CHAPTER 2:  
 
Volume I 
Report –Budget- Annex 
 
TREBALL DE FI DE GRAU 
 
“Fault detection in wind turbines using PCA and statistical hypothesis testing” 
 
TFG presentat per obtenir el títol de GRAU en ENGINYERIA MECÀNICA 
Per Josep Mª Serrahima de Cambra 
 
Barcelona, 8 de Juny de 2016 
 
Director: Francesc Pozo Montero 
Departament Matemàtica Aplicada III (MAIII) 
Universitat Politècnica de Catalunya (UPC)
TABLE OF CONTENTS

Table of contents ........................................................................................................... i
Report - Table of contents .............................................................................................. 3
List of Figures .................................................................................................................. 6
List of tables .................................................................................................................... 7
Resum ............................................................................................................................... 8
Resumen ........................................................................................................................... 8
Abstract ............................................................................................................................ 9
Acknowledgements ........................................................................................................ 10

CHAPTER 1: Introduction ............................................................................................... 11

CHAPTER 2: Wind power and wind turbines ................................................................. 13
  2.1. Wind power ........................................................................................................... 13
  2.2. Wind turbines ....................................................................................................... 15
    2.2.1. Main parts of a WT ......................................................................................... 15
    2.2.2. Location ......................................................................................................... 16
  2.3. Structural health monitoring ................................................................................ 17
    2.3.1. Control of wind turbines .............................................................................. 17

CHAPTER 3: Reference wind turbine ............................................................................ 20
  3.1. Reference WT ....................................................................................................... 20
    3.1.1. Wind modelling ............................................................................................ 22
    3.1.2. Generator-converter actuator model and pitch actuator model ................. 22
  3.2. Fault description ................................................................................................. 23
  3.3. Simulated files ..................................................................................................... 25

CHAPTER 4: Simple inference ....................................................................................... 26
  4.1. Theoretical foundations ....................................................................................... 26
    4.1.1. Principal Component Analysis ..................................................................... 26
    4.1.2. Statistical significance .................................................................................. 27
  4.2. Mathematical model ............................................................................................ 28
    4.2.1. Data driven baseline modeling based on PCA ........................................... 28
    4.2.2. Group scaling .............................................................................................. 29
    4.2.3. Fault detection based on Hypothesis testing .............................................. 32
4.2.4. Test of equality of means ................................................................. 34
4.3. Simulations ..................................................................................... 36
   4.3.1. Type I and Type II errors ............................................................. 39
   4.3.2. Sensitivity and specificity ............................................................. 39
   4.3.3. Optimum or desired results .......................................................... 40
   4.3.4. Receiver Operating Characteristic curves ..................................... 41
4.4. Results ............................................................................................. 42
4.5. Detection time reduction analysis ..................................................... 44
   4.5.1. Size of the sample ....................................................................... 45
   4.5.2. Time instants per row ................................................................. 51
4.6. Sensor selection ............................................................................... 55
4.7. Fault detection with a reduced number of sensors and a reduced number of time instants .......................................................... 57
   4.7.1. 6 sensors and L=50 ....................................................................... 57
   4.7.2. 6 sensors and L=25 ....................................................................... 58

CHAPTER 5: Simple inference in MATLAB ........................................... 59
   5.1.1. Load the healthy and faulty samples ............................................. 60
   5.1.2. Group scaling .............................................................................. 61
   5.1.3. PCA ............................................................................................ 61
   5.1.4. Statistical hypothesis testing ....................................................... 61
   5.1.5. ROC curve .................................................................................. 62

CHAPTER 6: Multivariate inference ......................................................... 64
   6.1. Mathematical Model ...................................................................... 65
      6.1.1. Group scaling and PCA ............................................................. 65
      6.1.2. Multivariate test ...................................................................... 65
   6.2. Simulations .................................................................................... 68
   6.3. Results ........................................................................................... 69

CHAPTER 7: Multiple inference in MATLAB ......................................... 74
   7.1.1. Creation of population mean vector and sample mean vector ... 74
   7.1.2. Creation of the sample covariance matrix .................................. 75
   7.1.3. Hotelling’s $T^2$ ....................................................................... 75

CHAPTER 8: Simple vs Multiple inference ............................................. 77
   8.1. For a fixed significance ................................................................. 78
   8.2. For a range of significances ............................................................ 79
      8.2.1. First score of the simple inference ........................................... 79
      8.2.2. Second and third scores in simple inference ......................... 80
8.2.3. Multiple inference with s=2 ............................................... 81
8.2.4. Multiple inference with s=7 ............................................... 82
8.2.5. Multiple inference with s=12 ............................................. 83
8.2.6. Total effectivity .............................................................. 84

CHAPTER 9: Conclusions ................................................................ 87

CHAPTER 10: Bibliography ............................................................. 90
  10.1. Bibliographic references ...................................................... 90
  10.2. General bibliography .......................................................... 90

Budget ......................................................................................... 95
  Software costs ............................................................................ 95
  Engineering costs ..................................................................... 95
  Total budget ............................................................................. 96

Annex ........................................................................................... 99
  A.1. Basic program simple inference ......................................... 99
  A.2. Variation of the size of the sample (Simple inference) ........ 101
  A.3. Time instants per row (simple inference) ............................ 106
  A.4. Basic program multiple inference ....................................... 111
“Fault detection in wind turbines using PCA and statistical hypothesis testing”

TFG presentat per obtenir el títol de GRAU en ENGINYERIA MECÀNICA
Per Josep Mª Serrahima de Cambra

Barcelona, 8 de Juny de 2016

Director: Francesc Pozo Montero
Departament Matemàtica Aplicada III (MAIII)
Universitat Politècnica de Catalunya (UPC)
REPORT - TABLE OF CONTENTS

List of Figures ........................................................................................................... 6
List of tables ................................................................................................................. 7
Resum ............................................................................................................................. 8
Resumen ......................................................................................................................... 8
Abstract .......................................................................................................................... 9
Acknowledgements ....................................................................................................... 10

CHAPTER 1: Introduction .......................................................................................... 11

CHAPTER 2: Wind power and wind turbines ......................................................... 13
  2.1. Wind power ........................................................................................................ 13
  2.2. Wind turbines ..................................................................................................... 15
    2.2.1. Main parts of a WT ..................................................................................... 15
    2.2.2. Location ..................................................................................................... 16
  2.3. Structural health monitoring ............................................................................. 17
    2.3.1. Control of wind turbines .......................................................................... 17

CHAPTER 3: Reference wind turbine ..................................................................... 20
  3.1. Reference WT ..................................................................................................... 20
    3.1.1. Wind modelling ......................................................................................... 22
    3.1.2. Generator-converter actuator model and pitch actuator model ..... 22
  3.2. Fault description .................................................................................................. 23
  3.3. Simulated files .................................................................................................... 25

CHAPTER 4: Simple inference ............................................................................... 26
  4.1. Theoretical foundations ..................................................................................... 26
    4.1.1. Principal Component Analysis .................................................................. 26
    4.1.2. Statistical significance .............................................................................. 27
  4.2. Mathematical model ........................................................................................... 28
    4.2.1. Data driven baseline modeling based on PCA ...................................... 28
    4.2.2. Group scaling ............................................................................................ 29
    4.2.3. Fault detection based on Hypothesis testing .......................................... 32
    4.2.4. Test of equality of means ......................................................................... 34
  4.3. Simulations ......................................................................................................... 36
4.3.1. Type I and Type II errors ........................................ 39
4.3.2. Sensitivity and specificity ........................................ 39
4.3.3. Optimum or desired results ................................... 40
4.3.4. Receiver Operating Characteristic curves ................. 41
4.4. Results ........................................................................ 42
4.5. Detection time reduction analysis ................................ 44
  4.5.1. Size of the sample ................................................ 45
  4.5.2. Time instants per row .......................................... 51
4.6. Sensor selection ......................................................... 55
4.7. Fault detection with a reduced number of sensors and a reduced number of time instants .............................................................. 57
  4.7.1. 6 sensors and L=50 ............................................. 57
  4.7.2. 6 sensors and L=25 ............................................ 58

**CHAPTER 5: Simple inference in MATLAB** ........................ 59
  5.1.1. Load the healthy and faulty samples ....................... 60
  5.1.2. Group scaling .................................................... 61
  5.1.3. PCA ............................................................... 61
  5.1.4. Statistical hypothesis testing ................................. 61
  5.1.5. ROC curve ....................................................... 62

**CHAPTER 6: Multivariate inference** .................................. 64
  6.1. Mathematical Model ............................................... 65
    6.1.1. Group scaling and PCA ..................................... 65
    6.1.2. Multivariate test ............................................. 65
  6.2. Simulations ........................................................... 68
  6.3. Results ................................................................. 69

**CHAPTER 7: Multiple inference in MATLAB** ....................... 74
  7.1.1. Creation of population mean vector and sample mean vector ...... 74
  7.1.2. Creation of the sample covariance matrix .................. 75
  7.1.3. Hotelling’s $T^2$ .................................................. 75

**CHAPTER 8: Simple vs Multiple inference** .......................... 77
  8.1. For a fixed significance ......................................... 78
  8.2. For a range of significances ..................................... 79
    8.2.1. First score of the simple inference ....................... 79
    8.2.2. Second and third scores in simple inference .......... 80
    8.2.3. Multiple inference with $s=2$ ............................. 81
    8.2.4. Multiple inference with $s=7$ ............................ 82
8.2.5. Multiple inference with s=12 .......................................................... 83
8.2.6. Total effectivity .................................................................................. 84

CHAPTER 9: Conclusions ................................................................................. 87

CHAPTER 10: Bibliography .............................................................................. 90

10.1. Bibliographic references ................................................................. 90
10.2. General bibliography ................................................................. 90
LIST OF FIGURES

Figure 1 Wind farm (Source: DNV-GL).................................................................................. 14
Figure 2 Parts of a wind turbine (Source: Office of Energy efficiency and renewable energy)................................................................................................. 15
Figure 3 Yaw system (Source: International Energy Agency) ............................................. 15
Figure 4 Real control of wind turbines (Source: [1]) .......................................................... 18
Figure 5 Speed signal with 10% turbulence (Source [1]) ................................................... 22
Figure 6 Example of PCA ..................................................................................................... 27
Figure 7 Acceptation and rejection areas .......................................................................... 28
Figure 8 (A) Baseline projection in the 3 first principal components, and (B) Baseline and faults 1, 4 and 7 .......................................................................................... 33
Figure 9 Process of diagnosis ............................................................................................. 37
Figure 10 PCA and statistical testing ................................................................................... 38
Figure 11 Example of a ROC curve .................................................................................... 41
Figure 12 ROC curve .......................................................................................................... 43
Figure 13 Specificity as a function of the size of the sample............................................. 47
Figure 14 Specificity and sensitivity of the first score as a function of the size. 48
Figure 15 Specificity and sensitivity of the: (A) second score, (B) third score, (C) fourth score ..................................................................................................................... 49
Figure 16 ROC curves for (A) \(v_{10} = 50\), (B) \(v_{8} = 40\), (C) \(v_{6} = 30\), (D) \(v_{4} = 20\) and (E) \(v_{2} = 10\) .................................................................................................................... 50
Figure 17 Specificity and sensitivity as a function of the number of time instants (L) ................................................................................................................................. 53
Figure 18 Sensitivity and specificity of the second score as a function of L....... 53
Figure 19 ROC curves as a function of L: (A) L=500, (B) L=400, (C) L=300, (D) L=200 and (E) L=100 ........................................................................................................... 54
Figure 20 MATLAB screenshot .......................................................................................... 59
Figure 21 Specificity for different multiple inferences as a function of the significance ................................................................................................................................. 72
Figure 22 Sensitivity as a function of the significance ......................................................... 73
Figure 23 First score as a function of the significance......................................................... 79
Figure 24 Second and third scores as a function of the significance ......................... 80
Figure 25 Multiple inference with s=2 as a function of the significance ........... 81
Figure 26 Multiple inference with s=7 as a function of the significance ........... 82
Figure 27 Multiple inference with s=12 as a function of the significance ........... 83
Figure 28 Simple and multiple inferences as a function of the significance...... 85
LIST OF TABLES

Table 1 Properties of the wind turbine ................................................................. 21
Table 2 Available sensors .................................................................................... 21
Table 3 Fault types ............................................................................................... 23
Table 4 Organization of simulated files ............................................................... 25
Table 5 Example of table of results ...................................................................... 39
Table 6 Specificity and sensitivity ....................................................................... 40
Table 7 Optimum results ..................................................................................... 40
Table 8 Optimum results of the first four scores .................................................. 40
Table 9 Results ................................................................................................... 42
Table 10 Sensitivity and specificity .................................................................... 43
Table 11 Number of comparisons for each size of the sample ......................... 46
Table 12 Results as a function of the size of the sample .................................... 46
Table 13 Results for the different number of time instants ............................... 52
Table 14 Six sensors with more information ....................................................... 56
Table 15 Results with 6 sensors .......................................................................... 56
Table 16 Effectivity with six sensors .................................................................. 57
Table 17 Results with six sensors and L=50 ....................................................... 58
Table 18 Effectivity (sensitivity-specificity) with six sensors and L=50 .......... 58
Table 19 Results when using six sensors and L=25 ........................................... 58
Table 20 Example of the table of results ............................................................ 69
Table 21 Results when we pick the s=2 first principal components ................. 70
Table 22 Results when we pick the s=7 first principal components ................. 70
Table 23 Results when we pick the s=12 first principal components ............... 71
Table 24 Comparative results when $\alpha = 0.1$ .................................................... 78
RESUM

L’augment de la mida dels aerogeneradors per la generació d’electricitat i la seva construcció en llocs remots per maximitzar la producció suposa un augment en costos de manteniment i operació. Per tal de reduir aquests costos, eliminar manteniments programats i millorar la seguretat, apareix la necessitat de sistemes de control a distància. *Structural health monitoring* és el procés d’implantació d’una estratègia de detecció de fallades a l’estructura. Aplicat als aerogeneradors, fins i tot en condicions de vent canviants és necessària la detecció de dany. La primera part del projecte millora una metodologia prèviament aplicada als aerogeneradors (inferència estadística simple) reduint el temps de detecció; el segon mètode aplicat utilitza la inferència múltiple per detectar dany. Ambdós mètodes són provats per 24 mostres d’aerogeneradors en diferents condicions (sanes i danyades), i els resultats són encoratjadors: utilitzant la inferència simple el temps de detecció és reduït fins obtenir una detecció gairebé instantània; alhora, aquest projecte serveix com a prova pilot amb la inferència múltiple utilitzada per la detecció de dany en aerogeneradors, amb una correcta diagnòstic d’estructures sanes i danyades.

RESUMEN

El aumento de tamaño de los aerogeneradores para generar electricidad y su construcción en lugares remotos para maximizar la producción supone un aumento en los costes de mantenimiento y operación. Para reducir estos costes, eliminar mantenimientos programados y mejorar la seguridad, aparece la necesidad de sistemas de control a distancia. *Structural health monitoring* es el proceso de implementación de una estrategia de detección de fallos para una estructura o sistema, es decir, detección en línea de fallos en la estructura. Aplicado a los aerogeneradores, incluso con condiciones cambiantes de viento es necesaria la detección de daño. La primera parte del proyecto mejora una metodología previamente aplicada a los aerogeneradores (inferencia estadística simple) reduciendo el tiempo de detección; el segundo método aplicado utiliza la inferencia múltiple para detectar fallos. Ambos métodos son probados por 24 muestras de aerogeneradores en diferentes condiciones (sanas y dañadas), y los resultados son alentadores: utilizando inferencia simple el tiempo de detección es reducido para obtener detección casi instantánea; simultáneamente, este proyecto funciona como una prueba piloto con la inferencia múltiple usada para la detección de daño en aerogeneradores, con una diagnóstico correcta de estructuras sanas y dañadas.
ABSTRACT

The increase in size of wind turbines (WT) to generate electricity and its construction in remote places to maximize the production has led to high maintenance and operation costs. In order to reduce these costs, avoid scheduled maintenance and improve safety considerations, there is a need of a distant monitoring system. Structural health monitoring is the process of implementing a fault (or damage) detection strategy for a structure or system, that is, the online detection of faults on a structure. When applied to wind turbines, even with changing wind conditions damages must be detected. The first part of this project improves a methodology that had previously been applied to WT (statistical simple inference) reducing the damage detection time; the second method applied uses multivariate inference to detect faults. Both methods are tested with a set of 24 data samples of WT in different healthy and faulty conditions, and the results in each case are encouraging: using simple inference the detection time can be reduced to get almost instant damage detection; simultaneously this project works as a pilot test with the multivariate inference used in the damage detection of wind turbines, with also correct diagnose of healthy and faulty WT.
ACKNOWLEDGEMENTS

First of all I want to thank my project’s director, Francesc Pozo, who has helped me and guided me to do this project in the best way possible. Moreover, I want to thank him for considering that my results were good enough to be part of an article, and allowing me to coauthor it with him. Also, I would like to thank Yolanda Vidal, who provides me with the data samples, and also coauthors the article with Francesc and myself.

I want to thank as well all the Departament de Matemàtica Aplicada, for allowing me to do this project with them; and to finish with, all the professors I have had through all four years of my degree, who have prepared me for this project and the years to come.
CHAPTER 1: INTRODUCTION

Electric energy consumption is the usage of electricity as a form of energy. The consumption of electricity all around the globe has not stopped from increasing since the Industrial Revolution. The main sources from where this electricity has been historically generated are fossil fuels, like coal or petroleum, hydraulic and more recently nuclear energy.

Using fossil fuels as a source of energy enhances the greenhouse effect, as well as having a price dependency of the countries that extract these products. As a result, since the 1970s the World has realized that there is a necessity of clean energy to prevent the increase on the average temperature, and that is when renewable energies began to increase its importance. There are many sources: wind, direct sun radiation, waves of the sea... but 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.

There has been a lot of investment regarding the technological advance of wind turbines. Hence, wind turbines are now larger, with really high towers and long blades to maximize the generation of electricity; also, they are placed in the best locations to generate electricity all day long. However, these improvements on the size and placement have a number of disadvantages.

First of all, the height of the tower hinders the inspections and the maintenance work, as the danger of accidents increases. Also, maintenance costs that large wind turbines have are higher as they increase its size, so there must be a way to prevent damages before the reparations are too expensive. In addition to all of this, new wind farms are built in the windiest areas, in order to maximize the generation, and this implies building them in remote areas, and even in the middle of the sea, as with the offshore wind turbines. Then, there is a new challenge that is the monitoring of the turbines.

In order to reduce maintenance costs and the loss of generated electricity while a turbine is stopped, there is a need of a system that allows control of wind
turbines in the distance. Structural health monitoring is the process of implementing a damage detection strategy for a structure or system, that is, the online detection of faults on a structure.

Structural health monitoring (SHM) of wind turbines allows distant and reliable damage detection in order to avoid catastrophic failures in the system, while it allows online control that can improve the life-cycle of the turbine, while reducing the maintenance expenses and prevent unnecessary inspections or replacements. Additionally, new turbines have lots of different sensors that collect data from all around the turbine, which makes available the needed data for its control. Then, it is a topic of interest in the near future, and it must be studied deeply.

The objective of this project is to study the SHM of wind turbines. However, this is a really complicated study: the excitation of wind turbines, that is, the source of the movement of the blades of a turbine is the wind that is constantly changing, and its turbulences are different in each moment. The method developed must be capable of detecting the failures even with changing wind conditions, in order to make detection of damage for real situations.

This project has several parts. First of all there is a description of how large wind turbines work and their main parts; after this section a deeper definition on SHM, its properties and the application on wind turbines is done. After these descriptive sections there is the application of SHM to wind turbines.

Two different methodologies are discussed, developed and applied to wind turbines. Both of them use the data collected by the sensors of the turbine, use statistical methods (PCA and hypothesis testing) but have many differences between them. The main first step is the creation of a pattern of a known turbine working under healthy and normal conditions, which will thereafter be compared with the turbine to diagnose. Then, there are 24 different data samples of turbines working under healthy or faulty conditions to test if the method works.

First there is an application of the simple or univariate inference. This method has been previously applied to wind turbines [1]. Firstly, the PCA extracts the most relevant information from the big data, and then there is statistical comparison with the healthy pattern in order to classify the turbine as healthy or faulty. Afterwards, this project follows a second purpose on this method: reduction of the detection time, in order to make the diagnosis time as short as possible to make an almost online detection.

The second method that is applied is the multiple or multivariate inference. As opposed to the simple inference, this method had only been applied to simple structures with constant excitations, and as it is known, wind is constantly changing, and thus the excitation is varying all the time. Hence, this is a pilot test to study the detection of damage using the multivariate inference. The result of this test is also to diagnose a turbine as healthy or faulty.

The last section of this project is the comparison of results and the differences between the two detection methodologies exposed before, the simple and multiple inferences.
CHAPTER 2: WIND POWER AND WIND TURBINES

Renewable energy is defined as the energy that is collected from resources that are naturally replenished and clean. For example wind, the energy from the sun and geothermal heat, among many others, are renewable energies.

2.1. Wind power

The International Energy Agency defines the wind energy as the kinetic energy of the wind exploited for electricity generation in wind turbines. The movement of the wind works as the source of energy that is thereafter converted to electricity by the movement of the blades of a wind turbine and its generator. Then, wind power is a plenty renewable energy, as well as distributed all around the globe, is clean and therefore produces no greenhouse emissions during its operation, and reduces the fossil fuels dependency.

The Global Wind Energy Council, in its annual publication of global wind statistics from 2015 states that the worldwide global installed wind power capacity is 432MW, and it has been increasing almost at a nonstop rhythm since 2000, when the total installed wind capacity was 17MW. The use of the wind power to generate electricity is considered to be one of the most important in the near future, because of their clean and almost unfinishable source, and must be studied and improved to maximize the generation. Therefore, as the evolution of the installed capacity has been increasing and will continue increasing, it is a
necessity to explain how turbines work, how they generate energy and how to control them in order to avoid premature breakdowns or damages.

To study wind turbines it is necessary to understand how they work. Wind is a variation on the atmospheric pressure that creates a circulation of large masses of air. It is associated to high and low pressures and is generated at really large scales. These big masses of moving air are the ones that have the enough kinetic energy to move the blades of turbines, and then this movement is used to generate electricity in wind turbines.

The generation of electricity directly depends on the wind’s direction and velocity. Large big turbines need a minimum approximated velocity of 5 m/s to work properly and begin generating electricity. However, there is also a maximum wind speed, that is, the cut-out speed, when the turbine stops from working in order to avoid possible breaks in the blades or the transmission parts. The direction is also a really important factor in the generation of electricity, as turbines need to be facing the wind in order to obtain the maximum of the kinetic energy.

The turbulence, that is always associated to a moving fluid, is counter-productive to the generation of electricity, as it creates vibrations and tensions that can be damaging for the structural integrity. These turbulences are created mostly because of the ground around the turbine. Then, the higher the wind is, the lesser turbulences are, and that is why WT are located in isolated flat places, with really high towers.
2.2. Wind turbines

There are two main wind turbines types: horizontal and vertical axis WT. In this 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.

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. Large scale wind turbines usually have three blades.

![Figure 2 Parts of a wind turbine](Source: Office of Energy efficiency and renewable energy)

2.2.1. Main parts of a WT

The main parts of commercial large horizontal-axis wind turbines are the following ones:

- In order to be always facing the wind’s direction, WT have an wind vane that controls the direction of the wind, and a servomotor or yaw motor, which allows a 360° turn so that the hub always faces the wind (Figure 3). This ensures the maximum production of electricity, as when the turbine faces the wind the maximum kinetic energy is transmitted the rotor.

- Large industrial wind turbines have three blades

![Figure 3 Yaw system](Source: International Energy Agency)
connected to the hub, which together make up the rotor, which convert the kinetic energy of the wind to rotation energy that can be used to generate electricity. Blades have an aerodynamic profile, which is studied to obtain the maximum energy from the wind and at the same time control the aerodynamic forces that generate undesired tensions.

- The rotation energy obtained by the blades is transmitted by a shaft, called low-speed shaft, as the angular velocity is low (30-60 rotations per minute), to the gearbox, which converts the low speeds to high rotational speed of around 1000-1800 rpm. The gearbox is one of the most expensive parts of the system.

- In order to obtain a constant velocity in the generator there is a pitch system. The pitch angle is the angle that the blades have when facing the wind, as the variation of the angle of the blades changes the rotational velocity of the rotor. Furthermore, when the wind velocity is too high, the pitch angle is set to the angle of minimum absorption of energy of the wind, and therefore stopping the turbine from working.

- Additionally, there is a braking system that stops the rotor mechanically, electrically or hydraulically in emergencies.

- The nacelle is the “big box” that contains the gearbox, low and high-speed shafts, generator, controllers and brake.

- The last part of the WT is the tower, which supports the structure of the turbine, and gives the necessary height for the wind to be constant, powerful and with the minimum turbulence.

The controller of the turbine is formed by the computer and systems that collect data from the sensors, organize it, and allow the startup and stop of the turbine, among other functions.

2.2.2. Location

Wind turbines are usually organized in wind farms to generate electricity altogether. These farms can be on land or at the sea:

Land-based wind turbines refer to the group of turbines that generate electricity when they are based in the mainland. This term can also be referred as onshore. Most of the installed wind farms in Europe are based in-land nowadays. Onshore wind turbines are the most used, as the construction of the tower and the reparations are easier than sea-based towers.

Offshore wind energy is the energy generated by wind turbines deployed in the sea. Turbines in the sea take advantage of better wind resource, as the wind is not disturbed by mountains or man-built structures that are inland. Then, they work for more hours and therefore generate more electricity. The installation can be in floating structures or with towers in not really profound seas. Construction and maintenance costs are much more expensive than in land farms, but production rate is higher. That’s when control of structures plays a major role.
2.3. Structural health monitoring

The process of implementing a damage detection strategy for aerospace, civil and mechanical engineering systems is referred as structural health monitoring. It is necessary then to define what damage or fault is.

As stated by Farrar and Worden [2], a **fault** can be defined as intentional or unintentional change to the material and/or geometric properties of a system, including changes to the boundary conditions and system connectivity, which adversely affect the current or future performance of this system.

Then, **failure** occurs when the damage progresses to a point where the system no longer can perform its intended function. Often failure is defined as in terms of exceeding some strength, stability or deformation-related performance criterion.

These two words describe the main purpose of this project. Most engineered systems with some fault or some kind of damage in its structure can often continue to perform its intended purpose for a while, but usually the performance is at some reduced level. However, this reduction in the performance can affect the overall function, as for example a stuck pitch in one of the blades in a wind turbine still allows the production in electrical energy, but can derive to a broken blade- which is a failure and therefore forces its stop.

Then, an early detection of faults avoids a major repair. Structural health monitoring is the technology that allows maintenance of systems based on a sensing system on the structure that monitors it and notifies if there is a damage or degradation. This kind of control provides with safety and economic benefits, as it ensures the structural healthy and avoids unexpected failures, while improving turbine’s reliability and reducing maintenance costs by detecting faults before they convert into failures, and also by eliminating scheduled maintenance.

The aim of this project is the creation of a method that detects the faults in wind turbines before the problem is too big and needs big reparations.

Nevertheless, there is a big difference between the usual fault detection procedures that are used in many of the diagnosis cases of shells, beams or simple structures [2], when a healthy structure is excited by a signal to create a pattern. Then, the new structure to diagnose is excited by the same signal and the dynamic response is compared with the pattern. This is known as guided waves in structures for structural health monitoring. Structural control of wind turbines does not follow a procedure of guided waves.

2.3.1. Control of wind turbines

Traditionally, condition monitoring systems would focus on the control of the main bearing, generator and gearbox to detect failures, as they are the most costly parts of the whole system. However, as we are going to use a method to detect more failures, not just the ones on these specific parts, we are going to use more information than what was needed before.
Nowadays, wind turbines have an enormous number of sensors that collect all kinds of data to support its operation. All turbines permanently collect big data from hundreds of sensors, from gearbox oil temperature to stresses in the blade root. All control actions use the data collected by this sensors as inputs, and with the increase in complexity of the amount of data available, there is a need on clever processing of these data.

As said in the previous section, most structural health monitoring problems are solved by guided waves, when the same excitation is applied to the healthy structure, from where a pattern is created, and the structure to diagnose, and its result is compared with the pattern. However, the excitation of wind turbines is the wind and its turbulence. Therefore, guided waves cannot be applied, as the excitation signal is never constant. Then, the **fault detection strategy used in wind turbines** states that, **even with different wind conditions, the test is able to detect if there is some fault** or damage. This is visually shown in Figure 4.

![Figure 4 Real control of wind turbines (Source: [1])](image)

Most industrial wind turbines are manufactured with an integrated system that can control various turbine parameters. These monitored data are collated and stored via a Supervisory Control and Data Acquisition (SCADA) system that archives the information in a convenient manner. However, as the sensors are collecting data every few instants, accumulations of data can easily be produced, and therefore its analysis can be really complicated. Too much information stored can mean difficulties to study it, and thus problems on the control of failures and damages. That is why a system of online and automatic detection is needed.

SCADA systems are almost always integrated in the control systems of the wind turbines, and therefore there is no need of new sensor’s addition in the system. Then, the system studied in this project can be applied in most of the industrial wind turbines that are found worldwide.

Then, among others, the main benefits or having a control or fault detection system in a wind turbine are as follows [4]:

- **Supervision at remote sides and remote diagnosis**: large turbines are usually built in remote sites, both onshore and offshore. They may not be accessible for parts of the year, and then a wrong decision of a reparation or scheduled maintenance can result in big economic losses. Then, it is
necessary that WTs have a fault detection system that can alert remotely if there is any problem.

- **Avoidance of premature breakdown**: prevent catastrophic failures and secondary effects, as the entire turbine is monitored.

- **Reduced maintenance cost**: inspection interval can be increased with online inspection.

- **Improvement of capacity factor**: with early warning of impending failures, repair action can be taken during low wind seasons and hence will not affect the capacity factor.

- **Support for further development of a turbine**: the data can be used to improve designs for future turbines.

Then, for these explained reasons, having a structural control of wind turbines improves significantly the life expectancy of turbines, and therefore must be studied in detail.
CHAPTER 3: REFERENCE WIND TURBINE

In this project the data used is not from a real-life installed and already functioning wind turbine. As there is a lack of contact between my project and a real distributor of wind turbines in Catalonia that could provide me with real data, a simulation is used.

3.1. Reference WT

Nevertheless, the purpose of this project is neither the simulation of the wind turbine, nor the development or study of the parameters explained in this section. CoDALab provides with the data of the already simulated turbine, with all these parameters already used. Therefore, this section just provides background of where the data that will be used in the control of the WT (the purpose of this project) comes from.

A numerical simulation of an onshore WT (a turbine prepared to be placed on the ground inland) is used, which simulates a large wind turbine that is representative of a typical utility-scale multimegawatt turbine. The simulation has been done via software:

- FAST software, by the National Renewable Energy Laboratory (NREL), from the U.S. Department of Energy.
- It is a CAE tool for simulating the coupled dynamic response of wind turbines. FAST joins aerodynamics models, hydrodynamics models for offshore structures, control and electrical system (servo) dynamics models, and structural (elastic) dynamics models to enable coupled
nonlinear aero-hydro-servo-elastic simulation in the time domain. The FAST tool enables the analysis of a range of wind turbine configurations, including two- or three-blade horizontal-axis rotor, pitch or stall regulation, rigid or teetering hub, upwind or downwind rotor, and lattice or tubular tower. The wind turbine can be modeled on land or offshore on fixed-bottom or floating substructures. FAST is based on advanced engineering models--derived from fundamental laws, but with appropriate simplifications and assumptions, and supplemented where applicable with computational solutions and test data. (This information can be found in the main website of the NREL.)

Our simulated wind turbine is a conventional three-bladed, upwind, variable-speed, variable blade-pitch-to-feather-controlled turbine of 5 MW. This simulated wind turbine has measures that commercial WT have nowadays. All the information about the simulated turbine is in the following table.

**Table 1 Properties of the wind turbine**

<table>
<thead>
<tr>
<th>Reference Wind Turbine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated power</td>
</tr>
<tr>
<td>Number of blades</td>
</tr>
<tr>
<td>Rotor/Hub diameter</td>
</tr>
<tr>
<td>Hub height</td>
</tr>
<tr>
<td>Cut-in, rated, cut-out Wind speed</td>
</tr>
<tr>
<td>Rated generator speed</td>
</tr>
<tr>
<td>Gearbox ratio</td>
</tr>
</tbody>
</table>

The data collected from the sensors of this wind turbine is the same as it would be in any commercial WT. The available data from our sensors is the following:

**Table 2 Available sensors**

<table>
<thead>
<tr>
<th>Number</th>
<th>Sensor type</th>
<th>Symbol</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Generated electrical power</td>
<td>$P_{e,m}$</td>
<td>kW</td>
</tr>
<tr>
<td>2</td>
<td>Rotor speed</td>
<td>$\omega_{r,m}$</td>
<td>Rad/s</td>
</tr>
<tr>
<td>3</td>
<td>Generator speed</td>
<td>$\omega_{g,m}$</td>
<td>Nm</td>
</tr>
<tr>
<td>4</td>
<td>Generator torque</td>
<td>$\tau_{c,m}$</td>
<td>deg</td>
</tr>
<tr>
<td>5</td>
<td>First pitch angle</td>
<td>$\beta_{1,m}$</td>
<td>deg</td>
</tr>
<tr>
<td>6</td>
<td>Second pitch angle</td>
<td>$\beta_{2,m}$</td>
<td>deg</td>
</tr>
<tr>
<td>7</td>
<td>Third pitch angle</td>
<td>$\beta_{3,m}$</td>
<td>m/s²</td>
</tr>
<tr>
<td>8</td>
<td>Fore-aft acceleration at tower bottom</td>
<td>$a_{f,a,m}^b$</td>
<td>m/s²</td>
</tr>
<tr>
<td>9</td>
<td>Side-to-side acceleration at tower bottom</td>
<td>$a_{ss,m}^b$</td>
<td>m/s²</td>
</tr>
<tr>
<td>10</td>
<td>Fore-aft acceleration at mid-tower</td>
<td>$a_{f,a,m}^m$</td>
<td>m/s²</td>
</tr>
<tr>
<td>11</td>
<td>Side-to-side acceleration at mid-tower</td>
<td>$a_{ss,m}^m$</td>
<td>m/s²</td>
</tr>
<tr>
<td>12</td>
<td>Fore-aft acceleration at tower top</td>
<td>$a_{f,a,m}^c$</td>
<td>m/s²</td>
</tr>
<tr>
<td>13</td>
<td>Side-to-side acceleration at tower top</td>
<td>$a_{ss,m}^c$</td>
<td>m/s²</td>
</tr>
</tbody>
</table>
This table represents the order of the data received, as the numeration of the sensors will follow this distribution in the entire project.

3.1.1. Wind modelling

The simulation has used the FAST design code with the TurbSim stochastic inflow turbulence tool [1]. Both codes together allow driving simulations of advanced turbine designs with simulated inflow turbulence environments that incorporate many of the important fluid dynamic features known to adversely affect the wind turbine.

The TurbSim tool allows creating a turbulence model with intensity set to 10%, with logarithmic profile wind type, mean speed set to 18.2 m/s (with values in a range from 12.91 m/s up to 22.57 m/s) and simulated at hub height. The roughness factor is set to 0.01 m.

Each sample simulated is run with a different wind data set. An example of a wind speed signal with the turbulence intensity set to 10% is Figure 5.

3.1.2. Generator-converter actuator model and pitch actuator model

The generator-converter and the pitch actuators are modeled apart from the embedded FAST code, with the objective to ease the model of different type of faults on these parts of the wind turbine.

On the one hand, the generator-converter can be modeled by a first order differential system:
\[
\frac{\tau_r(s)}{\tau_c(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}}
\]

In this equation \(\tau_r\) and \(\tau_c\) are the real generator torque and its reference (given by the controller) respectively, and we set \(\alpha_{gc} = 50\) \cite{1}. The power produced by the generator can be modelled by:

\[
P_c(t) = \eta_g \omega_g(t) \tau_r(t)
\]

where \(\eta_g\) is the efficiency of the generator, and \(\omega_g\) is the generator speed. The efficiency used is \(\eta_g = 0.98\).

On the other hand, the three pitch actuators are modeled as a second order linear differential equation, where pitch angle \(\beta_1(t)\) and its reference \(u(t)\) (given by the collective pitch controller):

\[
\frac{\beta_1(s)}{u(s)} = \frac{\omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2}
\]

In the last equation \(\omega_n\) and \(\xi\) are the natural frequency and the damping ratio, respectively. In the fault free case these values are \(\omega_n = 11.11\) rad/s and \(\xi = 0.6\).

### 3.2. Fault description

The definition of a fault is done in Section 2.3.

The faults that are going to be considered in this project are the ones explained in \cite{1}. These faults cover different parts of the wind turbine, different fault types and classes, and different levels of severity.

<table>
<thead>
<tr>
<th>Fault</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pitch actuator</td>
<td>Change in dynamics: high air content in oil ((\omega_n = 5.73, \text{rad/s}, \xi = 0.45))</td>
</tr>
<tr>
<td>2</td>
<td>Pitch actuator</td>
<td>Change in dynamics: pump wear ((\omega_n = 7.27, \text{rad/s}, \xi = 0.75))</td>
</tr>
<tr>
<td>3</td>
<td>Pitch actuator</td>
<td>Change in dynamics: hydraulic leakage ((\omega_n = 3.42, \text{rad/s}, \xi = 0.9))</td>
</tr>
<tr>
<td>4</td>
<td>Generator speed sensor</td>
<td>Scaling (gain factor equal to 1.2)</td>
</tr>
<tr>
<td>5</td>
<td>Pitch angle sensor</td>
<td>Stuck (fixed value equal to 5 deg)</td>
</tr>
<tr>
<td>6</td>
<td>Pitch angle sensor</td>
<td>Stuck (fixed value equal to 10 deg)</td>
</tr>
<tr>
<td>7</td>
<td>Pitch angle sensor</td>
<td>Scaling (gain factor equal to 1.2)</td>
</tr>
<tr>
<td>8</td>
<td>Torque actuator</td>
<td>Offset (offset value equal to 2000 Nm)</td>
</tr>
</tbody>
</table>

Faults in the pitch actuator are considered in the hydraulic system, which result in changed dynamics due to either a high air content in oil (fault 1) or a drop in
pressure in the hydraulic supply system due to pump wear (fault 2) or hydraulic leakage (fault 3).

Pump wear (fault 2) is an irreversible slow process over the years that result in low pump pressure. As this wear is irreversible, the only possibility to fix it is to replace the pump which will happen after pump wear reaches a certain level. Meanwhile, the pump will still be operating and the system dynamics is slowly changing, while the turbine structure should be able to withstand the effects of this fault. Pump wears after approximately 20 years if operation might result in pressure reduction to 75% of the rated pressure, which is reflected by the faulty natural frequency and a fault damping ratio \( \omega_n = 5.73 \frac{\text{rad}}{\text{s}}, \xi = 0.45 \), respectively.

Hydraulic leakage (fault 3) is another irreversible incipient fault, but it is introduced considerably faster than the pump wear. When this fault reaches a certain level, system reparation is necessary, and if the leakage is too fast, it will lead to a pressure drop and the preventive procedure is then to shut down the turbine before the blade is stuck in an undesired position. The fast pressure drop is easily detected and requires immediate reaction, because if the hydraulic pressure if too low, the hydraulic system will not be able to move the blades, which will cause the actuator to stick in its current position, resulting in blade seize.

On the contrary, high air content in the oil (fault 1) is an incipient reversible process, which means that the air content in the oil may disappear without any necessary repair to the system. The nominal value of the air content in the oil is 7%, whereas the high air content in the oil represents a 15%.

The generator speed measurement is done using encoders. The gain factor fault (fault 4) 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.

Faults in the pitch positions are important. The origin of these faults is either electrical or mechanical and it can result in either a fixed value (faults 5 and 6) or a changed gain factor (fault 7) on the measurements. In particular, the fixed value fault should be easily detected and therefore it is important that a fault detection, isolation and accommodation scheme can deal with this fault. If it is not handled correctly, these faults will influence the pitch reference position as the pitch controller is based on these pitch position measurements.

Finally, the last fault is a converter torque offset. It is difficult to detect this fault internally (by the electronics of the converter controller). Yet, from a wind turbine level, it can be detected, isolated and accommodated, as it changes the torque in the wind turbine power train.

These descriptions of the most common faults in wind turbines are important in this project, as these are the faults that are going to be tried to detect. However, it is necessary to point out that in this project there is no isolation of each fault, but there is only detection of a faulty state. Then, the result of each diagnose of a structure is to catalog the turbine as healthy or faulty.
3.3. Simulated files

As it is already stated, the files of data that are used in this project come from a simulated wind turbine. Then, all the files come from the FAST software, and it is necessary to explain how the files are organized in order to be able to study them.

All the data files are organized in MATLAB files (*.mat), with * being the name of the file. The files consist of a table will the data collected by the 13 sensors explained in Table 2.

The simulated files follow the next structure:

Table 4 Organization of simulated files

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>p_{e,m}</th>
<th>\omega_{r,m}</th>
<th>\omega_{g,m}</th>
<th>\tau_{c,m}</th>
<th>\beta_{1,m}</th>
<th>\beta_{2,m}</th>
<th>\beta_{3,m}</th>
<th>a_{r,g,m}^b</th>
<th>a_{s,s,m}^b</th>
<th>a_{r,a,m}^m</th>
<th>a_{s,s,m}^m</th>
<th>a_{r,a,m}^t</th>
<th>a_{s,s,m}^t</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\Delta t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 \cdot \Delta t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 \cdot \Delta t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>600 s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The files are organized following this table: the first column represents the time vector, that is, the instant when the data from each sensor is saved. It begins at 0s and has an increase of \( \Delta t \) seconds until the last value, which is 600 seconds. Then, there are the other 13 columns that represent the 13 sensors whose information is going to be used.

In this project 24 simulations are used:

- 16 simulated files or samples of healthy turbines, with different wind conditions
- 8 simulated faulty files, each of them with one of the faults explained in the previous section.
4.1. Theoretical foundations

Before explaining the mathematical model that is followed in this project it is necessary to explain what the different processes that are used are. Then, in this section there is an explanation on what PCA and significance from the statistical hypothesis testing are.

4.1.1. Principal Component Analysis

Principal component analysis (PCA) is a standard method of multivariate statistics used in many different types of problems. The objective of the PCA is to obtain relevant information from confusing data sets.

The PCA’s objective is to reveal information from data sets. This is achieved by reducing a complex data set to a lower dimension to reveal the structure that lies beneath it. The PCA algorithm seeks to project, by linear transformation, the data into a new $p$-dimensional set of Cartesian coordinates $z_1, z_2, \ldots, z_p$ called the principal components or scores.
The new coordinates have the following property, which is why this method is used: the first principal component $z_1$ is the linear combination of the original set of data with the maximal variance; the second score $z_2$ is the linear combination that has the maximum of the remaining variance, and so on. Therefore, the first score has the maximum information of the original data set; the second score has less information than the first one, but more than the third, and so on.

Image 6 represents a two dimensional data set, with feature 1 and feature 2 as axis references. The orange points are the data obtained from the sensors that collect them. As it can be seen, the projection of the points into these two axes is a number of points. Then, if a PCA is used, we obtain two scores, represented as well in the figure. The first score is the blue arrow that represents the maximal variance, which has a direction (the eigenvector) and a magnitude (eigenvalue). The second score is the red arrow that represents the second maximal variance. Then, the PCA reduces the original data set to the two most important directions, with the magnitude that represents its variance. These directions can be understood as a pattern: the data set has its maximum variability in the directions of the score 1, the second maximum variability in the direction of the second principal component, and so on in cases with more dimensions.

If a PCA strategy is used in a data set that is known to be healthy, or undamaged, then a pattern is obtained. It can be afterwards compared with other data sets to see if these have similar principal components or not and therefore be able to compare them.

4.1.2. Statistical significance

Statistical significance ($\alpha$) plays a major role in statistical hypothesis testing, as it is used to determine whether a null hypothesis should be rejected or retained.
Type I errors are bound to happen sometimes. Statistical term of significance is the term that determines how often these errors occur. It is a value that is decided by the person who makes the hypothesis testing. In the figure above, it can be seen that we draw the line that separates the acceptance and the rejection regions in the significance that has been chosen. Then, for a value if it falls in the acceptance region we accept the hypothesis as true, but if it falls in the rejection region we can do nothing but to reject it.

Then, the probability of rejecting a hypothesis when in fact it is true (Type I error, explained furtherly in Section 4.3.1) is the same value as the significance. The smaller $\alpha$ is, the greater the significance of the test.

The previous image exemplifies how a significance of $\alpha = 5\%$ works. As it is a two-tailed test, the rejection section is partitioned to both ends of the sampling distribution, and the addition of both makes up the total percentage.

### 4.2. Mathematical model

This mathematical model is based on the article “Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing” by Francesc Pozo and Yolanda Vidal [1].

#### 4.2.1. Data driven baseline modeling based on PCA

First of all, to start the PCA methodology we need the data measured from the wind turbine organized in a matrix form. The data collected from a sensor on a period of time $(n \cdot L - 1)\Delta t$ seconds, where $\Delta t$ is the sampling time, $n$ are the number of healthy experiments, $L$ are the number of time instants studied and $n, L \in \mathbb{N}$. The discrete measurements of the sensor form a real vector,

$$ (x_{11} \ x_{12} \ \cdots \ x_{1L} \ x_{21} \ x_{22} \ \cdots \ x_{2L} \ \cdots \ x_{n1} \ x_{n2} \ \cdots \ x_{nL}) \in \mathbb{R}^{nL} \quad (4) $$
where the real number $x_{ij}, i = 1, ..., n, j = 1, ..., L$ corresponds to the measure of the sensor at time $(i - 1)L + (j - 1)\Delta t$ seconds. This collected data can be then rearranged in matrix form as follows:

$$
\begin{pmatrix}
    x_{11} & x_{12} & \cdots & x_{1L} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{n1} & x_{n2} & \cdots & x_{nL}
\end{pmatrix} \in M_{n \times L}(\mathbb{R})
$$

This matrix $M_{n \times L}(\mathbb{R})$ is the vector space of $n \times L$ matrices over $\mathbb{R}$. In this matrix the information is distributed: each row represents all the values measured in a given experiment, while each column represents a different time instant.

However, we do not have just one sensor. If instead of having one sensor we have $N \in \mathbb{R}$ also collecting data during $(n \cdot L - 1)\Delta t$ seconds, and the sensed data from each sensor is arranged as explained in equation (2), we can create the new matrix $X \in M_{n \times (N \cdot L)}(\mathbb{R})$ as follows:

$$
X = \begin{pmatrix}
    x_{11} & x_{12} & \cdots & x_{1L} \\
    x_{11} & x_{12} & \cdots & x_{1L} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{11} & x_{12} & \cdots & x_{1L} \\
    x_{n1} & x_{n2} & \cdots & x_{nL}
\end{pmatrix} = (X^1 | X^2 | \cdots | X^N)
$$

In this matrix, the super index $k = 1, ..., N$ of each element $x_{ij}^k$ represents the number of a given sensor.

The objective of the principal component analysis is to find a linear transformation orthogonal matrix $P \in M_{(N \cdot L) \times (N \cdot L)}(\mathbb{R})$, an orthogonal matrix is a matrix such that $P \cdot P^T = P^T \cdot P = I \Rightarrow P^T = P^{-1}$, that will be used to project the original data matrix $X$ according to the next matrix product:

$$
T = XP \in M_{n \times (N \cdot L)}(\mathbb{R})
$$

Where $T$ is a matrix that has a diagonal covariance matrix.

### 4.2.2. Group scaling

To understand this methodology it is necessary to recall what we are calculating. The data from matrix $X$ comes from a number of different sensors, that can have different ranges, scales and magnitudes, and what is more important, come from the wind, that is neither constant nor empty of turbulence. Then, we need to do a preprocessing before the calculus begins. Therefore, we need to do a rescaling of the data before using the data as a hole.
In this project, the standardization is going to be used as the group scaling procedure. As stated by Farrar and Worden [2], standardization is a mandatory step before PCA, as it eliminates the possibility that some score will be dominant just because its coordinates have larger amplitude. This step makes the value in the data have a zero-mean (or mean-centered, when subtracting the mean in the enumerator, proof in equation (9)) and unit-variance.

Then, we will subtract at each number the mean of all measurements in the same column in $X$, and divide by the standard deviation of the measurements for each sensor. More precisely:

$$\mu_j^k = \frac{1}{n} \sum_{i=1}^{n} x_{ij}^k, \quad j = 1, ..., L$$

$$\mu^k = \frac{1}{nL} \sum_{i=1}^{n} \sum_{j=1}^{L} x_{ij}^k$$

$$\sigma^k = \sqrt{\frac{1}{nL} \sum_{i=1}^{n} \sum_{j=1}^{L} (x_{ij}^k - \mu^k)^2}$$

These last three formulae represent: $\mu_j^k$ the mean of the measures placed in the same column, that is, the mean of the $n$ measures of sensor $k$ in matrix $X^k$ at time $((i-1)L + (j-1))\Delta t$ seconds. The mean of all elements in sensor $k$ is represented by $\mu^k$; and the last parameter, $\sigma^k$, is the standard deviation of all the measures of sensor $k$.

The group scaling has the purpose of creating a new matrix $\tilde{X}$ that contains the same information as the original matrix, but without values that represent physical magnitudes with its different scales and units. This new matrix is formed by the scaled values of the previous one by following the next equation:

$$\tilde{x}_{ij}^k = \frac{x_{ij}^k - \mu_j^k}{\sigma^k}, \quad i = 1, ..., n, j = 1, ..., L, k = 1, ..., N$$

With the scaled values we create the new matrix. However, from now on in this project the scaled matrix will be called simply $X$. We call it this simpler way, as all the information from the original matrix is still in this new matrix and this is the matrix we are going to be using from now on.

As mentioned before, the mean of each column vector of the new matrix can be computed:
Fault detection in wind turbines using PCA and statistical hypothesis testing

\[
\frac{1}{n} \sum_{i=1}^{n} \hat{x}_{ij}^k = \frac{1}{n} \sum_{i=1}^{n} \frac{x_{ij}^k - \mu_j^k}{\sigma_j^k} = \frac{1}{n\sigma_j^k} \sum_{i=1}^{n} (x_{ij}^k - \mu_j^k) = \frac{1}{n\sigma_j^k} \left[ \sum_{i=1}^{n} (x_{ij}^k - n\mu_j^k) \right]
\]

\[
= \frac{1}{n\sigma_j^k} (n\mu_j^k - n\mu_j^k) = 0
\]

Since the scaled matrix is a \( X \) mean-centered matrix, it is possible to calculate its covariance matrix as follows:

\[
C_X = \frac{1}{n-1} X^T X \in M_{(N \cdot L) \times (N \cdot L)}(\mathbb{R})
\]

If the matrix was not mean-centered it would be much more complicated to calculate the covariance matrix. This covariance matrix is a \((N \cdot L) \times (N \cdot L)\) symmetric matrix that measures the degree of linear relationship within the data set between all possible pair of columns.

The subspaces in PCA are defined by the eigenvalues and the eigenvectors of the covariance matrix as follows:

\[
C_X P = P \Lambda
\]

Where the columns of \( P \in M_{(N \cdot L) \times (N \cdot L)}(\mathbb{R}) \) are the eigenvectors of \( C_X \). Diagonal terms of matrix \( \Lambda \in M_{(N \cdot L) \times (N \cdot L)}(\mathbb{R}) \) are the eigenvalues \( \lambda_i, i = 1, \ldots, N \cdot L \), of \( C_X \), whereas the off-diagonal terms are zero, that is,

\[
\Lambda_{ii} = \lambda_i, i = 1, \ldots, N \cdot L
\]

\[
\Lambda_{ij} = 0, i = 1, \ldots, N \cdot L, i \neq j
\]

The eigenvectors \( p_j, j = 1, \ldots, N \cdot L \), representing the columns of the transformation matrix \( P \) are classified according to the eigenvalues in descending order and they are called the principal components or the loading vectors of the data set. The eigenvector with the highest eigenvalue, called the first principal component, represents the most important pattern in the data with the largest quantity of information.

Matrix \( P \) is called the principal components of the data set or the loading matrix; matrix \( T \) is the transformed or projected matrix to the principal component space, also called score matrix. Using all \( N \cdot L \) principal components, that is, the full dimensional case, the orthogonality of \( P \) implies that \( PP^T = I \), where \( I \) is the \((N \cdot L) \times (N \cdot L)\) identity matrix.

\[
T = XP
\]
The objective of the PCA is to reduce the dimensionality of the data set \( X \). We select only a limited number \( \ell < N \cdot L \) of principal components, that is, only the eigenvectors related to the \( \ell \) highest eigenvalues. Therefore, the reduced \( P \) matrix is:

\[
P = (p_1|p_2| \cdots |p_\ell) \in M_{(N \cdot L) \times \ell}(\mathbb{R})
\]

(17)

The transformed matrix, with only the first \( \ell \) principal components:

\[
\hat{T} = X\hat{P} \in M_{n \times \ell}(\mathbb{R})
\]

(18)

Opposite to \( T \), \( \hat{T} \) is not invertible, as we just choose the first \( \ell \) principal components. Therefore, we cannot recover all the information of the initial matrix if we use the transposed \( \hat{T}^T \), and there will always be some error.

The key point of using the PCA is that initially our information had a physical meaning, being data collected by several sensors. However, after the group scaling and the PCA the scores do not have a physical meaning and are just scores that can be compared and a pattern can be obtained from them.

4.2.3. Fault detection based on Hypothesis testing

The current wind turbine to diagnose is subjected to wind and turbulences. The data measured by the \( N \in \mathbb{N} \) sensors during \((v \cdot L - 1)\Delta t\) seconds constructs a new matrix:

\[
Y = \\
\begin{pmatrix}
  y^1_{i1} & y^1_{i2} & \cdots & y^1_{iL} \\
  \vdots & \vdots & \ddots & \vdots \\
  y^N_{i1} & y^N_{i2} & \cdots & y^N_{iL}
\end{pmatrix}
= (Y^1|Y^2|\cdots|Y^N)
\]

(19)

\[
Y \in M_{\nu \times (N \cdot L)}(\mathbb{R})
\]

The number of rows in the matrix \( Y \), that is the natural number \( \nu \), is not necessarily equal to the number of rows in the \( X \) matrix \( (n) \). However, and this is mandatory, the number of columns of \( Y \) must be equal to that number in \( X \). This fact means that the number of sensors and the number of time instants per row must be maintained.

We have to follow the first step as we did before: the new matrix \( Y \) with information must be projected into the vector space spanned by the eigenvectors in \( \hat{P} \) (equation 11), and must be scaled by the values in equations 5, 6 and 7:
Fault detection in wind turbines using PCA and statistical hypothesis testing

\[ \tilde{y}_{ij}^k = \frac{y_{ij}^k - \mu_{ij}^k}{\sigma_k}, i = 1, ..., n, j = 1, ..., L, k = 1, ..., N \]  \hspace{1cm} (20)

The most important thing is that the values \( \mu_{ij}^k \) and \( \sigma_k \) are the same values that were used to scale \( \tilde{X} \). This way, the new projected matrix is set in the same vector space of the eigenvectors in \( P \) (or in our reduced matrix \( \tilde{P} \)), thus making them comparable.

The projection of the scaled matrix \( \tilde{Y} \) into the spanned space by the eigenvectors in \( \tilde{P} \) follows the following equation:

\[ T_d = \tilde{Y}\tilde{P} \in M_{n \times \ell}(\mathbb{R}) \]  \hspace{1cm} (21)

The matrix to be diagnosed, that is, to find out if the sample is working in a healthy or faulty state is \( T_d \). Its components can be calculated also with the following vector multiplication:

\[ t^i = r^i \cdot \tilde{P} \in \mathbb{R}^\ell \]  \hspace{1cm} (22)

The projections of each row vector \( r^i = \tilde{Y}(i,:) \in \mathbb{R}^{N \times L}, i = 1, ..., \nu \) of matrix \( \tilde{Y} \) into the space spanned by the eigenvectors in \( \tilde{P} \) are all vectors \( t^i \in \mathbb{R}^\ell, i = 1, ..., \nu \). The first component the vector \( t^i \) is called the first principal component or score 1; the second is called second principal component or score 2, and so on.

There are many cases where there can be visual separation between the baseline healthy sample and the damaged structure. However, when we plot the three first principal components of the healthy and damaged samples it is impossible to differentiate them. As it can be seen in the following figures, both healthy and faulty samples are mixed together, so it is impossible to distinguish visually the damaged from the healthy samples.

\[ \text{Figure 8 (A) Baseline projection in the 3 first principal components, and (B) Baseline and faults 1, 4 and 7} \]
There are methods that help to differentiate the healthy from the faulty data sets visually, for example the $Q$ index (also called SPE, square prediction error) and the Hotelling’s $T^2$ index. However, in our wind turbine case there is no possibility of a visual separation of the healthy and the faulty samples. Therefore, the next step is the statistical hypothesis testing.

The turbulent wind is considered a random process, hence vectors $r^i$ and $t^i$ are considered from now on random variables.

4.2.4. Test of equality of means

In Chapter 2.3 all different faults are described. In order to examine whether a sample of data is healthy or faulty, there is still a last step after the ones described in the previous chapter.

We have a PCA model (matrix $\hat{P}$ built in equation (14)) with data that comes from a fully functional healthy wind turbine. Then, for each principal component $j = 1, ..., \ell$ the baseline sample is defined as the set of $n$ real numbers (number of rows at $X$ matrix) are computed as the $j$-th component of the vector multiplication $(i,:) \cdot \hat{P}$. Then, we define the baseline sample as the set of numbers, where $e_j$ is the $j$-th canonical basis:

$$
\tau^i_j = (X(i,:) \cdot \hat{P})(j) = X(i,:) \cdot \hat{P} \cdot e_j, i = 1, ..., n
$$

Similarly, for each principal component $j = 1, ..., \ell$, the sample of the current wind turbine to diagnose is defined as the set of $\nu$ real numbers (this is the number of rows in matrix $\hat{Y}$) computed as the $j$-th component of the $t^i$ vector in equation (19). We then define the sample to diagnose as the set of numbers:

$$
\tau^i_j = t^i \cdot e_j, i = 1, ..., \nu
$$

As the goal of this method is to obtain a fault detection of wind turbines, there must be a comparison. The current sample of the turbine to be diagnosed is compared with the baseline sample that works in a healthy state, and the result is that either a healthy state is detected, or otherwise a fault is detected.

The test of equality of means will be the comparison test. We consider that:

1. The baseline sample is a random sample of random variables having a normal distribution with unknown mean $\mu_X$ and unknown standard deviation $\sigma_X$.

2. The random sample of the current wind turbine is also normally distributed with unknown mean $\mu_Y$ and unknown standard deviation $\sigma_Y$. 
We consider that the variances of both samples are not equal. The problem we consider is whether the means are equal, that is, \( \mu_X = \mu_Y \). This statement leads to the hypothesis of the test:

\[
H_0: \mu_X - \mu_Y = 0 \quad \text{versus} \quad H_1: \mu_X - \mu_Y \neq 0
\]  

(25)

Then, the hypotheses are:

- **The null hypothesis**: the sample of the wind turbine to diagnose is distributed as the baseline sample.
- **Alternative hypothesis**: 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, then the current wind turbine is categorized as healthy. Otherwise, if the null hypothesis is rejected in favor of the alternative hypothesis, then there is an indication of a fault in the wind turbine.

The test is based on the Welch-Satterthwaite method, exposed by Ugarte and Militino [3]. When random samples of size of \( n \) and \( v \), respectively, are taken from two normal distributions \( N(\mu_X, \sigma_X) \) and \( N(\mu_Y, \sigma_Y) \) and the population variances are unknown, the random variable:

\[
W = \frac{(\bar{X} - \bar{Y}) + (\mu_X - \mu_Y)}{\sqrt{\left(\frac{S_X^2}{n} + \frac{S_Y^2}{v}\right)}}
\]  

(26)

This random variable can be approximated with a \( t \)-distribution with \( \rho \) degrees of freedom, that is:

\[
W \sim t_{\rho}
\]  

(27)

The degrees of freedom are calculated with:

\[
\rho = \left\lfloor \frac{(S_X^2/n) + (S_Y^2/v)^2}{(S_X^2/n)/(n-1) + (S_Y^2/v)/(v-1)} \right\rfloor
\]  

(28)

In these last formulae, \( \bar{X} \) and \( \bar{Y} \) are the sample means as random variables; \( S^2 \) is the sample variance as random variable; \( s^2 \) is the variance of a sample; and \( \lfloor \cdot \rfloor \) is the floor function.
The value of the standardized test statistic using this method, that is, the value that we are really looking for, is defined as:

\[
t_{obs} = \frac{\bar{x} - \bar{y}}{\sqrt{\left(\frac{s_x^2}{n} + \frac{s_y^2}{v}\right)}}
\]  \hspace{1cm} (29)

In this last equation, \( \bar{x}, \bar{y} \) are the mean of a particular sample. The quantity \( t_{obs} \) is the fault indicator. We can construct the following test:

\[
|t_{obs}| \leq t^* \Rightarrow \text{Fail to reject } H_0
\]  \hspace{1cm} (30)

\[
|t_{obs}| > t^* \Rightarrow \text{Reject } H_0
\]  \hspace{1cm} (31)

And \( t^* \) is such that:

\[
P(t_p < t^*) = 1 - \frac{\alpha}{2}
\]  \hspace{1cm} (32)

The last parameter in this study is \( \alpha \), which is the chosen risk (significance) level for the test. The significance is the probability that the test procedure will result in a Type 1 error.

As a conclusion, the null hypothesis is rejected if \( |t_{obs}| > t^* \) (this would indicate the existence of a fault in the wind turbine). Otherwise, if \( |t_{obs}| \leq t^* \) there is no statistical evidence to suggest that both samples are normally distributed but with different means, thus indicating no fault in the wind turbine has been found.

4.3. Simulations

The previous chapter explains the mathematical model that is followed to study structures from wind turbines to find if they are in a healthy state or damaged (or in a faulty state). The summary of the process is graphically explained in Figure 9.
However, that is a general explanation without specific information. Then, in this project the concrete information that follows this model is the following.

- First of all, we have a data sample from a healthy, undamaged wind turbine, from which we obtain:
  - \( X \in M_{n \times (N \cdot L)}(\mathbb{R}) \)
  - \( n = 50 \), that is, the size of the sample or number of rows.
  - \( N = 13 \) sensors.
  - \( L = 500 \) time instants, that is, the number of columns per sensor.
  - From the \( X \) matrix we calculate from equations [8, 9, 10] \( \mu_j^k \), \( \mu^k \) and \( \sigma^k \), which respectively are the mean of all elements in the same column, the mean from all elements from the same sensor, and the standard deviation from all elements from each sensor. These values are going to be saved and are going to be the ones that do the group scaling to all samples to be diagnosed.
  - We rescale the \( X \) matrix with the equation [11] with the previous values.
  - The PCA is made from this matrix to obtain \( P \).
  - With equation [16] we calculate \( T \).

The baseline healthy wind turbine is used to obtain a pattern that will be the base from the future statistical comparison to find out the state of the structure to be diagnosed.

Similarly to this baseline sample, we have another group of 24 samples that are going to be studied. They are classified following as follows:

- We have 16 healthy samples, where the wind turbine is working under changing conditions, as the wind is not constant.
- Idem 8 faulty samples. We have one sample for each of the different fault scenarios, or most common damaged turbines, already described/explained in Table 3 in Chapter 3.2.
- All 24 samples are organized forming this matrix:
  o \( Y \in M_{v \times (N \cdot L)}(\mathbb{R}) \)
    - \( v = 50 \), that is, the size of the sample or number of rows.
    - \( N = 13 \) sensors.
    - \( L = 500 \) time instants, that is, the number of columns per sensor.
    - The time step between two measures (that is, two columns) is \( \Delta t = 0.0125 \) seconds.
    - Total time to fulfill the matrix is \( (v \cdot L - 1)\Delta t = 312.4875 \) seconds, arranged as in equation [19].

The objective of this chapter is to compare statistically each of these 24 samples with the pattern created with the baseline healthy sample. Then, the goal is to obtain a total recognition of the faulty and the healthy samples. Graphically, the comparison that is done mathematically for each of the 24 samples is explained in the following figure.

In this figure it can be seen that the original \( X \) matrix, which is formed by the 13 sensors, each of them filled with information to cover 50 rows is rescaled, and from it, using a PCA, the matrix \( P \) is found. This matrix, multiplied by the scaled \( X \) matrix, gives the values that will be compared with the values of the rescaled \( Y \) that is later multiplied by \( P \) as well.

**Figure 10** PCA and statistical testing
After both the baseline sample and the matrix with the information to diagnose have passed the PCA, the fault detection step must be carried out, which is done by statistical simple inference as previously explained in the mathematical model.

4.3.1. Type I and Type II errors

The result of the comparison depends on the number of principals components chosen. In this part of the project the number of principal components used is $\ell = 4$. Then, we will have four different scores with the detection results. As it is known from the PCA, the first component has more information than the second; the second has more information that the third, and so on. Hence, the first score is expected to have better results than the others.

The expected results, that is, the wanted or optimum results would be to have a total recognition of the 16 healthy samples, and the classification of the 8 faulty simulations as faulty. The best way to expose these results from now on is with the following table.

<table>
<thead>
<tr>
<th>Fail to reject $H_0$</th>
<th>Correct decision (1)</th>
<th>Type II error (missing fault) (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reject $H_0$</td>
<td>Type I error (false alarm) (2)</td>
<td>Correct decision (4)</td>
</tr>
</tbody>
</table>

The results are organized in this table as follows: the 16 healthy (undamaged) samples can be classified as healthy in position (1), or damaged in position (2), which is a Type I error, that is, to classify as faulty one sample that is in healthy state. For the 8 faulty samples, they can be classified in position (3), which is a Type II error (classify as healthy a structure that is damaged), or position (4), that is to catalog as faulty a damaged sample.

4.3.2. Sensitivity and specificity

There are two other statistical measures that can study the performance of the test: sensitivity and specificity. The sensitivity, also called the power of the test, is defined as the proportion of samples from the faulty wind turbines that are correctly identified as such. Thus, it can be computed as $1 - \gamma$. The specificity of the test is defined as the proportion of samples from the healthy structure that are correctly identified as healthy. It can be expressed as $1 - \alpha$.

Then, our table of results can be also expressed with the specificity and sensitivity:
Table 6 Specificity and sensitivity

<table>
<thead>
<tr>
<th></th>
<th>Undamaged Sample ($H_0$)</th>
<th>Damaged Sample ($H_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail to reject $H_0$</td>
<td>Specificity ($1 - \alpha$)</td>
<td>False negative rate ($\gamma$)</td>
</tr>
<tr>
<td>Reject $H_0$</td>
<td>False positive rate ($\alpha$)</td>
<td>Sensitivity ($1 - \gamma$)</td>
</tr>
</tbody>
</table>

The parameter that draws the line on whether a sample should or should not be rejected is the significance ($\alpha$). It is worth mentioning that type I errors are frequently considered to be more serious than type II errors. However, in this application, a type II error is related to a missing fault whereas a type I error is related to a false alarm. In consequence, type II errors should be reduced. Therefore a small level of significance of 1%, 5% or even 10% would lead to a reduced number of false alarms but to a higher rate of missing faults. That is the reason of the choice of a level of significance of $\alpha = 36\%$ in the hypothesis test.

4.3.3. Optimum or desired results

The optimum result is the shown in Table 7, where there are no errors in the classification and all samples are correctly categorized. The sum of the two rows in the undamaged samples must sum 16, as there are 16 healthy samples; and the addition of the rows in the damaged samples must add up 8.

Table 7 Optimum results

<table>
<thead>
<tr>
<th></th>
<th>Undamaged Sample ($H_0$)</th>
<th>Damaged Sample ($H_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail to reject $H_0$</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>Reject $H_0$</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>

However, there is not just one result, as there is a set of results for each diagnose. As we pick the first four principal components (scores), the table with the results will have four different columns with results, one per score. Then the optimum result organized in a table would be:

Table 8 Optimum results of the first four scores

<table>
<thead>
<tr>
<th>Score 1 ($H_0$)</th>
<th>Score 2 ($H_0$)</th>
<th>Score 3 ($H_0$)</th>
<th>Score 4 ($H_0$)</th>
<th>Score 1 ($H_1$)</th>
<th>Score 2 ($H_1$)</th>
<th>Score 3 ($H_1$)</th>
<th>Score 4 ($H_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail to reject $H_0$</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>Reject $H_0$</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

However, simulation results are never as good as the optimum ones. The results coming from the simulation are exposed in Chapter 4.4.
4.3.4. **Receiver Operating Characteristic curves**

An additional way to express the results is the Receiver Operating Characteristic curves (or ROC curves). The ROC curve is used in statistics to graphically plot the true positive rate (TPR) and the false positive rate (FPR) at various levels of significance.

The curves represent the TPR against the FPR, with the True Positive Rate being equal to the sensitivity. The Y-axis represents the true positive rate, while the X-axis represents the false positive rate.

The figure in the right exemplifies what a ROC curve is.

To draw a ROC curve we consider 49 levels of significance in the range $\alpha = 0.02, ..., 0.98$ with a difference of 0.02. Then, for each curve there will be 49 points to represent for each of the magnitudes to represent. These points have a pair of numbers such as:

\[(\text{False positive rate, True positive rate}) \in [0,1] \times [0,1] \subset \mathbb{R}^2\]  \(33\)

The optimum result is indicated by a curve that approaches the upper left corner, indicating a higher TPR with an associated FPR. The closer the curve gents to the dotted diagonal line of the ROC space, the less accurate the test is.
4.4. Results

As it has already been explained, in our project there are 24 samples of data organized as follows:

- 16 samples of healthy wind turbines.
- 8 samples of faulty wind turbines, each of them simulated with one of the typical faults.

Following the entire mathematical model explained in the Section 4.2 we obtain the following results.

<table>
<thead>
<tr>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H₀)</td>
<td>(H₁)</td>
<td>(H₀)</td>
<td>(H₁)</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Fail to reject H₀</td>
<td>11</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Reject H₀</td>
<td>0</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

In Table 9 there are the results of the analysis of the 24 samples using PCA and statistical hypothesis testing. We can obtain some conclusions of this analysis:

- The first principal component, or score 1, has 100% of effectivity in the detection:
  - Out of the 16 healthy samples, all 16 are catalogued as healthy structures. The meaning of this is that all samples of structures in normal working state are detected to be working in good conditions.
  - All 8 different faulty samples are identified as damaged, that is, the system detects that these structural samples are not working under normal undamaged conditions.

- The effectivity decreases with each score:
  - Score 2 detects only 12 out of the 16 healthy samples, and there is a Type II error (missing fault).
  - The third principal component detects 11 of the healthy and only 3 of the damaged samples.
  - The fourth score detects only 9 of the 16 healthy samples, with one missing fault.

- This decrease in the effectivity of the scores corresponds with the theory: the PCA extracts more information in the first score than in any of the others; the second principal component has more information than the third, and so on.
Another way to rewrite the previous table with the results is to write as the effectivity in percentage, what is the same as the sensitivity and specificity table:

**Table 10 Sensitivity and specificity**

<table>
<thead>
<tr>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H_0)</td>
<td>(H_1)</td>
<td>(H_0)</td>
<td>(H_1)</td>
</tr>
<tr>
<td>(H_0)</td>
<td>1.00</td>
<td>0.00</td>
<td>0.75</td>
</tr>
<tr>
<td>(H_1)</td>
<td>0.00</td>
<td>1.00</td>
<td>0.25</td>
</tr>
</tbody>
</table>

As it can be inferred from the table, specificity and sensitivity has a 100% in the first principal component, and decreases with the other scores.

Then, we can conclude by saying that the detection of healthy and faulty samples when the samples are formed by \(v = 50\) rows, \(L = 500\) time instants per row, that is, 500 columns; and these rows and columns for each of the \(N = 13\) sensors, is 100% effective with the first score. This effectivity is what we desire for a method, because we just need to detect the faults with the score that holds more information.

Apart from representing the results in tables, there is another way to represent the outcomes in a more visual way. This is a ROC curve. As already explained in Simulations section, a ROC curve studies the overall accuracy of the method, that is, it represents the True positive rate (or sensitivity) against the false positive rate, for 49 values of the significance \((\alpha)\) within the range \([0.02,0.98]\) with a 0.02 jump.

**Figure 12 ROC curve**

Figure 12 represents a ROC curve for the first four principal components, when the analysis is done with the conditions explained above. As it can be clearly seen, score 1 (red line) has amazing results for all the range of the significance. Being the optimum result the upper-left corner, this first score’s line lies on its
corner, showing that for a really big range of significances the first score has a really good relation between trues and false positive rates.

Scores 2 and 4 have acceptable results, while the third score cannot be considered good because it lies beneath the diagonal line almost at all points, with this line being considered the line that divides the plane into satisfactory and unsatisfactory results.

It can be seen that the results are prefect when using the first score, as there is a detection of all the faults, without missing faults or false alarms. Therefore, the detection using PCA and statistical hypothesis testing is a great way to analyze structures of the wind turbines to detect if they are working in normal conditions, or they are working in faulty conditions and damage is about or already produced, and thus needing reparations.

However, in order to fulfil the $Y \in M_{\nu \times (N \cdot L)}(\mathbb{R})$ matrix there is a need to collect data from the sensors during a total time of $(\nu \cdot L - 1)\Delta t = 312.4875$ seconds, as explained in Section 4.3 (Simulations). This is the time needed to create the matrix, and then there is the calculation time. Hence, this process is not completely on-line. Then, the next step is to try to reduce the detection time in order to need less time to detect the faults.

The next sections develop this idea, and try to find the minimum time for a completely effective detection.

### 4.5. Detection time reduction analysis

The process to detect if a structure of a wind turbine is damaged or not is the following: the sensors collect information that is stored in a matrix, then this matrix is scaled, a PCA is done and there is a comparison with statistical hypothesis testing. As all the steps but the first one are calculations, this time is considered to be much inferior to the sensors recollection of time, and will be decreasing as the power of computers increase.

Then, the **detection time** is the time of collection of data to create the matrix to be diagnosed. The goal of this section is to reduce the detection time in order to make it as shorter as possible without losing effectivity on the detection of faults. The detection time follows this equation:

$$Detection\ time = (\nu \cdot L - 1)\Delta t \ [\text{seconds}]$$  \hspace{1cm} (34)

By default, in the previous section we have been using the matrix $Y \in M_{\nu \times (N \cdot L)}(\mathbb{R})$, with a number of rows and columns of:

- $\nu = 50$, that is, the size of the sample or number of rows.
- The number of time instants is $L = 500$, that is, the number of columns per sensor.
With $N = 13$ sensors, that will be maintained in all the section.

As it has already been stated in equation [31], the total time to fill the $Y$ matrix depends on $\nu$ and $L$, as $\Delta t$ is a value that is fixed as the time of sensing all the properties.

Then, in order to reduce the detection time there are three different possibilities:

1. Reduce the number of rows ($\nu$) in $Y$.
2. Reduce the number of columns ($L$).
3. Reduce both $\nu$ and $L$.

The next subsections will work on each of these possibilities. It must be taken into account that the 13 sensors are maintained in all the section.

### 4.5.1. Size of the sample

The first way to study the detection time is the reduction of the size of the sample. The size of the sample has a physical meaning in our system: the information stored in the matrix is the data collected by the sensors, and therefore the smaller the size is, the lesser time the data must be acquired. Therefore, if the size of the sample is reduced, the total diagnosing time is reduced.

We have studied ten different scenarios to analyze the effect of the variation of the size of the sample. These ten scenarios correspond to 10 different values of $\nu$.

$$\nu_1 = 5, \nu_2 = 10, \ldots, \nu_i = 5 \cdot i, \ldots, \nu_{10} = 50$$

Then, we are going to study how a decrease in the number of rows affects the analyzing process, from the initial $\nu = 50$ to $\nu = 5$, with a step of 5.

Taking into account that we have 16 samples of healthy structures, and each sample has originally $\nu = 50$ rows, the total number of rows is:

$$N_{rowH} = 50 \cdot 16 = 800 \text{ rows}$$

Likewise, if we have 8 samples of faulty structures, the total number of rows of faulty samples is:

$$N_{rowF} = 50 \cdot 8 = 400 \text{ rows}$$

Then, the new number of samples to compare and try to detect if they are working under a healthy or faulty state is:

$$\left[ \frac{N_{rowH}}{\nu_i} \right], \text{for healthy samples}$$

$$\left[ \frac{N_{rowF}}{\nu_i} \right], \text{for faulty samples}$$
\( \nu_i = 5, 10, \ldots, 45, 50 \), and \( \lfloor \cdot \rfloor \) being the floor function.

The following table summarizes the number of comparisons for each \( \nu_i \):

**Table 11 Number of comparisons for each size of the sample**

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( \nu_{10} = 50 )</th>
<th>( \nu_{9} = 45 )</th>
<th>( \nu_{8} = 40 )</th>
<th>( \nu_{7} = 35 )</th>
<th>( \nu_{6} = 30 )</th>
<th>( \nu_{5} = 25 )</th>
<th>( \nu_{4} = 20 )</th>
<th>( \nu_{3} = 15 )</th>
<th>( \nu_{2} = 10 )</th>
<th>( \nu_{1} = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy Samples</td>
<td>16</td>
<td>17</td>
<td>20</td>
<td>22</td>
<td>26</td>
<td>32</td>
<td>40</td>
<td>53</td>
<td>80</td>
<td>160</td>
</tr>
<tr>
<td>Faulty Samples</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>11</td>
<td>13</td>
<td>16</td>
<td>20</td>
<td>26</td>
<td>40</td>
<td>80</td>
</tr>
</tbody>
</table>

As it can be seen in table 11, if the number of rows is half of the initial we have the double of analyzed samples (32 healthy samples and 16 faulty samples); if the number of rows per sample is decreased to 5, then we have ten more times of samples to analyze (160 healthy, 80 faulty).

Iteration has been created, where the entire mathematical procedure explained in Section 4.2 is followed. As the number of healthy and faulty samples changes in each iteration, as explained in the previous table, it has no meaning to express the result as the absolute number of good or wrong decisions (for \( \nu_3 = 15 \) there can be maximum of 53 correct decisions in the healthy samples and 26 in the faulty, for example) the results of this section are going to be expressed as percentages with respect its maximum possible outcome.

Then, results of this calculus are expressed in the following table:

**Table 12 Results as a function of the size of the sample**

<table>
<thead>
<tr>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu )</td>
<td>healthy</td>
<td>faulty</td>
<td>healthy</td>
</tr>
<tr>
<td>50</td>
<td>100,00</td>
<td>100,00</td>
<td>75,00</td>
</tr>
<tr>
<td>45</td>
<td>88,24</td>
<td>100,00</td>
<td>76,47</td>
</tr>
<tr>
<td>40</td>
<td>80,00</td>
<td>90,00</td>
<td>65,00</td>
</tr>
<tr>
<td>35</td>
<td>68,18</td>
<td>100,00</td>
<td>68,18</td>
</tr>
<tr>
<td>30</td>
<td>61,54</td>
<td>92,31</td>
<td>65,38</td>
</tr>
<tr>
<td>25</td>
<td>53,13</td>
<td>100,00</td>
<td>62,50</td>
</tr>
<tr>
<td>20</td>
<td>45,00</td>
<td>95,00</td>
<td>70,00</td>
</tr>
<tr>
<td>15</td>
<td>47,17</td>
<td>88,46</td>
<td>54,72</td>
</tr>
<tr>
<td>10</td>
<td>38,75</td>
<td>80,00</td>
<td>65,00</td>
</tr>
<tr>
<td>5</td>
<td>49,38</td>
<td>75,00</td>
<td>59,38</td>
</tr>
</tbody>
</table>
Table 12 shows the results for every value of \( \nu \); however, it is difficult to extract conclusions with the results exposed like this. Then, it is easier to see the results in a graphical way.

*Image 13 is a graphical representation of specificity of the three first principal components. As it has been explained in Section 4.3.2 (Sensitivity and specificity), specificity is defined as the proportion of samples from the healthy structure that are correctly identified as healthy.*

**Figure 13 Specificity as a function of the size of the sample**

Then, this plot represents how the effectivity of the detection of samples working under normal circumstances changes when the number of rows is decreased. Let’s study each case in a different plot to see the effects in each score.
In this image it can be clearly seen that sensitivity and specificity of the first principal component degrade drastically when varying $\nu$. Therefore, there is a direct connection between the correct decisions and the size of the sample. It can be seen that when the size ($\nu$) decreases, the specificity decreases rapidly from its maximum (a 100% effectivity at $\nu_{10} = 50$) to values around 50% when the size is half its initial value. Therefore, the results get worse as soon as the size of the sample decreases from 50. However, the sensitivity (how many of the faults are detected to be damaged) maintains a pretty good effectivity from sizes between 25 and 50, but then decreases to approximately 75% accuracy.

It is known that the first principal component has more information that the other scores. Thus, the other scores should have worse results when the size of the sample is decreased. The results obtained agree with the theory, as it can be seen in the following graphs.

Figure 14 Specificity and sensitivity of the first score as a function of the size
In images 15(a), 15(b) and 15(c) the results from the previous iteration are represented. The decrease in the size affects all scores, but it can be observed better in the second’s score plot (a). Specificity decreases from 75% when the sample has 50 rows to effectivity of 60% when the number of rows is reduced. Sensitivity has two peaks, one that has a 100% of success in the detection in $\nu_9 = 45$ and another in $\nu_5 = 25$, with a sensitivity of 87.5%. However, the general tendency of the line is to decrease. The third and fourth principal components ((b) and (c) respectively) are not representatively, as the effectivity on the detections is averagely of a 60%, therefore having lots of missing faults and false alarms.

**Figure 15** Specificity and sensitivity of the: (A) second score, (B) third score, (C) fourth score
Another way to study the effect of the decrease in the number of rows of the matrix $\mathbf{Y}$ is by using the ROC curves, described in Section 4.3.4 (ROC Curves). In this curves there is a study of the detection for a whole range of significances ($\alpha = 0.02, ..., 0.98$). A ROC curve has been studied for $v_2 = 10, v_4 = 20, v_6 = 30, v_8 = 40$ and $v_{10} = 50$. Just to recall, the optimum result in a Receiver Operating Characteristic curve is that the curve must be in its upper-left corner.

If we examine the ROC curves, in figure 16 (A, B, C, D and E) there is one conclusion that can be easily extracted. When $v_{10} = 50$ the first score has a perfect performance (the red line is always in contact with the upper-left corner and the total true positive rate (there are no errors). Then, when the size of the sample is decreased (images (B), (C), (D) and (E)) it can be appreciated a degradation on all the scores. It can be even more easily seen in the red line (score 1) as it begins in (A) being in the upper-left corner, but there is a separation that increases each time the size changes.

![Figure 16 ROC curves](image)

**Figure 16** ROC curves  
for (A) $v_{10} = 50$,  
(B) $v_8 = 40$, (C) $v_6 = 30$,  
(D) $v_4 = 20$ and (E)  
$v_2 = 10$
There is one clear conclusion to the study of the reduction of the size of the sample to reduce the total time of the detection. This conclusion is that as soon as the size of the sample decreases there is a degradation in the detection process for all scores. Then, if the size of the sample (that is, the number of rows in the original matrix) decreases from $v = 50$ the effectivity rapidly decreases, and therefore making the method non-valid for the detection of faults in wind turbines.

After this procedure, we conclude that the number of rows in the original matrix will be maintained from now on to $v = 50$, in order to prevent a worsening in the detection.

4.5.2. Time instants per row

The second possibility to reduce the time of diagnose is to find the optimum number of columns in the original $\mathbf{Y}$ matrix (equation [19]). Each column represents a different time instant. Therefore, the bigger number of columns the matrix needs, the more time instants the sensor must collect information before processing it. The reduction is made keeping the number of rows to $v = 50$.

To analyze the effect of the overall performance of the fault detection procedure with a reduced number of columns, we will study a total of 19 different scenarios, corresponding to 19 different values of $L$:

\[
L_1 = 5, L_2 = 10, L_3 = 15, \ldots, L_{10} = 50
\]
\[
L_{11} = 100, L_{12} = 150, L_{13} = 200, \ldots, L_{19} = 500
\]

Then, we will be analyzing a decrease of 50 columns from the initial $L_{19} = 500$ until $L_{10} = 50$; then, the decrease will be of 5 columns per iteration.

In order to study this reduction process we will study the same 16 healthy and 8 faulty samples that are explained in section 4.3.

In this study there is a big difference with the previous one (related to the size of the sample): when studying the reduction of time instants per row, the number of samples to study is always maintained to 16 healthy and 8 faulty samples. The reduction implies a smaller matrix (as there are less time instants per row, the number of columns decrease) but there will be no extra samples to study. Therefore, the maximum result for the healthy will be 16, while the maximum available result for the faulty will be 8. However, to make an easier representation the table will be written in a percent way.

The results can be expressed in the following table:
Table 13 Results for the different number of time instants

<table>
<thead>
<tr>
<th>L</th>
<th>Score 1 healthy</th>
<th>Score 1 faulty</th>
<th>Score 2 healthy</th>
<th>Score 2 faulty</th>
<th>Score 3 healthy</th>
<th>Score 3 faulty</th>
<th>Score 4 healthy</th>
<th>Score 4 faulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>100</td>
<td>100</td>
<td>75</td>
<td>88</td>
<td>69</td>
<td>38</td>
<td>56</td>
<td>87.5</td>
</tr>
<tr>
<td>450</td>
<td>100</td>
<td>100</td>
<td>63</td>
<td>88</td>
<td>63</td>
<td>13</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>400</td>
<td>100</td>
<td>100</td>
<td>50</td>
<td>88</td>
<td>63</td>
<td>38</td>
<td>75</td>
<td>37.5</td>
</tr>
<tr>
<td>350</td>
<td>100</td>
<td>100</td>
<td>44</td>
<td>88</td>
<td>69</td>
<td>0</td>
<td>69</td>
<td>62.5</td>
</tr>
<tr>
<td>300</td>
<td>100</td>
<td>100</td>
<td>56</td>
<td>88</td>
<td>75</td>
<td>25</td>
<td>50</td>
<td>75</td>
</tr>
<tr>
<td>250</td>
<td>94</td>
<td>100</td>
<td>38</td>
<td>100</td>
<td>56</td>
<td>13</td>
<td>31</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>94</td>
<td>100</td>
<td>44</td>
<td>100</td>
<td>69</td>
<td>13</td>
<td>31</td>
<td>100</td>
</tr>
<tr>
<td>150</td>
<td>100</td>
<td>100</td>
<td>44</td>
<td>100</td>
<td>75</td>
<td>13</td>
<td>31</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
<td>63</td>
<td>100</td>
<td>56</td>
<td>0</td>
<td>81</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>100</td>
<td>69</td>
<td>100</td>
<td>94</td>
<td>38</td>
<td>69</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>100</td>
<td>100</td>
<td>75</td>
<td>100</td>
<td>94</td>
<td>13</td>
<td>81</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td>100</td>
<td>88</td>
<td>100</td>
<td>94</td>
<td>75</td>
<td>81</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>100</td>
<td>100</td>
<td>94</td>
<td>100</td>
<td>100</td>
<td>38</td>
<td>88</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>25</td>
<td>88</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>88</td>
<td>94</td>
<td>75</td>
<td>63</td>
<td>25</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>13</td>
<td>100</td>
<td>63</td>
<td>100</td>
<td>63</td>
<td>69</td>
<td>12.5</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>25</td>
<td>100</td>
<td>38</td>
<td>81</td>
<td>12.5</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>0</td>
<td>88</td>
<td>88</td>
<td>100</td>
<td>88</td>
<td>94</td>
<td>12.5</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>63</td>
<td>75</td>
<td>75</td>
<td>88</td>
<td>100</td>
<td>94</td>
<td>100</td>
</tr>
</tbody>
</table>

The results can be expressed as well in a graphical way that leads to an easier understanding and also to clearer conclusions.

To begin with, we will use the score that brings better results for our study, the first score. Figure 17 represents the variation of the specificity and sensitivity as a function of the number of columns, that is, the number of columns. The x-axis represents the number of columns, while the vertical axis represents the percentage of effectivity.
This figure shows the behavior of the correct decisions in identifying the healthy and faulty samples, with a zoom in the range of $L = 5, \ldots, 50$ in order to make the visual study clearer. The results shown are extraordinary: if the number of columns is reduced from the original $L = 500$ to $L = 25$ the sensitivity is maintained to 100%, which means that all the faults are detected even if there is a reduction of columns. Specificity (detection of the healthy samples) is maintained to values of 100% as well, except for $L = 250$ and $L = 200$, where there are 6.25% of false alarms, which can still be considered a really good result.

As it is known, the consecutive scores after the first one have less information than the first, hence their results are worse. Figure 18 shows the behavior of the effectivity of the detection of both healthy and faulty samples using the second score. In this plot it can be seen that the second score’s results are nothing like the ones from the first, as the detection is not maintained to 100% in almost none of the points for the specificity, while the specificity has some good rates of detection at $L = 50, \ldots, 250$, but bad
results in all the other values of the number of time instants per row. Then, the second score is not able to distinguish if a sample is working under good or bad conditions, thus it will not be used.

Third and fourth score have both of them worse results than the first and second scores, so they are not presented as figures as they are not worth it.

The ROC curves for the different $L = 100, 200 ..., 500$ are represented below.

![ROC curves](image)

**Figure 19** ROC curves as a function of $L$: (A) $L=500$, (B) $L=400$, (C) $L=300$, (D) $L=200$ and (E) $L=100$

In the previous ROC curves it can be inferred that the red line (always representing the first score) keeps in all five cases a really good performance; scores three and four (blue and pink) have their effectivity decreased and therefore their performance, as expected, is worse than the first.
To summarize the conclusions extracted from this study:

- There is **no direct connection** between the decrease in the number of time instants per row and the specificity and sensitivity. Hence, the detection is maintained to values of 100% in almost all the number of rows from the original 500 to 25.

- It can be also observed that the first principal component has a perfect recognition of the faulty samples of the wind turbine when $L \geq 25$.

- Detection time can be reduced a lot:

  \[
  Detection\ time = (v \cdot L - 1)\Delta t
  \]

  \[
  Detection\ time = (50 \cdot 25 - 1) \times 0.0125 = 15.6\ seconds
  \]

From an original diagnose time of 312.4875 seconds (Section 4.3) we can have a **95% reduction of the time of diagnose** of a sample if we reduce the number of time instants per row in our sample ($L$).

### 4.6. Sensor selection

In the article “On Real-Time Fault Detection in Wind Turbines: Sensor Selection Algorithm and Detection Time Reduction Analysis”, from authors F. Pozo, Y. Vidal and myself, J. M. Serrahima, there is an algorithm that is followed to find the most important 6 sensors to detect the faults.

The goal of this article is to find the six sensors that collect the most important information that allows a detection of the faults. A reduced number of sensors imply a cheaper detection system, and also faster detection, as the calculus time will be inferior due to the amount of information stored.

The final result are the six sensors that separate the most the data coming from the healthy wind turbine and the data coming from the faulty one.

The results are:

- The **most important sensors are 1, 2, 4, 5, 6 and 7**.

Then, the new sensors exposed in the table as it has done before:
Table 14 Six sensors with more information

<table>
<thead>
<tr>
<th>Number</th>
<th>Sensor type</th>
<th>Symbol</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Generated electrical power</td>
<td>$P_{e,m}$</td>
<td>kW</td>
</tr>
<tr>
<td>2</td>
<td>Rotor speed</td>
<td>$\omega_{r,m}$</td>
<td>Rad/s</td>
</tr>
<tr>
<td>4</td>
<td>Generator torque</td>
<td>$\tau_{c,m}$</td>
<td>deg</td>
</tr>
<tr>
<td>5</td>
<td>First pitch angle</td>
<td>$\beta_{1,m}$</td>
<td>deg</td>
</tr>
<tr>
<td>6</td>
<td>Second pitch angle</td>
<td>$\beta_{2,m}$</td>
<td>deg</td>
</tr>
<tr>
<td>7</td>
<td>Third pitch angle</td>
<td>$\beta_{3,m}$</td>
<td>m/s²</td>
</tr>
</tbody>
</table>

Then, what we are measuring is the generated electrical power, rotor speed, generator torque, and the first, second and third pitch angles.

Using these six sensors, and the following information:

- All 24 samples (16 healthy and 8 faulty) are organized forming this matrix:
  
  \[ Y \in M_{\nu \times (N \cdot L)}(\mathbb{R}) \]

  - \( \nu = 50 \), that is, the size of the sample or number of rows.
  - \( N = 6 \) sensors.
  - \( L = 500 \) time instants, that is, the number of columns per sensor.
  - The time step between two measures (that is, two columns) is \( \Delta t = 0.0125 \) seconds.
  - Total time to fulfill the matrix is \( (\nu \cdot L - 1)\Delta t = 312.4875 \) seconds, arranged as in equation [16].

Then, we are going to study the same 24 samples as we have been studying, with a reduced number of 6 sensors and maintaining the \( L = 500 \) time instants and \( \nu = 50 \).

The results of this study:

Table 15 Results with 6 sensors

<table>
<thead>
<tr>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H₀)</td>
<td>(H₁)</td>
<td>(H₀)</td>
<td>(H₁)</td>
</tr>
<tr>
<td>(H₀)</td>
<td>(H₀)</td>
<td>(H₀)</td>
<td>(H₁)</td>
</tr>
<tr>
<td>Fail to reject ( H₀ )</td>
<td>16</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Reject ( H₀ )</td>
<td>0</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

These results expose that when using the six sensors that contain more information, full detection of faults can be done if we consider the first score.
Table 16 Effectivity with six sensors

<table>
<thead>
<tr>
<th></th>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(H₀)</td>
<td>(H₁)</td>
<td>(H₀)</td>
<td>(H₁)</td>
</tr>
<tr>
<td>Fail to reject H₀</td>
<td>1.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.12</td>
</tr>
<tr>
<td>Reject H₀</td>
<td>0.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.88</td>
</tr>
</tbody>
</table>

As what happened when the 13 sensors where studied, the second, third and fourth principal components do not have a good characterization of the samples. However, as the first score has a perfect recognition of the faults and there is neither Type I nor type II errors, we can conclude that this sensor selection is working fine.

When considering other combinations of sensors the results were not as good as with this combination. That is why we are using these six sensors.

4.7. Fault detection with a reduced number of sensors and a reduced number of time instants

This section is a conclusion of all the sections from this Chapter on Simple inference.

In this Chapter the effects of different variations have been studied, with different conclusions:

- The number of rows (size of the sample) cannot be reduced from \( v = 50 \), as it produces degradation on the fault detection procedure.

- The number of time instants can be reduced to \( L = 50 \) or \( L = 25 \) with a perfect recognition of the faults.

- The number of sensors can be reduced to 6 (following table 14).

Then, the last study to be done is the fault detection of a method that uses a reduced number of sensors and a reduced number of time instants.

4.7.1. 6 sensors and \( L=50 \)

For the first case (\( L = 50 \)), the results of the fault detection strategy are summarized in Tables 17-18. These results clearly exposed that the first principal component is capable of detecting all the faulty samples, and at the same time it is capable to state that all 16 healthy samples come from a wind turbine working on their normal condition. Thus, there are neither missing faults, which is a major problem in wind turbines, nor false alarms.
Then, the first score is capable of diagnose correctly all the samples.

### 4.7.2. 6 sensors and L=25

The last case studied for the simple inference is a study with a reduced number of sensors (6 sensors) and a reduced number of time instants \( L = 25 \).

The results are summarized in the following table, which shows just the first score (as it is the most important one) and shows directly the detection and its specificity and sensitivity.

<table>
<thead>
<tr>
<th></th>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( (H_0) )</td>
<td>( (H_1) )</td>
<td>( (H_0) )</td>
<td>( (H_1) )</td>
</tr>
<tr>
<td>Fail to reject ( H_0 )</td>
<td>16</td>
<td>0</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Reject ( H_0 )</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 18 Effectivity (sensitivity-specificity) with six sensors and L=50

<table>
<thead>
<tr>
<th></th>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( (H_0) )</td>
<td>( (H_1) )</td>
<td>( (H_0) )</td>
<td>( (H_1) )</td>
</tr>
<tr>
<td>Fail to reject ( H_0 )</td>
<td>1.00</td>
<td>0.00</td>
<td>0.31</td>
<td>0.44</td>
</tr>
<tr>
<td>Reject ( H_0 )</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.56</td>
</tr>
</tbody>
</table>

The conclusion of all these studies is summarized in the previous table. Using a simple inference method, involving a PCA and statistical hypothesis testing, there can be a reduction on sensors and time instants and still obtain a total recognition of healthy and faulty samples.

Summarized:

- The **number of rows** (size of the sample) cannot be reduced from \( v = 50 \).
- The **number of time instants** can be reduced to \( L = 25 \).
- The **number of sensors** can be reduced to **6** (following table 14).

With these parameters there is a **100% detection of faults**.
CHAPTER 5: SIMPLE INFERENCE IN MATLAB

In this project there is a necessity to use a powerful, fast and reliable program, which can easily deal with large matrices. The best software to work with matrices that is available and that I have used in some subjects throughout the degree is MATLAB.

MATLAB is a program that makes numerical computations, and at the same time is a programming language, that allows coding whatever is needed. Its developer is MathWorks, and every year there is a new upgrade of the program. In my project I am using MATLAB r2015a.

A screenshot of the program:
There are several windows: command window (on the right in the figure) that allows to display results, enter variables, among many other properties; the editor is where the code is written and can be saved (functions can also be written here); the workspace show all the variables that are stored and can be loaded or used; and the command folder is the place where the program is saved in the computer, and saves or can load any file that is there.

In this section there are some examples of parts of the code used in my project.

5.1.1. Load the healthy and faulty samples

As it is explained in Section 3.3, the simulated files have a specific order, and then the files must be loaded to MATLAB and then rearranged the way we want:

```matlab
N = 13;
n = 50;
nfallos = 8; % The different faults
L = 48001; % This implies a step time of 0.0125 s.
            % I will consider the last 6.25 s, that is, 500 elements.
rL = 500; % The number of time instants is reduced to 500.
mostra = 16;
for i=1:n
    var=strcat('SimulacioSaludableRegio3_',num2str(i));
    var=strcat(var,'.mat');
    load(var);
    OutData(:,5)=[];
    for j=1:N
        for k=1:(mostra+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:

```matlab
for fallo=1:nfallos
    var=strcat('SimulacioFallo',num2str(fallo));
    var=strcat(var,'.mat');
    load(var);
    OutData(:,5)=[];
    for j=1:n
        for k=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
```
5.1.2. Group scaling

As it is explained in Section 4.2.2, the data must be rescaled with the mean and the standard deviation of the baseline healthy wind turbine:

```plaintext
for i=1:N % We just compute the standard deviation of the first 50 rows
    dt(i)=std(reshape(X(1:n,(i-1)*rL+1:i*rL),1,n*rL));
end
for j=1:rL % number of columns per block
    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 % We just compute the standard deviation of the first 50 rows
    dt(i)=std(reshape(X(1:n,(i-1)*rL+1:i*rL),1,n*rL));
end
for j=1:rL % number of columns per block
    YT(:,(i-1)*rL+j) = (Y(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
end
end

% YT is the Y (faulty) to diagnose, after the group scaling
```

5.1.3. PCA

All the equations explained in Section 4.2.3 can be summarized in one MATLAB command, which allows us to compute the entire PCA to obtain the \( P \) matrix (named `coeff` in the program)

```plaintext
% PCA

[coeff,score,latent] = princomp(XT(1:50,:),'econ');
T = XT*coeff; %matrix in the new coordinates
Td = YT*coeff; %matrix in the new coordinates
```

5.1.4. Statistical hypothesis testing

The statistical comparison is made with the following iteration, and it follows the idea explained in Section 4.2.4, where the test of equality of means is exposed.

```plaintext
sample = 50; % size of the matrix
[nrow,ncol] = size(T);
```
iter1 = floor((nrow-50)/sample);
sr = 1;
clear CH
total=zeros(1,4);
[nrowy,ncoly]= size(Td);
iter2 = floor((nrow)/sample);
CH = zeros(iter1+iter2,1);
for sr = 1:4
    % STANDART DEVIATION
    sh2 = std(T(1:50,sr))^2;
    for i=1:iter1
        sc2 = std(T(n+sample*(i-1)+1:n+sample*i,sr))^2;
dof = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+1:n+sample*i,sr))/sqrt(std(T(1:50,sr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,sr))^2/sample);
tobs = (mean(T(1:50,sr))-mean(T(n+sample*(i-1)+1:n+sample*i,sr)))/sqrt(std(T(1:50,sr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,sr))^2/sample);
tstar10 = tinv(.82,dofy); % significance=36%
    CH(i,sr) = (abs(tobs)<=tstar10);
end
end

for sr=1:4
    % STANDART DEVIATION
    sh2 = std(T(1:50,sr))^2;
    for i=1:iter2
        sc2 = std(Td(1+sample*(i-1):sample*i,sr))^2;
dofy = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+1:n+sample*i,sr))/sqrt(std(Td(1+sample*(i-1):sample*i,sr))^2/n+std(Td(1+sample*(i-1):sample*i,sr))^2/sample);
tstar10y = tinv(.82,dofy); % significance=36%
    CH(i+iter1,sr) = (abs(tobsy)<=tstar10y);
end
end

suma=zeros(1,8);
for sr = 1:4
    suma(1,sr*2-1) = sum(CH(1:iter1,sr));
    suma(1,sr*2) = iter2-sum(CH(iter1+1:iter1+iter2,sr));
end
display(suma);

The suma matrix is the matrix that shows the results of detections of both healthy and faulty samples.

5.1.5. ROC curve

To create a ROC curve as it is explained in Section 4.3.4, to show the results for a range of levels of significance:

CHroc = zeros(iter1+iter2,1);
for alpha = 1:49
    for sr = 1:4
        % STANDART DEVIATION
        sh2 = std(T(1:50,sr))^2;
        for i=1:iter1
            sc2 = std(T(n+sample*(i-1)+1:n+sample*i,sr))^2;
dof = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+1:n+sample*i,sr))/sqrt(std(T(1:50,sr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,sr))^2/sample);
tobs = (mean(T(1:50,sr))-mean(T(n+sample*(i-1)+1:n+sample*i,sr)))/sqrt(std(T(1:50,sr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,sr))^2/sample);
tstar10 = tinv(.82,dofy); % significance=36%
        CH(i,sr) = (abs(tobs)<=tstar10);
    end
end

for sr=1:4
    % STANDART DEVIATION
    sh2 = std(Td(1+sample*(i-1):sample*i,sr))^2;
    for i=1:iter2
        sc2 = std(Td(1+sample*(i-1):sample*i,sr))^2;
dofy = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+1:n+sample*i,sr))/sqrt(std(Td(1+sample*(i-1):sample*i,sr))^2/n+std(Td(1+sample*(i-1):sample*i,sr))^2/sample);
tstar10y = tinv(.82,dofy); % significance=36%
    CH(i+iter1,sr) = (abs(tobsy)<=tstar10y);
end
end

suma=zeros(1,8);
for sr = 1:4
    suma(1,sr*2-1) = sum(CH(1:iter1,sr));
    suma(1,sr*2) = iter2-sum(CH(iter1+1:iter1+iter2,sr));
end
display(suma);
sh2 = std(T(1:50,scr))^2;
for i=1:iter1
    sc2 = std(T(n+sample*(i-1)+1:n+sample*i,scr))^2;
    dof = floor(((sh2/n+sc2/sample)^2)/(sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1));
    tobs = (mean(T(1:50,scr)) - mean(T(n+sample*(i-1)+1:n+sample*i,scr))) / sqrt(std(T(1:50,scr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,scr))^2/sample);
    tstar10 = tinv(1-0.01*alpha,dof);
    CHroc(i,scr) = (abs(tobs)<=tstar10);
end
FPR(alpha,scr) = (iter1- sum(CHroc(1:iter1,scr)))/(iter1);
end
for alpha=1:49
    for scr=1:4
        % STANDARD DEVIATION
        sh2 = std(T(1:50,scr))^2;
        for i=1:iter2
            sc2 = std(Td(1+sample*(i-1):sample*i,scr))^2;
            dofy = floor(((sh2/n+sc2/sample)^2)/(sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1));
            tobsy = (mean(T(1:50,scr)) - mean(Td(i+sample*(i-1):sample*i,scr))) / sqrt(std(T(1:50,scr))^2/n+std(Td(i+sample*(i-1):sample*i,scr))^2/sample);
            tstar10y = tinv(1-0.01*alpha,dofy);
            CHroc(i+iter1,scr) = (abs(tobsy)<=tstar10y);
        end
        TPR(alpha,scr) = (iter2- sum(CHroc(iter1+1:iter1+iter2,scr)))/(iter2);
    end
end
fig = 1;
figure(fig)
% plotting options
set(gcf,'DefaultLineMarkerSize',8);
set(gcf,'DefaultLineWidth',2);
set(gcf,'DefaultAxesFontSize',12);
set(gcf,'DefaultAxesFontName','times');
plot(FPR(:,1),TPR(:,1),'r-o','MarkerFaceColor','red','MarkerSize',8)
hold on
plot(FPR(:,2),TPR(:,2),'g-^','MarkerFaceColor','green','MarkerSize',8)
plot(FPR(:,3),TPR(:,3),'b-s','MarkerFaceColor','blue','MarkerSize',8)
plot(FPR(:,4),TPR(:,4),'m-','MarkerFaceColor','magenta','MarkerSize',8)
plot([0 1],[0 1],'k--')
xlabel('False positive rate','Interpreter','latex')
ylabel('True positive rate','Interpreter','latex')
legend('score 1','score 2','score 3','score 4','Location','southeast')
title(strcat('#L=$',num2str(rL),')'),'Interpreter','latex')
hold off
CHAPTER 6: MULTIVARIATE INFERENCE

The second methodology to study the fault detection in wind turbines that is used in this project is the multivariate (or multiple) statistical inference. In this methodology, as opposed to the simple inference, we do not have different scores, because what we are comparing are vectors.

This method had previously been applied to the detection of structural damage in shells with piezoelectric sensors [5], on springs, columns and beams [6] or simple structures. However, in all those cases, the excitation is constant, and there are structural changes, and the objective was to determine these changes with the same initial excitation. In the situations that are studied in this project, the initial excitation is not constant: wind is a random process, as it has variable conditions. Then, the application of the multivariate inference is much more complicated as just the application of the method. This is a test to see if this methodology is capable of detecting faults with changing conditions. Then, this section of the project applies a well-known method to study simple structures, to the study of fault detection in a complex wind turbine.
6.1. Mathematical Model

6.1.1. Group scaling and PCA

The first steps are the same as they were from the simple inference:
- Creation of the matrix \( X \in M_{n \times (N \cdot L)}(\mathbb{R}) \).
- Group scaling.
- Creation of the new matrix \( \bar{X} \).
- Calculate of the Principal Component Analysis matrix \( P \).
- Find the baseline pattern \( T \).
- With the sample to diagnose create matrix \( Y \in M_{v \times (N \cdot L)}(\mathbb{R}) \).
- Scale the matrix \( \bar{Y} \) with the values from the original matrix.
- The projection of the scaled matrix \( \bar{Y} \) into the spanned space by the eigenvectors in \( P \).

The first steps are the same as explained in Section 4.2.1 of Simple Inference, and to calculate the previous matrices, we must follow equations [4] to [7].

6.1.2. Multivariate test

The objective of this test is to determine if the distribution of the multivariate random samples that are obtained from the wind turbine to be diagnosed (undamaged or not) is connected to the distribution of the baseline [5].

Let \( s \in \mathbb{N} \) be the number of principal components that are going to be considered jointly. We also consider that:
- The baseline projection is a multivariate random sample of a multivariate random variable following a multivariate normal distribution with known population mean vector \( \mu_h \in \mathbb{R}^s \) and known variance-covariance matrix \( \Sigma \in M_{s \times s}(\mathbb{R}) \).
- The multivariate random sample of the structure to be diagnosed also follows a multivariate normal distribution with unknown multivariate mean vector \( \mu_c \in \mathbb{R}^s \) and known variance-covariance matrix \( \Sigma \in M_{s \times s}(\mathbb{R}) \).

Then, what we are really calculating is the mean vector from the initial healthy \( \bar{X} \). Then, we suppose that this value is the same as the population mean vector.

\[
\bar{X} \rightarrow T = \bar{X} \cdot P
\]
As it is known, $T$ is the principal component matrix, and has 50 rows (as the original $\bar{X}$ matrix had 50 rows) and $s$ columns ($s$ are the number of principal components).

Then, for $i = 1, \ldots, s$ being the principal components of $T$, and $t_i$ being each of the columns of $T$, the population mean vector is:

$$
\mu_h = \frac{1}{s} \sum_{i=1}^{n} t_i = \begin{bmatrix} 
\mu^1_h \\
\mu^2_h \\
\vdots \\
\mu^s_h 
\end{bmatrix}
$$

(41)

That is, the average of all the values of the $i = 1, \ldots, s$ principal component. Then we have a vector of size $s$, that we consider that is the same as the population mean vector in order to be able to continue with the calculation.

Then, the problem that we are going to consider is to determine whether a given $s$-dimensional vector $\mu_c$ is a plausible value for the mean of a multivariate normal distribution $N_s(\mu_h, \Sigma)$.

$$
H_0: \mu_h = \mu_c \text{ versus } H_1: \mu_h \neq \mu_c
$$

(42)

- **Null hypothesis**: the multivariate random sample of the wind turbine is distributed as the baseline projection. If the result is that the null hypothesis is not rejected the WT is categorized as healthy.
- Otherwise, if the null hypothesis is rejected in favor of the **alternative hypothesis** this would indicate the presence of some structural change and therefore the presence of a fault.

The test that this section uses is based on the Hotelling’s $T^2$ statistic [7]. When a multivariate random sample of size $v \in \mathbb{N}$ is taken from a multivariate normal distribution $N_s(\mu_h, \Sigma)$, the random variable $T^2$:

$$
T^2 = v(\bar{Y} - \mu_h)^T S^{-1}(\bar{Y} - \mu_h)
$$

(43)

Is distributed as:

$$
T^2 \sim \frac{(v - 1)s}{v - s} F_{s,v-s}
$$

(44)

In this equation is $\bar{Y}$ the sample mean of the matrix to $T_d$ diagnose (that is, the mean of each row of the matrix):
\( T_d = \bar{Y} P \in M_{n \times r}(\mathbb{R}) \) \hspace{1cm} (45)

\[ Y = \frac{1}{s} \sum_{i=1}^{n} t d_i = \begin{bmatrix} \bar{y}_d^1 \\ \bar{y}_d^2 \\ \vdots \\ \bar{y}_d^s \end{bmatrix} \] \hspace{1cm} (46)

The term \( F_{s,v-s} \) denotes a random variable with and F-distribution with \( s \) and \( v - s \) degrees of freedom.

\( S \) is the sample covariance matrix, calculated as:

\[ S = \frac{1}{v} \sum_{i=1}^{v} (Y_i - \mu_h)(Y_i - \mu_h)^T, \quad S \in M_{s \times s} \] \hspace{1cm} (47)

The parameter that we are going to compare, for each sample, is:

\[ t_{obs}^2 = v(\bar{Y} - \mu_h)^T S^{-1}(\bar{Y} - \mu_h) \] \hspace{1cm} (48)

This parameter is going to be compared with:

\[ t_{obs}^2 \leq \frac{(v-1)s}{v-s} F_{s,v-s}(\alpha) \rightarrow \text{Fail to reject } H_0 \] \hspace{1cm} (49)

\[ t_{obs}^2 > \frac{(v-1)s}{v-s} F_{s,v-s}(\alpha) \rightarrow \text{Reject } H_0 \] \hspace{1cm} (50)

The upper \((100\alpha)\)th percentile of the \( F_{s,v-s} \) distribution is \( F_{s,v-s}(\alpha) \).

To calculate this expression we need the inverse function:

\[ F_{s,v-s}(\alpha) = finv(1 - \alpha, s, v - s) \] \hspace{1cm} (51)
6.2. Simulations

The simulated files used in the multivariate statistical inference are the same that were used in the previous analysis with the simple inference. This way, the results are comparable one with the other.

Then, from an undamaged wind turbine we have:

- \( X \in M_{n \times (N \cdot L)}(\mathbb{R}) \)
- \( n = 50 \), that is, the size of the sample or number of rows.
- \( N = 13 \) sensors.
- \( L = 500 \) time instants, that is, the number of columns per sensor.
- From the \( X \) matrix we calculate from equations [5, 6, 7] \( \mu^j_k \), \( \mu^k \) and \( \sigma^k \), which respectively are the mean of all elements in the same column, the mean from all elements from the same sensor, and the standard deviation from all elements from each sensor. These values are going to be saved and are going to be the ones that do the group scaling to all samples to be diagnosed.
- We rescale the \( X \) matrix with the equation [8] with the previous values.
- The PCA is made from this matrix to obtain \( P \).
- With equation [13] we calculate \( T \).

We also have 24 samples of turbines to diagnose: 16 healthy samples with different conditions, and 8 faulty samples, one for each of the typical faults.

- \( Y \in M_{\nu \times (N \cdot L)}(\mathbb{R}) \)
- \( \nu = 50 \), that is, the size of the sample or number of rows.
- \( N = 13 \) sensors.
- \( L = 500 \) time instants, that is, the number of columns per sensor.

The results are exposed in the following section.
6.3. Results

This is a method that has almost never been applied to structural control of wind turbines. Then, there are a lot of parameters that can vary and need to be studied.

First of all, as it is a multivariate inference we need to decide which is:

- The best number of principal components to use. That is, the number $s$ that defines the length of the population mean vector.
- The significance that allows a good detection of faults without making neither Type I nor Type II errors.

As there is no function to optimize and find the maximum, this will be an iteratively method to find the best result.

The study that this part of the project handles is the study of the number of principal components that are going to be used. The number of principal components is the length that the population mean vector will have:

$$\mu_h = \left[ \begin{array}{c} \mu_h^1 \\ \mu_h^2 \\ \vdots \\ \mu_h^s \end{array} \right]$$

This length will be equal to the length of the sample mean vector, in order to be able to compare them.

As it can be inferred, the bigger $s$ is the more information we have, as the multiple inference will use all the components from the first to the $s$-th component. However, at the same time using more and more components will increase the calculation time, so there is a need to find a value that allows detection but is no too large.

As it has been used in all the previous sections, the results will be represented in a table as follows:

<table>
<thead>
<tr>
<th></th>
<th>Undamaged Sample ($H_0$)</th>
<th>Damaged Sample ($H_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail to reject $H_0$</td>
<td>Correct decision</td>
<td>Type II error (missing fault)</td>
</tr>
<tr>
<td>Reject $H_0$</td>
<td>Type I error (false alarm)</td>
<td>Correct decision</td>
</tr>
</tbody>
</table>

For example, some of the tests that have been done have these results:
When the number of principal components is \( s = 2 \):

**Table 21** Results when we pick the \( s=2 \) first principal components

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>Undamaged Sample (( H_0 ))</th>
<th>Damaged Sample (( H_1 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>2%</td>
<td>Fail to reject ( H_0 ) 15</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 1</td>
<td>8</td>
</tr>
<tr>
<td>6%</td>
<td>Fail to reject ( H_0 ) 14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 2</td>
<td>8</td>
</tr>
<tr>
<td>14%</td>
<td>Fail to reject ( H_0 ) 12</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 4</td>
<td>8</td>
</tr>
<tr>
<td>22%</td>
<td>Fail to reject ( H_0 ) 11</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 5</td>
<td>8</td>
</tr>
<tr>
<td>38%</td>
<td>Fail to reject ( H_0 ) 9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 7</td>
<td>8</td>
</tr>
</tbody>
</table>

This study can be continued until the significance reaches really high values; however, it can be seen a tendency where the detection of undamaged samples decreases, while the detection of the damaged samples is always perfect.

When the number of principal components is \( s = 7 \):

**Table 22** Results when we pick the \( s=7 \) first principal components

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>Undamaged Sample (( H_0 ))</th>
<th>Damaged Sample (( H_1 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>2%</td>
<td>Fail to reject ( H_0 ) 16</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 0</td>
<td>8</td>
</tr>
<tr>
<td>6%</td>
<td>Fail to reject ( H_0 ) 16</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 0</td>
<td>8</td>
</tr>
<tr>
<td>14%</td>
<td>Fail to reject ( H_0 ) 12</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 4</td>
<td>8</td>
</tr>
<tr>
<td>22%</td>
<td>Fail to reject ( H_0 ) 7</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 9</td>
<td>8</td>
</tr>
<tr>
<td>38%</td>
<td>Fail to reject ( H_0 ) 1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 ) 15</td>
<td>8</td>
</tr>
</tbody>
</table>
When the number of principal components is \( s = 12 \):

**Table 23** Results when we pick the \( s=12 \) first principal components

<table>
<thead>
<tr>
<th></th>
<th>Undamaged Sample ((H_0))</th>
<th>Damaged Sample ((H_1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha = 2% )</td>
<td>Fail to reject ( H_0 )</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha = 6% )</td>
<td>Fail to reject ( H_0 )</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha = 14% )</td>
<td>Fail to reject ( H_0 )</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha = 22% )</td>
<td>Fail to reject ( H_0 )</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 )</td>
<td>4</td>
</tr>
<tr>
<td>( \alpha = 38% )</td>
<td>Fail to reject ( H_0 )</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Reject ( H_0 )</td>
<td>8</td>
</tr>
</tbody>
</table>

As a conclusion of these tables, it can be seen that there is a tendency of decreasing the number of right decisions when healthy samples are diagnosed: when the number of principal components is maintained, if the significance increases there are more Type I errors (false alarms, detection of a fault when there is no fault). However, the sensitivity (detection of the faults) is always maximum for all the cases of the multivariate inference with \( s = 2, 7 \text{ and } 12 \).

The results can be better understood graphically (Figure 21):
When using the 12 first principal components the effectiveness in the detection of the healthy samples begins with a 100% and starts decreasing when the significance is $\alpha > 15\%$.

The multiple inference when using $s=7$ has worse results than when using 12, as the degradation in the detection starts much before. At the same time, the multiple inference when using $s=2$ is even worse, as there is never a 100% effectivity in the detection of the healthy samples.

Figure 21 Specificity for different multiple inferences as a function of the significance
The sensitivity of the multiple inference offers extraordinary results:

- Even if we increase the significance, the detection is maintained always to 100% of effectivity for all the groups studied: principal components 1-2, 1-7 and 1-12.

Then, we can conclude that out of the different values studied, the results in multivariate inference is the best when the number of principal components is set to 12, as the range of significances that gives a total detection is bigger than on the others.

The conclusions to the multivariate inference are that:

- When the number of principal components increase, the range of significances where there is a perfect detection (all faulty and healthy samples are correctly categorized) is bigger. For example, for s=12 the range of significances is $\alpha = 1, \ldots, 14\%$.

- If the number of principal components used is small (for example s=2) there is never a total recognition of the healthy samples (specificity).

- Sensitivity is maintained to a 100% when the number of principal components used goes from 2 to 12, and for a range of significances of $\alpha = 1, \ldots, 40\%$. This means that the mean vectors of the faulty samples are really different from the healthy one, and therefore can always be detected as faulty.
CHAPTER 7: MULTIPLE INFERENCE IN MATLAB

As it has already been explained, the first steps to study the multiple inference are the same as the ones from the simple inference (import the data, normalization and the PCA). Then, the main parts of the multiple inference in MATLAB are the following ones:

For a number of principal components used together s=2.

7.1.1. Creation of population mean vector and sample mean vector

The population mean vector is calculated as explained in equation \[41\], and it contains information of the first s principal components. The sample mean vector is calculated following the same equation, but with the data from the sample to diagnose:

```matlab
s=2;
%Calculating of population mean vector (mu)
mu=zeros(s,1);
for scr = 1:s
    mu(scr,1)=sum(T(1:sample,scr))/50;
end
%Sample vector mean
[nrow,ncol] = size(T);
[nrowy,ncoly]= size(Td);
iter1=floor((nrow-50)/sample);
iter2=floor((nrowy)/sample);
vectormean=zeros(s,iter1+iter2);
for scr=1:s
    for i=1:iter1
        vectormean(scr,i)=mean(T(n+sample*(i-1)+1:n+sample*i,scr));
    end
end
```
end
for scr=1:s
    for i=1:iter2
        vectormean(scr,iter1+i)=mean(Td(sample*(i-1)+1:sample*i,scr));
    end
end

7.1.2. Creation of the sample covariance matrix

The sample covariance matrix is the matrix that is calculated as equation [47], and uses information from the matrix after the PCA and the sample mean vector:

for iteracio=1:24 %there are 24 samples (16 healthy and 8 faulty)
    S=0;
    clear xi T2
    %Sample covariance matrix
    if iteracio<=16
        for i=1:sample
            %We create a vertical vector with the values of each row of T
            xi=T(n+i+sample*(iteracio-1),1:s)';
            %We create the matrix when transposing the second vector
            S=S+(xi-mu)*(xi-mu)';
        end
    else
        for i=1:sample
            %We create a vertical vector with the values of each row of T
            xi=Td(i+sample*(iteracio-17),1:s)';
            %We create the matrix when transposing the second vector
            S=S+(xi-mu)*(xi-mu)';
        end
    end
    %The sample covariance matrix must be multiplied by (1/sample)
    S=(1/sample)*S;

7.1.3. Hotelling’s $T^2$

The multivariate statistical hypothesis testing is made using the Hotelling’s method, and this method follows equations [48, 49, 50].

%Hotelling's T^2
T2=sample*(vectormean(:,iteracio)-mu)'*inv(S)*(vectormean(:,iteracio)-mu);

if T2<=(sample-1)*s/(sample-s)*finv(1-alpha,s,sample-s)
    CH3(iteracio,1)=1;
else
    CH3(iteracio,1)=0;
end
end
%Result vector:
    suma(1,1)=sum(CH3(1:16));
    suma(1,2)=iter2-sum(CH3(17:24));

The *suma* vector is the vector that has the results, that is, the number of diagnosed samples.
CHAPTER 8: SIMPLE VS MULTIPLE INFERENCE

In this project we have studied two separate ways to detect the state of a wind turbine, that is, if the turbine is on a healthy or a faulty state. Both processes begin identically:

- A set of healthy data ($\mathbf{X}$) from the structure create a pattern ($\mathbf{P}$), after being standardized with the mean and the standard deviation.

- A sample to diagnose ($\mathbf{Y}$) is standardized with the same mean and standard deviation as with the healthy data.

- The pattern from the healthy structure (PCA model, ($\mathbf{P}$)) is applied to both the healthy sample and the sample to diagnose.

When we have both matrices in the new set of coordinates (the principal component space), that is when the two methods differ.

1. Simple inference compares each principal score from the healthy and the structure to diagnose, thus we have as much results as principal components we choose for each comparison.

2. Multivariate inference creates a population mean sample with $s$ principal components from the healthy sample, which is compared with the sample mean (also with $s$ principal components) from the structure to diagnose. Then, we have only one result per comparison.

However, even if the two methods differ in the last part of the statistical comparison, they can still be compared after all: the ultimate result is a
classification of a sample as healthy or faulty. Then, as we know which samples are in a healthy state or damaged, we can compare their results.

Using the same 24 samples as we have been using in this project, classified as:

- **16 healthy samples**, with \( v = 50 \) rows, \( l = 500 \) time instants and \( n = 13 \) sensors.

- **8 faulty samples**, with \( v = 50 \) rows, \( l = 500 \) time instants and \( n = 13 \) sensors, one for each fault.

We will fix the significance and study the results for this given significance. Then, in order to study the effects of different significances in both methods, we are going to study a wide range of significances, for both simple and multiple inferences between:

1. First score of the simple inference
2. Second and third scores of the simple inference
3. Multiple inference with \( s = 2 \)
4. Multiple inference with \( s = 7 \)
5. Multiple inference with \( s = 12 \)

### 8.1. For a fixed significance

We are going to begin the comparison by choosing a random value for the significance, and compare the results obtained by the first and second scores from the simple inference; and multivariate inference with \( s = 2, s = 7 \) and \( s = 12 \).

For a value of significance of \( \alpha = 0.1 = 10\% \) we have the following results:

<table>
<thead>
<tr>
<th>Simple inference</th>
<th>Multiple inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score 1 ((H_0))</td>
<td>Score 2 ((H_0))</td>
</tr>
<tr>
<td>Fail to reject (H_0)</td>
<td>16</td>
</tr>
<tr>
<td>Reject (H_0)</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

For the same value for the significance \( (\alpha = 10\%)\), the results show that the **first score** is able to detect and **correctly classify all the 16 healthy samples**; however, one of the faulty samples is not classified as such, so there is **one missing fault**.

The **second score** has worse results, as expected, as there are **three false alarms and 7 missing faults**.
For the multiple there is an improvement on the correct classification in the healthy samples: when 12 scores are used all samples are correctly classified, while only 13 are classified in the Multi7 and 12 on the Multi2. Nevertheless, all faulty samples are correctly classified as faulty in the three cases.

However, there is a need of a most exhaustive study of the variation of the significance in the effectiveness of the diagnosis of samples with both simple and multiple inferences. This study is done in the following section.

8.2. For a range of significances

For a range of significances, that is:

\[ \alpha = 1\%, ..., 60\% \] (53)

We are going to study the effect of the different significances in both methods. As the significance in both methods is defined as the rejection region of the test, they are comparable one with the other.

8.2.1. First score of the simple inference

Using only the first score of the simple inference (just to recall, in the section of the Simple Inference we fixed the significance to \( \alpha = 36\% \)), the sensitivity and specificity varies as showed in the following figure:

![Figure 23 First score as a function of the significance](image)
It can be seen in figure 23 that the specificity, or proportion of the healthy samples that are correctly identified as such, begins with 100% of precision when the significance is really small, and decreases when the significance is bigger than $\alpha = 40\%$.

However, the sensitivity (detection of the faulty samples) begins with a really low effectivity when the significance is low, and there is a full detection when $\alpha > 30\%$.

Then, the first score has 100% effectivity in the method (correctly classifying both healthy and faulty samples):

$$EF \ (\text{score } 1) = [\alpha = 31\%, \alpha = 41\%]$$

Then, for an effective classification in the simple inference the significance must be **between 0.31 and 0.41** for the first score.

### 8.2.2. Second and third scores in simple inference

The second and third scores are represented altogether, as their results are not as good as the results when we used the first one.

![Figure 24 Second and third scores as a function of the significance](image)

As it can be inferred from the figure, the second score has bad specificity (correct classification of healthy samples), as just the initial values of the significance
there is a correct decision; its sensitivity is good only when the values of the significance are above 35%.

For the third score the specificity is better than with the second one, but the sensitivity is never 100%, thus it is not worth considering it.

Then, the effectivity of both scores is:

\[
EF (\text{score } 2) = [0]
\]

\[
EF (\text{score } 3) = [0]
\]

There is no interval where the total effectivity is 100% as there is never correct classification of both healthy and faulty samples.

Then, from now on it is not worth comparing them with the other methods.

8.2.3. **Multiple inference with s=2**

The first multiple inference that is studied is the multiple inference with s=2. This means that the information from scores 1 and 2 is used (equation [41]).

![Multiple inference s=2](image)

**Figure 25** Multiple inference with s=2 as a function of the significance

The results when using the first two scores together are good when considering the sensitivity, that is, the correct decisions for the faulty samples; however, the
correct classification of the healthy samples begins in an approximate rate of 95% of effectivity and then decreases as the significance increases. Then, the interval for 100% effectivity on the classification for both faulty and healthy samples is:

\[ EF \left( \text{Multi 2} \right) = [0] \]  

However, even with effectivity’s interval being zero, the result is better than the second and third scores on the simple inferences, as with the Multi2 the method is always detecting the faults in the structure.

8.2.4. Multiple inference with \( s=7 \)

The results of the multiple inference when using the scores 1 to 7 are supposed to be better than with \( s=2 \), as it is using information from more scores.

Figure 26 Multiple inference with \( s=7 \) as a function of the significance

The results of the Multi7 are better than the previous ones: sensitivity is always maintained to 100%, and the specificity begins with a total detection. Then, when using the first 7 principal components, the interval for 100% effectivity on the detection for both faulty and healthy samples is:

\[ EF \left( \text{Multi 7} \right) = [\alpha = 1\%, \alpha = 7\%] \]
Then, for \textit{significances between 0.01 and 0.07} there is \textit{correct detection} of both healthy and faulty samples when using \textit{Multi7}.

\subsection*{8.2.5. Multiple inference with $s=12$}

When using the first 12 principal components together, the results are better than with the other multivariate results.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure27}
\caption{Multiple inference with $s=12$ as a function of the significance}
\end{figure}

As the theory states, when using a multivariate inference methodology, the highest the number of principal components we use together, the better the results are. When using the first 12 scores altogether, there is a total detection of the faulty samples for all the values of significance; the correct classification of the healthy samples is a 100\% when the significance is low.

Then, the interval for 100\% effectivity for both faulty and healthy samples is:

\begin{equation}
EF (\text{Multi 12}) = [\alpha = 1\%, \alpha = 14\%]
\end{equation}

\begin{flushright}
(59)
\end{flushright}
The **multiple inference with** \( s = 7 \) **has a perfect classification** of both healthy and faulty samples when \( 0.01 < \alpha < 0.14 \).

### 8.2.6. Total effectivity

When comparing the simple and multivariate inference in the previous graphs there was a representation of both sensitivity and specificity. However, for the last comparison we are going to define a new variable, the **total effectivity**, which is the percentage of right decisions for both healthy and faulty samples:

\[
TEF(\%) = \frac{(Correct\ Healthy\ Det) + (Correct\ Faulty\ Det)}{Total\ number\ of\ samples} \cdot 100
\]

This equation expresses the total of correct decisions of the method to study, with values that are:

- As there are 16 healthy samples, the number of correct decisions must be between: \( Correct\ Healthy\ Det = (0,16) \).

- As there are 8 faulty samples that are diagnosed, the number of correct classification of these samples is: \( Correct\ Faulty\ Det = (0,8) \)

- **Total number of samples** = 16 + 8 = 24

Then, the total effectivity of **Score 1 and Score 2** (from simple inference), and **Multi 7 and Multi 12** (from multiple inference) is:
When we use the same 24 samples as we have been using in this project, classified as:

- **16 healthy samples**, with $\nu = 50$ rows, $L = 500$ time instants and $N = 13$ sensors.
- **8 faulty samples**, with $\nu = 50$ rows, $L = 500$ time instants and $N = 13$ sensors, one for each fault.

We have used the same samples in order to have comparable results, and from these results we can obtain really good and clear conclusions.

- First of all, from Figure 23 we can conclude that the **first score** (from the simple inference) has a perfect performance to diagnose a wind turbine if and only if the significance is $0.31 < \alpha < 0.41$. If we recall the section of the simple inference, we have been using a significance of 36%, which lies in this interval.
- Then, **scores 2 and 3** from the simple inference have never a perfect effectivity in the diagnosis of wind turbine’s structures.

**Figure 28** Simple and multiple inferences as a function of the significance
The higher the number of scores that we are using altogether in the multiple inference the better the results are.

The Multi12 from the multivariate inference has a total effectivity of 100% for an interval of significances of $0.1 < \alpha < 0.14$.

Now, in order to compare the results of the last figure it is necessary to recall the meaning of the significance. The level of significance of a test is the number that determines the rejection region. It is the probability that the test will result in a Type I error.

Then, for small levels of significance the number of missing faults is inferior. Therefore, for the multiple inference we can have a smaller significance, and hence a smaller number of missing faults (Type I error) than when using the simple inference.

In figure 28 it can be seen that for $0.1 < \alpha < 0.14$ the Multi12 has a total effectivity of 100%, as there is a correct classification of both healthy and faulty samples; score 1, however, increases its total effectivity from a 60 to a 95% but is never perfect until the significance is 0.30.

As we can empirically see that the multiple inference always detects the damaged structures, the lower the significance is the better the method is. Then, for low significances the multivariate statistical inference has better results than the simple inference.
CHAPTER 9: CONCLUSIONS

This project had one clear object of study. This objective was to study the structural health monitoring (SHM) of wind turbines. SHM is a type of structural control that allows distant damage detection of a system when it is working, due to the data collected by the thirteen sensors placed on the structure.

In order to control a structure it is necessary to have a healthy baseline from which to extract a healthy pattern that will thereafter compared with other structures. However, wind turbines are working under changing conditions: the wind and its associated turbulence are constantly varying, thus it is difficult to extract a pattern. The method developed must be capable of detecting the failures even with changing wind conditions, in order to make detection of damage for real situations.

Then, in this project a statistical comparison is done between the pattern and the structure that must be diagnosed. Using principal component analysis (PCA) and hypothesis testing, a diagnosis is done to a structure: it can be catalogued as healthy (when the structure is working under normal circumstances) or faulty (there is some kind of fault or damage in the system).

As it has been explained in this project, the first method used is the simple inference. In this method, there is firstly a group scaling, a PCA and then simple or univariate statistical inference. As it is explained, this method had been previously applied to wind turbines. In this project, this method obtains good results: all data samples from structures in different conditions are correctly catalogued as healthy or faulty. Then, the first objective of this project has been accomplished: simple inference can be used to diagnose wind turbines to detect if there is some kind of structural damage.

However, the time of the diagnosis was really high in the previous study. When a real turbine is controlled, there is a need of a fast diagnosis, that is, an almost on-line detection of faults, if there is any. Then, the second objective of this
project was to reduce the detection time without losing effectiveness. This section consists of two different studies: reduction of the size of the data sample (the number of rows of the original matrix with the data) and the reduction of time instants (the number of columns of the matrix); also, an extra study is done regarding sensor selection, with the six sensors that contain more information.

The first study that has been applied is the reduction of the size of the sample. As it has been explained in its section of the project, the decrease on the size of the sample implies degradation on the diagnosis of both healthy and faulty structures. Then, there is a clear conclusion to this study: the number of rows of the matrix cannot be changed from the original one, as the effectiveness of the method disappears.

The second study on the reduction of the detection time in the simple inference that has been done is the reduction of time instants saved in the data sample. Each column of the data sample is a measure from a sensor of a different time instant, hence, the bigger the matrix is, the more time the sensors must be collecting data. Then, a reduction of this parameter directly decreases the total diagnosis time. The result of this study is outstanding: the time instants per row can be reduced from the original $L = 500$, to $L = 25$ columns with 100% effectivity of correct diagnosis. Then, the total diagnosis time can be reduced a 95% from the original detection time. This implies that the fault detection can be on real time, that is, an almost online fault detection of a structure.

Moreover, using only the six sensors that contain more information, and with the reduction of the detection time (number of columns), all faulty and healthy samples are correctly identified, with a reduction of 95% of the detection time and a 54% reduction on the number of sensors.

All these results regarding simple inference (reduction of the detection time, both size of the sample and time instants, and the sensor selection), are the base of the article “On Real-Time Fault Detection in Wind Turbines: Sensor Selection Algorithm and Detection Time Reduction Analysis”, that I am coauthoring with Dr. Francesc Pozo and Dr. Yolanda Vidal, that is at this moment being reviewed for its future publication in Energies.

The second method that is applied in this project is the multivariate (multiple) inference. Unlike simple inference, multiple inference, as far as I am aware of, had not been previously applied to wind turbines. This is then a pilot trial, as in the other applications the excitation was constant, while in the turbine’s case the excitation (wind) is not constant but always changing. The multivariate inference obtains results that can be summarized as follows: the higher the number of principal components working altogether, the better. In this project the best number of scores working together turns out to be 12, which gives a perfect effectivity while diagnosing healthy and faulty samples on bigger ranges of significances than any of the other multiple inferences tested.

The multivariate inference works for a different range of significances than the one that gives perfect recognition on the simple inference. This is the main conclusion that we can obtain from the section that compares them. Multiple inferences with a big number of principal components working together always detect the damaged structures when significance is small (significance is the probability of a type I error, that is, a false alarm), while the simple inference
has good functionality when significances are bigger. Then, for low significances the multivariate statistical inference has better results than the simple inference.

This project has improved a method that had already been tested on wind turbines, and it has applied for the first time the multivariate inference to detect damages in wind turbines. However, there are things that future projects can improve and continue.

Regarding the data samples, future projects can use the same methodology and study more different samples of data of wind turbines in different conditions. Moreover, it can be a good improvement to test the method with data from a real, already installed and functioning turbine, to make sure that the method is able to diagnose it as well.

Additionally, the detection time for the multivariate inference is too long. Future works can also follow the line to reduce the detection time when using the multiple inference, as it has been done in the simple inference.

Another important future line of work is the detection and diagnose of each fault. In this project, the result of each study is whether a wind turbine is working under healthy or faulty conditions; then, a future project could try to diagnose which fault is happening at each set of data samples. This algorithm would ease the maintenance works as well as decrease the costs.

The main objectives of the project have been achieved while at the same time they leave doors open for future research.
CHAPTER 10: 
BIBLIOGRAPHY

10.1. Bibliographic references


10.2. General bibliography


Fault detection in wind turbines using PCA and statistical hypothesis testing


Lent, Craig S. Learning to program with MATLAB (Building GUI tools). John Wiley & Sons, Ltd, 1989

Oetiker, Tobias; Partl, Hubert; Hyna, Irene, and Schlegl, Elisabeth. The not so short Introduction to LaTeX. Free Software Foundation, Inc., 2011.


Budget

“Fault detection in wind turbines using PCA and statistical hypothesis testing”

TFG presentat per obtenir el títol de GRAU en ENGINYERIA MECÀNICA
Per Josep Mª Serrahima de Cambra

Barcelona, 8 de Juny de 2016

Director: Francesc Pozo Montero
Departament Matemàtica Aplicada III (MAIII)
Universitat Politècnica de Catalunya (UPC)
BUDGET

Any engineering project has its associated economic costs. Considering that this is a research project, this budget will be divided in software costs, engineering costs, indirect expenses and benefit.

Software costs

Software costs are related to the costs of the usage of different software, which need a license to function. Costs have been proportionally included for only the 6 months this project has been worked.

<table>
<thead>
<tr>
<th>Software</th>
<th>License annual cost</th>
<th>6 months’ cost</th>
<th>Cost [€]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB r2015a</td>
<td>2,000</td>
<td>1,000</td>
<td>1,000</td>
</tr>
<tr>
<td>Microsoft Office Professional</td>
<td>594</td>
<td>297</td>
<td>297</td>
</tr>
<tr>
<td><strong>SUBTOTAL</strong></td>
<td><strong>1,297</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additionally to the software costs, there is a need to add the cost of the computer where all the work has been done, and the Internet signal:

<table>
<thead>
<tr>
<th>Software</th>
<th>Price</th>
<th>6 months’ cost</th>
<th>Cost [€]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBTOTAL 1</td>
<td></td>
<td>1,297</td>
<td></td>
</tr>
<tr>
<td>HP Envy 15.6</td>
<td>1,215</td>
<td>1,215</td>
<td></td>
</tr>
<tr>
<td>Internet</td>
<td>45€/month</td>
<td>270</td>
<td>270</td>
</tr>
<tr>
<td><strong>SUBTOTAL</strong></td>
<td><strong>2,782</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Engineering costs

The engineering costs include the study of the methodology of the structural health monitoring, application of the method and programming, and study the results. As this is a research project, these are the main costs, but at the same time they are theoretical costs, as this is the price someone would pay for an engineer to investigate about this topic. Additionally, the time cost of my advisor, who has been helping and reviewing the project, is considered.
<table>
<thead>
<tr>
<th>Worker</th>
<th>Salary (€/hour)</th>
<th>Number of hours</th>
<th>Cost [€]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junior Engineer</td>
<td>20</td>
<td>600</td>
<td>12,000</td>
</tr>
<tr>
<td>Doctor</td>
<td>80</td>
<td>30</td>
<td>2,400</td>
</tr>
<tr>
<td><strong>SUBTOTAL 2</strong></td>
<td></td>
<td></td>
<td><strong>14,400</strong></td>
</tr>
</tbody>
</table>

**Total budget**

The indirect costs are approximated as a 13% of the engineering and software costs, and project benefit is estimated at 6% of the other costs.

<table>
<thead>
<tr>
<th>Worker</th>
<th>Value</th>
<th>Cost [€]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtotal 1</td>
<td></td>
<td>2,782</td>
</tr>
<tr>
<td>Subtotal 2</td>
<td></td>
<td>14,400</td>
</tr>
<tr>
<td><strong>Indirect costs</strong></td>
<td>13%</td>
<td>2,234</td>
</tr>
<tr>
<td><strong>TOTAL COSTS</strong></td>
<td></td>
<td><strong>19,416</strong></td>
</tr>
<tr>
<td><strong>Benefit</strong></td>
<td>6%</td>
<td><strong>1,165</strong></td>
</tr>
<tr>
<td><strong>TOTAL BUDGET</strong></td>
<td></td>
<td><strong>20,581</strong></td>
</tr>
</tbody>
</table>

The total budget of this project is **TWENTY THOUSAND FIVE HUNDRED AND EIGHTY ONE EUROS (Taxes excluded)**.
Annex

“Fault detection in wind turbines using PCA and statistical hypothesis testing”

TFG presentat per obtenir el títol de GRAU en ENGINYERIA MECÀNICA
Per Josep Mª Serrahima de Cambra

Barcelona, 8 de Juny de 2016

Director: Francesc Pozo Montero
Departament Matemàtica Aplicada III(MAIII)
Universitat Politècnica de Catalunya (UPC)
ANNEX

This annex is formed by some of the programs that I have created to obtain the solutions that are explained through the entire project.

A.1. Basic program simple inference

This program loads the samples, scales them, does the PCA and the statistical comparison between the baseline and the sample to diagnose.

```matlab
clear all
clc
N = 13;
n = 50;
nfallos = 8; % The different faults
L = 48001; % This implies a step time of 0.0125 s.
% I will consider the last 6.25 s, that is, 500 elements.
rL =500; % The number of time instants is reduced to 500.
mostra = 16;
for i=1:n
    var=strcat('SimulacioSaludableRegio3_',num2str(i));
    var=strcat(var,'.mat');
    load(var);
    OutData(:,5)=[];
    for j=1:N
        for k=1:(mostra+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
for i=1:N % We just compute the standard deviation of the first 50 rows
    % Standard deviation of all the healthy measures of sensor 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
        % princomp centers X by subtracting off column means
        XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
    end
end
% PCA
[coeff,score,latent] = princomp(XT(1:50,:),'econ');
T = XT*coeff; %matrix in the new coordinates

% FALLOS
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

for i=1:N  % We just compute the standard deviation of the first 50 rows
    dt(i)=std(reshape(X(1:n,(i-1)*rL+1:i*rL),1,n*rL));
end

% XT is the scaled matrix after the group-scaling
YT(:,(i-1)*rL+j) = (Y(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);

% YT is the Y (faulty) to diagnose, after the group scaling
Td = YT*coeff; % matrix in the new coordinates

sample = 50;  % size of the matrix
[nrow,ncol] = size(T);
iter1 = floor((nrow-50)/sample);
scr = 1;
clear CH
total=zeros(1,4);
[nrowy,ncoly]= size(Td);
iter2 = floor((nrowy)/sample);
CH = zeros(iter1+iter2,1);
for scr = 1:4
    % STANDART DEVIATION
    sh2 = std(T(1:50,scr))^2;
    for i=1:iter1
        sc2 = std(T(n+sample*(i-1)+1:n+sample*i,scr))^2;
        dof = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
        tobs = (mean(T(1:50,scr)) - mean(Td(1+sample*(i-1)+1:n+sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(Td(1+sample*(i-1)+1:n+sample*i,scr))^2/sample);
        tstar10 = tinv(.82,dof); % significance=36%
        CH(i,scr) = (abs(tobs)<=tstar10);
    end
end

for scr=1:4
    % STANDART DEVIATION
    sh2 = std(T(1:50,scr))^2;
    for i=1:iter2
        sc2=std(Td(1+sample*(i-1):sample*i,scr))^2;
        dof= floor (((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
        tobsy=(mean(T(1:50,scr)) - mean(Td(1+sample*(i-1):sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(Td(1+sample*(i-1):sample*i,scr))^2/sample);
        tstar10y = tinv(.82,dof); % significance=36%
end
Fault detection in wind turbines using PCA and statistical hypothesis testing

CH(i+iter1,scr) = (abs(tobsy)<=tstar10y); %CH is the matrix that saves if a sample is healthy (1) or faulty (0)
end
end

suma=zeros(1,8);
for scr = 1:4
    suma(1,scr*2-1) = sum(CH(1:iter1,scr)); %the 16 first samples are healthy, so suma should be 16 if all samples are correctly diagnosed
    suma(1,scr*2) = iter2-sum(CH(iter1+1:iter1+iter2,scr)); %8 samples are faulty, so suma should be 8 if all samples are correctly diagnosed
end
display(suma); %suma is the vector with the results of the detections
display(CH);

A.2. Variation of the size of the sample (Simple inference)

This code calculates the efficiency on the method when we reduce the number of rows (size of the sample). Additionally, it plots the results and calculates the ROC curves for five different sizes.

% This program solves iteratively our problem, maintaining all the time the % same number of time instants per row (rL=500), and changing the length of % the sample that is compared with the pattern (sample=50,45,...,5).
clear all
clc
N = 13;
n = 50;
nfallos = 8; % The different faults
L = 48001; % This implies a step time of 0.0125 s.
rL=500; %The number of time instants is fixed in this program
suma=zeros(10,8);
mostra = 16;
for resultats=1:10
    clear X Y XT YT Td
    % Import healthy simulations and fill matrix X
    for i=1:n
        var=strcat('SimulacioSaludableRegio3_',num2str(i));
        var=strcat(var,'.mat');
        load(var);
        OutData(:,5)=[];
        for j=1:N
            for k=1:(mostra+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
    % Group scaling of X
    for i=1:N
        % We just compute the standard deviation of the first 50 rows
        % Standard deviation of all the healthy measures of sensor 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
    % princomp centers X by subtracting off column means
    XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
end
% PCA
[coeff,score,latent] = princomp(XT(1:50,:), 'econ');
% T is the matrix to diagnose with all the healthy simulations
T = XT*coeff;
% Import faulty simulations and fill matrix Y
for fallo=1:nfallos
    var=strcat('SimulacioFallo',num2str(fallo));
    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
% Group scaling of Y
for i=1:N
    % We just compute the standard deviation of the first 50 rows
    % Standard deviation of all the healthy measures of sensor 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
        % princomp 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)))/dt(i);
    end
end
% T is the matrix to diagnose with all the healthy simmulations after the
% group scaling
Td = YT*coeff;
% Statistical inference
clear CH
clear nrow ncol nrowy ncoly
% sample is the variable that changes in each iteration
sample=50-5*(resultats-1);
[nrow,ncol] = size(T);
iter1 = floor((nrow-50)/sample);
sr = 1;
[nrowy,ncoly]= size(Td);
iter2 = floor((nrowy)/sample);
CH = zeros(iter1+iter2,1);
% Creation of the CH matrix
for scr = 1:4
    % STANDART DEVIATION
    sh2 = std(T(1:50,scr))^2;
    for i=1:iter1
        sc2 = std(T(n+sample*(i-1)+1:n+sample*i,scr))%^2;
        dof = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
        tobs = (mean(T(1:50,scr))-mean(T(n+sample*(i-1)+1:n+sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,scr))^2/sample);
        tstar10 = tinv(.82,dof);  % nivell de significació: 36%
        CH(i,scr) = (abs(tobs)<=tstar10);
Fault detection in wind turbines using PCA and statistical hypothesis testing

end
end
for scr=1:4

% STANDART DEVIATION
sh2 = std(T(1:50,scr))^2;
for i=1:iter2
sc2=std(Td(1+sample*(i-1):sample*i,scr))^2;
dofy=floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
tobsy=(mean(T(1:50,scr))-mean(Td(1+sample*(i-1):sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(Td(1+sample*(i-1):sample*i,scr))^2/sample);
tstar10y = tinv(.82,dofy); % nivell de significació: 36%
CH(i+iter1,scr) = (abs(tobsy)<=tstar10y);
end

end
% To create the suma matrix with ten columns, to show results for every iteration
for scr = 1:4
suma(resultats,scr*2-1) = sum(CH(1:iter1,scr))/iter1*100;
suma(resultats,scr*2) = (iter2-sum(CH(iter1+1:iter1+iter2,scr)))/iter2*100;
end
display(suma);

%% Plots
valorsx=5:5:50;
for i=1:4
figure(i)
healthy=(suma(10:-1:1,i*2-1)');
faulty=(suma(10:-1:1,i*2)');
%plotting options
set(gcf,'DefaultLineMarkerSize',8);
set(gcf,'DefaultLineLineWidth',2);
set(gcf,'DefaultAxesFontSize',12);
set(gcf,'DefaultAxesFontName','times');
plot(valorsx, healthy,'g-o','MarkerFaceColor','green','MarkerSize',8);
hold on
title('Sensitivity and specificity as a function of the size of the sample ($\nu$)','$\nu$','interpreter','latex');
xlabel('size of the sample ($\nu$)','$\nu$','interpreter','latex');
ylabel('correct decisions ($\%$)','$\%$','interpreter','latex');
grid on
plot(valorsx, faulty, 'r-s','MarkerFaceColor','red','MarkerSize',8);
axis([5 50 0 100]);
legend('healthy (specificity)','faulty (sensitivity)','$\%,Location','southeast');
hold off
var = strcat('Grafic_r1500_variacionsample_score_',num2str(i));
var1 = strcat(var,'.eps');
var2= strcat(var,'.png');
print(var1,'-depsc2');
print(var2,'-dpng');
end

figure (5);
score1h=(suma(10:-1:1,1)');
score2h=(suma(10:-1:1,3)');
score3h=(suma(10:-1:1,5)');
hold on
set(gcf,'DefaultLineMarkerSize',8);
set(gcf,'DefaultLineLineWidth',2);
set(gcf,'DefaultAxesFontSize',12);

set(gcf,'DefaultAxesFontName','times');
plot(valorsx, score1h, 'g-o', 'MarkerFaceColor', 'green', 'MarkerSize', 8);
hold on
title('Specificity as a function of the size of the sample ($\nu$)', 'Interpreter','latex');
grid on;
xlabel('size of the sample ($\nu$)', 'Interpreter','latex');
ylabel('correct decisions ($%$)', 'Interpreter','latex');
plot(valorsx, score2h, 'm-s', 'MarkerFaceColor', 'm', 'MarkerSize', 8);
plot(valorsx, score3h, 'b-*', 'MarkerFaceColor', 'red', 'MarkerSize', 8);
axis([5 50 0 100]);
legend('Score 1', 'Score 2', 'Score 3', 'Location', 'southeast');
hold off

%% ROC curve for rl=500 and sample=10:10:50
AUC=zeros(5,4);
for sample=50:-10:10
    clear X Y XT YT T Td
    rL=500; %The number of time instants is fixed in this program
    mostra = 16;
    for i=1:n
        var=strcat('SimulacioSaludableRegio3_',num2str(i));
        var=strcat(var,'.mat');
        load(var);
        OutData(:,5)=[];
        for j=1:N
            for k=1:(mostra+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
    for i=1:N % We just compute the standard deviation of the first 50 rows
        % Standard deviation of all the healthy measures of sensor 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 princomp centers X by subtracting off column means
            XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
        end
    end
    % PCA
    [coeff,score,latent] = princomp(XT(1:50,:),',econ');
    T = XT*coeff;

    % FALLOS
    for fallo=1:nfallos
        var=strcat('SimulacioFallo',num2str(fallo));
        var=strcat(var,'.mat');
        load(var);
        OutData(:,5)=[];
        for j=1:N
            for k=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
end
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
        YT(:,(i-1)*rL+j) = (Y(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
    end
end
Td = YT*coeff;
% mida dels grups
[nrow,ncol] = size(T);
iter1 = floor((nrow-50)/sample);
scr = 1; % ara primer score, després entra en un for
total=zeros(1,4);
[nrowy,ncoly]= size(Td);
iter2 = floor((nrowy)/sample);
CHroc = zeros(iter1+iter2,1);
for alpha = 1:49
    for scr = 1:4
        % STANDART DEVIATION
        sh2 = std(T(1:50,scr))^2;
        for i=1:iter1
            sc2 = std(T(n+sample*(i-1)+1:n+sample*i,scr))^2;
            dof = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
            tobs = (mean(T(1:50,scr))-mean(T(n+sample*(i-1)+1:n+sample*i,scr)))/sqrt((sh2/n)^2/n+sc2/sample)^2/sample);
            tstar10 = tinv(1-0.01*alpha,dof); % nivell de significació: 36%
            CHroc(i,scr) = (abs(tobs)<=tstar10);
        end
        FPR(alpha,scr) = (iter1-sum(CHroc(1:iter1,scr)))/(iter1);
    end
end
for alpha=1:49
    for scr=1:4
        % STANDART DEVIATION
        sh2 = std(T(1:50,scr))^2;
        for i=1:iter2
            sc2 = std(Td(1+sample*(i-1):1+sample*i,scr))^2;
            dofy = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
            tobsy = (mean(T(1:50,scr))-mean(Td(1+sample*(i-1):1+sample*i,scr)))/sqrt((sh2/n)^2/n+sc2/sample)^2/sample);
            tstar10y = tinv(1-0.01*alpha,dofy); % nivell de significació: 36%
            CHroc(i+iter1,scr) = (abs(tobsy)<=tstar10y);
        end
        TPR(alpha,scr) = (iter2-sum(CHroc(i+iter1+iter2,scr)))/(iter2);
    end
end
fig = sample/10+4;
figure(fig)
%plotting options
set(gcf,'DefaultLineMarkerSize',8);
set(gcf,'DefaultLineLineWidth',2);
set(gcf,'DefaultAxesFontSize',12);
set(gcf,'DefaultAxesTickLabelInterpreter','latex');plot(FPR(:,1),TPR(:,1),'r-o','MarkerFaceColor','red','MarkerSize',8)
hold on
A.3. Time instants per row (simple inference)

This program solves iteratively the problem changing the number of columns in each matrix, i.e. the time instants saved are different in each iteration and different (500, 450, ..., 50). For each of them the group scaling, PCA and statistical inference is done. As a final result we obtain the matrix suma, where the number of right decisions for each mostra (16 samples of healthy simulations, 8 faulty simulations), and also four plots are obtained, one for each of the principal scores.

clear all
clc
N = 13;
n = 50;
nfallos = 8; % The different faults
L = 48001; % This implies a step time of 0.0125 s.
suma=zeros(10,8);
for resultats=1:10
    rL =500-(resultats-1)*50;
    mostra = 16;
    % Borrarem de la memoria totes les variables (matrius) per a no obtenir % resultats erroris
    clear X Y XT YT Td
    for i=1:n
        var=strcat('SimulacioSaludableRegio3_',num2str(i));
        var=strcat(var,'.mat');
        load(var);
        OutData(:,5)=[];
        for j=1:N
            for k=1:(mostra+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
end

% generació fitxer EPS
var = strcat('ROC_with_sample_',num2str(sample));
var1 = strcat(var,'.eps');
var2=strcat(var,'.png');
print(var1,'-depsc2');
print(var2,'-dpng');

%Area under the curve
for scr=1:4
    AUC(sample/10,scr)=trapz(FPR(:,scr),TPR(:,scr));
end

display(AUC);
Fault detection in wind turbines using PCA and statistical hypothesis testing

for i=1:N % We just compute the standard deviation of the first 50 rows
% Standard deviation of all the healthy measures of sensor 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
  % princomp centers X by subtracting off column means
  XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
end
% PCA
[coeff,score,latent] = princomp(XT(1:50,:),'econ');
T = XT*coeff;
% FALLOS
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
for i=1:N % We just compute the standard deviation of the first 50 rows
% Standard deviation of all the healthy measures of sensor 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
  % princomp centers X by subtracting off column means
  XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
end
% Td és la matriu Y (fallos) per a diagnosticar, després de fer el group
% scaling
Td = YT*coeff;
clear CH
clear nrow ncol nrowy ncoly
sample = 50; % mida dels grups
[nrow,ncol] = size(T);
iter1 = floor((nrow-50)/sample);
scr = 1; % ara primer score, després entra en un for
[nrowy,ncoly]= size(Td);
iter2 = floor((nrowy)/sample);
CH = zeros(iter1+iter2,1);
for scr = 1:4 % STANDART DEVIATION
  sh2 = std(T(1:50,scr))^2;
  for i=1:iter1
    sc2 = std(T(n+sample*(i-1)+1:n+sample*i,scr))^2;
    dof = floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1)));
    tobs = (mean(T(1:50,scr))-mean(T(n+sample*(i-1)+1:n+sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,scr))^2/sample);
    tstar10 = tinv(.82,dof); % nivell de significació: 36%
    CH(i,scr) = (abs(tobs)<=tstar10);
  end
end
end

end
end

for scr=1:4

% STANDART DEVIATION
sh2 = std(T(1:50,scr))^2;
for i=1:iter2
    sc2=std(Td(1+sample*(i-1):sample*i,scr))^2;
dofy=floor(((sh2/n+sc2/sample)^2)/((sh2/n)^2/(n-1)+(sc2/sample)^2/(sample-1))));
tobsy=(mean(T(1:50,scr))-mean(Td(1+sample*(i-1):sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(Td(1+sample*(i-1):sample*i,scr))^2/sample);
tstar10y = tinv(.82,dofy); % nivell de significació: 36%
    CH(i+iter1,scr) = (abs(tobsy)<tstar10y);
end
end

% La manera més eficaç de crear la matriu amb 10 columnes:
for scr = 1:4
    suma(resultats,scr*2-1) = sum(CH(1:iter1,scr))/iter1*100;
    suma(resultats,scr*2) = (iter2-sum(CH(iter1+1:iter1+iter2,scr)))/iter2*100;
end
end

% Una iteració amb resultats=1..9 per a calcular els valors amb rL=5..50
for resultats=1:9
    rL =50-(resultats)*5;
    mostra = 16;
    % Borrarem de la memoria totes les variables (matrius) per a no obtenir
    % resultats errornis
    clear X Y XT YT Td
    for i=1:n
        var=strcat('SimulacioSaludableRegio3_',num2str(i));
        var=strcat(var,'.mat');
        load(var);
        OutData(:,5)=[];
        for j=1:N
            for k=1:(mostra+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
end

for i=1:N % We just compute the standard deviation of the first 50 rows
    % Standard deviation of all the healthy measures of sensor 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
        % princomp centers X by subtracting off column means
        XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
    end
end

% PCA
[coeff,score,latent] = princomp(XT(1:50,:),'econ');
T = XT*coeff;
% STANDART DEVIATION
sh2 = std(T(1:50,1))^2;
% FALLOS
for fallo=1:nfallos
    var=strcat('SimulacioFallo',num2str(fallo));
    var=strcat(var,'.mat');
Fault detection in wind turbines using PCA and statistical hypothesis testing

```matlab
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
for i = 1:N
    % We just compute the standard deviation of the first 50 rows
    % Standard deviation of all the healthy measures of sensor i=1:N
    dt(i) = std(reshape(X(1:n,(i-1)*rL+1:i*rL),1,n*rL));
    for j = 1:rL
        % XT is the scaled matrix after the group-scaling
        % princomp 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)))/dt(i);
    end
end
% Td és la matriu Y (fallos) per a diagnosticar, després de fer el group
% scaling
Td = YT*coeff;
clear CH
clear nrow ncol nrowy ncoly
sample = 50; % mida dels grups
[nrow,ncol] = size(T);
iter1 = floor((nrow-50)/sample);
scr = 1; % ara primer score, després entra en un for
[nrowy,ncoly] = size(Td);
iter2 = floor((nrowy)/sample);
CH = zeros(iter1+iter2,1);
for scr = 1:4
    for i = 1:iter1
        sc2 = std(T(n+sample*(i-1)+1:n+sample*i,scr))^2;
        dof = floor(((sh2/n+sc2/sample)^2)/(sh2/n)^2/(n-1)+((sc2/sample)^2/(sample-1)));
        tobs = (mean(T(1:50,scr))-mean(T(n+sample*(i-1)+1:n+sample*i,scr)))/sqrt(std(T(1:50,scr))^2/n+std(T(n+sample*(i-1)+1:n+sample*i,scr))^2/sample);
        tstar10 = tinv(.82,dof); % nivell de significació: 36%
        CH(i,scr) = (abs(tobs)<=tstar10);
    end
end
for scr = 1:4
    for i = 1:iter2
        sc2 = std(Td(1+sample*(i-1):sample*i,scr))^2;
        dofy = floor(((sh2/n+sc2/sample)^2)/(sh2/n)^2/(n-1)+((sc2/sample)^2/(sample-1)));
        tobsy = (mean(Td(1:50,scr))-mean(Td(1+sample*(i-1):sample*i,scr)))/sqrt(std(Td(1:50,scr))^2/n+std(Td(1+sample*(i-1):sample*i,scr))^2/sample);
        tstar10y = tinv(.82,dofy); % nivell de significació: 36%
        CH(i+iter1,scr) = (abs(tobsy)<=tstar10y);
    end
end
for scr = 1:4
    suma(resultats+10,scr*2-1) = sum(CH(1:iter1,scr))/iter1*100;
    suma(resultats+10,scr*2) = (iter2-sum(CH(1:iter1+iter1+iter2,scr)))/iter2*100;
end
end
display(suma);```
%% Plot the different effectivity for each of the number of columns.
b=5:5:45;
valorsx=50:50:500;
valorsx=[b valorsx];
for i=1:4
    figure(i)
    healthy=(suma(19:-1:1,i*2-1)');
    faulty=(suma(19 :-1:1,i*2)');
    %plotting options
    set(gcf,'DefaultLineMarkerSize',8);
    set(gcf,'DefaultLineLineWidth',2);
    set(gcf,'DefaultAxesFontSize',12);
    set(gcf,'DefaultAxesFontName','times');
    plot(valorsx, healthy,'g-o','MarkerFaceColor','green','MarkerSize',8);
    hold on
    title({'Sensitivity and specificity as a function of;  
    the number of time instants ($L$) per row'},'Interpreter','latex');
    grid on;
    xlabel('number of time instants ($L$) per row','Interpreter','latex');
    ylabel('correct decisions ($\%$)');
    plot(valorsx, faulty, 'r-s', 'MarkerFaceColor', 'red', 'MarkerSize', 8);
    axis([5 500 0 100]);
    legend('healthy (specificity)','faulty (sensitivity)', 'Location', 'southeast');
    hold off
    var=strcat('Grafic_rL_500_a_5_variation_score_',num2str(i));
    var1 = strcat(var, '.eps');
    var2=strcat(var, '.png');
    print(var1, '-depsc2');
    print(var2, '-dpng');
end
a=5:5:50;
for i=1:4
    figure(i+10)
    healthy=(suma(19:-1:10,i*2-1)');
    faulty=(suma(19 :-1:10,i*2)');
    %plotting options
    set(gcf,'DefaultLineMarkerSize',8);
    set(gcf,'DefaultLineLineWidth',2);
    set(gcf,'DefaultAxesFontSize',12);
    set(gcf,'DefaultAxesFontName','times');
    plot(a, healthy,'g-o','MarkerFaceColor','green','MarkerSize',8);
    hold on
    title({'Sensitivity and specificity as a function of;  
    the number of time instants ($L$) per row'},'Interpreter','latex');
    grid on;
    xlabel('number of time instants ($L$) per row','Interpreter','latex');
    ylabel('correct decisions ($\%$)');
    plot(a, faulty, 'r-s', 'MarkerFaceColor', 'red', 'MarkerSize', 8);
    axis([5 50 0 100]);
    legend('healthy (specificity)','faulty (sensitivity)', 'Location', 'southeast');
    hold off
    var=strcat('Grafic_rL_de50a5_variation_score_',num2str(i));
    var1 = strcat(var, '.eps');
    var2=strcat(var, '.png');
    print(var1, '-depsc2');
    print(var2, '-dpng');
end
A.4. Basic program multiple inference

This program uses multiple inference to diagnose the samples. The first parts are the same as the simple inference (loading the data, group scaling and PCA). Additionally, this program makes the 3-dimensional plots of the baseline and the damages.

```matlab
% Program that uses multivariate statistical inference to study the structure of the Wind Turbine
clear all
clc
N = 13;
n = 50;
nfallos = 8; % The different faults
L = 48001; % This implies a step time of 0.0125 s.
rL=500;
mu=zeros(10,8);
mostra = 16;
% Import of the data and creation of the X matrix
for i=1:n
    var=strcat('SimulacioSaludableRegio3_',num2str(i));
    var=strcat(var,'.mat');
    load(var);
    OutData(:,5)=[];
    for j=1:N
        for k=1:(mostra+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
for i=1:N % We just compute the standard deviation of the first 50 rows
    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
        % princomp centers X by subtracting off column means
        XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
    end
end
% PCA
[coeff,score,latent] = princomp(XT(1:50,:),'econ');
T = XT*coeff;

% FALLOS
for fallo=1:nfallos
    var=strcat('SimulacioFallo',num2str(fallo));
    var=strcat(var,'.mat');
    load(var);
    OutData(:,5)=[];
    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
for i=1:N % We just compute the standard deviation of the first 50 rows
    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
        % princomp centers X by subtracting off column means
        XT(:,(i-1)*rL+j) = (X(:,(i-1)*rL+j)-mean(X(1:n,(i-1)*rL+j)))/dt(i);
    end
end
```

Fault detection in wind turbines using PCA and statistical hypothesis testing
\[ YT(:, (i-1)*rL+j) = (Y(:, (i-1)*rL+j) - \text{mean}(X(1:n, (i-1)*rL+j))) / \text{dt}(i); \]

end

Td = YT*coeff;

\%
\% 3d plots
\%
figure(1);
x3dsimple1=(T(1:200,1));
y3dsimple1=(T(1:200,2));
z3dsimple1=(T(1:200,3));
scatter3(x3dsimple1, y3dsimple1, z3dsimple1, 'MarkerEdgeColor','k', 'MarkerFaceColor',[0 .75 .75]);
view(-30,10); grid on;
legend('baseline', 'Location', 'southeast');
grid on;
xlabel('PC1', 'Interpreter', 'latex');
ylabel('PC2', 'Interpreter', 'latex');
zlabel('PC3', 'Interpreter', 'latex');

figure(2);
x3dsimple2damage1=(Td(1:50,1));
y3dsimple2damage1=(Td(1:50,2));
z3dsimple2damage1=(Td(1:50,3));
hold on;
scatter3(x3dsimple2damage1, y3dsimple2damage1, z3dsimple2damage1, 'MarkerEdgeColor','k', 'MarkerFaceColor',[0 .75 .75]);
scatter3(Td(201:250,1), Td(201:250,2), Td(201:250,3), 'MarkerEdgeColor','r');
scatter3(Td(301:350,1), Td(301:350,2), Td(301:350,3), 'MarkerEdgeColor','g');
view(-30,10); grid on;
legend('Damage 1', 'Damage 2', 'Damage 3', 'Location', 'southeast');
grid on;
xlabel('PC1', 'Interpreter', 'latex');
ylabel('PC2', 'Interpreter', 'latex');
zlabel('PC3', 'Interpreter', 'latex');
hold off;

\%
\% MULTI VARIATE INFERENCE
\%
% We will begin by picking a number of principal components (s=12)
% The number of samples that will be used to create the population
% mean vector will be (sample=50)
resultat=zeros(20,2);
suma=zeros(24,1);
\% for s=1:20
s=12;
sample=50;

% Calculatation of population mean vector (mu)
mu=zeros(s,1);
for scr = 1:s
mu(scr,1)=sum(T(1:sample,scr))/50;
end

% Sample vector mean
[nrow, ncol] = size(T);
[nrowy, ncoly]= size(Td);
iter1=floor((nrow-1)/sample);
iter2=floor((nrowy)/sample);
vectormean=zeros(s,iter1+iter2);
for scr=1:s
    for i=1:iter1
        vectormean(scr,i)=mean(T(n+sample*(i-1)+1:n+sample*i,scr));
    end
end
for scr=1:s
    for i=1:iter2
        vectormean(scr,iter1+i)=mean(Td(sample*(i-1)+1:sample*i,scr));
    end
end
for iteracio=1:24
    S=0;
    clear xi T2
    %Sample covariance matrix
    if iteracio<=16
        for i=1:sample
            %We create a vertical vector with the values of each row of T
            xi=T(n+i+sample*(iteracio-1),1:s)';
            %We create the matrix when transposing the second vector
            S=S+(xi-mu)*(xi-mu)';
        end
    else
        for i=1:sample
            %We create a vertical vector with the values of each row of T
            xi=Td(i+sample*(iteracio-17),1:s)';
            %We create the matrix when transposing the second vector
            S=S+(xi-mu)*(xi-mu)';
        end
    end
    %The sample covariance matrix must be multiplied by (1/sample)
    S=(1/sample)*S;
    %Hotelling's T^2
    T2=sample*(vectormean(:,iteracio)-mu)'*inv(S)*(vectormean(:,iteracio)-mu);
    alpha=10; %we select the significance to be equal to 10%
    if T2<=(sample-1)*s/(sample-s)*finv(1-alpha/100,s,sample-s)
        suma(iteracio,1)=1;
    else
        suma(iteracio,1)=0;
    end
    resultat(s,1)=sum(suma(1:16));
    resultat(s,2)=sum(suma(17:24));
end
display(resultat);