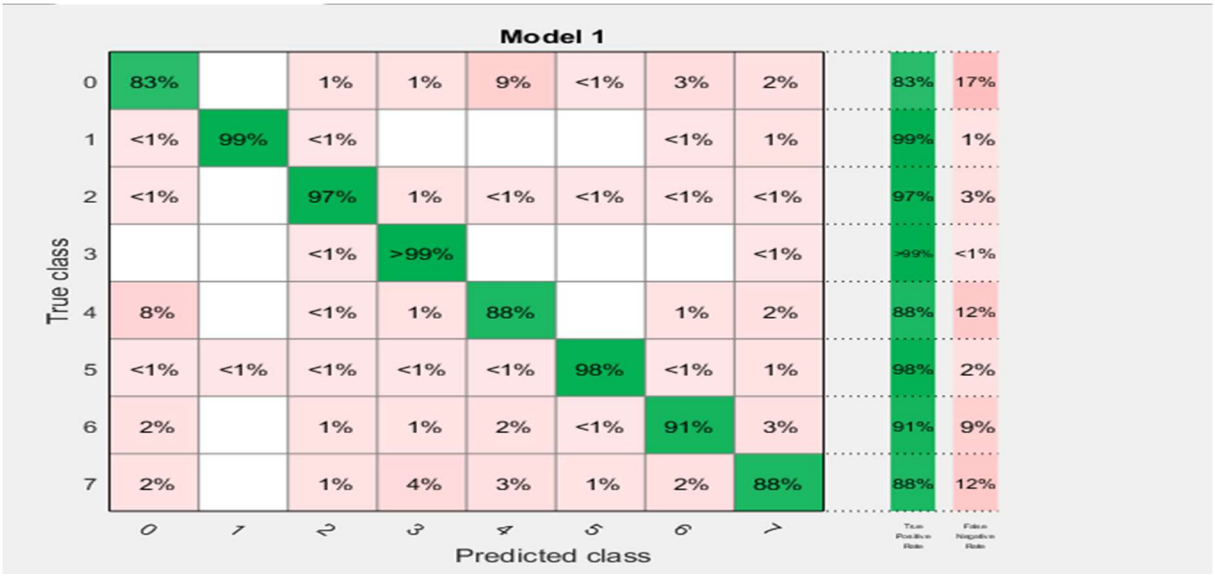


Figure 28: PCA-SVM model training results

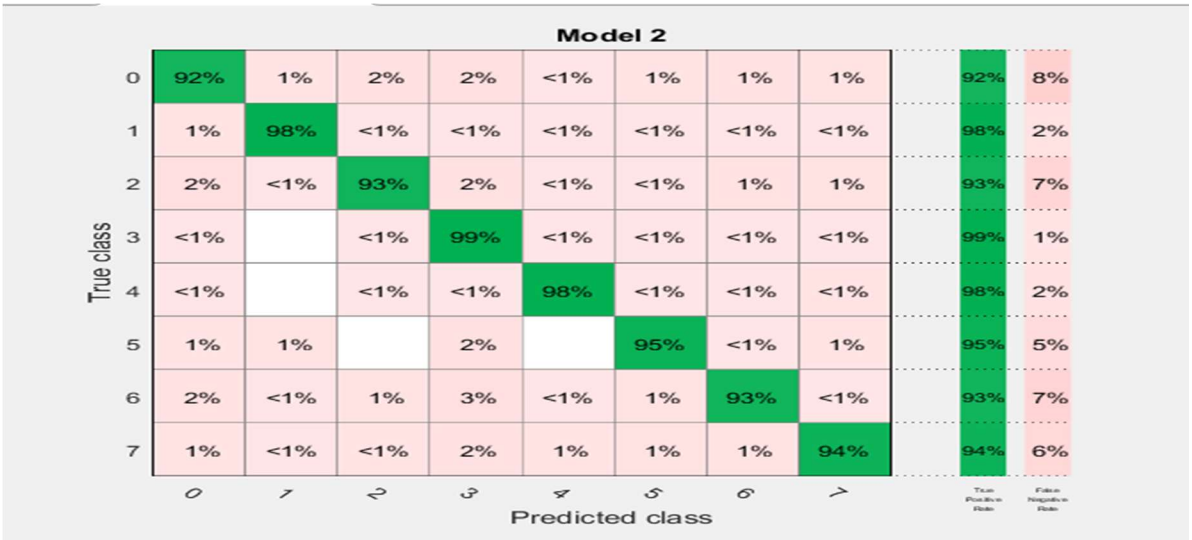


Figure 29: KICA-SVM model training results.

Table 3.6: fault detection accuracy for PCA-SVM and KICA-SVM models.

	PCA-SVM		KICA-SVM	
	Train	**Testing**	**Training**	**Testing**
Overall model accuracy	83 %	79.72%	92%	86%
Helthy case F0	96%	91%	96.6%	92%
F1	99 %	93.88%	98 %	95.04%
F2	97 %	92.84%	93 %	89.96%
F3	99 %	96%	99 %	97%
F4	88 %	83.19%	98 %	95.86%
F5	98 %	90.66%	95 %	91.73%
F6	91 %	82.03%	93 %	87.24%
F7	88 %	79.6%	94 %	80.7%

3.5.2- Discussion:

The confusion matrices above represents the accuracy of classification for each fault with both models. These results were successfully validated as the testing results show.

As expected the fault detection accuracy was the highest when detecting fault like F1, F3 and F5 as it reached over 90% in both models. Moreover, SVM could easily detect the controller's faults F6 and F7 with high accuracy.

The SVM model had shown good accuracy in detecting and classifying the different faults that occurred in the PV system. PCA based SVM had shown a significant improvement compared to the normal PCA threshold method. It was also able to detect faults that could not be detected with T2 and SPE indicators.

KICA based SVM was more effective in detecting faults in general, is the accuracy of the model reached 92%. We can say that this method has achieved the goal of this work in detecting GCPV system faults with high accuracy, especially the controller faults (F6,F7), which we could not detect using the classical PCA approach.

3.6- Conclusion:

In chis chapter PCA based SVM and KICA based SVM methods were investigated for fault detection and diagnosis of a grid connected PV system. This investigation aimed to detect as much faults as possible with assuring an optimal time detection and a lower false alarms rate. The evaluation criteria FDR and FAR has shown a significant superiority of the KICA method in detection faults, especially the controller faults that the PCA method had failed to diagnose.

The SVM model is then built for classifying the faults using the features extracted from PCA and KICA. Again, KICA based SVM were more efficient in diagnosing the different faults with high overall accuracy compared to PCA-SVM model.

GENERAL CONCLUSION AND FUTUR WORK

Although power generation through GCPV systems may become the main alternative for fuel based power systems, we cannot ignore the many challenges facing this technology. This process is greatly exposed to internal and external faults, depends on environmental parameters, not to mention that PV panels are quite costly.

In order to reduce the cost, improve efficiency and reliability in GCPV systems, many fault detection system can be implemented to prevent damages of the GCPV. Both Multi statistical process monitoring and ML techniques had shown good results detecting and identifying faults in different complex process.

In this work, KICA based threshold based on independent fault indicator I^2 and SPE is considered as an alternative for the classical PCA for fault detection. The approach is designed mainly to deal with nonlinear characteristic of the GCPV system and overcome the Gaussian assumptions that PCA technique rely on. The obtained ICs are then introduced as features for a one-vs-one SVM classifier for a more accurate fault diagnosis.

The evaluation of this approach demonstrated satisfactory results in term of fault detection rate and detection delay, presented good results for almost all the faults that this paper concerns with and successfully detected the faults injected at the controller, which PCA failed to detect. Although the robustness of these devices to external factors, I^2 fault indicator shows good sensitivity to faults. To classify the types of faults and overcome the false alarms, SVM multiclass classifier was introduced. Our model had shown even better accuracy for fault diagnosis even in compared to the PCA based SVM.

For future work, an adaptive fault detection technique based on KICA may be considered to prevent the considerable false alarm rate shown in some of the faulty scenarios. The shading fault specifically had proved quite a challenge as it showed the highest FAR, so developing an alternative approach for this kind of faults should be a priority.

References

[1] Lana El Chaar Ph.D., in Power Electronics Handbook (Third Edition), 2011

[2] Mark Feskin, John A. Dutton, e-Education Institute, Utility Solar Power and concentration

[3] Sudha Bansa, Design of a DC-DC Converter for Photovoltaic Solar systems, 2012

[4] Abul Kalam Ajad, Analysis of P&O and INC MPPT Techniques for PV Array Using MATLAB, 2016

[6] How does temperature and irradiance affect I-V curves?, Accessed: 20/03/2022, https://www.seaward.com/gb/support/solar/faqs/00797-how-does-temperature-and-irradiance-affect-i-v-curves/

[7] Mohammadreza Aghaei, Solar PV systems design and monitoring, 2020

[8] Photovoltaic system performance monitoring–Guidelines for measurement, data exchange and analysis. 1998

[9] Venkat Venkatasubramanian , Raghunathan Rengaswamy, Kewen Yin , Surya N. Kavuri, A review of process fault detection and diagnosis Part I: Quantitative model-based methods, 2002

[10] journals.sagepub.com,'Measurement and Control'2021. [Online]. Accesed :15/04/2022 https://journals.sagepub.com/doi/full/10.1177/0020294013510471

[11] Alexandros Mouzakitis**,** Classification of Fault Diagnosis Methods for Control Systems, 2013

[12] Dunia and Qin, Loss-of-Main Monitoring and Detection for Distributed Generations Using Dynamic Principal Component Analysis, 1998

[13] Amal Hichri, Mansour Hajji, Fault detection and diagnosis in grid connected photovoltaic systems, 2020

[14] Hua Han Zhikun Cao Bo Gu Neng Ren, PCA-SVM-Based Automated Fault Detection and Diagnosis (AFDD) for Vapor-Compression Refrigeration Systems, January 28, 2010

[15] A.Bakdi, Real-time fault detection in PV systems under MPPT using PMU and high frequency multi-sensor data through online PCA-KDE-based multivariate KL divergence, 2021

[16] W.Cho, kernel PCA for fault detection, 2005

[17] Bin Shams, Fault Identification using Kernel Principle Component Analysis, 2011

[18] M. Kallas, G. Mourot, D. Maquin, and J. Ragot, Fault estimation of nonlinear processes using kernel principal component analysis, 2014

[19] Mi F. Ren, Yan Liang, and Ming Y. Gong An Improved PCA-based Fault Detection Method for non-Gaussian Systems Using SIP Criterion, 2019

[20] Wook Choi, Elaine B. Martin,* A. Julian Morris, and In-Beum Lee, Adaptive Multivariate Statistical Process Control for Monitoring Time-Varying Processes Sang , 2006

[21] Julianna Delua, SME, IBM Analytics, Data Science/Machine Learning.

[22] Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms, 2019

[23] BART DE KETELAERE, MIA HUBERT, and ERIC SCHMITT, Overview of PCA-Based Statistical Process-Monitoring Methods for Time-Dependent, High-Dimensional Data ,2020

[24] *James H. Steiger* , Principal Components Analysis *February 16, 2015*

[25] D. Zumoffen and M. Basualdo, "From large chemical plant data to fault diagnosis integrated to decentralized fault tolerant control: pulp mill process application," Industrial & Engineering Chemistry Research, vol. 47, pp. 1201-1220, 2007.

[26] Tracy N.D., Young J.C., Mason, R.L."Multivariate Control Charts for Individual observations". Journal of Quality Technology, 24(2), 88-95.(1995)

[27] PENG PENG, YI ZHANG , FENG LIU, HONGWEI WANG, AND HEMING ZHANG1, A Robust and Sparse Process Fault Detection Method Based on RSPCA, 2002

[28] R.Raich and A.Cinar, a new combined static based fault detection based on Mahalanobis distance using SPE and T^2,2010

[29] A.Haykin, kernel mapping, 1999

[30] A.Mansouri, Kernel PCA- and Kernel PLS-based generalized likelihood ratio tests for fault detection, 2020.

[31] Sch Jolkopf,kernel PCA monitoring, 1998.

[32] Alberto Garc´ıa-Gonz´alez, Antonio Huerta, Sergio Zlotnik and Pedro D´ıez, A kernel Principal Component Analysis (kPCA) digest with a new backward mapping (pre-image reconstruction) strategy, 2021

[33] Jutten Hérault , Independent component analyses,2000

[34] Alaa Tharwat, Independent component analysis: An introduction, 2002

[35] Jicong Fan, Youqing Wang , Fault detection and diagnosis of non-linear non-Gaussian dynamic processes using kernel dynamic independent component analysis ,2018

[36] Yen-Chi Chen , Density Estimation: Histogram and Kernel Density Estimator,2016

[37] Junjie Wang, Dedong Gao, Shaokang Zhu, Shan Wang & Haixiong Liu, Fault diagnosis method of photovoltaic array based on support vector machine,2008

[38] Shen Yin, Xin Gao, Hamid Reza Karimi, and Xiangping Zhu, Study on Support Vector Machine-Based Fault Detection in Tennessee Eastman Process,2014

[40] Julianna Delua, Supervised vs. Unsupervised Learning: What's the Difference?, SME, IBM Analytics, Data Science/Machine Learning

[41] machine Junjie Wang, Dedong Gao, Shaokang Zhu, Shan Wang & Haixiong Liu, Fault diagnosis method of photovoltaic array,2010

[42] Fang Wu, Shen Yin, and Hamid Reza Karimi, Fault Detection and Diagnosis in Process Data Using Support Vector Machines,2013

APPENDIX A

Appendix A represents the PV system's measurements and how they change in each fault scenario. This appendix only demonstrated the most effected parameters in each case.

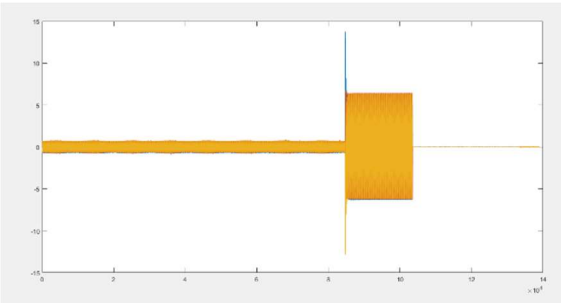


Fig.A.1 Grid current Iabc for inverter fault F1

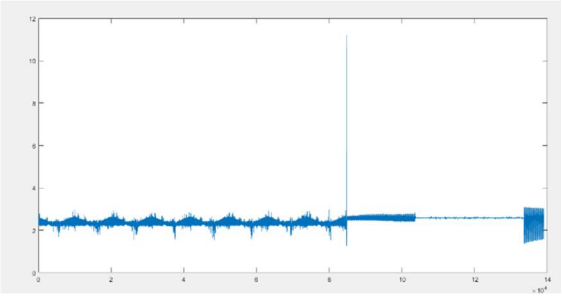


Fig.A.2 PV current IPV for inverter fault F1

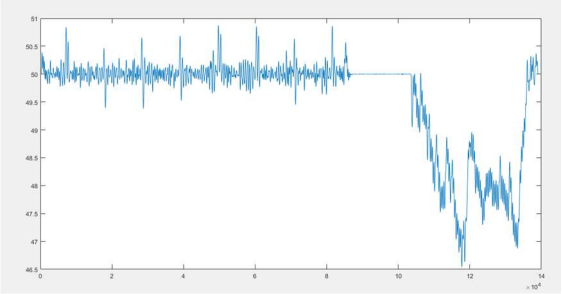


Fig.A.3 Grid frequency fi for inverter fault F1

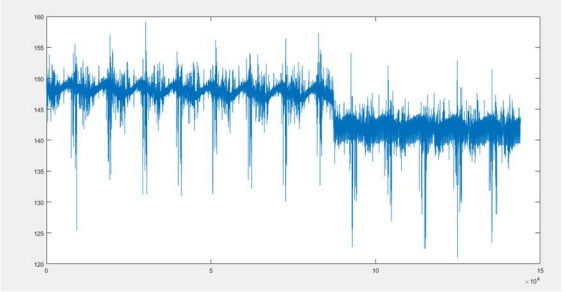


Fig.A.4 Dc voltage Vdc for sensor fault F2

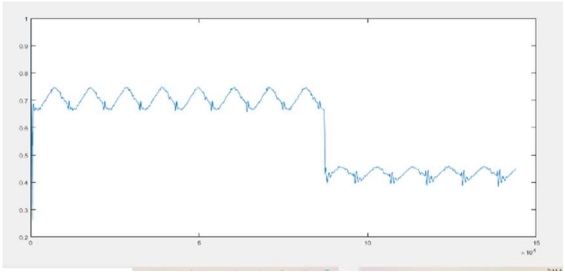


Fig.A.5 Rms Grid current |Iabc| for sensor fault F2

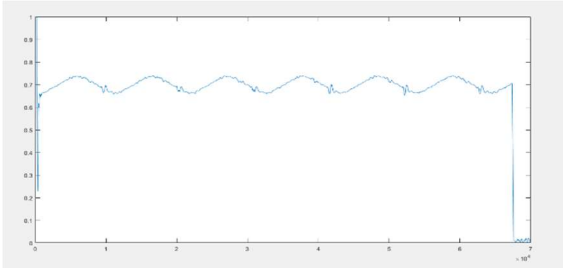


Fig.A.6 RMS Grid current |Iabc| for fault F3

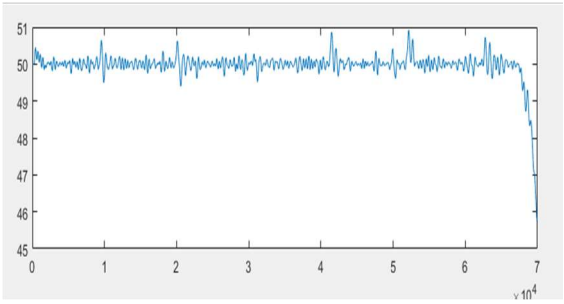


Fig.A.7 Grid frequency fi for fault F3

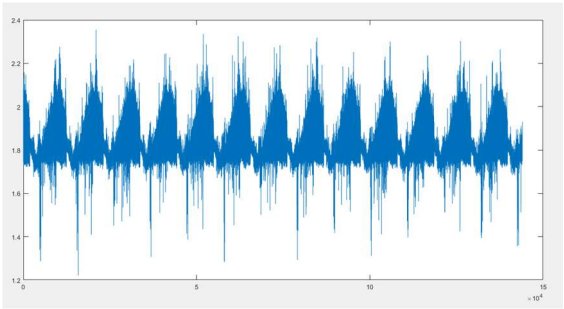


Fig.A.8 PV current Ipv for shading fault F4

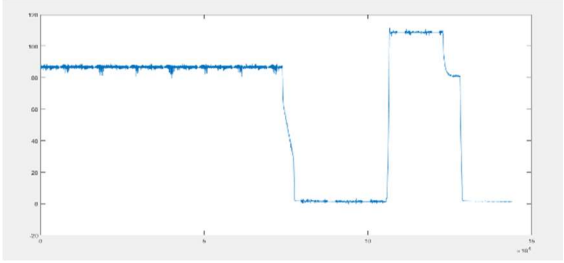


Fig.A.9 PV voltage Vpv for inverter fault F5

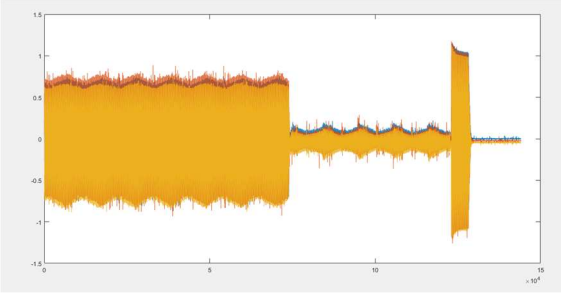
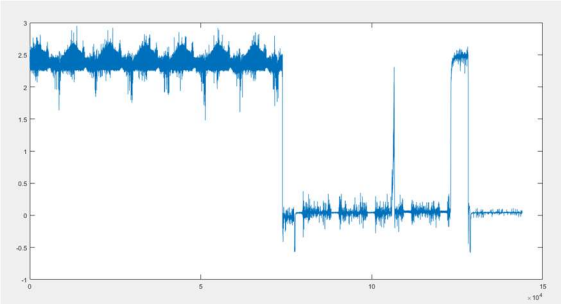


Fig.A.10 Grid current Iabc for open circuit fault F5



FigA.11 PV Grid current for open circuit fault F5

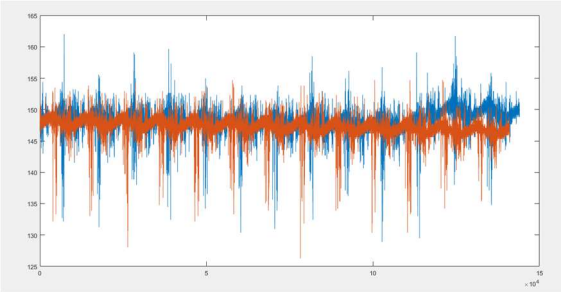


Fig.A.12 DC voltage vdc for inverter fault F6 vs normal case.