MASTER THESIS

Master's Degree in Automatic Control and Robotics

Fault detection in structures (wind turbine) through statistical techniques, singular spectrum analysis (SSA) and frequency-based methods.

Autor: Victor Patricio Verdezoto Pereira

Director: Francesc Pozo Montero

Convocatòria: October 2016

Escola Tècnica Superior

d'Enginyeria Industrial de Barcelona

Abstract

This master thesis aims to demonstrate the effectiveness of a fault detection strategy in the structure of the wind turbine through a new strategy involving singular spectrum analysis (SSA), statistical methods and methods based on frequency.

Based on a dynamic model of wind turbine, the time series behavior was obtained in the first instance without presenting any system failure and a second instance of system failure. From these series we can intervene with SSA and with statistical methods (variance, means, covariance, Fisher criteria) to design the fault detection system. The baseline will be designed with 850 healthy samples and a total sampling time of 6.25 seconds.
This baseline which will provide us with the components to be compared so that the system can detect various faults that occur (fault types: fixed value, gain factor, offset and dynamics changed) in an efficient way.

The results show that a sensor fault system with a high percentage of effectiveness was designed. The greater or lesser effectiveness thereof will depend on the established base line and the components used.

:

Summary

List of Figures

List of Tables

1. Introduction

This chapter introduces the project’s motivation, its objectives, the followed methodology, the problem that we are tackling and the simulation.

1.1. Project Motivation

Since several years ago, about 150, one of the biggest problems of mankind is the change in the natural structure of the atmosphere, in other words environmental pollution. One of the ways in which it seeks to mitigate this problem is by changing non-renewable energy sources by renewable energy sources.

Within all this context of renewable energy sources are wind turbines that are driven by the wind force. The wind power is the most common due to the maturity of its technology, infrastructures are well-known and its cost is competitive with other sources.
In a general framework the wind turbines have: a gondola, shell that protects its internal mechanism, and blades that rotate, according to technology, at a constant speed or variable speed, where the rotor speed varies with the wind speed to achieve greater efficiency. It is at this point sought efficiency of resources where we can intervene to ensure the normal operation of the turbines.

The turbines are currently monitored by specialized equipment. The challenge that is presented with the monitoring is to optimize resources in the maintenance of the equipment and maintain the quality of the product that will be delivered to the end user (energy). At present the data you provide to us the sensors placed in this type of turbines are diverse and in great quantity which enables us to work with the same.
From these data is to design a system of detection of failure that can indicate whether there has been a structural damage to the turbine, which will have a planned preventative maintenance of equipment already that usually sensors cannot detect this type of damage directly or when the operators of the equipment they realize it is too late and has to be executed directly actions of corrective maintenance. To be able to perform a planned preventative maintenance costs for repairs are reduced and the end user will be affected minimally or simply will not be affected while if you performed a corrective maintenance costs rise, the damages to the user are inevitable and the computers suffer greater wear.

On the other hand the techniques to implement SSA, statistical methods and methods based on frequency are relatively recent in the field of detection of faults. Although the SSA has been used before, has been especially used in the analysis of time series in the field weather. The method of SSA applied in the climatological field is attractive and has strength so great that may allow appropriate approximations in the analysis of systems, including non-linear. The statistical methods and analysis in frequency can act as an important complement to strengthen the implementation of the SSA.

The use of these novel methods in the investigation of failure detection opens a range of possibilities, where even the most negative results may be used in any useful conclusions from system.

1.2. Objective

The objective of this Master thesis is to try to develop system detector failures in the structure of the turbine by applying the techniques of singular spectrum analysis, statistical methods and frequency-based methods.

System data, used for the analysis will be taken of the simulations carried out by CoDAlab(control, dynamics and applications). Them provides with the data of the already simulated turbine, with all measured parameters.

It must be made clear that this work is not intended to design the simulation of the wind turbine or the analysis of the sensor module used.

1.3. Methodology

The methodology we will follow for the project is composed of 4 phases:

1. Documentation,

2. Design of the algorithm,

3. Simulation results,

4. Analysis of the results.

The documentation regards reviewing the state of the art in the failure detection in structures, SSA, statistical methods and analysis in frequency. Next the algorithm will be designed, identifying a way to detect a problem in the structure of the wind turbine. After the design, the algorithm will be tested in a simulated environment (Chapter 3). Finally the results will be analyzed in order to have some conclusions about the proposed system for the detection of faults in the structures.

1.4. Problem Approach

In this section the approach to solve the failure detection problem is described. The strategy for solving the problem begins with the identification of the data provided for the analysis. This identification is obtained the data without fault and with fails; each one of the failures is adopted in the simulation software and will be detailed later. Then divide the data without fault, in such a way that a part are intended to develop a base line that we serve as an indicator of a correct performance and another part of the same will be used in the tests of the detector system faults.

The base line is obtained by applying the methods already mentioned, it is part of the algorithm that was designed. From this point the algorithm is complemented looking for the best way to detect faults. Once completed the algorithm proceed to the next phase that are the tests.

In this section we will proceed to test the algorithm designed. The data is not used to develop the base line and data with fault shall be tested with the algorithm to check its efficiency.

In this last section we will proceed to define the success or otherwise of the algorithm designed as well as the causes that have led to this performance.

1.5. Contribution

The contributions of this thesis are:

- The application of a novel technique in the field of condition monitoring of wind turbines.
- The combination with statistical techniques to strengthen the performance of the systems.

2. Previous Works

In this chapter we present some previous works regarding the SSA, fault detection and wind turbines.

2.1. Singular Spectrum Analysis (SSA) definitions

The beginnings of SSA is usually associated with the publication of papers by Broomhead (e.g. Broomhead and King, 1986) while the ideas of SSA were independently developed in Russia (St. Petersburg, Moscow) and in several groups in the UK and USA. [1] SSA as a data analysis method has been used for years in digital signal processing. [2]

The areas of application of SSA are very different, from the economy through geology to reach mathematics or physical applications to name a few of the fields, where we can say that this method works in a better way as a procedure of analysis of temporary series.

SSA is in its most basic form a linear analysis and prediction method. It's superiority over classical spectral methods in the sense in which it can use your concepts in a successful way in the nonlinear dynamics. SSA can provide useful information physical and modest predictability in the medium term from a few hundred data points. [3] It was introduced into nonlinear dynamics by Broomhead and King. [4]

The aim of SSA is to make a decomposition of the original series into the sum of a small number of independent and interpretable components such as a slowly varying trend, oscillatory components and a structure less noise. SSA makes a pretty good noise reduction but if you want something specific must be clear that noise reduction is a signal processing problem. [5]
The SSA also describes the variability of a time series, in terms of the structure of auto covariance shifted in time.

At the present the SSA method is a very useful tool which can be used for solving the following problems: 1) finding trends of different resolution; 2) smoothing; 3) extraction of seasonality components; 4) simultaneous extraction of cycles with small and large periods; 5) extraction of periodicities with varying amplitudes; 6) simultaneous extraction of complex trends and periodicities; 7) finding structure in short time series; and 8) change-point detection.

In brief, systems that require time series analysis are those for which the equations that govern the physical system are unknown. Normally these kinds of systems are composed of interacting subsystems that feed into a complex plot.

Since it is clear that behind the analysis of a temporary record the dynamic behavior of the physical system hangs with respect to the variable under study, it is also true that the register is the result of all interactions between variables and therefore in principle you contains information regarding the dynamics of all important variables involved in the system evolution.

2.2. SSA, Wind Turbines and Fault Detection

Now we will mention the case studies that carried out the UPC with SSA, wind turbines and fault detection.

- First of all to mention the: A multivariate data analysis approach towards vibration analysis and vibration-based damage assessment: Application for delamination detection in a composite beam.
  This study trade to introduce a novel methodology for structural vibration analysis and vibration based monitoring which utilizes the SSA. Through two case studies the importance of this new methodology is demonstrated. The first study is a numerical demonstration of an example for a two degree-of-freedom (2DoF) and spring-mass damper system with nonlinear stiffness. The second study is an experiment where the method is based on the decomposition of the frequency domain structural variation response using new variables. [6]
  In this work used the multichannel singular spectrum analysis (MSSA). The MSSA is a natural extension of the SSA for multivariate systems. MSSA applies SSA onto several time series. In this case there is more than one time series.
  In this study only claim crude capacity was tested of SSA method for the location and to estimate damage.[6]
  This work proved the first capabilities of SSA method for location and estimating damage. This is a pioneering study which introduces the application of SSA and MSSA in the frequency domain for damage assessment in structures.

- Now we mention: Wind turbine fault detection through principal component analysis and statistical hypothesis testing.
  This project addresses the problem of online fault detection of an advanced wind turbine benchmark under actuators (pitch and torque) and sensors (pitch angle measurement) faults of different type: fixed value, gain factor, offset and changed dynamics. [7]

A comprehensive statistical analysis is performed for structural health monitoring. The analysis starts by obtaining the baseline principal component analysis (PCA) model and projections using measurements from the healthy or undamaged structure. PCA is used in this framework as a way to compress and extract information from the sensor data stored for the structure which summarizes most of the variance in a few (new) variables into the baseline model space. These new variables are used for comparison (hypothesis) with each new experiment and so to determine whether these data belong to healthy or damaged structure. [7]

The goal is to obtain a fault detection method such that when the distribution of the current sample is related to the distribution of the baseline sample a healthy state is predicted and otherwise a fault is detected. For this reason statistical hypothesis testing is used and consists in two hypotheses: the null hypothesis is “the sample of the wind turbine to be diagnosed is distributed as the baseline sample” and the alternative hypothesis is “the sample of the wind turbine to be diagnosed is not distributed as the baseline sample”. In other words, if the result of the test is that the null hypothesis is not rejected, the current wind turbine is categorized as healthy. Otherwise, if the null hypothesis is rejected in favor of the alternative, this would indicate the presence of some faults in the wind turbine[7].

The contribution of this work is that the projection on the first component is not always the best option to detect and distinguish damage. The problem is that the first component (model PCA) captures the maximal variance of the data. However, when new data are projected into this model, there is no longer a guarantee of the existence of maximal variance in these new data.

- Another important project development is: Detection of structural changes through principal component analysis and multivariate statistical inference.
  The characterize of this work are three parts: a) The nature of the data, vectors of principal component analysis projections are used instead of all data structure. b) The size of the data (two random samples). c) The samples come from a multidimensional variable [8].

  In the experiments, four piezoelectric sensors discs were attached to the surface of a thin aluminum plate. As a response to an electrical excitation, sensors produce a mechanical vibration, propagating, in this case, across the plate. 1000 samples were analyzing. 500 experiments were performed over the healthy structure, and another 500 experiments were performed over the damaged structure with five damage types. The framework of multivariate statistical inference is used with the objective of the classification of structures in healthy or damaged.[8]

The main contribution of this work is that the classification of healthy or damaged structures is achieved by multivariate statistical inference. One of the most important contributions is that regardless of all tests fail univariate, multivariate statistical inference is able to make a wise decision and improve the performance of fault detection.

3. Wind Turbine

3.1. Definition

A wind turbine is a mechanical device that converts wind energy into electricity. Wind turbines designed to convert energy from wind motion (kinetic energy) into mechanical energy, movement of an axis. Then in the turbine generator, this mechanical energy is converted into electricity. The electricity generated can be stored in batteries or used directly.[21]

3.2. Characteristics

There are two main wind turbines types: horizontal and vertical (axis). In the project the turbine used is a horizontal one, as they are the most common large wind turbines; the summary of the basic functional parts that is done in this section is related to this kind of turbines.

Reference turbine used in the simulations (FAST) has the following characteristics:

- Rated power: 5 MW
- Numbers of blades: 3
- Rotor diameter: 126 m
- Hub height: 90 m
- Cut-in wind speed: 3 m/s
- Rate wind speed: 11.4 m/s
- Cut-out wind speed: 25 m/s
- Rated gen. speed: 1173.7 rpm
- Gearbox ratio: 97

Horizontal-axis wind turbines can have a reduced number of blades (two or three) when their purpose is the generation of electricity, or a big number of blades when they are used to do mechanical work.

3.3. Components

The main components of a vertical axis turbine are:

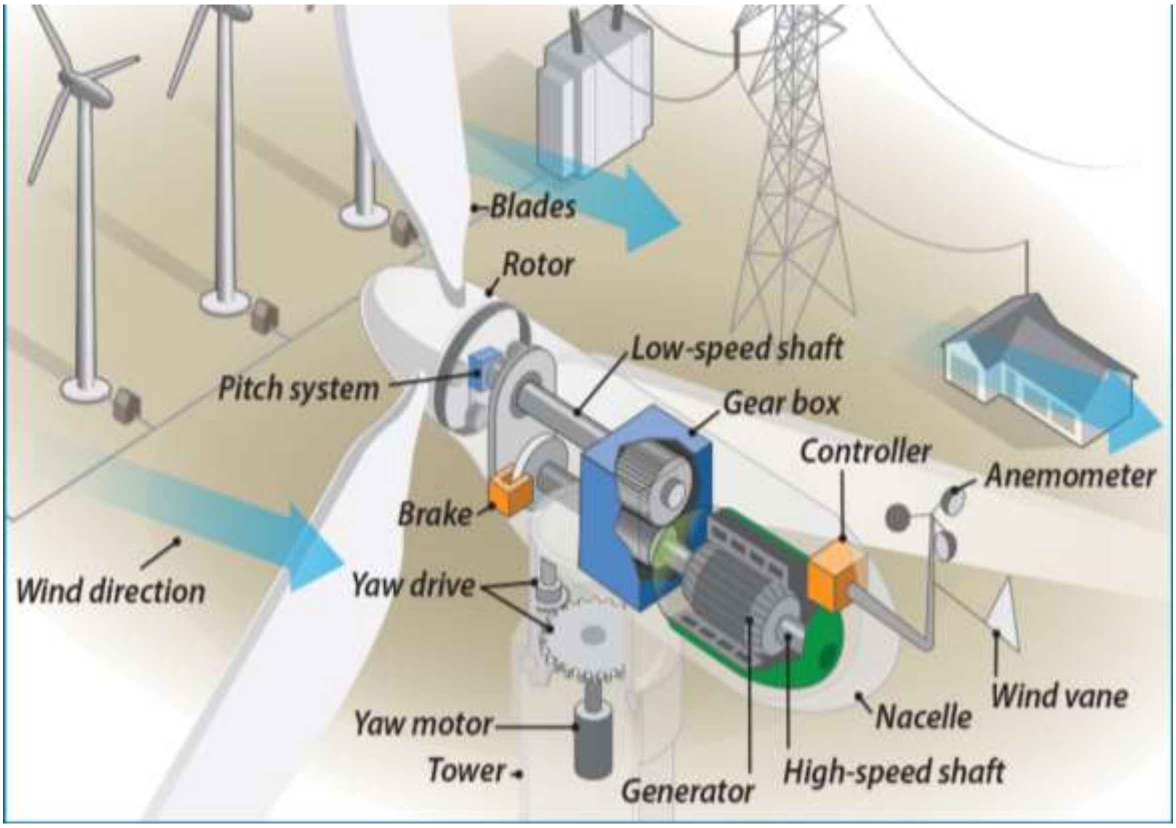


Figure1.1: Parts of a wind turbine

(Source: Office of Energy efficiency and renewable energy[24])

- **Foundation:** Consist of an underground reinforced concrete foundation, suitable to the terrain and wind loads, on which a tower rises. [9]
- **Tower:** The structural element that supports the entire weight of the wind turbine and maintains high ground the turbine blades. It is usually made of steel and hollow inside to allow access to the nacelle. This is usually typically tubular steel or reinforced concrete (now typically used composite structures in which the bottom is made of concrete and steel top). Raise the turbine enough to be able to access higher wind speeds, in contrast to the low speeds close to terrain points and the existence of turbulences. At the end of the tower a rotating nacelle steel or fiberglass is fixed.
- Rotor and Blades: Modern turbines usually consist of two or three blades, normal being the use of three by softness in turn it provides. The blades are made of a

polymer matrix composite material (polyester) with a fiber reinforcement glass or carbon to toughen. They can measure lengths in the range from 1 meter to 100 meters and are connected to the rotor hub. [10]

The rotor is that converts the kinetic energy of the wind into mechanical energy which is used to drive in the power generator. It consists of vanes or blades (blades), the hub (hub) where the blades are assembled, and the nose, which is the front end in cone shape, and which is used to avoid turbulence in the center of the rotor.[22]

- **Mechanical transmission system:** It is the main axis or low speed shaft compound, the gearbox and the high speed shaft. The main axis is transmitting the aerodynamic rotor torque generator system. The gearbox (gear box) is what makes the rotor speed is low, at a high speed for a conventional generator can produce electricity. The high speed shaft which delivers mechanical power to the generator directly.[22]
- **Electrical Generator:** The responsible for converting mechanical energy into electrical energy. In the SCEE they have been used both asynchronous and synchronous generators. [11]
- Guidance system: The guidance system is generally composed of a servo-mechanism that rotates the nacelle in the direction of the wind sensed by a weathervane.
- Control system: It consists of sensors, actuators and a main controller that has different functions: power control, speed control, voltage control, start and stop the machine direction of the turbine control other variables such as temperature and vibration.[23]
- Security system: The security system generally has the function to take the wind turbine to a secure and stable, for people and for the same equipment condition. It comprises braking systems, detection systems high temperatures, pressures and vibrations.
- Nacelle: The capsule or enclosure that protects the generator, transmission systems and orientation and to other components. It attaches to the tower and the rotor.
- Tower: The support of the nacelle and rotor design is robust to withstand all the dynamics of the wind turbine.

3.4. Data to sense in the wind turbine

Our data are composed of a series of experiments. Each experiment has a duration of 600 (s) and the sampling rate is 0.0125 (s).

3.4.1. Healthy Scenarios (sensed variables)

The variables taken into account at the simulation will be in the table1:

Number Sensor / Symbol	Sensor Type	Units
1 / Pe	Generated electrical power	kW
2 / wr	Rotor speed	rad/s
3 / wg	Generator speed	rad/s
4 / Tc	Generator torque	Nm
5 / u	Wind speed	rpm
6 / B_1	First pitch angle	deg
7 / B_2	Second pitch angle	deg
8 / B_3	Third pitch angle	deg
9 / ax_1	fore-aft acceleration at tower bottom	m/s^2
10 / ay_1	side-to-side acceleration at tower bottom	m/s^2
11 / ax_2	fore-aft acceleration at mid-tower	m/s^2
12 / ay_2	side-to-side acceleration at mid-tower	m/s^2
13 / ax_3	fore-aft acceleration at tower top	m/s^2
14 / ay_3	side-to-side acceleration at tower top	m/s^2

Table1. Available measurements

3.4.2. Fault Scenarios

Damage scenarios were represented by the failures in table 2:

Fault	Type	Description
F1	Pitch actuator	High air content in oil
F2	Pitch actuator	Pump wear
F3	Pitch actuator	Hydraulic leakage
F4	Generator speed sensor	Gain factor(Wg*1.2)
F5	Pitch angle sensor	Stuck(fixed value = 5 deg)
F6	Pitch angle sensor	Stuck(fixed value = 10 deg)
F7	Pitch angle sensor	Scaling(gain factor = 1.2)
F8	Torque actuator	Offset(value = 2000 Nm)

Table2. Fault scenarios

ETSEIB

3.4.2.1. Failure scenario 1, 2 and 3

The fluid power subsystem has lower failure rates and better capability of handling extreme loads than the electrical systems. Therefore, fluid power pitch systems are preferred on multi-MW size and offshore turbines. The most common problems such as leaks, contamination and electrical failures occurs which causes abnormal behavior of the entire system.

The first stage fault occurs due to high air content in the oil, resulting in a change in the system dynamics. Foaming is the most severe form of air pollution in a lubrication system. This usually occurs when the fluid surface tension is too high to allow air bubbles break after form and rise to the surface of the fluid. The foam reduces the effectiveness of the lubricant, resulting in accelerated wear, overheating and cleaning problems in severe cases.

The second stage of failure occurs when a high drop in pressure in the hydraulic supply system by wear and tear of the pump occurs. This wear of the pump causes a change in the dynamics of the system. . As this wear is irreversible, the only possibility to fix it is to replace the pump, which will happen after pump wear reaches certain level.

The third scenario fails, it happens to have a leak in the hydraulic system. Leakage of pitch cylinders can be internal or external and if this failure is not solved in time the system may collapse. The leak in the hydraulic system causing a pressure drop that affects the dynamic behavior of the system.

3.4.2.2. Failure scenario 4 (gain factor in Wg)

The generator speed measurement is done using encoders. The gain factor fault is introduced when the encoder reads more marks on the rotating part than actually present, which can happen as a result of dirt or other false markings on the rotating part.

3.4.2.3. Failure scenario 5, 6 and 7

The faults in the pitch position are the most important failures found in the actual systems. The origin of these faults is electrical or mechanical and it can result in fixed value as failures 5 and 6. This kind of failures can also cause the change in the gain factor as in the fault scenario 7.

The importance of detecting such failures is that the pitch controller is based on these measures for proper operation of the system.

3.4.2.4. Failure scenario 8

In this last scenario a converter torque offset fault is considered. This fault is possible detect because the change in this actuator cause that the torque balance in the wind turbine power train.

4. Methodology

The methodology used is detailed in the next steps together with the main ideas.

4.1. Singular Spectrum Analysis (SSA)

SSA is the decomposition of a time series into oscillatory components and noise. SSA tries to catch the underlying regular dynamical behavior which is the signature of the dynamical system.

We detail below the steps to implement the SSA:

- The time series stored in the vector X:

$$X=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{20} \end{bmatrix} \qquad \text{(Ec. 4.1)}$$

  The time series data are 20, then N =20. N is the number of elements in X.
  If we assume as a condition that this series of sample data was mean = 0 and standard deviation = 1. It is no necessary to normalize.

- Obtain the covariance matrix C. It is necessary compute the covariance between the values X(t) and X(t+K), where k is a delay.
  To obtain the covariance matrix we need create a new matrix Y that contains the original time series in the first column, a lag-1 shifted version of that time series in the second column, etc. We choose the windows size M, which represent the lags to consider for the matrix Y

  $M = 4 \rightarrow k = (0,1,2,3)$ (Ec.4.2)

$$Y=\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_3 & x_4 & x_5 \\ x_3 & x_4 & x_5 & x_6 \\ \vdots & \vdots & \vdots & \vdots \\ x_{19} & x_{20} & 0 & 0 \\ x_{20} & 0 & 0 & 0 \end{bmatrix} \qquad \text{(Ec.4.3)}$$

- Compute the covariance matrix C

$$C = \frac{Y^T Y}{N} \text{ (mean = 0 and variance = 1)} \quad \text{(Ec.4.4)}$$

$$C = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 \\ C_2 & C_5 & C_6 & C_7 \\ C_3 & C_6 & C_8 & C_9 \\ C_4 & C_7 & C_9 & C_{10} \end{bmatrix}$$

The diagonal of the matrix C contains the variance of each column.

- Compute the eigenvalues (λ) and eigenvectors (P) of the matrix C.

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} \qquad P = \begin{bmatrix} \rho_1 & \rho_5 & \rho_9 & \rho_{13} \\ \rho_2 & \rho_6 & \rho_{10} & \rho_{14} \\ \rho_3 & \rho_7 & \rho_{11} & \rho_{15} \\ \rho_4 & \rho_8 & \rho_{12} & \rho_{16} \end{bmatrix}$$

The columns of the matrix P are the eigenvectors. The eigenvalue in the first row; corresponds to the eigenvector is in the first column of P. The second eigenvalue is in the second row of λ and its corresponding eigenvector in the second column, and so on.

- Principal components

The eigenvectors of matrix C can be used to construct the principal components of the time series.

$$PC = YP \quad \text{(Ec.4.5)}$$

The equation 4.5 represents the matrix Y (embedded time series) projected onto the eigenvectors.

$$PC = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \delta_{1,3} & \delta_{1,4} \\ \delta_{2,1} & \delta_{2,2} & \delta_{2,3} & \delta_{2,4} \\ \delta_{3,1} & \delta_{3,2} & \delta_{3,3} & \delta_{3,4} \\ \vdots & \vdots & \vdots & \vdots \\ \delta_{20,1} & 0 & 0 & 0 \end{bmatrix}$$

The four columns of the matrix PC are the principal components PC1, PC2, PC3 and PC4. They are ordered in the same way as the eigenvectors are ordered in the matrix ρ ; so the 1st column is PC1, the 2nd column is PC2, etc.

4.2. Multivariable Spectrum Analysis (MSSA)

To explain the MSSA we follow the same steps as in the case of SSA pointing out the differences.

- The 2 dimensional time series stored in the vector X:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \\ \vdots & \vdots \\ x_{20,1} & x_{20,2} \end{bmatrix} \qquad \text{(Ec. 4.6)}$$

The time series data are 20 in each series, then N =20 (number of the elements in columns of X).

- In MSSA is necessary and obligatory normalizing the time series.

$$X_N = \begin{bmatrix} x_{N(1,1)} & x_{N(1,2)} \\ x_{N(2,1)} & x_{N(2,2)} \\ x_{N(3,1)} & x_{N(3,2)} \\ \vdots & \vdots \\ x_{N(20,1)} & x_{N(20,2)} \end{bmatrix} \qquad \text{(Ec 4.7)}$$

Normalize means to remove the mean value and to divide it by the standard deviation for each series.

- Compute the covariance matrix C.
To obtain the covariance matrix we need create a new matrix Y that contains the time delayed versions of the first and the second time series. We choose the windows size M, which represent the lags to consider for the matrix Y for each series.

M = 4 → k = (0,1,2,3) (Ec.4.8)

$$Y=\begin{bmatrix} \overset{1}{x}_{N(1,1)} & \overset{1}{x}_{N(2,1)} & \overset{1}{x}_{N(3,1)} & \overset{1}{x}_{N(4,1)} & \overset{2}{x}_{N(1,1)} & \overset{2}{x}_{N(2,1)} & \overset{2}{x}_{N(3,1)} & \overset{2}{x}_{N(4,1)} \\ \overset{1}{x}_{N(2,1)} & \overset{1}{x}_{N(3,1)} & \overset{1}{x}_{N(4,1)} & \overset{1}{x}_{N(5,1)} & \overset{2}{x}_{N(2,1)} & \overset{2}{x}_{N(3,1)} & \overset{2}{x}_{N(4,1)} & \overset{2}{x}_{N(5,1)} \\ \overset{1}{x}_{N(3,1)} & \overset{1}{x}_{N(4,1)} & \overset{1}{x}_{N(5,1)} & 0 & \overset{2}{x}_{N(3,1)} & \overset{2}{x}_{N(4,1)} & \overset{2}{x}_{N(5,1)} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overset{1}{x}_{N(20,1)} & 0_{N(20,1)} & 0_{N(20,1)} & 0_{N(20,1)} & \overset{2}{x}_{N(20,1)} & 0_{N(20,1)} & 0_{N(20,1)} & 0_{N(20,1)} \end{bmatrix} \qquad \text{(Ec.4.9)}$$

The firs 4 column s of Y represent the time delayed embedding of the first time series, the second four columns represent the time delayed embedding of the second column.

- Compute the covariance matrix C

$$C=\frac{Y^T Y}{N} \qquad \text{(Ec.4.10)}$$

$$C=\begin{bmatrix} C_1 & C_5 & C_9 & C_{13} & C_1^2 & C_5^2 & C_9^2 & C_{13}^2 \\ C_2 & C_6 & C_{10} & C_{14} & C_2^2 & C_6^2 & C_{10}^2 & C_{14}^2 \\ C_3 & C_7 & C_{11} & C_{15} & C_3^2 & C_7^2 & C_{11}^2 & C_{15}^2 \\ C_4 & C_8 & C_{12} & C_{16} & C_4^2 & C_8^2 & C_{12}^2 & C_{16}^2 \end{bmatrix}$$

- Compute the eigenvalues (λ) and eigenvectors (P) of the matrix C.

$$\lambda=\begin{bmatrix} \lambda_1^1 \\ \lambda_2^1 \\ \lambda_3^1 \\ \lambda_4^1 \\ \lambda_1^2 \\ \lambda_2^2 \\ \lambda_3^2 \\ \lambda_4^2 \end{bmatrix}$$

$$P=\begin{bmatrix} \rho_1^1 & \rho_5^1 & \rho_9^1 & \rho_{13}^1 & \rho_1^2 & \rho_5^2 & \rho_9^2 & \rho_{13}^2 \\ \rho_2^1 & \rho_6^1 & \rho_{10}^1 & \rho_{14}^1 & \rho_2^2 & \rho_6^2 & \rho_{10}^2 & \rho_{14}^2 \\ \rho_3^1 & \rho_7^1 & \rho_{11}^1 & \rho_{15}^1 & \rho_3^2 & \rho_7^2 & \rho_{11}^2 & \rho_{15}^2 \\ \rho_4^1 & \rho_8^1 & \rho_{12}^1 & \rho_{16}^1 & \rho_4^2 & \rho_8^2 & \rho_{12}^2 & \rho_{16}^2 \end{bmatrix}$$

The columns of the matrix λ contain the eigenvalues, four for each series. The columns of P contain the eigenvectors; the first 4 columns (components) correspond to the first time series and the second 4 columns (components).

- Principal components

  The eigenvectors of matrix C can use to construct the principal components of the time series. Each series with their components.

$$PC = YP \tag{Ec.4.11}$$

  The equation 4.11 represents the matrix Y (embedded time series) projected onto the eigenvectors.

$$PC = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \delta_{1,3} & \delta_{1,4} & \delta_{1,5} & \delta_{1,6} & \delta_{1,7} & \delta_{1,8} \\ \delta_{2,1} & \delta_{2,2} & \delta_{2,3} & \delta_{2,4} & \delta_{2,5} & \delta_{2,6} & \delta_{2,7} & \delta_{2,8} \\ \delta_{3,1} & \delta_{3,2} & \delta_{3,3} & \delta_{3,4} & \delta_{3,5} & \delta_{3,6} & \delta_{3,7} & \delta_{3,8} \\ \delta_{20,1} & \delta_{20,2} & \delta_{20,3} & \delta_{20,4} & \delta_{20,5} & \delta_{20,6} & \delta_{20,7} & \delta_{20,8} \end{bmatrix}$$

  Each PC contains characteristics of both time series. Unlike the EOFs or the matrices Y and C.

4.3. Statistical methods

Fisher defined separation between two distributions the proportion of the variance between classes, between the variance within the classes. This index will be useful because it will allow us in a final step to distinguish healthy samples of faulty samples.[19]

We review the basic concepts of mean, variance and covariance that are part of Fisher criteria. The equation will be explained in Chapter 5.

The arithmetic average is the statistic that provides a measure of location or position of a set of numbers. This arithmetical average is called mean [13].

The covariance between the two variables can be considered as the average product of deviations from each variable's respective mean [20].

Additionally don't forget that the matrices containing data time series have valuable information to work with them. The information can be obtained by calculating summary numbers, called descriptive statistics.[20]

These different statistical techniques allow us to improvise the performance of detector system failures.

5. Implementation

A change in physical properties due to structural changes or damage will cause detectable changes in dynamical responses.

The steps performed in the algorithm for detecting faults in the structure of the wind turbine are presented.

5.1. Data files

The system data, as mentioned above, come from the research group CoDAlab(control, dynamics and applications) of the UPC. Each file is a table with data from 14 sensors collected for 600 seconds. The sampling time is 0.0125 seconds.

Used 13 of the 14 data collected by the sensors. The data of wind speed is definitely something random so no we will consider in the analysis of the system.

Should be load the healthy samples in a matrix X (50 samples) and the faulty samples in a matrix Y (8 samples). These two matrices with the original data are structured as follows.

	X = SENSORS DATA											
	S1				S2......................................				........................S13			
	1	**2**	....	**500**	**501**	**502**	....	**1000**	....	....	....	**6500**
1	x(1)	x(2)	....	x(500)	x(1)	x(2)	....	x(500)	x(1)	x(2)	....	x(500)
2	x(2)											
3	x(3)											
:	:	healthy sample										
16	x(16)											
17	x(1)											
18	x(2)											
19	x(3)											
:	:	healthy sample										
32	x(16)											
:	:											
:	x(1)											
:	x(2)											
:	x(3)	healthy sample										
:	:											
850	x(16)											

Fig.5.1: Structure matrix X (sensor data)

	Y = DATA SENSORS(fault)												
	S1				S2..				........................S13				
	1	2	....	500	501	502	....	1000	....	....	....	6500
1	x(1)	x(2)	....	x(500)	x(1)	x(2)	....	x(500)	x(1)	x(2)	....	x(500)
2	x(2)											
3	x(3)											
:	:	faulty sample										
50	x(50)											
51	x(1)											
52	x(2)											
53	x(3)											
:	:	faulty sample										
100	x(50)											
:	:											
:	x(1)											
:	x(2)											
:	x(3)	faulty sample										
:	:											
400	x(50)											

Fig.5.2: Structure matrix Y (sensor data)

5.2. Data Normalization

The first is to calculate the standard deviation of the samples from the different sensors, for purposes of the calculations keep even the differentiation between healthy samples and samples with failure. After obtaining the standard deviation, the average is calculated for each of the moments that makes the time series (total sampling time 6.25 seconds) and a sampling time of 0.0125 secs. These are the variables necessary to normalize the data series.

A healthy original data we subtract the average value and divide for the standard deviation which is obtained with the resulting data are standardized. The normalized data are stored in a new variable called XT keeps the structure shown in Figure 5.1.

A faulty original data we subtract the mean value and divided for standard deviation to thereby obtain the resulting data are standardized. The normalized data are stored in a new variable called YT maintaining the structure shown in Figure 5.2.

5.3. Covariance Matrix (baseline)

The main covariance matrix is calculated, i.e. seeks to establish a baseline behavior which use only healthy data.
We started by calculating the mean of all experiments at each instant of time. These new data we store in a variable named XT_aux having the following structure:

	XT_aux = Compacted Data											
	S1	S2	S3	S4	S5	S6	S7	S8	S9	....	....	S13
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	....	....	**13**
1	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)	x(9)	....	....	x(13)
2	x(2)											
3	x(3)											
4	x(4)											
5	x(1)											
6	x(2)											
7	x(3)											
8	x(4)	healthy sample										
9	x(1)											
10	x(2)											
:	:											
:	:											
:	:											
:	:											
:	:											
500	x(500)											

Fig.5.3: Structure matrix XT_aux (compacted data).

Then you must select the new length of the data series and the number of delays with which we worked.

- Length of the series → N_cov = 50
- Delays → M_lag = 10

Organized delays matrix for each of the series of different sensors and store the data in the Mat_lag variable. The delays matrix structure show below.

	Mat_lag = Delayed Matrix											
	S1				S2				....	S13		
	1	**2**	....	**10**	**11**	**12**	....	**20**	....	**121**	....	**130**
1	x(1)	x(t-1)	....	x(t-9)	x(1)	x(t-1)	....	x(t-9)	....	x(1)	....	x(t-9)
2	x(2)											
3	x(3)											
4	x(4)											
5	x(5)											
6	x(6)											
7	x(7)											
8	x(8)	healthy sample										
9	x(9)											
10	x(10)											
:	:											
:	:											
:	:											
:	:											
:	:											
500	x(500)											

Fig.5.4: Structure matrix Mat_lag (delayed matrix).

Applying the formula of equation 4.4 we obtain the matrix C.

$$C = \frac{Mat\_lag^{T} Mat\_lag}{L}$$

The structure of matrix C shown in figure 5.5:

	C = Covariances Matrix											
	C1				C2				....	C13		
	1	**2**	....	**10**	**11**	**12**	....	**20**	....	**121**	....	**130**
1	x(1)	x(2)	....	x(10)	x(1)	x(2)	....	x(10)	....	x(1)	....	x(10)
2	x(2)											
3	x(3)											
4	x(4)											
5	x(5)											
6	x(6)											
7	x(7)											
8	x(8)											
9	x(9)											
10	x(10)											
:	:											
:	:											
:	:											
:	:											
:	:											
130	x(130)											

Fig.5.5: Structure matrix C(covariance matrix).

5.4. Eigenvalues and eigenvectors

Now we obtain the eigenvalues and eigenvectors of the covariance matrix. These eigenvalues and eigenvectors is obtain with the commando eig (C) of MATLAB.

The eigenvectors obtained were stored in the V_ssa variable, while the eigenvalues obtained were stored in the D_ssa variable. The structures of these new variables are presented below:

	D_ssa = eigenvalues											
	1	**2**	....	**10**	**11**	**12**	....	**20**	....	**121**	....	**130**
1	x(1)	0	0	0	0	0	0	0	0	0	0	0
2	0	x(2)	0	0	0	0	0	0	0	0	0	0
3	0	0	x(3)	0	0	0	0	0	0	0	0	0
4	0	0	0	....	0	0	0	0	0	0	0	0
5	0	0	0	0	....	0	0	0	0	0	0	0
6	0	0	0	0	0	....	0	0	0	0	0	0
7	0	0	0	0	0	0	....	0	0	0	0	0
:	0	0	0	0	0	0	0	....	0	0	0	0
:	0	0	0	0	0	0	0	0	....	0	0	0
:	0	0	0	0	0	0	0	0	0	....	0	0
:	0	0	0	0	0	0	0	0	0	0	x(129)	0
130	0	0	0	0	0	0	0	0	0	0	0	x(130)

Fig.5.6: Structure matrix D_ssa(eigenvalues).

	V_ssa = eigenvectors											
	C1				C2				.....	C13		
	1	2	.....	10	11	12	.....	20	.....	121	.....	130
1	x(1)	x(2)	.....	x(10)	x(11)	x(12)	.....	x(20)	.....	x(121)	.....	x(130)
2	x(2)											
3	x(3)											
4	x(4)											
5	x(5)											
6	x(6)											
7	x(7)											
8	x(8)											
9	x(9)											
10	x(10)											
:	:											
:	:											
:	:											
:	:											
:	:											
130	x(130)											

Fig.5.7: Structure matrix V_ssa (eigenvectors).

In the end of these steps we get the V_ssa matrix based on health data, so we can provide the basis for future comparison.

5.5. Data for testing

We now proceed to organize data to make appropriate tests and verify whether or not the detection of damage is possible.

We repeat what was done in 5.2 and 5.3 with healthy data, until obtain the delays matrix. Keep in mind that now use the data that were not part in obtaining the base-line covariance matrix. Unhealthy to not be used in obtaining data C and V_ssa are ready to be part of the test.

16 samples with failures (2 of each type of error) and 16 healthy samples were organized. The structure of the delays matrix of healthy samples is equals to the presented in Figure 5.4 while the structure of the delays matrix with data failure is as follows:

	Y_Mat_lag = Delayed Matrix											
	S1				S2				.....	S13		
	1	2	.....	10	11	12	.....	20	.....	121	.....	130
1	x(1)	x(t-1)	.....	x(t-9)	x(1)	x(t-1)	.....	x(t-9)	.....	x(1)	.....	x(t-9)
2	x(2)											
3	x(3)											
4	x(4)											
5	x(5)											
6	x(6)											
7	x(7)											
8	x(8)	faulty sample										
9	x(9)											
10	x(10)											
:	:											
:	:											
:	:											
:	:											
:	:											
500	x(500)											

Fig.5.8: Structure matrix Mat_lag (delayed matrix)

5.6. Graphical Results

The next step is to apply equation 4.5, which will use the eigenvectors obtained with healthy data (V_ssa) and matrix data lags obtained for use in the tests (Fig 5.8.).

In the first instance we show the results obtained with the covariance matrix base and its respective delay matrix.

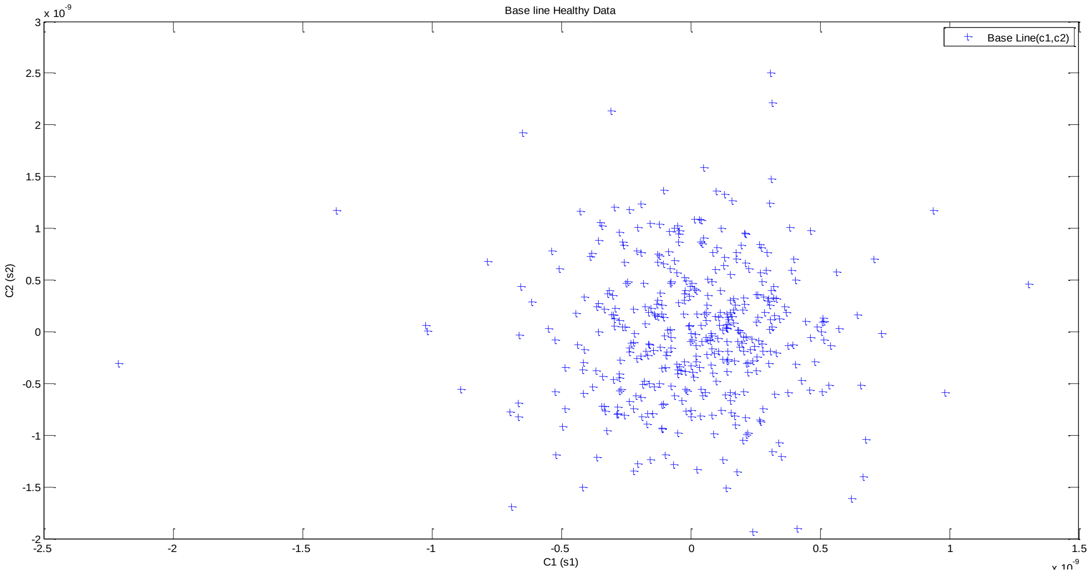


Fig.5.9: Baseline projection in the 2 first principal components

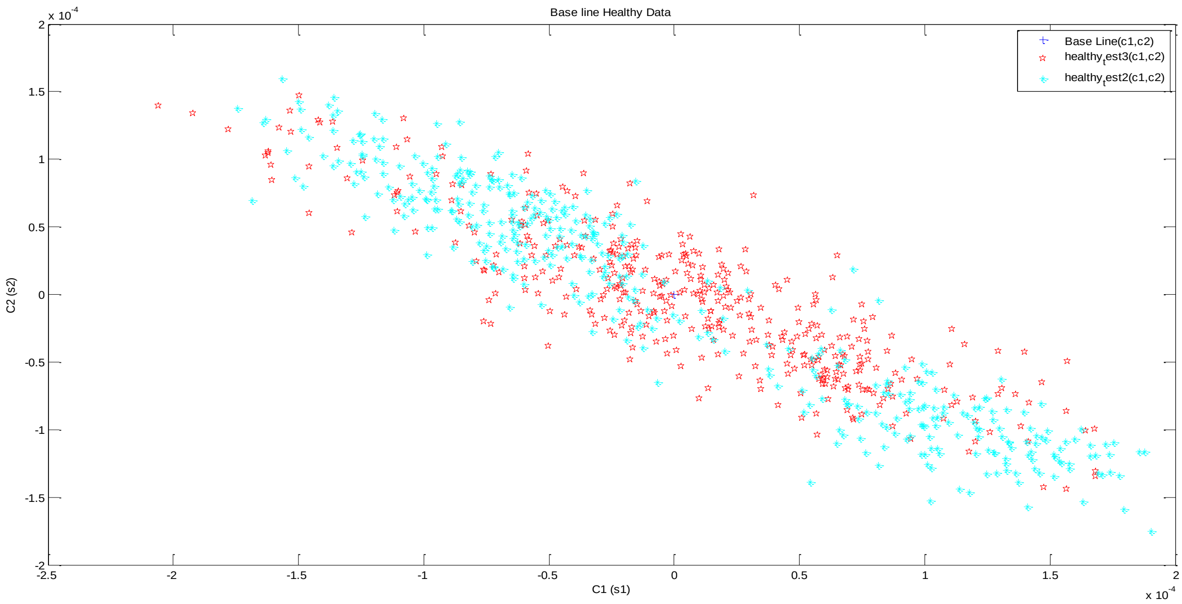


Fig.5.10.a: Projection in the 2 first principal components (baseline, healthy test1 and healthy test2)

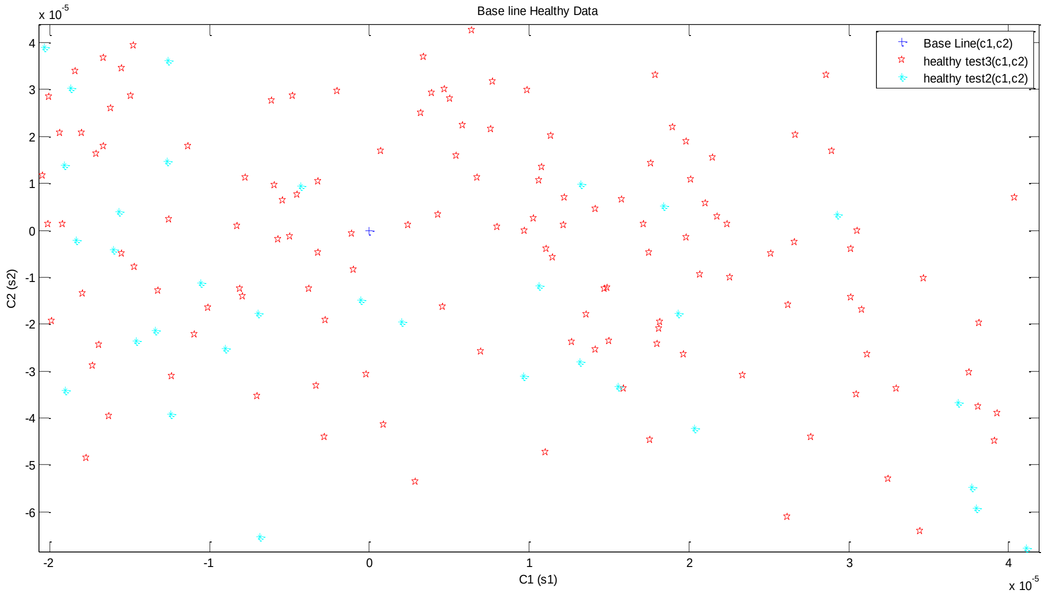


Fig.5.10.b: Zoomed area to the projection in the 2 first principal components (baseline, healthy test1 and healthy test2)

In the graphs 5.10.a and 5.10.b we can see how the healthy samples are around of the sample set as a baseline.

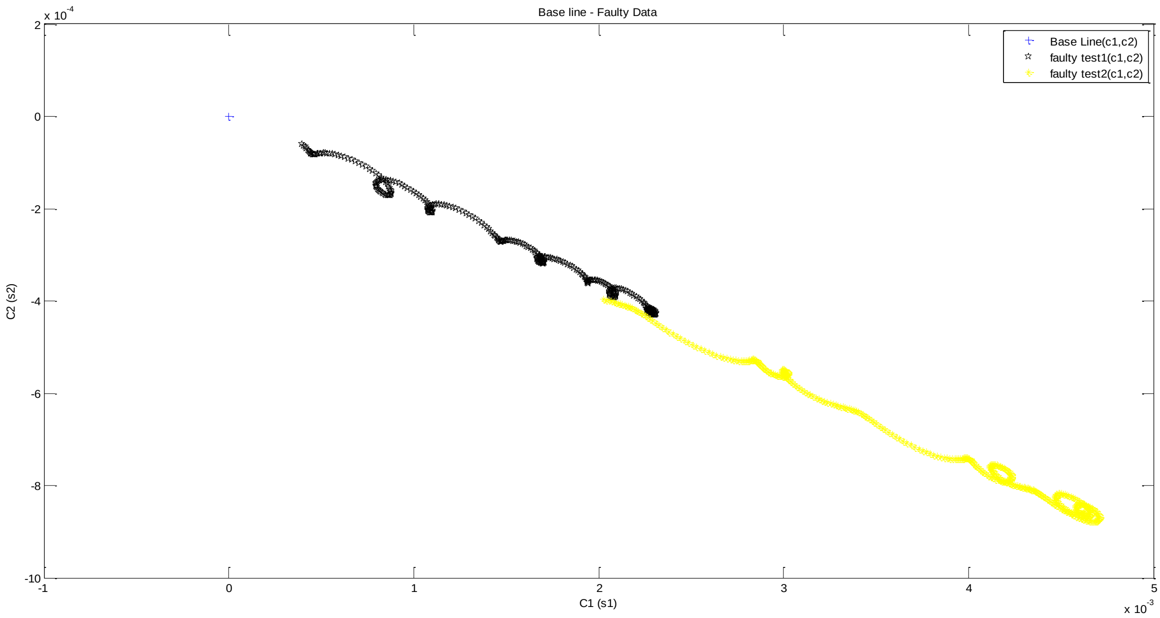


Fig.5.11.a: Faulty projection in the 2 first principal components (Baseline, faulty test1 and faulty test2)

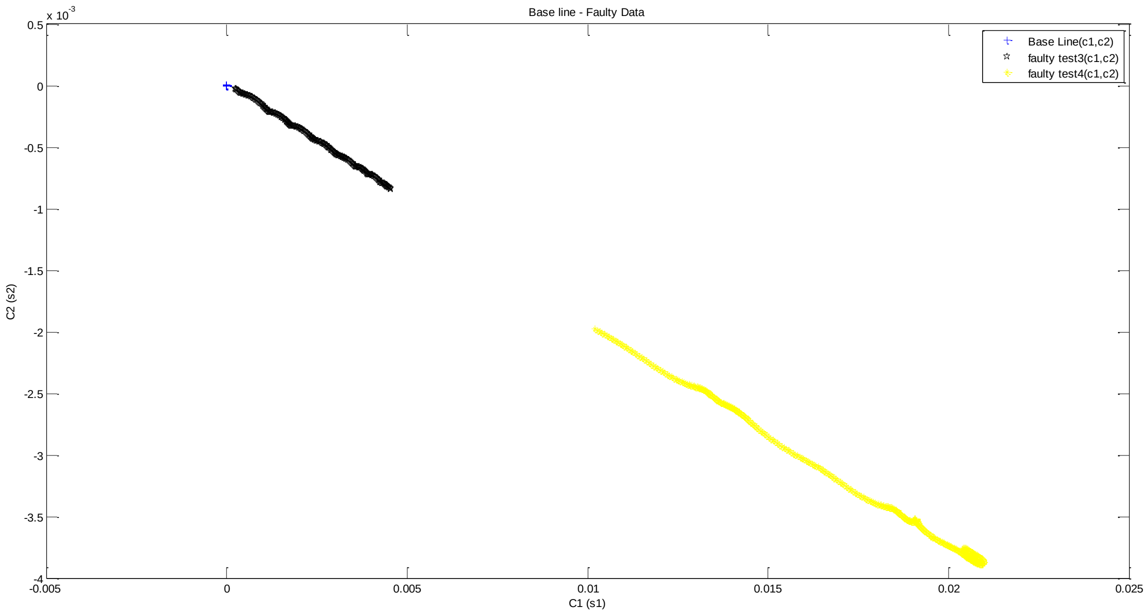
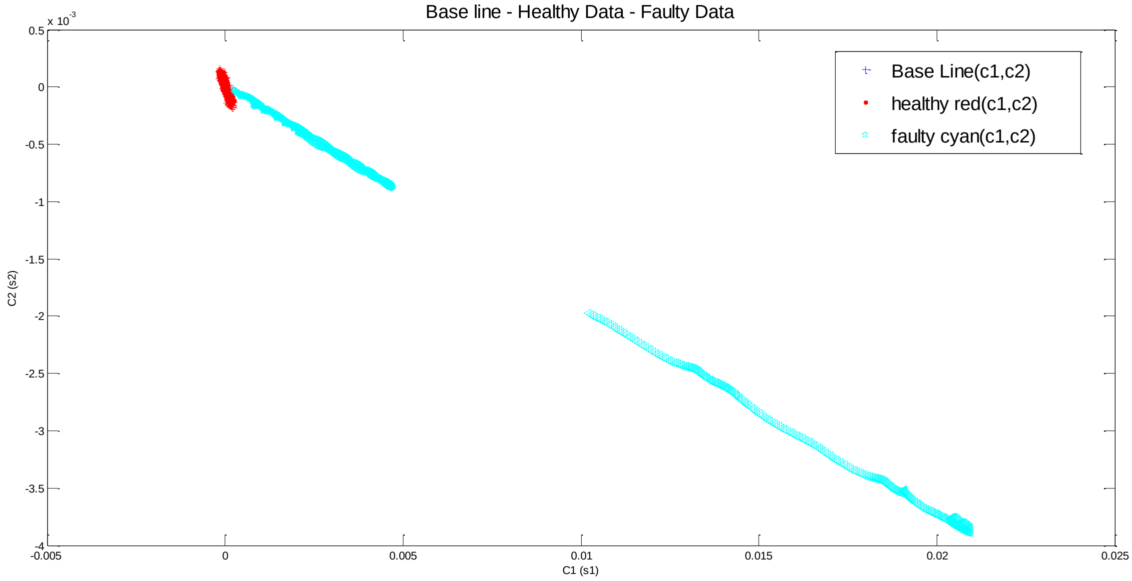


Fig.5.11.b: Faulty projection in the 2 first principal components (Baseline, faulty test3 and faulty test4)

In the graphs 5.11.a and 5.11.b we can see how faulty data can be distinguished without problem of the baseline.



. **Fig.5.12.a:** Healthy, faulty and baseline projection in the 2 first principal components

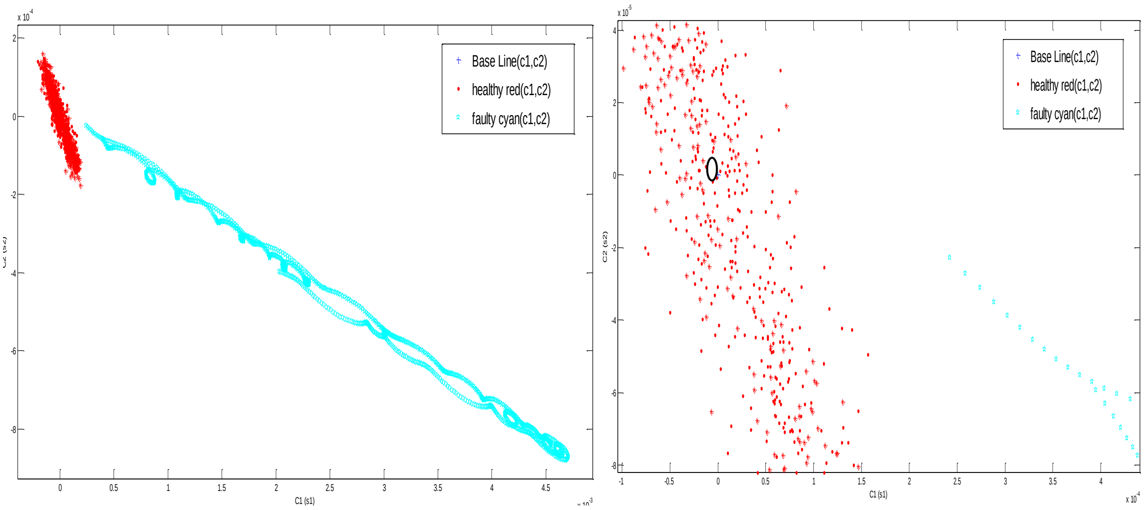


Fig.5.12.b: Approach of healthy**,** faulty and base-line projection in the 2 first principal components (Baseline (blue) – Healthy (red) – Faulty (cyan))

In graphic Fig.5.12.a and 5.12.b we can see how healthy samples are around the baseline while samples with the failure can distinguish that are separated from the baseline. The failure samples are the first 3 types of fault (F1, F2, F3 reference Table 2).

Then more results will be displayed with healthy samples and damaged samples and the projection in the first two components.

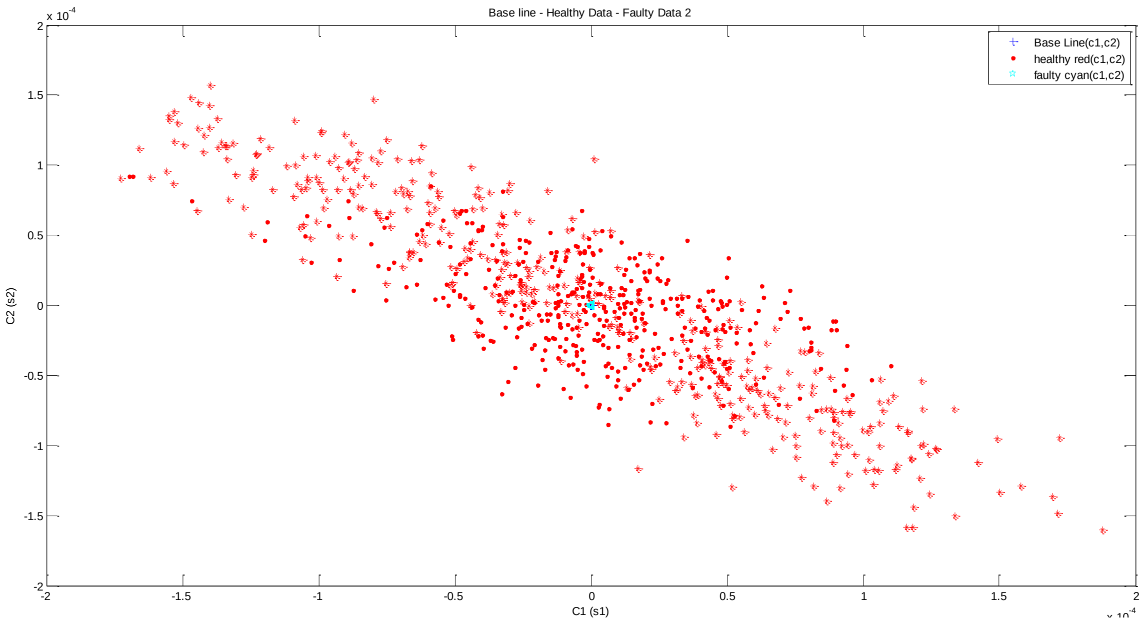


Fig.5.13.a: Projection of damaged, healthy and base-line data in the first two components (Baseline (blue) – Healthy (red) – Faulty (cyan))

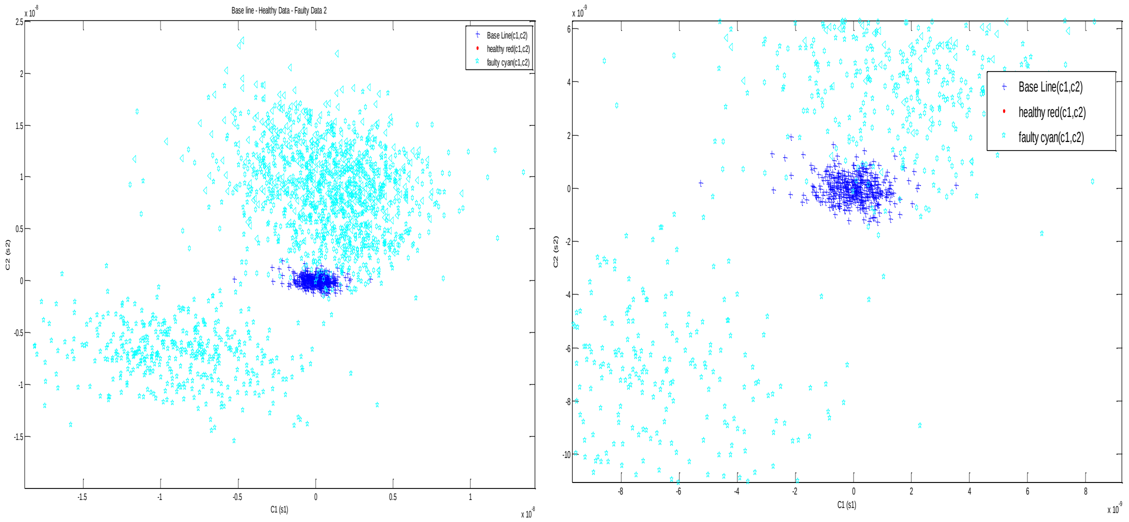


Fig.5.13.b: Approach to projection of damaged, healthy and baseline data in the first two components (Baseline (blue) – Healthy (red) – Faulty (cyan))

In Figure 5.13.a we look healthy data around the baseline, but to make a first approach we realize that we also found around faulty data.

When making a greater approach, see figure 5.13.b, we can realize that the samples with errors are found along the baseline and not around so they can be identified.

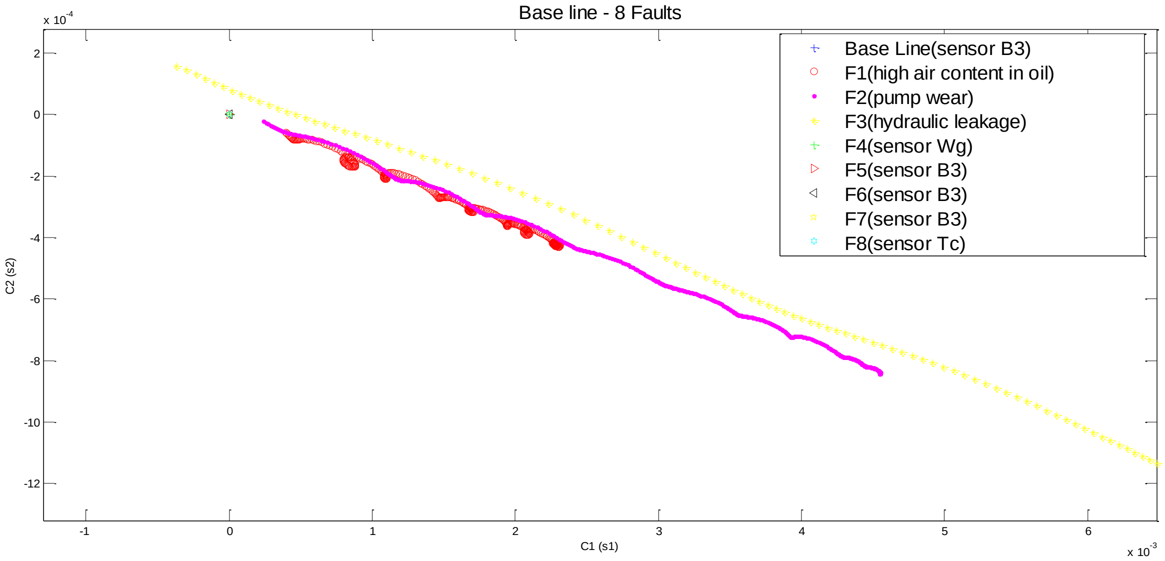


Fig.5.14.a: Projection of 8 faults and base-line data in the first two components (Baseline (blue) – Healthy (red) – Faulty (cyan))

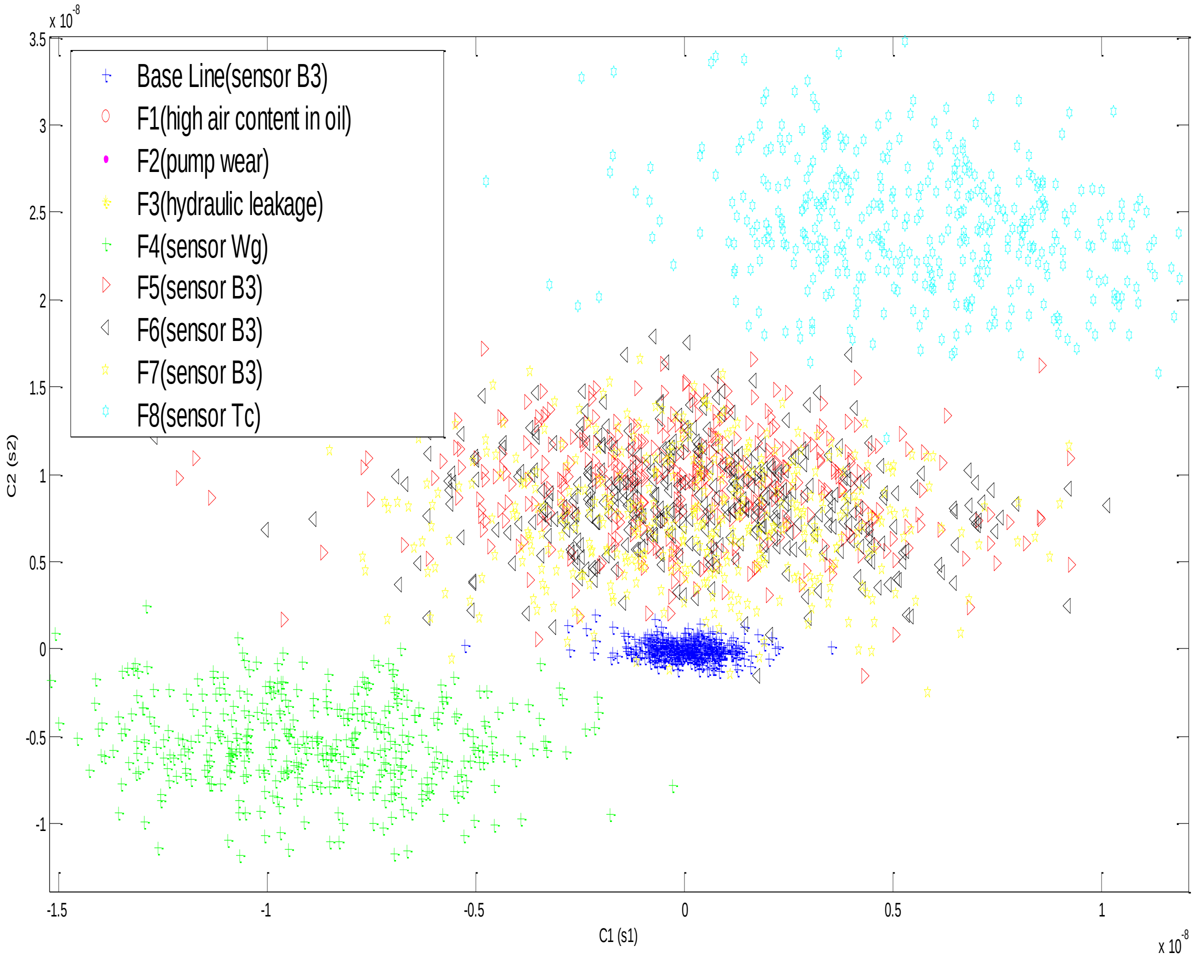


Fig.5.14.b: Approach to projection of 8 faults and base-line data in the first two components (Baseline (blue) – Healthy (red) – Faulty (cyan))

In figure 5.14.a we can see directly that 3 of the 8 faults are separated from the base line so its classification is simple.

The figure 5.14.b that is an approach of Figure 5.14.a can observe how the remaining failures also can be identified of the baseline. The failure F4 and failure F8 can be differentiated of the failures 5, 6 and 7 which are very close to each other. What happens with failures 5, 6 and 7 it is that are similar faults difference in magnitude.

5.7. Fisher Criteria

The Fisher criteria simply imply a procedure of linear correction which is applicable to any form of price and quantity index numbers. [25]

This approach will allow us to distinguish between those data sets that are separate from our baseline, in other words it allows us to distinguish between healthy samples and samples with failure .

Below we present the formula to be implemented for obtaining this index define classification:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \qquad \text{(Ec 5.1)}$$

J(w) is a measure of the difference between class means normalized by a measure of the within-class scatter matrix.
Assume we have two sets of data contained in the vectors X1 and X2. Where we will first obtain the average:

$$u1 = mean(X1)' \qquad \text{(Ec 5.2)}$$

$$u2 = mean(X2)' \qquad \text{(Ec 5.3)}$$

Then we obtain these data covariance:

$$S1 = \mathrm{cov}(X1) \qquad \text{(Ec 5.4)}$$

$$S2 = \mathrm{cov}(X2) \qquad \text{(Ec 5.5)}$$

Now we get the within-class scatter matrix

$$Sw = S1 + S2 \qquad \text{(Ec 5.6)}$$

Now we get the between-class scatter matrix

$$S_B = (u1 - u2) * (u1 - u2)' \qquad \text{(Ec 5.7)}$$

Finally we proceed to obtain W:

$$aux_W = inv(S_W)$$
$$aux\_Sb = inv(S_W) * S_B \quad \text{(Ec 5.8)}$$
$$[V, D] = eig(aux\_Sb)$$

$$w = V(:,1) \quad \text{(Ec 5.9)}$$

Substituting Equations 5.9, 5.7 and 5.6 in Equation 5.1 obtain the desired index as an indicator of the detection.

The higher the index more separate baseline is. The indexes are always calculated between the baseline and the test sample.

In this part we will test 16 healthy samples and 16 samples with failures (2 faults of each type). If the sample is healthy receive a score of 1 whereas if the sample is fault will receive a score of 0. The criteria used to allocate these markers is based on that if the achieved distance by Fisher criterion, if the index is greater 1.5 the sample is unhealthy while if the sample is less than 1.5 will be healthy.

At the end of the algorithm we obtain an array called classification where the first column is healthy data and the third column is unhealthy data. While the second and fourth columns are the markers obtained using the algorithm.

5.8. Classification Results

The final detection algorithm was:

	Samples	**Healthy**	**Faulty**
Healthy	16	**14**	**2**
Faulty	16	**0**	**16**

Table 3. Detection results (baseline c1 and c2)

This gives us a percentage of 93.75% classification which is acceptable.

5.9. Changes in baseline

We shall now proceed to change the components of the base-line to test the robustness of the classifier developed under different scenarios. The same 32 samples were used to test the first classifier were used.

Change the components of the baseline that are now component 3 and component 4.

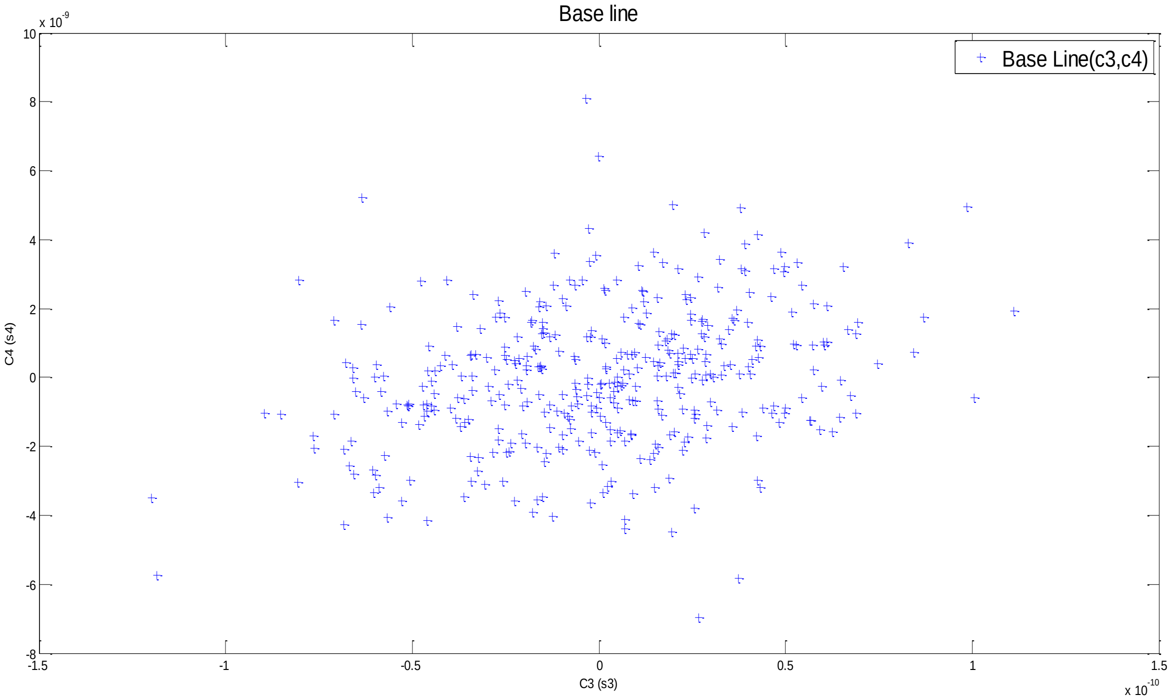


Fig.5.15: Baseline projection in the 3 and 4 principal components

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	16	**14**	**2**
Faulty	16	**0**	**16**

Table 4. Detection results (base-line c3 and c4)

As in the previous case we obtained a percentage of 93.75% classification which is acceptable.

Change the components of the baseline that are now component 5 and component 6.

The new base-line show in the next figure:

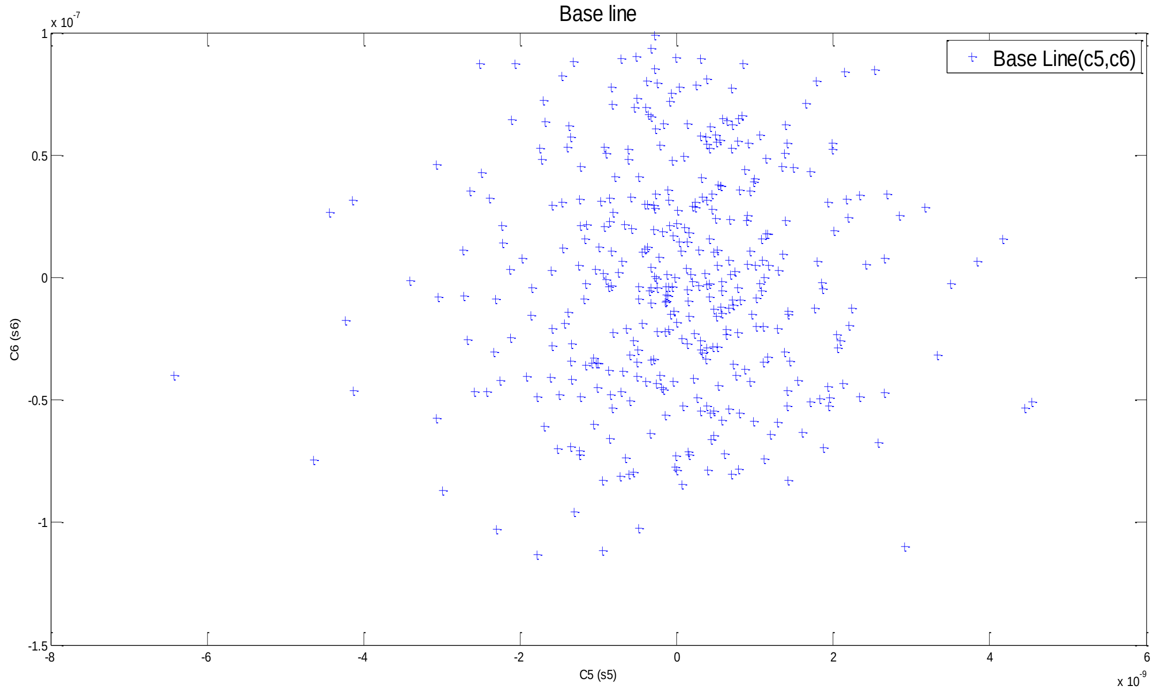


Fig.5.16: Baseline projection in the 5 and 6 principal components

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	16	**14**	**2**
Faulty	16	**10**	**6**

Table 5. Detection results (baseline c5 and c6)

We obtained a percentage of classification with a value of 62.5%, which is acceptable.

It is much less effective than in the two previous cases but in general we can say that is a relatively accepted value.

Change the components of the baseline that are now component 7 and component 8.

The new base-line show in the next figure:

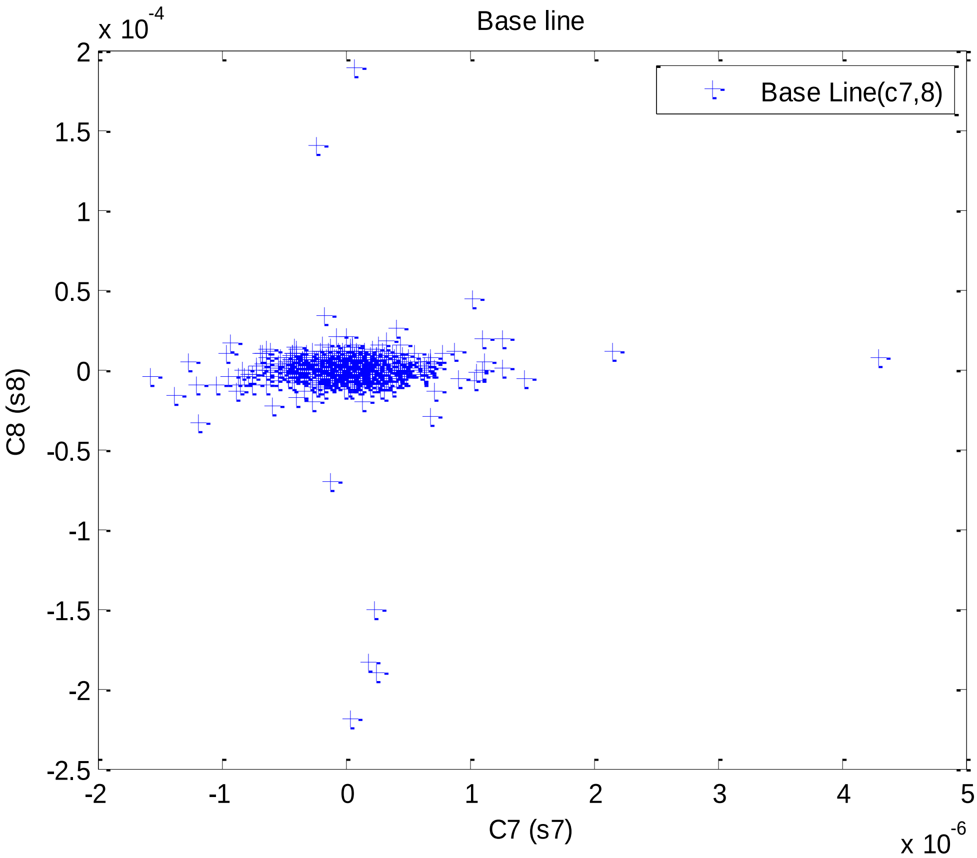


Fig.5.17: Baseline projection in the 7 and 8 principal components

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	16	**14**	**2**
Faulty	16	**10**	**6**

Table 6. Detection results (baseline c7 and c8)

We obtained a percentage of classification with a value of 62.5%, which is acceptable. It is much less effective than in the two first cases but it has equal performance in the case above. In general we can say that is a relatively accepted value.

Change the components of the baseline that are now component 9 and component 10.

The new base-line show in the next figure:

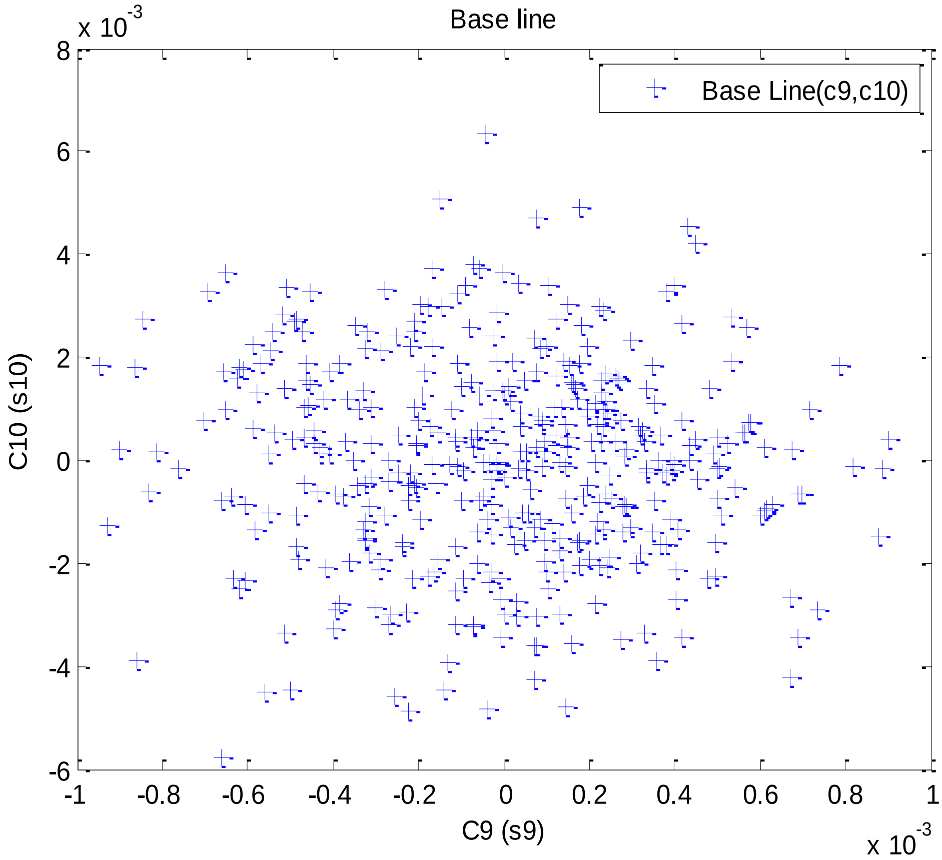


Fig.5.18: Baseline projection in the 9 and 10 principal components

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	16	**14**	**2**
Faulty	16	**12**	**4**

Table 7. Detection results (baseline c9 and c10)

We obtained a percentage of classification with a value of 56.25%, which is not good. It is much less effective than in the four cases above.

Change the components of the baseline that are now component 11 and component 12.

The new base-line show in the next figure:

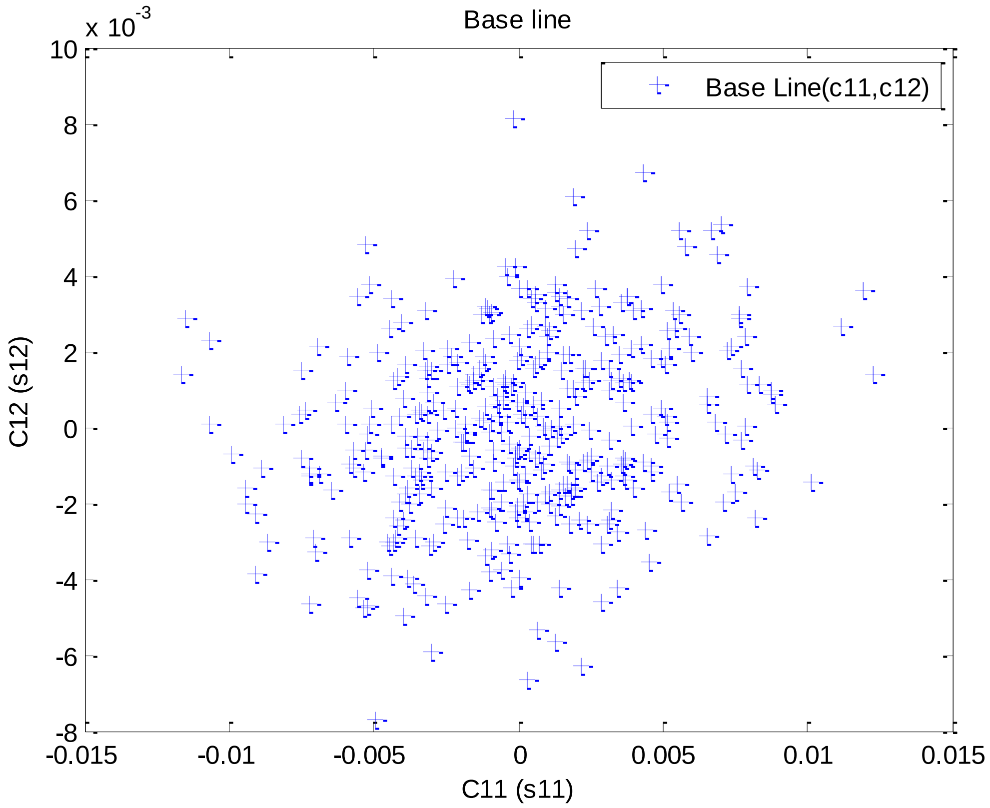


Fig.5.19: Baseline projection in the 11 and 12 principal components

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	16	**14**	**2**
Faulty	16	**14**	**2**

Table 8. Detection results (baseline c11 and c12)

We obtained a percentage of classification with a value of 50%, which is not good. It is much less effective than in the other cases above.

ETSEIB

Change the components of the baseline that are now component 13 and component 1.

The new base-line show in the next figure:

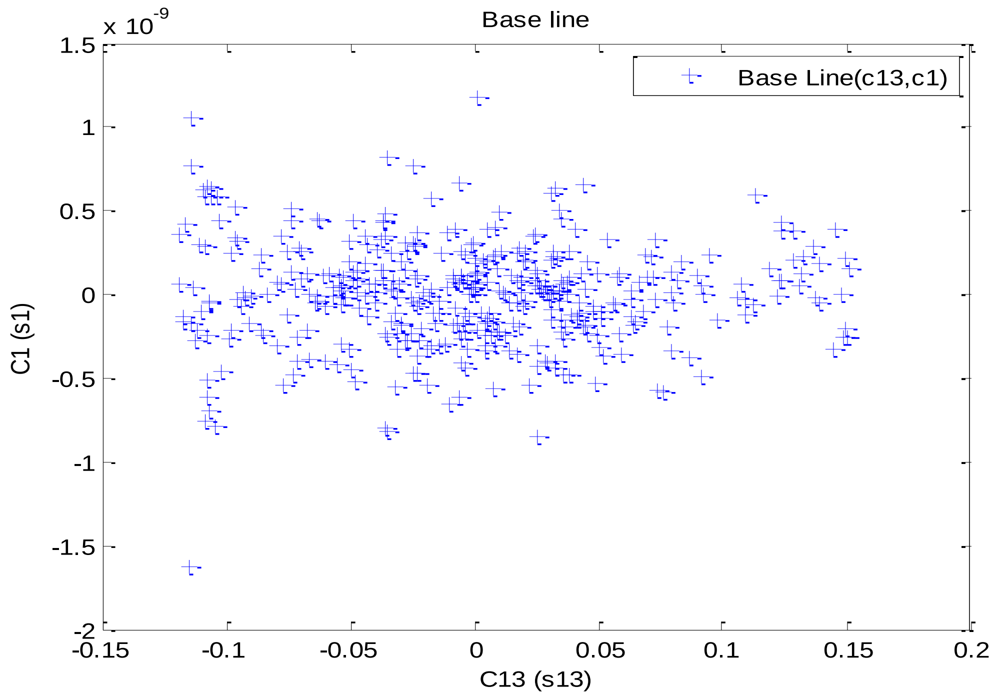


Fig.5.20: Baseline projection in the 13 and 1 principal components

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	16	**14**	**2**
Faulty	16	**16**	**0**

Table 9. Detection results (baseline c13 and c1)

We obtained a percentage of detection with a value of 43.75%, which is bad. It is the worst case of classification.

5.10. Increase Samples

The number of samples increased to 64. Composed of 32 healthy and 32 with failure (4 samples for each type of fault).

The classification with this change was:

	Samples	Healthy	Faulty
Healthy	32	**29**	**3**
Faulty	32	**30**	**2**

Table 10. Detection results 64 samples(baseline c1 and c2)

We obtained a percentage of detection with a value of 92.19%, which is good result. The efficiency of the detector system failures still good despite the increase of samples (increase 100% of samples).

The following table shows the result of the fault detection system(92.19% detecttion) sample by sample.

Sample	Fisher Criteria (healthy sample)	Detection	Fisher Criteria (faulty sample)	Detection
1	2,4507	0	14,3158	0
2	2,8550	0	7,5171	0
3	0,0008	1	22,4570	0
4	0,0102	1	4,8622	0
5	0,0090	1	28,7861	0
6	0,0123	1	3,6554	0
7	0,0024	1	10,7209	0
8	0,0136	1	7,1021	0
9	0,0064	1	2,9999	0
10	0,0068	1	6,6050	0
11	0,0038	1	3,7863	0
12	0,0460	1	5,4646	0
13	0,0001	1	6,7721	0
14	0,0754	1	3,9565	0
15	0,0026	1	37,4981	0
16	0,0829	1	35,2889	0
17	0,3871	1	21,1978	0
18	2,7973	0	0,5150	1
19	0,0141	1	51,7902	0
20	0,0036	1	1,0871	1
21	0,0137	1	75,9321	0
22	0,0009	1	1,5237	0
23	0,0098	1	4,8416	0
24	0,0036	1	5,4866	0
25	0,0263	1	2,4361	0
26	0,0038	1	17,1295	0
27	0,0420	1	4,0726	0
28	0,0019	1	20,5976	0
29	0,0433	1	5,2506	0
30	0,0078	1	28,8167	0
31	0,0444	1	31,7586	0
32	0,0019	1	97,5539	0

Table 11. Detection results 64 samples(sample by sample)

Now we change one component of the baseline to prove the components one by one and the results are present below:

	Samples	%Effectiveness	Time (seconds)	Baseline
1	64	**92.19**	**11.6**	**c1 – c3**
2	64	**92.19**	**11.9**	**c1 – c4**
3	64	**92.19**	**11.3**	**c1 – c5**
4	64	**60.94**	**11.7**	**c1 – c6**
5	64	**60.94**	**11.4**	**c1 – c7**
6	64	**60.94**	**11.7**	**c1 – c8**
7	64	**60.94**	**11.9**	**c1 – c9**
8	64	**54.69**	**12.2**	**c1 – c10**
9	64	**50**	**11.4**	**c1 – c11**
10	64	**53.13**	**11.4**	**c1 – c12**
11	64	**45.31**	**11.5**	**c1 – c13**

Table 12. Detection results 64 samples (change components)

6. Environmental, economic and social impact

This chapter presents the commitment that the developed project faces with the community. It is intended to present an estimation of the cost of the project and the impact that this work proposes in a social environment.

6.1. Environmental Impact

According to the Directive 2011/92/EU of the European Parliament and of the Council on the assessment of the effects of certain public and private projects on the environment [16], this developed work not has significant effects on the environment. The resources used for developing this are different kind of software and this is why this project does not present any menace to the environment.

The project presented in this master thesis is a project of research with no direct applications in industry, it has no significant effects on the environment and an environmental impact analysis is not required.

6.2. Economical Impact

The work developed in this thesis is not a product to be sold and have no direct economic benefits. This thesis had the aim to investigate a method in order to contribute with the current state of the art in detection of faults in wind turbines. The impact of this work is increasing the amount of knowledge and contributions created in the UPC, this increases the UPC's notoriety.

6.2.1. Human resources

Since I am a student which is doing a thesis and I do not have any grant the human resources cost is 0. For seek of completeness we hypothesize this work could be done by a private institution and we want to analyses the costs of the human resources of the project. The total numbers of hours are an approximation of the hours I invested in the project.

Time(hours)	Task	Hour Cost(€)	Cost(€)
200	Literature Review	19,44	3888
160	Algorithm Programming	19,44	3110.4
80	Data Processing / Test	19,44	1555.2

Table 13. Cost of human resources

For software and hardware analysis, the amortization amount that corresponds to the weeks of use of each resource was considered. For some elements that their lifetime are unknown, information from 27/2014 law for corporate tax in Spain was taken [17]. It considers a maximum lifetime of 6 years for software, 10 years for electronic devices

6.2.2. Hardware Resources

Units	Resource	Unit Cost (€)	Life Time (years)	Amortization (€)
1	Desktop	350	6	5.83
1	Laptop Sony Vaio I7	1400	10	14.1
1	Wireless/Router	115	10	1.16

Table 14. Cost of hardware resources

Finally, the software considered for this project are those used in algorithm development, data processing and documentation.

6.2.3. Software Resources

Units	Resource	Unit Cost (€)	Life Time (years)	Amortization (€)
1	Matlab Standard	2000	6	10.95

Table 15. Cost of software resources

6.2.4. Total Cost

Considering the costs of the hardware, software and human resources we obtain the total cost of the project. We consider also a margin of contingencies of 15% in order to consider additional costs and the error of the estimation of the costs.

Concept	Cost (€)
Human Resources	8553.6
Harwdware	1865
Software	2000
Subtotal	12418.6
Margin (15%)	1862.79
Total	14281.4

Table 16. Total costs

6.3. Social Impact

They are the first work done in the field of troubleshooting with these techniques, new in the field, so at this point you cannot speak of an immediate impact.

Now what you can say is the future impact because having a system efficient and effective detection of failures we can reduce the costs of maintenance of turbines and reduce unexpected shutdown of them.

This means reduced maintenance costs it means reducing the cost per kilowatt / h produced which generally will be reflected in the bill of customers. Another advantage for the future is that the quality of the product delivered to consumers will be improved.

Conclusions

The combination of different techniques such as SSA, MSSA and statistical techniques allow us to reach an acceptable outcome with respect to the detection of healthy samples and samples with failure.

A change in physical properties due to structural changes or damage will cause detectable changes in dynamical responses. Even with different wind turbulence, the fault detection can able to detect some damage, fault or misbehavior because the wind is a random variable that it was not considered by its nature.

The best detector of faults is comprised of the first components of the baseline. Its effectiveness is the 93.75% which allows us to infer that it has developed a good fault detection system.

One of the main problems when working with such large time series is the computational cost that means working with them. Should optimize the code and have a good computer to minimize the response time of the system.

Being different types of failures the characterizing of the line base for all type of failure has a high level of complexity, so selecting the components of the baseline will be crucial for the proper performance of the detector or system failures. This could be verified in the tests when changing the components of the baseline the performance of the system fell down.

The combination of SSA with statistical techniques, such as the criterion of Fisher, helped strengthen the detector system failures given the variety of different failure scenarios

The number of samples is increased by the detector fault system and its efficacy was reduced slightly (1.66%), which is normal considering that the number of samples increase in a 100%.

The method of Singular Spectrum Analysis (SSA) is a variant of principal component analysis applied to time series analysis. The procedure involves making a multivariate statistical study from a simple time series, generating multiple series obtained lag applied to the original series, with which the so-called delayed covariance matrix is constructed.

Using the technique of SSA and Fisher criterion we obtained acceptable results to analyze and not even used frequency based methods.

Future Work

The first thing to look for is the optimization of code designed for fault detection system, that leads to improve the processing time of the detector system since the amount of data that must be managed is very large.

Another aspect to be addressed in the future is to deepen the implementation of these new techniques in the field of fault detection as SSA and seek to complement them with more intervention techniques such as statistics to improvise system operation. One technique that could intervene is the neural networks, for example could provide in the phase of construction of the baseline. In this way it could be handled in a better way the various failure scenarios at the same time, and the number of variables you have in each.

Acknowledgments

First of all I thanks to my parents who always stood by me and believed in me when I no longer believed even. They are the pillars of this new achievement even giving me all the support being separated by many kilometers.

I thanks to my brother for being that person who is always willing to help in everything without explanations and sharing so many good and bad moments in this adventure.

I thank to my director Francesc Pozo for all the shared knowledge and the unconditional support given to me for the development of this master thesis.

I thank to the Polytechnic University of Catalonia (UPC) , its entire administrative staff and teachers by all the knowledge given, for the help and facilities to have the best two years of an unforgettable experience.

I thank my classmates with whom we had the opportunity to share this experience because they became a very important support in difficult times and that being so far from home they became a kind of family.

Finally my love to one little angel named Rocio that accompanied me throughout this journey, in good and bad times but especially in those moments where i needed recharge and free the mind to move forward.

.

Bibliography

References

[1] Steven M. LaValle. Planning Algorithms. Cambridge University Press, May 2006.

[2] R. Kumaresan and D.W. Tufts, Data-adaptative principal component signal processing, in: IEEE Proc. Conf. on Decision and Control (Albuquerque,1980) 949.

[3] Vautard, R., Yiou, P., & Ghil, M. (1992). Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena*, *58*(1), 95-126.

[4] K. Fraedrich, J.Atmos. Sci.43(1986) 419.

[5] Hassani, H. (2007). Singular spectrum analysis: methodology and comparison.

[6] Garcia, D., & Trendafilova, I. (2014). A multivariate data analysis approach towards vibration analysis and vibration-based damage assessment: Application for delamination detection in a composite beam. *Journal of Sound and Vibration*, *333*(25), 7036-7050.

[7] Pozo, F., Arruga, I., Mujica, L. E., Ruiz, M., & Podivilova, E. (2016). Detection of structural changes through principal component analysis and multivariate statistical inference. *Structural Health Monitoring*, 1475921715624504.

[8] Pozo, F., & Vidal, Y. (2015). Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing. *Energies*, *9*(1), 3.

[9] Visible body: alternative energy / wind turbines.(15/07/2005). Recovered http://www.textoscientificos.com/energia/turbinas.

[10] MSc. Jorge Mírez.(01/01/2012).Visible body: Explanation of parts of a wind generator. Recovered of https://jmirez.wordpress.com/2012/01/01/j389-explicacion-de-partes-de-un- generador-eolico/.

[11] Multilingual/WordPress.Visible body: COMPONENTES DE UN AEROGENERADOR. Recovered of http://energiadoblecero.com/energias-renovables/energia-eolica/componentes-de- un-aerogenerador.

[12] Pozo, F., Vidal, Y., & Serrahima, J. M. (2016). On Real-Time Fault Detection in Wind Turbines: Sensor Selection Algorithm and Detection Time Reduction Analysis. *Energies*, *9*(7), 520.

[13] `http://hdl.handle.net/123456789/4341`

[14] J.B.Elsner and Anastasios A Tsonis.(1996).Visible body: Singular Spectrum Analysis. Springer Science+Business Media, LLC. Recovered https://books.google.es/books?id=5sfSBwAAQBAJ&dq=SSA+singular+spectral+analysis+&lr=&hl=es&source=gbs_navlinks_s.

[15] Rukhin, A. L. (2002). Analysis of time series structure SSA and Related techniques. *Technometrics*, *44*(3), 290-290.

[16] European Parliament and Council of the European Union. Environmental Impact Assessment (EIA) Directive (2011/92/EU). Official Journal of the European Union. Amended on April 2014. 2011.

[17] Spain (2014). Ley 27/2014, de 27 de noviembre, del Impuesto sobre Sociedades. Boletín Oficial del Estado, viernes 28 de noviembre de 2014. Available on https://www.boe.es/boe/dias/2014/11/28/pdfs/BOEA-2014-12328.pdf

[18] *NWTC Information Portal (FAST).* https://nwtc.nrel.gov/FAST. Last modified 19-March-2015 ; Accessed 04-October-2016

[19] Aly A. Farag Shireen Y. Elhabian.(02/10/2008).Visible body: A tutorial on data reduction. University of Louisville (CVIP LAB). Recovered of https://www.lsv.uni-saarland.de/fileadmin/teaching/dsp/ss15/DSP2016/matdid437773.pdf

[20] DEL PINO, S. B., & UNIVERSITARIA, P. U. ESTADÍSTICA DESCRIPTIVA E INFERENCIAL.

[21] Muljadi, E., & Butterfield, C. P. (2001). Pitch-controlled variable-speed wind turbine generation. *IEEE Transactions on Industry Applications*, *37*(1), 240-246.

[22] Martínez, E., Sanz, F., Pellegrini, S., Jiménez, E., & Blanco, J. (2009). Life cycle assessment of a multi-megawatt wind turbine. *Renewable Energy*, *34*(3), 667-673.

[23] Muyeen, S. M., Takahashi, R., Murata, T., & Tamura, J. (2010). A variable speed wind turbine control strategy to meet wind farm grid code requirements.*IEEE Transactions on power systems*, *25*(1), 331-340.

[24] Energy.gov. Visible body: The Inside of a Wind Turbine. Office of Energy Efficiency & Renewable Energy. Recovered of http://www.energy.gov/eere/wind/inside-wind-turbine-0

[25] Jazairi, N. (1972). Fisher Ideal Index Reformulated. *International Statistical Review / Revue Internationale De Statistique, 40*(1), 47-51. doi:1. Retrieved from http://www.jstor.org/stable/1402705 doi:1

7. ANNEXES

7.1. MATLAB CODE

```
close all;
clear all;
clc;
time_inicio=cputime;
N = 13; %numero de sensores utilizados
n = 50; %muestras saludables
nfallos = 8; %numero de fallos
L = 48001; % muestras (paso de 0.0125 s).
rL =500; % numero de muestras reducida 500 (6.25 segundos).
muestra = 16; %muestras a considerar de cada experimento
%load healthy samples
for i=1:n
    var=strcat('SimulacioSaludableRegio3_',num2str(i));
    var=strcat(var,'.mat');
    load(var);
    OutData(:,5)=[]; %velocidad del viento (debido a ser
aleatoria no interesa)
    for j=1:N
        for k=1:(muestra+1)
            X(i+(k-1)*n,((j-1)*rL+1):j*rL) = OutData((L-
k*rL+1):L-(k-1)*rL,j)';
        end
    end
end

%To load the faulty samples:
for fallo=1:nfallos
var=strcat('SimulacioFallo',num2str(fallo));
var=strcat(var,'.mat');
load(var);
OutData(:,5)=[];
    for k=1:n
        for j=1:N
            Y((fallo-1)*n+k,((j-1)*rL+1):j*rL) = OutData((L-
k*rL+1):(L-(k-1)*rL),j)';
        end
    end
end
%Normalized
for i=1:N
    dt(i)=std(reshape(X(1:n,(i-1)*rL+1:i*rL),1,n*rL));
    for j=1:rL % number of columns per block
        %XT is the scaled matrix after the group-scaling
        %centers X by subtracting off column means
```

ETSEIB

```
        XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-
1)*rL+j)))/dt(i);
    end
end

for i=1:N
    dty(i)=std(reshape(X(1:n,(i-1)*rL+1:i*rL),1,n*rL));
    for j=1:rL % number of columns per block
    % XT is the scaled matrix after the group-scaling
    % centers X by subtracting off column means
    YT(:,(i-1)*rL+j) = (Y(:,(i-1)*rL+j)-mean(X(1:n,(i-
1)*rL+j)))/dty(i);
    end
end

%% SSA
for j=1:13
    for i=1:500
    XT_aux(i,j)=mean(XT(:,i+j*500-500));
    end
end
N_cov=50;
M_lag=10;
new_XT=zeros(N_cov,M_lag);
M_cov=12;
Mat_lag=zeros(rL,M_lag*(M_cov+1));
for i=0:M_cov
    for j=1:M_lag
        new_XT((1:500-j+1),j) = XT_aux((1:500-j+1)+j-1,i+1);
    end
Mat_lag(:,(i*M_lag+1):(i*M_lag+10))=new_XT(:,:);
end

C=Mat_lag'*Mat_lag/L;
[V_ssa,D_saa]=eig(C);
PC=Mat_lag*V_ssa;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Test%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

 muestra2=33;
 for i=1:n
 var=strcat('SimulacioSaludableRegio3_',num2str(i));
 var=strcat(var,'.mat');
 load(var);
 OutData(:,5)=[]; %velocidad del viento(debido a ser
aleatoria no interesa)
    for j=1:N
        for k=17:(muestra2)
        X_test(i+(k-1-16)*n,((j-1)*rL+1):j*rL) = OutData((L-
k*rL+1):L-(k-1)*rL,j)';
```

ETSEIB

```
        X_test2(i+(k-1-16)*n,((j-1)*rL+1):j*rL) = OutData((L-2*k*rL+1):L-(2*k-1)*rL,j)';
        end
    end
 end

%Normalize
for i=1:N
dt(i)=std(reshape(X_test(1:n,(i-1)*rL+1:i*rL),1,n*rL));
dt(i)=std(reshape(X_test2(1:n,(i-1)*rL+1:i*rL),1,n*rL));
    for j=1:rL % number of columns per block
        %XT is the scaled matrix after the group-scaling
        %centers X by subtracting off column means
        XT_test(:,(i-1)*rL+j) = (X_test(:,(i-1)*rL+j)-mean(X_test(1:n,(i-1)*rL+j)))/dt(i);
        XT_test2(:,(i-1)*rL+j) = (X_test2(:,(i-1)*rL+j)-mean(X_test2(1:n,(i-1)*rL+j)))/dt(i);
    end
end

%%%Ordenando matriz de muestras saludables%%%%
for j=0:12
    for i=1:500
    XTest_aux(:,i+j*500)= XT_test(i,(1+j*500):(j*500+500))';
    XTest_aux2(:,i+j*500)=XT_test2(i,(1+j*500):(j*500+500))';
    end
end

N_covt=50;
M_lagt=10;
new_XTest=zeros(N_covt,M_lagt);
new_XTest2=zeros(N_covt,M_lagt);
M_covt=12;
Mat_lagt=zeros(rL,M_lagt*(M_covt+1));
Mat_lagt2=zeros(rL,M_lagt*(M_covt+1));
for i=0:M_covt
    for j=1:M_lagt
        new_XTest((1:500-j+1),j) = XTest_aux((1:500-j+1)+j-1,i+1);
        new_XTest1((1:500-j+1),j) = XTest_aux((1:500-j+1)+j-1,i+2);
        new_XTest2((1:500-j+1),j) = XTest_aux((1:500-j+1)+j-1,i*500+51);
        new_XTest3((1:500-j+1),j) = XTest_aux((1:500-j+1)+j-1,i*500+52);    %H3
        new_XTest4((1:500-j+1),j) = XTest_aux((1:500-j+1)+j-1,i*500+101);   %H4
        new_XTest5((1:500-j+1),j) = XTest_aux((1:500-j+1)+j-1,i*500+102);   %H5
```

```
        new_XTest6((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+151);   %H6
        new_XTest7((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+152);   %H7
        new_XTest8((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+201);   %H8
        new_XTest9((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+202);   %H9
        new_XTest10((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+251);   %H10
        new_XTest11((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+252);   %H11
        new_XTest12((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+301);   %H12
        new_XTest13((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+302);   %H13
        new_XTest14((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+351);   %H14
        new_XTest15((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+352);   %H15

%%%%%%%%%%%%%% more data for test
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        new_XTest16((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i+3);         %h16
        new_XTest17((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i+4);         %h17
        new_XTest18((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+53);  %h18
        new_XTest19((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+54);  %H19
        new_XTest20((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+103); %H20
        new_XTest21((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+104); %H21
        new_XTest22((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+153); %H22
        new_XTest23((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+154); %H23
        new_XTest24((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+203); %H24
        new_XTest25((1:500-j+1),j)  = XTest_aux((1:500-j+1)+j-
1,i*500+204); %H25
        new_XTest26((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+253);  %H26
        new_XTest27((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+254);  %H27
        new_XTest28((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+303);  %H28
        new_XTest29((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+304);  %H29
```

```
        new_XTest30((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+353);  %H30
        new_XTest31((1:500-j+1),j)= XTest_aux((1:500-j+1)+j-
1,i*500+354);  %H31


    end
Mat_lagt(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest(:,:);
Mat_lagt1(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest1(:,:);
Mat_lagt2(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest2(:,:);
Mat_lagt3(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest3(:,:);
Mat_lagt4(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest4(:,:);
Mat_lagt5(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest5(:,:);
Mat_lagt6(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest6(:,:);
Mat_lagt7(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest7(:,:);
Mat_lagt8(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest8(:,:);
Mat_lagt9(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest9(:,:);
Mat_lagt10(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest10(:,:);
Mat_lagt11(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest11(:,:);
Mat_lagt12(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest12(:,:);
Mat_lagt13(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest13(:,:);
Mat_lagt14(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest14(:,:);
Mat_lagt15(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest15(:,:);

%%% segundo grupo de datos
Mat_lagt16(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest16(:,:);

Mat_lagt17(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest17(:,:);

Mat_lagt18(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest18(:,:);

Mat_lagt19(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest19(:,:);

Mat_lagt20(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest20(:,:);
Mat_lagt21(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest21(:,:);
Mat_lagt22(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest22(:,:);
Mat_lagt23(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest23(:,:);
Mat_lagt24(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest24(:,:);
Mat_lagt25(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest25(:,:);
Mat_lagt26(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest26(:,:);
Mat_lagt27(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest27(:,:);
Mat_lagt28(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest28(:,:);
Mat_lagt29(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest29(:,:);
Mat_lagt30(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest30(:,:);
Mat_lagt31(:,(i*M_lagt+1):(i*M_lagt+10))=new_XTest31(:,:);


end

%%%%%%%%healthy base
line%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Base_line=Mat_lag*V_ssa; %all variables healthy used for the
coviarance matrix for the base line

%%%%%%%%%%%%%TEST WITH OTHERS HEALTHY
SAMPLES%%%%%%%%%%%%%%%%%%%%%
t=Mat_lagt*V_ssa;
t1=Mat_lagt1*V_ssa;
t2=Mat_lagt2*V_ssa;
t3=Mat_lagt3*V_ssa;
t4=Mat_lagt4*V_ssa;
t5=Mat_lagt5*V_ssa;
t6=Mat_lagt6*V_ssa;
t7=Mat_lagt7*V_ssa;
t8=Mat_lagt8*V_ssa;
t9=Mat_lagt9*V_ssa;
t10=Mat_lagt10*V_ssa;
t11=Mat_lagt11*V_ssa;
t12=Mat_lagt12*V_ssa;
t13=Mat_lagt13*V_ssa;
t14=Mat_lagt14*V_ssa;
t15=Mat_lagt15*V_ssa;
t16=Mat_lagt16*V_ssa;
t17=Mat_lagt17*V_ssa;
t18=Mat_lagt18*V_ssa;
t19=Mat_lagt19*V_ssa;
t20=Mat_lagt20*V_ssa;
t21=Mat_lagt21*V_ssa;
t22=Mat_lagt22*V_ssa;
t23=Mat_lagt23*V_ssa;
t24=Mat_lagt24*V_ssa;
t25=Mat_lagt25*V_ssa;
t26=Mat_lagt26*V_ssa;
t27=Mat_lagt27*V_ssa;
t28=Mat_lagt28*V_ssa;
t29=Mat_lagt29*V_ssa;
t30=Mat_lagt30*V_ssa;
t31=Mat_lagt31*V_ssa;

%%%test
faulty%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
for j=0:12
    for i=1:400
        YTest_aux(:,(i+j*400))=YT(i,(1+j*500):(j*500+500))';
    end
end

 Y_N_covt=50;
 Y_M_lagt=10;
 Y_M_covt=12;
 Y_Mat_lagt=zeros(rL,Y_M_lagt*(Y_M_covt+1));
 for i=0:Y_M_covt
     for j=1:Y_M_lagt
        new_YTest((1:500-j+1),j)  = YTest_aux((1:500-j+1)+j-
1,i*400+1);     %f1
        new_YTest1((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+2);     %f1
        new_YTest2((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+51);    %f2
        new_YTest3((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+52);    %f2
        new_YTest4((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+101);   %f3
        new_YTest5((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+102);   %f3
        new_YTest6((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+151);   %f4
```

```
        new_YTest7((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+152);   %f4
        new_YTest8((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+201);   %f5
        new_YTest9((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+202);   %f5
        new_YTest10((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+251);   %f6
        new_YTest11((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+252);   %f6
        new_YTest12((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+301);   %f7
        new_YTest13((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+302);   %f7
        new_YTest14((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+351);   %f8
        new_YTest15((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+352);   %f8


        %%segundo set de muestras
        new_YTest16((1:500-j+1),j)  = YTest_aux((1:500-
j+1)+j-1,i*400+3);      %f1
        new_YTest17((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+4);     %f1
        new_YTest18((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+53);    %f2
        new_YTest19((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+54);    %f2
        new_YTest20((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+103);   %f3
        new_YTest21((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+104);   %f3
        new_YTest22((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+153);   %f4
        new_YTest23((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+154);   %f4
        new_YTest24((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+203);   %f5
        new_YTest25((1:500-j+1),j) = YTest_aux((1:500-j+1)+j-
1,i*400+204);   %f5
        new_YTest26((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+253);   %f6
        new_YTest27((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+254);   %f6
        new_YTest28((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+303);   %f7
        new_YTest29((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+304);   %f7
        new_YTest30((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+353);   %f8
```

```
        new_YTest31((1:500-j+1),j)= YTest_aux((1:500-j+1)+j-
1,i*400+354);    %f8


     end
 Y_Mat_lagt(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest(:,:);


Y_Mat_lagt1(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest1(:,:)
;

Y_Mat_lagt2(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest2(:,:)
;

Y_Mat_lagt3(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest3(:,:)
;

Y_Mat_lagt4(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest4(:,:)
;

Y_Mat_lagt5(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest5(:,:)
;

Y_Mat_lagt6(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest6(:,:)
;

Y_Mat_lagt7(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest7(:,:)
;

Y_Mat_lagt8(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest8(:,:)
;

Y_Mat_lagt9(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest9(:,:)
;

Y_Mat_lagt10(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest10(:,
:);

Y_Mat_lagt11(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest11(:,
:);

Y_Mat_lagt12(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest12(:,
:);

Y_Mat_lagt13(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest13(:,
:);

Y_Mat_lagt14(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest14(:,
:);
```

```
Y_Mat_lagt15(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest15(:,
:);

 %%%%%%% segundo set de datos

Y_Mat_lagt16(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest16(:,
:);

Y_Mat_lagt17(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest17(:,
:);

Y_Mat_lagt18(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest18(:,
:);

Y_Mat_lagt19(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest19(:,
:);

Y_Mat_lagt20(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest20(:,
:);

Y_Mat_lagt21(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest21(:,
:);

Y_Mat_lagt22(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest22(:,
:);

Y_Mat_lagt23(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest23(:,
:);

Y_Mat_lagt24(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest24(:,
:);

Y_Mat_lagt25(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest25(:,
:);

Y_Mat_lagt26(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest26(:,
:);

Y_Mat_lagt27(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest27(:,
:);

Y_Mat_lagt28(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest28(:,
:);

Y_Mat_lagt29(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest29(:,
:);

Y_Mat_lagt30(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest30(:,
:);
```

```
Y_Mat_lagt31(:,(i*Y_M_lagt+1):(i*Y_M_lagt+10))=new_YTest31(:,
:);



 end
tf=Y_Mat_lagt*V_ssa;
%f1
tf1=Y_Mat_lagt1*V_ssa;
%f1
tf2=Y_Mat_lagt2*V_ssa;
%f2
tf3=Y_Mat_lagt3*V_ssa;
%f2
tf4=Y_Mat_lagt4*V_ssa;
%f3
tf5=Y_Mat_lagt5*V_ssa;
%f3
tf6=Y_Mat_lagt6*V_ssa;
%f4
tf7=Y_Mat_lagt7*V_ssa;
%f4
tf8=Y_Mat_lagt8*V_ssa;
%f5
tf9=Y_Mat_lagt9*V_ssa;
%f5
tf10=Y_Mat_lagt10*V_ssa;
%f6
tf11=Y_Mat_lagt11*V_ssa;
%f6
tf12=Y_Mat_lagt12*V_ssa;
%f7
tf13=Y_Mat_lagt13*V_ssa;
%f7
tf14=Y_Mat_lagt14*V_ssa;
%f8
tf15=Y_Mat_lagt15*V_ssa;
%f8
tf16=Y_Mat_lagt16*V_ssa;
tf17=Y_Mat_lagt17*V_ssa;
tf18=Y_Mat_lagt18*V_ssa;
tf19=Y_Mat_lagt19*V_ssa;
tf20=Y_Mat_lagt20*V_ssa;
tf21=Y_Mat_lagt21*V_ssa;
tf22=Y_Mat_lagt22*V_ssa;
tf23=Y_Mat_lagt23*V_ssa;
tf24=Y_Mat_lagt24*V_ssa;
tf25=Y_Mat_lagt25*V_ssa;
tf26=Y_Mat_lagt26*V_ssa;
tf27=Y_Mat_lagt27*V_ssa;
tf28=Y_Mat_lagt28*V_ssa;
tf29=Y_Mat_lagt29*V_ssa;
tf30=Y_Mat_lagt30*V_ssa;
tf31=Y_Mat_lagt31*V_ssa;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Graficos%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%base line and helathy
samples%%%%%%%%%%%%%%%%%%%%%%%%%%5
figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
hold on
```

```
plot(t3(1:400,1),t3(1:400,11),'rp');
hold on
plot(t2(1:400,1),t2(1:400,11),'c*');
legend('Base Line(c1,c2)','healthy test3(c1,c2)','healthy
test2(c1,c2)');
xlabel('C1 (s1)');
ylabel('C2 (s2)');
title('Base line - Healthy Data');


%%%%%%%%%%%%%%%%%%%%%%%%Base line and faulty samples
%%%%%%%%%%%%%%%%%%%%%%
figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
hold on
plot(tf1(1:400,1),tf1(1:400,11),'kp');
hold on
plot(tf2(1:400,1),tf2(1:400,11),'y*');
legend('Base Line(c1,c2)','faulty test1(c1,c2)','faulty
test2(c1,c2)');
xlabel('C1 (s1)');
ylabel('C2 (s2)');
title('Base line - Faulty Data');


%%%%%%%%%%%%%%%%%%%%Base line and another faulty
samples%%%%%%%%%%%%%%%%%
figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
hold on
plot(tf3(1:400,1),tf3(1:400,11),'kp');
hold on
plot(tf4(1:400,1),tf4(1:400,11),'y*');
legend('Base Line(c1,c2)','faulty test3(c1,c2)','faulty
test4(c1,c2)');
xlabel('C1 (s1)');
ylabel('C2 (s2)');
title('Base line - Faulty Data 2');

%%%%%%%%%%%%%%%%%% Baseline, helthy and faulty data
%%%%%%%%%%%%%%%%%%%%%%%

figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
hold on
plot(t3(1:400,1),t3(1:400,11),'r.');
hold on
plot(tf1(1:400,1),tf1(1:400,11),'cp');
hold on
plot(t2(1:400,1),t2(1:400,11),'r*');
hold on
```

```
plot(tf2(1:400,1),tf2(1:400,11),'ch');
hold on
plot(tf3(1:400,1),tf3(1:400,11),'cp');
hold on
plot(tf4(1:400,1),tf4(1:400,11),'c<');
legend('Base Line(c1,c2)','healthy red(c1,c2)','faulty
cyan(c1,c2)');
xlabel('C1 (s1)');
ylabel('C2 (s2)');
title('Base line - Healthy Data - Faulty Data');


%%%%%%%%%%%%%%%%%% Baseline, helthy and faulty data 2
%%%%%%%%%%%%%%%%%%%%%%

figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
hold on
plot(t9(1:400,1),t9(1:400,11),'r.');
hold on
plot(tf7(1:400,1),tf7(1:400,11),'cp');
hold on
plot(t10(1:400,1),t10(1:400,11),'r*');
hold on
plot(tf8(1:400,1),tf8(1:400,11),'ch');
hold on
plot(tf10(1:400,1),tf10(1:400,11),'cp');
hold on
plot(tf12(1:400,1),tf12(1:400,11),'c<');
legend('Base Line(c1,c2)','healthy red(c1,c2)','faulty
cyan(c1,c2)');
xlabel('C1 (s1)');
ylabel('C2 (s2)');
title('Base line - Healthy Data - Faulty Data 2');

%%%%%%%%%%%%%%%%%% Baseline and eigth faulty data
%%%%%%%%%%%%%%%%%%
figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
hold on
plot(tf1(1:400,1),tf1(1:400,11),'ro');
hold on
plot(tf3(1:400,1),tf3(1:400,11),'m.');
hold on
plot(tf5(1:400,1),tf5(1:400,11),'y*');
hold on
% hold on
plot(tf6(1:400,1),tf6(1:400,11),'g+');
hold on
plot(tf9(1:400,1),tf9(1:400,11),'r>');
hold on
```

```
plot(tf11(1:400,1),tf11(1:400,11),'k<');
hold on
plot(tf13(1:400,1),tf13(1:400,11),'yp');
hold on
plot(tf14(1:400,1),tf14(1:400,11),'ch');
legend('Base Line(sensor B3)','F1(high air content in
oil)','F2(pump wear)','F3(hydraulic leakage)','F4(sensor
Wg)','F5(sensor B3)','F6(sensor B3)','F7(sensor
B3)','F8(sensor Tc)');
xlabel('C1 (s1)');
ylabel('C2 (s2)');
title('Base line - 8 Faults');

%%%%%%%%%%%%%%%only base line with you delayed
matrix%%%%%%%%%%%%%%%%
figure
plot(Base_line(1:400,1),Base_line(1:400,11),'b+');
legend('Base Line(c7,8)');
xlabel('C7 (s7)');
ylabel('C8 (s8)');
title('Base line');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%

%%%indice the Fisher base line %%%%%%%%%%%%%%%%%%%%
w1=[Base_line(1:400,1),Base_line(1:400,121)];

%%% fisher vector healthy samples %%%%%%%%%%%%
wh=[t(1:400,1),t(1:400,11)];
wh16=[t16(1:400,1),t16(1:400,11)];
wh1=[t1(1:400,1),t1(1:400,11)];
wh17=[t17(1:400,1),t17(1:400,11)];
wh2=[t2(1:400,1),t2(1:400,11)];
wh18=[t18(1:400,1),t18(1:400,11)];
wh3=[t3(1:400,1),t3(1:400,11)];
wh19=[t19(1:400,1),t19(1:400,11)];
wh4=[t4(1:400,1),t4(1:400,11)];
wh20=[t20(1:400,1),t20(1:400,11)];
wh5=[t5(1:400,1),t5(1:400,11)];
wh21=[t21(1:400,1),t21(1:400,11)];
wh6=[t6(1:400,1),t6(1:400,11)];
wh22=[t22(1:400,1),t22(1:400,11)];
wh7=[t7(1:400,1),t7(1:400,11)];
wh23=[t23(1:400,1),t23(1:400,11)];
wh8=[t8(1:400,1),t8(1:400,11)];
wh24=[t24(1:400,1),t24(1:400,11)];
wh9=[t9(1:400,1),t9(1:400,11)];
wh25=[t25(1:400,1),t25(1:400,11)];
wh10=[t10(1:400,1),t10(1:400,11)];
wh26=[t26(1:400,1),t26(1:400,11)];
```

```
wh11=[t11(1:400,1),t11(1:400,11)];
wh27=[t27(1:400,1),t27(1:400,11)];
wh12=[t12(1:400,1),t12(1:400,11)];
wh28=[t28(1:400,1),t28(1:400,11)];
wh13=[t13(1:400,1),t13(1:400,11)];
wh29=[t29(1:400,1),t29(1:400,11)];
wh14=[t14(1:400,1),t14(1:400,11)];
wh30=[t30(1:400,1),t30(1:400,11)];
wh15=[t15(1:400,1),t15(1:400,11)];
wh31=[t31(1:400,1),t31(1:400,11)];

%%%% fisher vector faulty samples;
wf=[tf(1:400,1),tf(1:400,11)];
wf16=[tf16(1:400,1),tf16(1:400,11)];
wf1=[tf1(1:400,1),tf1(1:400,11)];
wf17=[tf17(1:400,1),tf17(1:400,11)];
wf2=[tf2(1:400,1),tf2(1:400,11)];
wf18=[tf18(1:400,1),tf18(1:400,11)];
wf3=[tf3(1:400,1),tf3(1:400,11)];
wf19=[tf19(1:400,1),tf19(1:400,11)];
wf4=[tf4(1:400,1),tf4(1:400,11)];
wf20=[tf20(1:400,1),tf20(1:400,11)];
wf5=[tf5(1:400,1),tf5(1:400,11)];
wf21=[tf21(1:400,1),tf21(1:400,11)];
wf6=[tf6(1:400,1),tf6(1:400,11)];
wf22=[tf22(1:400,1),tf22(1:400,11)];
wf7=[tf7(1:400,1),tf7(1:400,11)];
wf23=[tf23(1:400,1),tf23(1:400,11)];
wf8=[tf8(1:400,1),tf8(1:400,11)];
wf24=[tf24(1:400,1),tf24(1:400,11)];
wf9=[tf9(1:400,1),tf9(1:400,11)];
wf25=[tf25(1:400,1),tf25(1:400,11)];
wf10=[tf10(1:400,1),tf10(1:400,11)];
wf26=[tf26(1:400,1),tf26(1:400,11)];
wf11=[tf11(1:400,1),tf11(1:400,11)];
wf27=[tf27(1:400,1),tf27(1:400,11)];
wf12=[tf12(1:400,1),tf12(1:400,11)];
wf28=[tf28(1:400,1),tf28(1:400,11)];
wf13=[tf13(1:400,1),tf13(1:400,11)];
wf29=[tf29(1:400,1),tf29(1:400,11)];
wf14=[tf14(1:400,1),tf14(1:400,11)];
wf30=[tf30(1:400,1),tf30(1:400,11)];
wf15=[tf15(1:400,1),tf15(1:400,11)];
wf31=[tf31(1:400,1),tf31(1:400,11)];


%%%Compute fisher index %%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%primer set de datos con falla          %%%%segundo set de
datos con falla
indicef(1,1)=distancia(w1,wf);
indicef(17,1)=distancia(w1,wf16);
indicef(2,1)=distancia(w1,wf1);
indicef(18,1)=distancia(w1,wf17);
indicef(3,1)=distancia(w1,wf2);
indicef(19,1)=distancia(w1,wf18);
indicef(4,1)=distancia(w1,wf3);
indicef(20,1)=distancia(w1,wf19);
indicef(5,1)=distancia(w1,wf4);
indicef(21,1)=distancia(w1,wf20);
indicef(6,1)=distancia(w1,wf5);
indicef(22,1)=distancia(w1,wf21);
indicef(7,1)=distancia(w1,wf6);
indicef(23,1)=distancia(w1,wf22);
indicef(8,1)=distancia(w1,wf7);
indicef(24,1)=distancia(w1,wf23);
indicef(9,1)=distancia(w1,wf8);
indicef(25,1)=distancia(w1,wf24);
indicef(10,1)=distancia(w1,wf9);
indicef(26,1)=distancia(w1,wf25);
indicef(11,1)=distancia(w1,wf10);
indicef(27,1)=distancia(w1,wf26);
indicef(12,1)=distancia(w1,wf11);
indicef(28,1)=distancia(w1,wf27);
indicef(13,1)=distancia(w1,wf12);
indicef(29,1)=distancia(w1,wf28);
indicef(14,1)=distancia(w1,wf13);
indicef(30,1)=distancia(w1,wf29);
indicef(15,1)=distancia(w1,wf14);
indicef(31,1)=distancia(w1,wf30);
indicef(16,1)=distancia(w1,wf15);
indicef(32,1)=distancia(w1,wf31);


%%%%primer set de datos saludables      %%%%segundo set de
datos saludables
indiceh(1,1)=distancia(w1,wh);
indiceh(17,1)=distancia(w1,wh16);
indiceh(2,1)=distancia(w1,wh1);
indiceh(18,1)=distancia(w1,wh17);
indiceh(3,1)=distancia(w1,wh2);
indiceh(19,1)=distancia(w1,wh18);
indiceh(4,1)=distancia(w1,wh3);
indiceh(20,1)=distancia(w1,wh19);
indiceh(5,1)=distancia(w1,wh4);
indiceh(21,1)=distancia(w1,wh20);
indiceh(6,1)=distancia(w1,wh5);
indiceh(22,1)=distancia(w1,wh21);
```

```
indiceh(7,1)=distancia(w1,wh6);
indiceh(23,1)=distancia(w1,wh22);
indiceh(8,1)=distancia(w1,wh7);
indiceh(24,1)=distancia(w1,wh23);
indiceh(9,1)=distancia(w1,wh8);
indiceh(25,1)=distancia(w1,wh24);
indiceh(10,1)=distancia(w1,wh9);
indiceh(26,1)=distancia(w1,wh25);
indiceh(11,1)=distancia(w1,wh10);
indiceh(27,1)=distancia(w1,wh26);
indiceh(12,1)=distancia(w1,wh11);
indiceh(28,1)=distancia(w1,wh27);
indiceh(13,1)=distancia(w1,wh12);
indiceh(29,1)=distancia(w1,wh28);
indiceh(14,1)=distancia(w1,wh13);
indiceh(30,1)=distancia(w1,wh29);
indiceh(15,1)=distancia(w1,wh14);
indiceh(31,1)=distancia(w1,wh30);
indiceh(16,1)=distancia(w1,wh15);
indiceh(32,1)=distancia(w1,wh31);


%%% Final Designation

muestras_a_clasificar=[indiceh,indicef];
limit=length(muestras_a_clasificar);
aux=1;
for i=0:2:2

    for j=1:limit

            if muestras_a_clasificar(j,aux)<1.5
                detection(j,i+2)=1;%healthy

detection(j,i+1)=muestras_a_clasificar(j,aux);
            else detection(j,i+2)=0;%faulty

detection(j,i+1)=muestras_a_clasificar(j,aux);
            end
    end
aux=1+aux;
end

detection

time_final=cputime;

total_time=time_final-time_inicio
```

7.1.1. Function Fisher Criteria

```
function j=distancia(x1,x2)

%N1=length(x1);
%N2=length(x2);
%u1=[sum(x1(:,1)),sum(x1(:,2))]*1/N1;
%u2=[sum(x2(:,1)),sum(x2(:,2))]*1/N2;

u1=mean(x1);
u2=mean(x2);
s1=cov(x1);
s2=cov(x2);
sw=s1+s2;
SB=(u1-u2)*(u1-u2)';

invSw=inv(sw);
invSw_by_SB=invSw*SB;
[V,D]=eig(invSw_by_SB);
w=V(:,1);

dis=(w'*SB*w)/(w'*sw*w);
j=dis;
```