POLITECNICO DI MILANO

FACOLTA DI INGEGNERIA
DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA
DELL'AUTOMAZIONE

A tool for the integration of FMECA and diagnostic analysis

Relatore: Ing. Macchi Marco
Correlatore: Ing. Luca Fumagalli

Tesi di Laurea di:
Berno Paolo
Matr. n. 740495

Anno Accademico 2009/2010

Alla mia famiglia

Contents

List of figures

List of tables

List of attachments

Abstract

La Manutenzione negli anni recenti ha ottenuto spazi sempre più importanti nell'Ingegneria dei Sistemi, assumendo un ruolo di primo piano già nelle fasi preliminari della progettazione di impianti industriali ed infrastrutture. Questa maturazione nasce dall'esigenza di abbandonare la vecchia concezione della manutenzione intesa come "costo necessario" da minimizzare durante l'esercizio del sistema, per pervenire ad una nuova presa di coscienza da parte degli addetti ai lavori e soprattutto dei Manager che oggi riconoscono alla disciplina un ruolo di "strumento di profitto". La presente tesi ha lo scopo di mostrare come sia possibile, partendo da concetti presenti in letteratura riguardanti la gestione della manutenzione, sviluppare un software che riesca a supportare la diagnosi di diverse condizioni operative di una macchina a partire dalla sola firma elettrica, derivabile dalla misurazione dell'andamento della potenza assorbita nel tempo. Obiettivo principale del software è poi quello di mostrare l'integrazione della FMECA – un tipico strumento dell'ingegneria e pianificazione della manutenzione – con l'analisi diagnostica, allo scopo ultimo di ottenere informazioni sempre aggiornate riguardo a possibili guasti.

Questo permette di validare lo strumento sia come supporto per gli utilizzatori della macchina, in un'ottica tradizionale di gestione della manutenzione, sia come strumento a supporto di un servizio di manutenzione che può essere offerto dal costruttore della macchina stessa. Inoltre, collegando in maniera stretta la FMECA all'analisi diagnostica, è possibile dimostrare come sfruttare al meglio i dati del monitoraggio di macchina ai fini dell'ingegneria di manutenzione e, di più, come rendere "continue" – mentre, cioè, l'impianto sta funzionando – le analisi che, ad oggi, nel migliore dei casi, vengono fatte con frequenza bassa (es. una FMECA può essere fatta su base annua o addirittura a seguito di specifici episodi come la modifica d'impianto). Questo stretto legame tra FMECA ed analisi diagnostica è visto, nella letteratura scientifica più recente, come foriero di un miglioramento importante delle attuali pratiche di gestione della manutenzione.

Abstract

In order to be competitive, it is necessary for companies to continuously increase the effectiveness and efficiency of their production processes. Current production strategies demand high availability of production equipment in order to keep low cost and guarantee the customer satisfaction. Therefore, maintenance has gained in importance as a support function for ensuring equipment availability, quality products, on-time deliveries, and plant safety.

Well-performed maintenance implies seeing as few corrective maintenance actions as possible while performing as little preventive maintenance as possible, pursuing always an optimization of maintenance costs. This might seem as a utopia, but during the past decades strategies and concepts have evolved to support this. One of these is condition based maintenance. In condition based maintenance, critical item characteristics are monitored (through, for example, vibration or temperature monitoring) in order to gain early indications of an incipient failure. Research activity shows that condition based maintenance has not been implemented on a wide basis. Therefore, the purpose of this research is to investigate how a condition based maintenance approach can be implemented in an industrial setting. This is done thanks to the development of a method that can integrate FMECA - a typical engineering instrument about maintenance planning - with a failure diagnostic tool.

The software designed can be seen as support for the users of the machine and also as useful instrument for maintenance service, that can be offered by the manufacturers.

Executive Summary

La Manutenzione negli anni recenti ha ottenuto spazi sempre più importanti nell'Ingegneria dei Sistemi, assumendo un ruolo di primo piano già nelle fasi preliminari della progettazione di impianti industriali ed infrastrutture. Questa maturazione nasce dall'esigenza di abbandonare la vecchia concezione della manutenzione intesa come "costo necessario" da minimizzare durante l'esercizio del sistema, per pervenire ad una nuova presa di coscienza da parte degli addetti ai lavori e soprattutto dei Manager che oggi riconoscono alla disciplina un ruolo di "strumento di profitto" che, come tale, deve essere pianificato e progettato, sin dalla concezione del progetto, e quindi approfondito, corretto e migliorato durante tutto il ciclo di vita del sistema.
Questa crescita ha raggiunto livelli di maturazione diversa nei diversi settori industriali: in alcuni settori la disciplina è ormai consolidata (Aeronautico, Spaziale, Nucleare), in molti altri settori ha avuto una rapida crescita negli ultimi anni (trasporti, petrolchimica), in altri ancora è agli esordi ma anche in questi ambiti ci si aspetta nei prossimi anni un consolidamento (manifatturiero, energetico).
Il lavoro di questa tesi si propone di illustrare come sia possibile integrare più strumenti di analisi usati, fino ad oggi, nel campo della manutenzione in maniera separata; la concretizzazione del concetto di integrazione proposto nella tesi si avrà attraverso lo sviluppo di un corrispondente, unico strumento software.
Va detto innanzi tutto che quando si parla di guasto, se ne parla in senso lato includendo non solo i guasti dei componenti hardware (valvole, pompe, trasformatori, …), bensì considerando anche gli errori umani ("guasti" dell'operatore o del manutentore) o errori del software di gestione e controllo ("guasti" software), gli eventi esterni ambientali (alluvioni, sismi, incidenti esterni connessi ad attività antropiche, ecc.)
In genere per guasto si intende la perdita di una funzionalità, il mancato soddisfacimento di un obiettivo di impianto, sistema, sotto-sistema o componente.
Al fine di identificare tutti i guasti e le relative conseguenze si procede con un approccio di decomposizione del problema in problemi più semplici. Ad esempio se si sta ragionando in termini strutturali si può pensare di scomporre l'impianto in diversi sistemi, i sistemi in sottosistemi, i sottosistemi in componenti. Dopo di che si vanno ad esaminare in modo sistematico tutti i modi di guasto dei singoli componenti valutandone le conseguenze a livello locale ma anche sistemico. L'approccio qui indicato sinteticamente è noto come Failure Mode,

Effect and Criticality Analysis (FMECA) e rappresenta uno degli strumenti implementati nello strumento software sviluppato nel presente lavoro.

L'obiettivo principale di questo lavoro, infatti, è quello di mostrare come sia possibile integrare la FMECA con un'analisi diagnostica, per permettere di ottenere informazioni sempre aggiornate riguardo a possibili guasti.

L'analisi diagnostica è stata sviluppata partendo da un'analisi approfondita del materiale disponibile in letteratura.

La tecnica implementata è quella della Principal Component Analysis (PCA), un metodo non parametrico per estrarre l'informazione contenuta in un insieme di dati apparentemente confusi per ridondanza, rumore o inadatto riferimento.

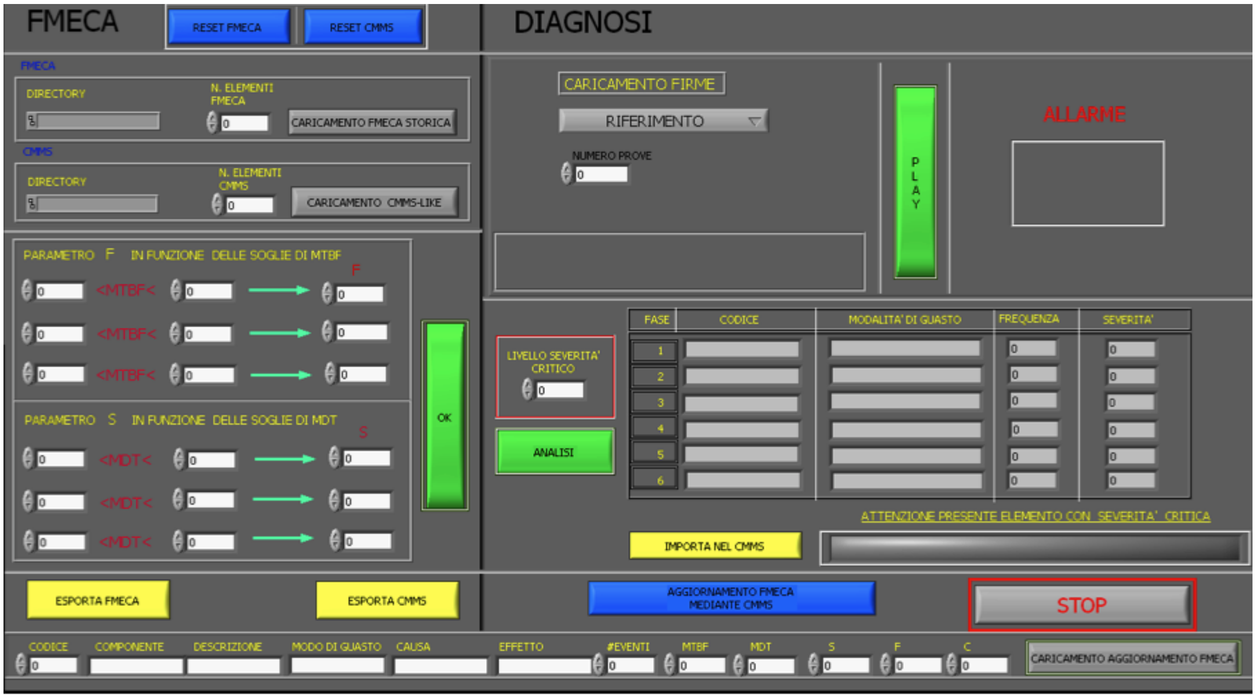
L'utilizzo della PCA in questo contesto consiste nel riconoscere se il dato corrente di funzionamento (es. di una macchina o equipaggiamento industriale) rientra in una partizione precedentemente definita come *normale* sulla base di dati ottenuti precedentemente. Se al contrario il dato corrente se ne discosta, verrà segnata un'anomalia al sistema di supervisione, attraverso l'interfaccia grafica sviluppata.

Il segnale analizzato all'interno di questo lavoro si riferisce alla potenza assorbita, graficata nel dominio del tempo. Questo tipo di informazione è legata al concetto di firma elettrica.

Attraverso l'implementazione congiunta della FMECA e dell'analisi diagnostica è stato così possibile sviluppare un tool in grado non solo di riconoscere la presenza di un modo di guasto, ma anche di isolarlo rispetto al componente degradato, identificarlo con precisione e mostrarne la severità all'operatore che utilizza lo strumento.

Lo strumento software proposto è stato sviluppato utilizzando due ambienti commerciali molto diffusi in ambito industriale: LabView 8.5 e Matlab2010.

Con il primo si è potuta ottenere un'interfaccia grafica (figura sotto) mentre con il secondo si sono gestiti i calcoli matematici e le varie funzioni su cui si basa l'analisi diagnostica legata alla PCA.



Lo strumento così ottenuto è stato quindi testato attraverso alcune prove svolte su una macchina utensile, con la quali è stato possibile validarlo.

Questo software è da considerarsi come supporto sia per gli utilizzatori della macchina, in un’ottica tradizionale di gestione della manutenzione industrale, sia come strumento a supporto di un servizio di manutenzione che può essere offerto dal costruttore della macchina stessa.

List of definitions

Term	Description
Condition based maintenance	Preventive maintenance based on performance and/or parameter monitoring and the subsequent actions
Condition based maintenance system	A system that uses performance and/or parameter monitoring to determine (and if possible schedule) maintenance actions autonomously or in interactions with other systems or humans
Conditional probability of failure	The probability that a failure will occur in a specific period provided that the item concerned has survived to the beginning of that period
Corrective maintenance	Maintenance carried out after fault recognition and intended to put an item into a state in which it can perform a required function
Diagnosis	Fault recognition and identification
Failure	Termination of the ability of an item to perform a required function
Failure consequence	The way in which a failure mode or a multiple failure matters
Failure effect	What happened when a failure mode occurs
Failure mode	A single event that causes a functional failure
Fault	State of an item characterized by inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources

Item	Any part, component, device, subsystem, functional unit, equipment or system that can be individually considered.
Maintenance	Combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function.
Monitoring	Activity, performed either manually or automatically, intended to observe the actual state of an item
Potential failure	A potential failure is an identifiable physical condition which indicates a functional failure is imminent
Predetermined maintenance	Preventive maintenance carried out in accordance with established intervals of time or number of units of use but without previous condition investigation.
Predictive maintenance	Condition based maintenance carried out following a forecast derived from the analysis and evaluation of significant parameters of the condition of the item.
Preventive maintenance	Maintenance carried out at predetermined intervals or according to prescribed criteria and intended to reduce the probability of failure or degradation of functioning of an item.
Prognosis	Prediction of when a failure may occur.

1
Production systems and maintenance

This chapter attempts to summarise the recent research and developments in diagnostics of mechanical systems implementing maintenance.
It presents theory and definitions regarding production and manufacturing systems, failure, faults and maintenance.

1.1 Background

Increased productivity is a key issue for manufacturing companies to stay competitive on a global market. Success, and even survival, in manufacturing requires continuous development and improvement in the way products are being produced.
The production systems of today are often guided by a complex production strategy and it is increasingly important that production is available to meet customer demand. As the trends of the new production strategies also imply working with fewer inventories, the production systems become even more vulnerable to unplanned unavailability.
Good product design is of course essential for products with high reliability. However, no matter how good the product design is, products deteriorate over time since they are operating under certain stress or load in the real environment, often involving randomness.
Maintenance as production support, has thus become very important to ensure equipment availability, quality products, on-time deliveries and plant safety. Even so, maintenance is still considered a cost center in many companies[ref.2].

Maintenance as a support function in production systems has been valued as a critical role and even as a prerequisite. This, of course, also implies that maintenance must be performed effectively, in other words, the correct maintenance action should be taken at the proper time.

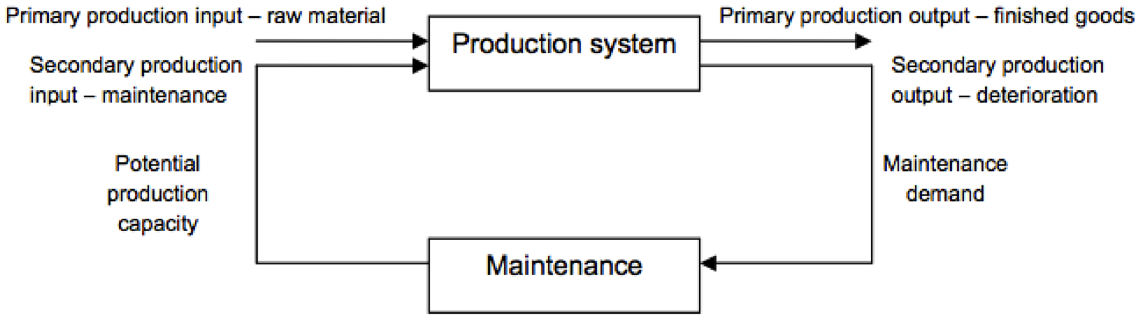


Figure 1. The need and effect of maintenance on a production system.
Obviously, additional support functions are necessary in order to run a production system.
However, as the Figure visualizes, maintenance plays a vital role in upholding production capacity.

According to Simeu-Abazi and Sassine (2001), the prime target of maintenance should be to ensure the system function of production equipment. Further, maintenance should provide the right parameters of: cost, reliability, maintainability, and productivity, for any automated manufacturing system.

Various approaches to performing maintenance exist. Also, various definitions of maintenance have been suggested through the years, the common point being that they have moved away from the traditional perception of maintenance to repair broken items. Maintenance is defined as a: “Combination of all technical, administrative, and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function.”

Indeed, until the advent of CNC machines maintenance was largely unplanned, it took place when a breakdown occurred. There is no doubt that it was inefficient. The machine could be out service at the most incovenient times, there had to be a larger invetories of work in progress in case of breakdown and a breakdown crew had to be always available.

The development of CNC machines with the possibility of unmanned production certainly was one issue that caused a review of maintenance strategies.

It is well known in the companies that maintenance can be performed in two major types: corrective maintenance and preventive maintenance. Both of the traditional maintenance types are widely used in practically all industrial sectors [ref.15].

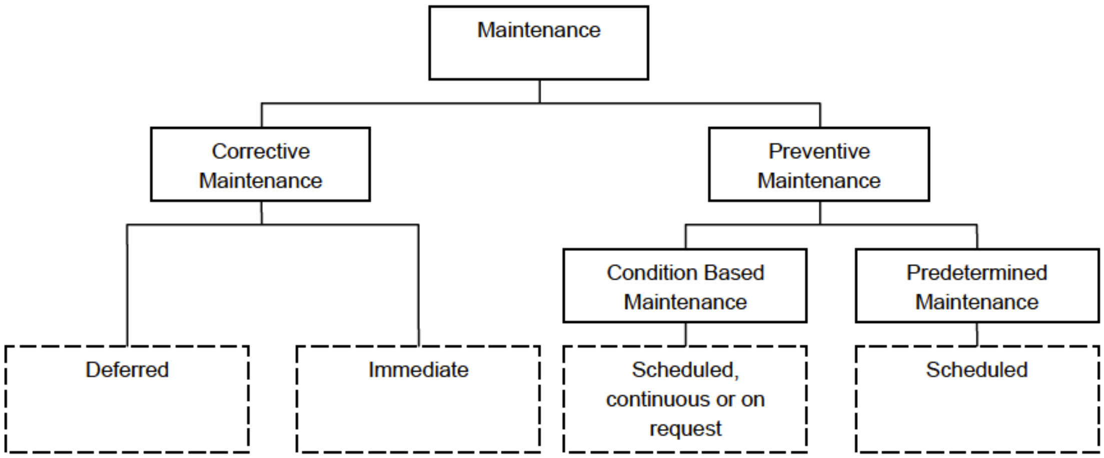


Figure 2. Overview of different maintenance types

Corrective maintenance, similar to repair work, is undertaken after a breakdown or when obvious failure has been located. Corrective maintenance is defined as: Maintenance carried out after fault recognition and intended to put an item into a state in which it can perform a required function. For repair work, some modeling approaches are available. With minimal repair, the failed item is only restored to its functioning state and the item continues as if nothing has happened. If the item instead is replaced by a new component of the same type, or if it is restored to an "as good as new" condition, the failure rate will decrease to the level of when the item was just put into use.

For failures on critical functions, corrective maintenance has to be performed immediately. However, for failures that have no or little consequence on the comprehensive system function, the maintenance can be deferred in time to a better-suited occasion.

Conversely, preventive maintenance has been defined as: Maintenance carried out at predetermined intervals or according to prescribed criteria and intended to reduce the probability of failure or the degradation of the functioning of an item.

Preventive maintenance is divided into two types, predetermined maintenance and condition based maintenance.

Predetermined maintenance is scheduled and planned without the occurrence of any monitoring activities. The scheduling can be based on the number of hours in use, the number of times an item has been used; the number of kilometers the items has been used, according to prescribed dates, and so on. Predetermined maintenance is best suited to an item that has a visible age or wearout characteristic and where maintenance tasks can be made at a time that for sure will prevent a failure from occurring.

In this strategy the machine is operated until a predetermined time when maintenance is carried out. The strategy has some advantages: it allows planning of maintenance resources, planning of the timing of downtime and planning for replacement.

Its main disadvantage lies with the value placed upon the predetermined time between maintenance procedures. It leads to possibly unnecessary downtime of the machine and the oversupply of replacement elements at scheduled maintenance periods.

Planning maintenance intervals based on age are not always the best approach; other alternatives should then be consulted. Although many failures are not related to age, most of them give incipient warnings that they are in the process of failing (Moubray, 1997). Thus, even if the great majority of machine tool users are still practising planned schedule maintenance, some users is wishing to implement condition based maintenance, that is the other preventive maintenance type.

Condition based maintenance does not utilize predetermined intervals and schedules [11]. Instead, it monitors the condition of items in order to decide on a dynamic preventive schedule.

Condition based maintenance is based on measuring of working data, on monitoring of variables that describe the degrade of equipment.

These measurements should allow the prediction of the time to failure for all elements and thus allow maintenance to be planned before any elements fail.

The need for condition based maintenance was revealed as early as in the 1960's through a study performed during the development of the preventive maintenance program for the Boeing 747. The study purpose was to determine the failure characteristics of aircraft components. The study was, at the request of the Department of Defense (USA), documented and published by Nowlan and Heap in 1978 [ref.23].

Condition based maintenance can be described in different way. A first attempt to explain it, can be through the description of the "Potential failure to Failure curve, P-F curve (see Figure). Consulting the P-F curve for a particular failure mode can give indications about what type of on-condition task is Appropriate [ref.24]. Obviously, in order to be effective, on-condition tasks must be performed in intervals shorter than the P-F interval.

Moubray defines an on-condition task as: “A scheduled task used to determine whether a potential failure has occurred.”, and further divides the on-condition techniques into four categories:

- condition monitoring technologies,
- techniques based on product quality,
- primary effects monitoring techniques, and
- inspection techniques based on the human senses.

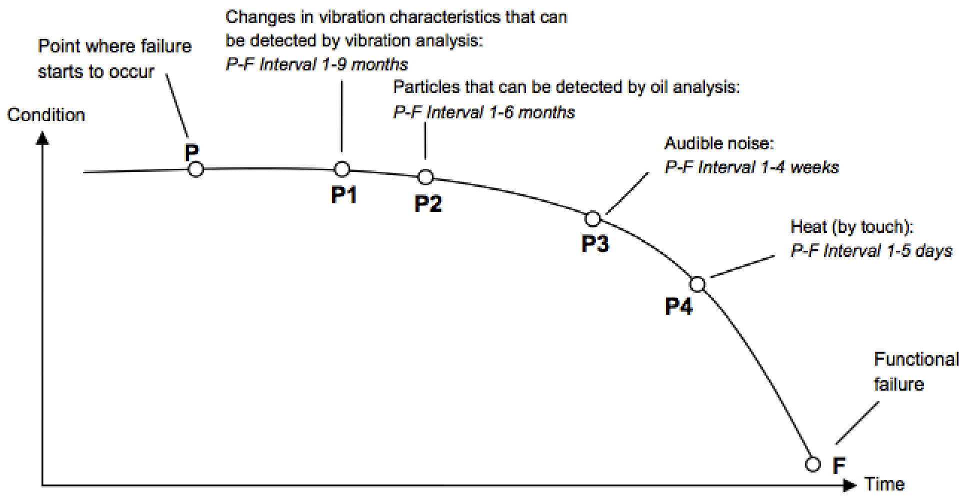


Figure 3.The potential failure to failure curve of a ball bearing

Often, different potential failure conditions can precede a failure mode. The P-F interval of these potential failure conditions can vary a great deal, choosing more than one potential failure condition as a warning can be a good idea. As an example, an incipient ball bearing failure might start with changes in high frequency vibration characteristics, followed by increasing particle content in lubricating oil, audible noise, and, finally, heat builds up in the bearing caps [ref.27].

Failures in production systems can create many inconveniences. It is possible to list possible problems resulting from system failure, all of which can generate massive costs [ref.28]:

• lost production time,
• volume of lost production,
• mass of harmful chemicals into the environment,
• lost customers,
• warranty payments,
• cost of mobilization of emergency resources, and
• insurance cost

A production system generates value when being utilized productively, and obviously costs money in an unproductive state.

1.2 The purpose of the thesis

The aim of this project is to develop a cheap diagnostic tool that can control the health of an industrial machine. The main idea of this research it to link the main aspects of maintenance in an unique software, based on LabView 8.5 and Matlab 2009.
The idea is to link diagnostic analysis with FMECA (Failure modes, effects and criticality analysis) and the analysis of failure history.

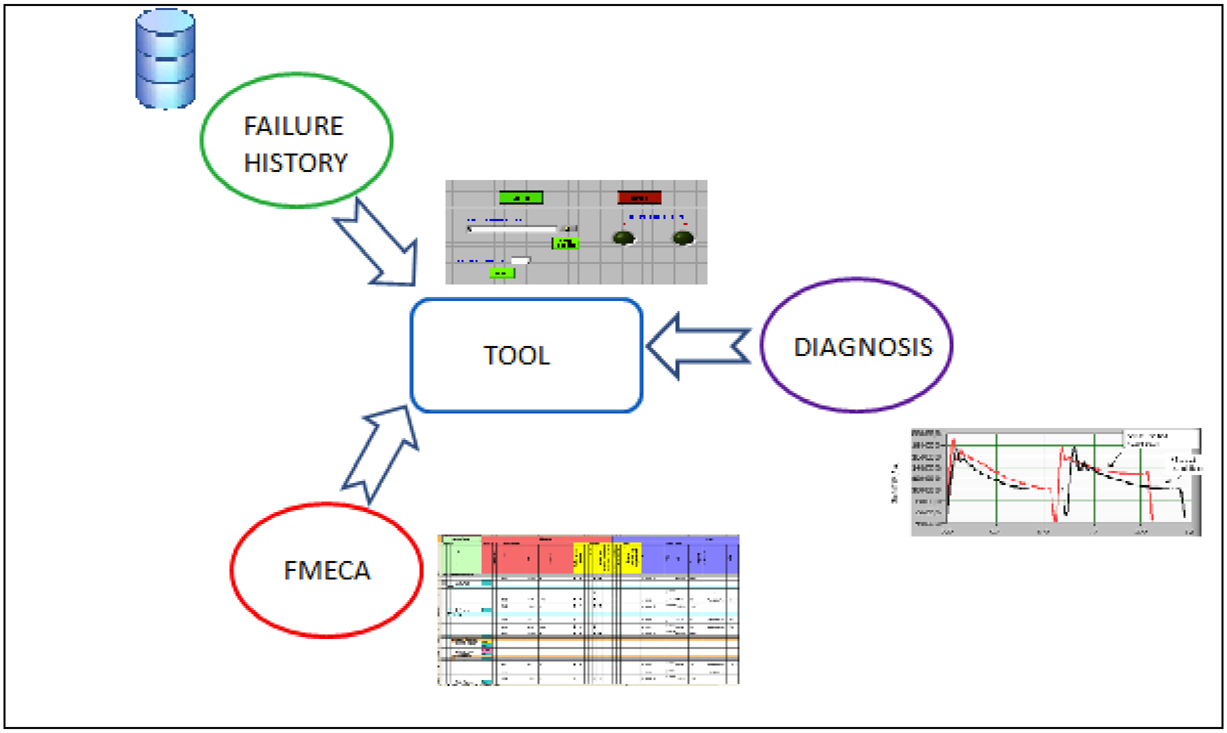


Figure 4. Purpose of thesis: integrate FMECA, DIAGNOSIS and FAILURE HISTORY

In fact, a good diagnosis and proper knowledge of failure history and plant criticality allow maintenance to be more accurate in analyzing the problems occurring on the production system. The reasons and the technique implemented to obtain this interaction will be explained in detail in the next chapters.

The project tries to analyze different fault conditions that could happen on an industrial machine (see case study used for validation that is presented in chapter 6).

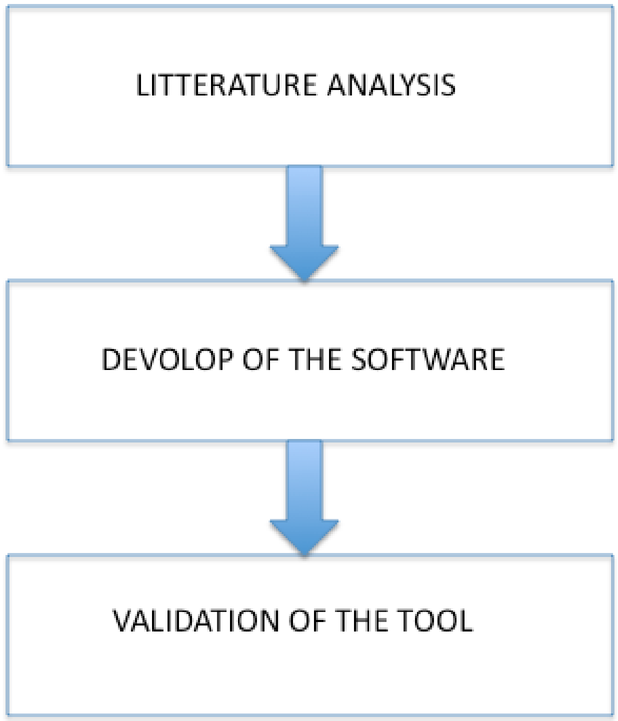


Figure 5. Overview of the project

Diagnosis, based on Principle component analysis (P.C.A), starts with the analysis of electrical power signal characteristics (E.S.A), acquired from industrial machine.
The reason of linking diagnosis with FMECA is based on the idea to have good indicators about severity and probability of diagnosed failure. Using these parameters, in fact the maintenance operator can understand how the failure is dangerous for the plant and she/he can decide if it is needed to make maintenance or postpone it. The particular aspect of this tool is the possibility to update FMECA according to the faults in a “real time” way.
In fact when a failure is found, it is added to a list of history of failure (CMMS). From these data, such as breakdown date and time duration of breakdown, it is possible to compute some parameters that are used to update regularly the indicators of FMECA.

Summarizing, the goal of this tool is therefore:

- to add important technical information about failure to diagnosis.
- to obtain relevant information from the failure history.
- to update FMECA in real time.

1.3 Structure of the thesis

The thesis is divided into 6 chapters. The present Chapter 1 contains the introduction, with a background and problem discussion, followed by the purposes of the research. Chapters 2 and 3 contain a theoretical framework. Chapter 2 introduces the structure of FMECA and why it is important in a maintenance problem. Meanwhile, Chapter 3 introduces the development of a Fault Detection and Isolation (FDI) system based on an adaptive Principal Component Analysis (PCA) algorithm. Chapter 4 presents what is discussed in literature about interaction between Diagnosis and FMECA through reference of papers.

Chapter 5 and 6 present the industrial machine of the case study and describes the project, the implementation method and the software. Chapter 7 presents a discussion of the conclusions and suggestions on future research.

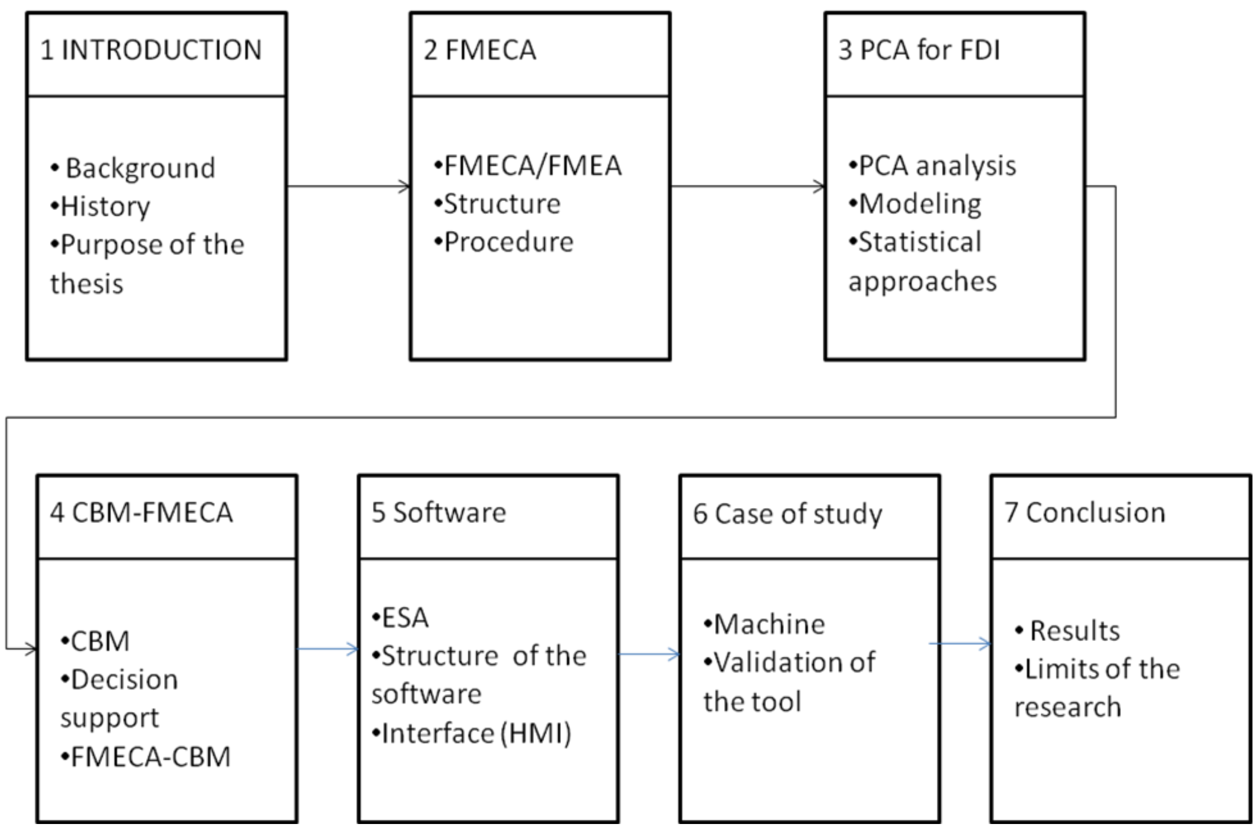


Figure 6. Structure of the thesis.

2

FMECA

This chapter attempts to analyze FMECA in all its parts: what FMECA, the procedure, the structure, etc.

2.1 Introduction

Traditionally, reliability has been achieved through extensive testing and use of techniques such as probabilistic reliability modeling. These are techniques done in the late stages of development. The challenge is to design in quality and reliability early in the development cycle of a product.

Therefore engineers introduced *Failure Modes and Effects Analysis* (FMEA). FMEA is a methodology for analyzing potential reliability problems early in the development cycle where it is easier to take actions to overcome these issues, thereby enhancing reliability through design. FMEA is used to identify potential failure modes, determine their effect on the operation of the product, and identify actions to mitigate the failures. A crucial step is anticipating what might go wrong with a product. While anticipating every failure mode is not possible, the development team should formulate as extensive a list of potential failure modes as possible.

The early and consistent use of FMEA in the design process allows the engineer to design out failures and produce reliable, safe, and customer pleasing products. FMEA does also capture historical information for use in future product improvement.[ref.16]

In this project it is used FMECA, an evolution of FMEA. The FMECA is composed of two separate analyses, the FMEA and the Criticality Analysis (CA). The FMEA must be completed prior to performing the CA. It provides the added benefit of showing a quantitative ranking of system and/or subsystem failure modes. The Criticality Analysis allows the analysts to identify reliability and severity related with particular components or systems. In this chapter how it is possible to complete FMEA analysis is presented, thus providing the basis from which perform FMECA.

2.1.1 *What is FMEA - Failure Modes and Effects Analysis?*

The National Aeronautics and Space Administration (NASA) define FMEA as a forward logic (bottom-up), tabular technique that explores the ways or modes in which each system element can fail and assesses the consequences of each of these failures. For them FMEA is a useful tool for cost and benefit studies to implement effective risk mitigation and countermeasure.

For Wikipedia, the free online encyclopedia, Failure Mode and Effect Analysis is a method that examines potential product or process failures, evaluates risk priorities, and helps determine remedial actions to avoid identified problems. It is an integral part of any ISO 9000 compliant quality system.

All these definitions have some terms in common. There is always a system and an examination of potential failures. After that follows an assessment of the identified failures [ref.4].

So, it is possible to define Failure Mode and Effects Analysis like an analysis technique which facilitates the identification of potential problems in the design or process by examining the effects of lower level failures.

Recommended actions or compensating provisions are made to reduce the likelihood of the problem occurring, and mitigate the risk, if in fact, it does occur.

The FMEA team determines, by failure mode analysis, the effect of each failure and identifies single failure points that are critical. It may also rank each failure according to the criticality of a failure effect and its probability of occurring (SAE J1739, section 5 *POTENTIAL FAILURE MODE AND EFFECTS ANALYSIS FOR TOOLING & EQUIPMENT).*

2.1.2 *What is FMECA - Failure Modes Effects and Criticality Analysis?*

The next step in the FMEA evolution was FMECA. FMECA is an acronym for Failure Modes and Effects Criticality Analysis. The American Society for Quality define it as a procedure that is performed after a failure mode effects analysis to classify each potential failure effect according to its severity and probability of occurrence [ref.7].

FMECA is used to support a number of different engineering activities. These include:

- **Reliability Analyses** – The FMECA should identify areas of greatest concern from a logistic and / or safety viewpoint. Such areas should then be targeted for possible design changes which may for instance improve reliability.
- **Maintainability Analyses** – The FMECA often records and / or highlights areas of the design which require some form of scheduled maintenance activity.
- **Testability Analyses** – FMECA often includes a detailed analysis of detection methods including any Built In Test (BIT).
- **Safety Analyses** – failure mode criticality results will often feed into the Fault Tree Analyses (FTA).

2.2 FMEA/FMECA Procedure

There are several different approaches to do a Failure Modes and Effects Analysis. One possible way is described in the following paragraph.

2.2.1 Preparation

Before undertaking an FMEA it is essential to undertake certain preparatory steps. The scope will depend on the complexity of the system being studied. First we have to define the system and its mission which should be analyzed. After that a description of the operation of the system has to be performed. And in the next steps the failure categories and the environmental conditions should be identified and described.

We start with describing the product or process and its function. An overall understanding of the product or process is very important. This understanding simplifies the process of analysis by helping the engineer identify those product/process uses that fall within the intended function and which ones fall outside. It is important to consider both intentional and unintentional uses since product failure often ends in litigation, which can be costly and time consuming.

2.2.2 Creating a Block Diagram

In the next step we are creating a Block Diagram of the product or process. This diagram shows major components or process steps as blocks connected together by lines that indicate how the components or steps are related. The diagram shows the logical relationships of components and establishes a structure around which the FMEA can be developed. The block diagram should always be included with the FMEA form.

2.2.3 Header of the FMEA Form worksheet

In this step we have to use a table like table n.1. If items are components, list them in a logical manner under their subsystem/assembly based on the block diagram.
After that we have to identify Failure Modes. A failure mode is defined as the manner in which a component, subsystem, system, process, etc. could potentially fail. A failure mode in one component can serve as the cause of a failure mode in another component. Each failure should be listed in technical terms.
At this point the failure mode should be identified whether or not the failure is likely to occur. Looking at similar products or processes and the failures that have been documented for them is an excellent starting point.
Then, it is needed to describe the effects of those failure modes. For each failure mode identified the engineer should determine what the ultimate effect will be. A failure effect is defined as the result of a failure mode on the function of the product/process as perceived by the customer. They should be described in terms of what the customer might see or experience should the identified failure mode occur.

The customer is to see as internal as well as external one. Some examples of failure effects are e.g. injury to the user, inoperability of the product or process, degraded performance, noise, etc.

Part	Function	Potential Failure Mode	Potential Effects of Failure	Severity	Potential causes of failure	Occurrence	How will the potetial failure be detected?	Detection	RPN	Action

Table 1. Header of FMECA

2.2.4 Severity

Severity is an assessment of the seriousness of the effect and refers directly to the potential failure mode being studied. The Customer in process FMEA is both the internal and where appropriate, external Customer. The severity ranking is also an estimate of how difficult it will be for the subsequent operations to be carried out to its specification in Performance, Cost, and Time. The Ranking and suggested criteria are listed in table n.2.

A common industry standard scale uses 1 to represent no effect and 10 to indicate very severe with failure affecting system operation and safety without warning. The intent of the ranking is to help the analyst determine whether a failure would be a minor nuisance or a catastrophic occurrence to the customer. This enables the engineer to prioritize the failures and address the real big issues first.

Effect	**Criteria**	**Severity of Effect**	**Rank**
None		No effect	**1**
Very minor	Minor disruption to production line	A portion of the product may have to be reworked. Defect not noticed by average customers; cosmetic defects.	**2**
Minor	Minor disruption to production line.	A portion of the product may have to be reworked. Defect noticed by average	**3**

		customers; cosmetic defects.	
Very low	Minor disruption to production line.	The product may have to be sorted and reworked. Defect noticed by average customers; cosmetic defects.	**4**
Low	Some disruption to product line.	100% of product may have to be reworked. Customer has some dissatisfaction. Item is fit for purpose but may have reduced levels of performance.	**5**
Moderate	Some disruption to product line.	A portion of the product may have to be scrapped. Customer has some dissatisfaction. Item is fit for purpose but may have reduced levels of performance	**6**
High	Some disruption to product line.	Product may have to be sorted and a portion scrapped. Customer dissatisfied. Item is useable but at reduced levels of performance.	**7**
Very high	Major disruption to production line.	100% of product may have to be scrapped. Loss of primary function. Item unusable. Customer very dissatisfied	**8**
Hazard with warning	May endanger machine or operator.	Failure occurs with warning. The failure mode affects safe	**9**

		operation and involves noncompliance with regulations	
Hazard without warning	May endanger machine or operator	Failure occurs without warning. The failure mode affects safe operation and involves noncompliance with regulations	**10**

Table 2. Standard scale to assess the effects of a fault

2.2.5 Causes of failure mode

Identify the causes for each failure mode. A failure cause is defined as a design weakness that may result in a failure. The potential causes for each failure mode should be identified and documented. The causes should be listed in technical terms and not in terms of symptoms. Examples of potential causes include improper torque applied, Improper operating conditions, too much solvent, improper alignment, excessive voltage etc.[18]

2.2.6 Criticality analysis: Occurrence

The Occurrence is the assessment of the probability that the specific cause of the Failure mode will occur. A numerical weight should be assigned to each cause that indicates how likely that cause is (probability of the cause occurring).

For that failure history is helpful in increasing the truth of the probability. Therefore historical data stored in databases can be used and questions like the following are very helpful to solve this problem.

- What statistical data is available from previous or similar process designs?
- Is the process a repeat of a previous design, or have there been some changes?

- Is the process design completely new?
- Has the environment in which the process is to operate changeable?
- Have mathematical or engineering studies been used to predict failure?

A common industry standard scale uses 1 to represent unlikely and 10 to indicate inevitable. The Ranking and suggested criteria are can seen in table n.3.

Notional probability of failure	Evaluated failure rates	Cpk	Rank
Remote: Failure is unlikely. No Failures ever associated with almost identical processes	1 in 1,500,000	>1.67	**1**
Very low: Only Isolated Failures associated with almost identical processes	1 in 150,000	1.50	**2**
Low: Isolated Failures associated with similar processes	1 in 15,000	1.33	**3**
Moderate: Generally associated with processes similar to previous processes Failures, but not in 'major' proportions	1 in 2,000	1.17	**4**
	1 in 400	1.00	**5**
	1 in 80	0.83	**6**
High: Generally associated with processes similar to previous processes that have often failed	1 in 20	0.67	**7**
	1 in 8	0.51	**8**
Very high: Failure is almost inevitable	1 in 3	0.33	**9**
	1 in 2	<0.33	**10**

Table 3. A common industry standard scale to assess the criticality of a fault

2.2.7 Detection

Here we have to distinguish between two types of detection. On one hand we have to identify Current Controls (design or process).

Current Controls (design or process) are the mechanisms that prevent the cause of the failure mode from occurring or which detect the failure before it reaches the Customer. The engineer should now identify testing, analysis, monitoring, and other techniques that can or have been used on the same or similar products/processes to detect failures. Each of these controls should be assessed to determine how well it is expected to identify or detect failure modes.

After a new product or process has been in use previously undetected or unidentified failure modes may appear. The FMEA\FMECA should then be updated and plans made to address those failures to eliminate them from the product/process.

The other thing is to assess the probability that the proposed process controls will detect a potential cause of failure or a process weakness. Assume the failure has occurred and then assess the ability of the Controls to prevent shipment of the part with that defect. Low Occurrence does not mean Low Detection - the Control should detect the Low Occurrence. Statistical sampling is an acceptable Control. Improving Product and/or Process design is the best strategy for reducing the Detection ranking - Improving means of Detection still requires improved designs with its subsequent improvement of the basic design. Higher rankings should question the method of the Control. The ranking and suggested criteria are shown in table n.4.

Detection	**The likelihood the Controls will detect a Defect**	**Rank**
Almost Certain	Current controls are almost certain to detect the Failure Mode. Reliable detection controls are known with similar processes.	**1**
Very High	Very High likelihood the current controls will detect the Failure Mode.	**2**
High	High likelihood that the current controls will detect the Failure Mode.	**3**
Moderately High	Moderately high likelihood that the current controls will detect the Failure Mode.	**4**
Moderate	Moderate likelihood that the current controls will detect the Failure Mode.	**5**
Low	Low likelihood that the current controls will detect the Failure Mode	**6**
Very Low	Very Low likelihood that the current controls will detect the Failure Mode	**7**

Remote	Remote likelihood that the current controls will detect the Failure Mode	**8**
Very Remote	Very Remote likelihood that the current controls will detect the Failure Mode	**9**
Almost Impossible	No known controls available to detect the Failure Mode.	**10**

Table 4. The ranking and different criteria to assess Detection

2.2.8 Risk Priority Numbers (RPN)

The Risk Priority Number is a mathematical product of the numerical Severity, Probability, and Detection ratings:

RPN = (Severity) x (Probability) x (Detection)

The RPN is used to prioritize items than require additional quality planning or action.

2.2.9 Actions

Determine Recommended Action(s) to address potential failures that have a high RPN. These actions could include specific inspection, testing or quality procedures; selection of different components or materials; de-rating; limiting environmental stresses or operating range; redesign of the item to avoid the failure mode; monitoring mechanisms; performing preventative maintenance; and inclusion of back-up systems or redundancy.
After that we have to assign Responsibility and a Target Completion Date for these actions. This makes responsibility clear-cut and facilitates tracking.
Update the FMECA as the design or process changes, the assessment changes or new information becomes known [ref.6]-[ref.4]-[ref.3].

2.3 FMEA/FMECA in this research

Strengths of FMECA include its comprehensiveness, the systematic establishment of relationships between failure causes and effects, and its ability to point out individual failure modes for corrective action in design of an equipment / machine. Main weaknesses include the extensive labor required, the large number of trivial cases considered, and the inability to deal with multiple-failure scenarios or unplanned cross-system effects such as sneak circuits.

Failure Modes, Effects, and Criticality Analysis is however an excellent hazard analysis and risk assessment tool, but it suffers from the above mentioned limitations. Therefore, FMECA should be used in conjunction with other analytical tools when developing reliability estimates.

Another main reason for which people do not fully utilize the FMEA/FMECA result is that they do not know how and when to link the FMEA/FMECA information with the process control functions in the industrial plant.

In this work one possible link between FMEA/FMECA in conjunction with other analytical tools based on diagnosis is analyzed and validated through a software tool in the context of maintenance of a machine tool.

More precisely, through FMEA/FMECA, we can have some important parameters like severity and probability of a failure. It is possible to use these parameters for example to show in real time to a technical operator the seriousness of a fault that is occurring.

Severity and probability will be linked to quantitative measures like mean down time associated to a failure and mean time between failures, so to allow a further automatic generation of FMECA indicators.

Moreover one can use information from historical failure to update FMEA/FMECA, in the next chapters all these links will be analyzed with more details.

3

Principal Component Analysis for Fault Detection and Isolation

This chapter describes the development of a Fault Detection and Isolation (FDI) system based on an adaptive Principal Component Analysis (PCA) algorithm, used to compare the current machine operation state with a "good behaviour" model based on a preliminary set of data.

3.1 Introduction

A progressing fault in machinery will affect certain parameters, such as vibration, noise, temperature, debris, etc. In machinery diagnostics it is essential to analyze external relevant information such as vibrations to judge the condition of internal components, which are usually inaccessible without dismantling the machines. In this work, an electric signal is used to detect incipient faults and the proposed tool is tested in the case proposed in Chapter 6.
Machine fault diagnostics is a procedure of mapping the information obtained in the measurement space and/or features in the feature space to machine faults in the fault space. This mapping process is also called pattern recognition. Traditionally, pattern recognition is done manually with auxiliary graphical tools such as power spectrum graph, phase spectrum graph, cepstrum graph, AR spectrum graph, spectrogram, wavelet scalogram, wavelet phase graph, etc. However, manual pattern recognition requires expertise in the specific area of the diagnostic application. Thus, highly trained and skilled personnel are needed. Therefore, automatic pattern recognition is highly desirable. This can be achieved by classification of signals based on the Information and/or features extracted from the signals. In the following sections, different machine fault diagnostic approaches are discussed with emphasis on statistical approaches and artificial intelligent approaches.

3.1.1 Statistical approaches

A common method of fault diagnostics is to detect whether a specific fault is present or not based on the available condition monitoring information without intrusive inspection of the machine. This fault detection problem can be described as a hypothesis test problem with null hypothesis H0: Fault A is present, against alternative hypothesis H1: Fault A is not present. In a concrete fault diagnostic problem, hypotheses H0 and H1 are interpreted into an expression using specific models or distributions, or the parameters of a specific model or distribution. Test statistics are then constructed to summarize the condition monitoring information so as to be able to decide whether to accept the null hypothesis H0 or reject it.

Cluster analysis, as a multivariate statistical analysis method, is a statistical classification approach that groups signals into different fault categories on the basis of the similarity of the characteristics or features they possess. It seeks to minimise within-group variance and maximise between-group variance. The result of cluster analysis is a number of heterogeneous groups with homogeneous contents: there are substantial differences between the groups, but the signals within a single group are similar.

A natural way of signal grouping is based on certain distance measures or similarity measure between two signals. These measures are usually derived from certain discriminant functions in statistical pattern recognition. Commonly used distance measures are Euclidean distance, Mahalanobis distance, Kullback–Leibler distance and Bayesian distance.

Baydar [ref. 8] investigated the use of an another multivariate statistical technique known as principal component analysis (PCA) for analysis of the time waveform signals in gear fault diagnostics, and this approach (used in this thesis) is discussed in the next paragraph 3.2.

3.1.2 Artifical Intelligence approaches

AI techniques have been increasingly applied to machine diagnosis and have shown improved performance over conventional approaches. In practice, however, it is not easy to apply AI techniques due to the lack of efficient procedures to obtain training data and specific knowledge, which are required to train the models. So far, most of the applications in the literature just used experimental data for model training. In the literature, two popular AI technique for machine diagnosis are artificial neural networks (ANNs) and expert systems (ESs) . Other AI techniques used include fuzzy logic systems, fuzzy–neural networks (FNNs), neural–fuzzy systems and evolutionary algorithms (EAs). An ANN is a computational model that mimics the human brain structure. It consists of simple processing elements connected in a complex layer structure which enables the model to approximate a complex non-linear function with multi-input and multi-output. A processing element comprises a node and a weight. The ANN learns the unknown function by adjusting its weights with observations of input and output. This process is usually called training of an ANN. There are various neural network models, each one with its pros and contras.

The ANN models usually use supervised learning algorithms which require external input such as the a priori knowledge about the target or desired output. For example, a common practice of training a neural network model is to use a set of experimental data with known (seeded) faults. This training process is supervised learning. In contrast to supervised learning, unsupervised learning does not require external input. An unsupervised neural network learns itself using new information available.

In contrast to neural networks, which learn knowledge by training on observed data with known inputs and outputs, ESs utilize domain expert knowledge in a computer program with an automated inference engine to perform reasoning for problem solving. Three main reasoning methods for ES used in the area of machinery diagnostics are rule-based reasoning, case-based reasoning   and model-based reasoning.

ESs and neural networks have their own limitations. One main limitation of rule-based ESs is combinatorial explosion, which refers to the computation problem caused when the number rule increases exponentially as the number of variables increases. Another main limitation is consistency maintenance, which refers to the process by which the system decides when some of the variables need to be recomputed in response to changes in other values. Two main limitations of neural networks are the difficulty to have physical explanations of the trained model and the difficulty in the training process. Obviously, combination of both techniques would significantly improve the performance.

In condition monitoring practice, knowledge from domain specific experts is usually inexact and reasoning on knowledge is often imprecise. Therefore, measures of the uncertainties in knowledge and reasoning are required for ES to provide more robust problem solving. Commonly used uncertainty measures are probability, fuzzy member functions in fuzzy logic theory and belief functions in belief networks theory.

Neural networks and ESs have also been combined with other AI techniques to enhance machine diagnostic systems.

3.1.3. Other approaches

Another class of machine fault diagnostic approaches is the model-based approaches. These approaches utilize physics specific, explicit mathematical model of the monitored machine. Based on this explicit model, residual generation methods such as Kalman filter, parameter estimation (or system identification) and parity relations are used to obtain signals, called residuals, which is indicative of fault presence in the machine. Finally, the residuals are evaluated to arrive at fault detection, isolation and identification. Model-based approaches can be more effective than other model-free approaches if a correct and accurate model is built. However, explicit mathematical modelling may not be feasible for complex systems since it would be very difficult or even impossible to build mathematical models for such systems.
Various model-based diagnostic approaches have been applied to fault diagnosis of a variety of mechanical systems such as gearboxes , bearings , rotors and cutting tools .
The information provided by these methods was shown to be very helpful to having more precise fault identification along with evaluating the confidence of a diagnostic decision.
Petri nets, as a general purpose graphical tool for describing relations existing between conditions and events, have recently been applied to machine fault detection and diagnostics.

3.2 PCA modeling

Principal component analysis is one of the most popular statistical methods, for extracting information from measured data, which finds the directions of significant variability in the data by forming linear combinations of variables.
Consider a data matrix $X \in R^{nxm}$ containing n samples of m process variables collected under normal operation.
This matrix must be normalized to zero mean and unit variance with the scale parameter vectors $\bar{x}$ and s as the mean and variance vectors respectively. Next step to calculate PCA is to construct the covariance matrix R:

$$R = \frac{1}{n-1} X^T X \qquad (1)$$

and performing the SVD decomposition on R:

$$R = V\Lambda V^T \tag{2}$$

where Λ is a diagonal matrix that contains in its diagonal the eigenvalues of R sorted in decreasing order ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$).). Columns of matrix V are the eigenvectors of R. The transformation matrix P $\in R^{mxa}$ is generated choosing a eigenvectors or columns of V corresponding to a principal eigenvalues. Matrix P transforms the space of the measured variables into the reduced dimension space.

$$T = XP \tag{3}$$

Columns of matrix P are called *loadings* and elements of T are called *scores*. Scores are the values of the original measured variables that have been transformed into the reduced dimension space. Operating in equation (3), the scores can be transformed into the original space.

$$\hat{X} = TP^T \tag{4}$$

The residual matrix E is calculated as:

$$E = X - \hat{X} \tag{5}$$

Finally the original data space can be calculated as:

$$X = TP^T + E \tag{6}$$

It is very important to choose the number of principal components a, because TP^T represents the principal sources of variability in the process and E represents the variability corresponding to process noise. There are several proposed procedures for determining the number of components to be retained in a PCA model as:

a) **The SCREE procedure**. It is a graphical method in which one constructs a plot of the eigenvalues in descending order and looks for the *knee* in the curve. The number of selected components are the components between the high component and the *knee*.
b) **Cumulative Percent Variance (CPV) approach**. It is a measure of the percent variance ($CPV(a) \geq 90\%$) captured by the first a principal components is adopted:

$$CPV(a) = \frac{\sum_{i=1}^{a} \lambda_i}{trace(R)} 100 \qquad (7)$$

c) **Cross validation**. [8]

3.2.1 *Statistics for monitoring*

Having established a PCA model based on historical data collected when only common cause variation are present, multivariate control charts based on Hotelling's T^2 and square prediction error (SPE) or Q can be plotted. The monitoring can be reduced to this two variables (T^2 and Q) characterizing two orthogonal subsets of the original space. T^2 represents the major variation in the data and Q represents the random noise in the data. T^2 can be calculated as the sum of squares of a new process data vector x:

$$T^2 = x^T P \Lambda_a^{-1} P^T x \qquad (8)$$

where Λ_a is a squared matrix formed by the first a rows and columns of Λ.
The process is considered *normal* for a given significance level a if:

$$T^2 \leq T_a^2 = \frac{(n^2-1)a}{n(n-a)} F_\alpha(a, n-a) \qquad (9)$$

where $F_\alpha(a, n-a)$ is the critic value of the Fisher-Snedecor distribution with n and n-a degrees of freedom and α the level of significance. α takes values between 90% and 95%. T^2 is based on the first a principal components so that it provides a test for derivations in the latent variables that are of greatest importance to the variance of the process. This statistic will only detect an event if the variation in the latent variables is greater than the variation explained by common causes.

New events can be detected by calculating the squared prediction error *SPE* or *Q* of the residuals of a new observation. *Q* statistic, is calculated as the sum of squares of the residuals. The scalar value Q is a measurement of *goodness of fit* of the sample to the model and is directly associated with the noise:

$$Q = r^T r \tag{10}$$

With:

$$r = (I - PP^T)x \tag{11}$$

The upper limit of this statistic can be computed as the next form:

$$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{12}$$

With:

$$T^2 = x^T P \Lambda_a^{-1} P^T x \tag{13}$$

where c_α is the value of the normal distribution with α the level of significance.

When an unusual event occurs and it produces a change in the covariance structure of the model, it will be detected by a high value of *Q*.

In the figure below it is possible to see in a graphic representation the two statistical parameter : T^2 and Q.

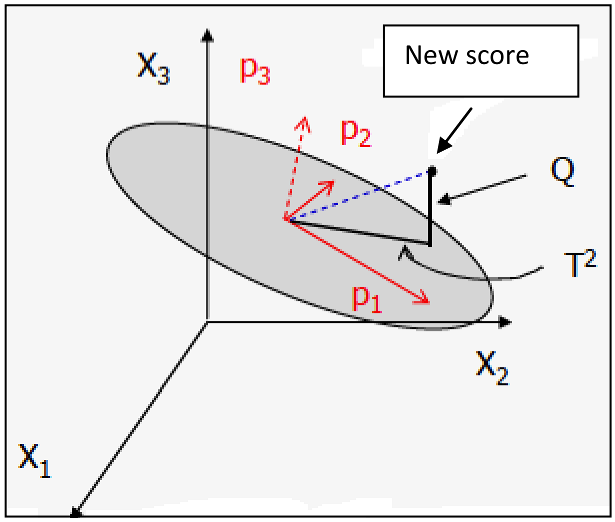


Figure 7. Graphical representation of PCA and parameters Q and T^2

To implement a monitoring and fault detection system based on PCA, it is necessary to consider two tasks:

- **OFF-LINE**

  1. Acquire training data which represents normal process operations.
  2. Scale the training data and obtain the scale parameter vectors 1x and s.
  3. Carry out SVD to obtain PCA model.
  4. Determine the number of principal components and the upper control limits for T^2 and Q statistics.

- **ON-LINE**

  1. Obtain the next testing sample x, and scale it using the scale parameter vectors and s.
  2. Evaluate the T^2 and Q statistics using the obtained PCA model. If one of these exceeds the upper limit, this measurement is considered an alarm. If there are some consecutive established number of alarms, an uncommon event has occurred.
  3. Repeat from step 2.[ref.3]-[ref.8]

3.3 Conclusion

PCA, explained in this chapter, is used in this project to detect a machine failure using Electric Signature Analysis (ESA).
It is decided to use ESA, because in this research detection starts from characteristic features of the time waveform signals as descriptive statistics such as:
- mean peak,
- peak to-peak interval,
- standard deviation,
- crest factor,
- skewness index,
- kurtosis index

The indicators are taken from a previous research activity, during a project that involves companies and Politecnico di Milano about the use of Electric Signature Analysis (ESA).

Using PCA, it is then possible to extract information from these data, and then to detect failure with multivariate control charts based on Hotelling's T^2 and square prediction error (SPE).
This procedure is implemented in Matlab code and integrated in the project with Labview8.5, as explained in Chapter 5.

4

CBM-FMECA

This chapter gives a litterature overview about Condition-based maintenance (CBM), namely the part related with diagnostic analysis, and its links with FMECA.

In literature, there is a great amount of contributions analyzing and proposing different maintenance processes and their applications using a myriad of technologies. From this plenty of diversified information, some authors have made the effort of classifying the maintenance processes and, in this way, contribute to structuring this discipline.

For example, Levrat et al (2008) determine four classes of possible maintenance applications: for strategy, for diagnosis and prognosis, for maintenance policy assessment and maintenance scheduling, and for deployment and implementation.
The proposed tool (showed in next Chapter 5) should contribute to the following (with respect to the categories mentioned by Levrat et al.(2008)): the CBM module has to do with diagnosis and prognosis, the FMECA module could support the maintenance scheduling and finally.
More oriented to the content management, Karim et al (2009) summarise and classify the multiples industrial and academic artefacts than can be used to the maintenance information exchange, useful for systems integration. These information exchange standards are classified in Maintenance Specific Contributions (like S1000D, MIMOSA, PROTEUS, etc) and in Generic Contributions (like XML, ISO 10303 and ISO 15531).

4.1 CBM - Condition based maintenance

Condition-based maintenance (CBM) consists of continuously evaluating the condition of a monitored machine and thereby successfully identifying faults before catastrophic breakdown occurs.

The decision to perform maintenance is reached by observing the "condition" of the system and/or its components. The condition of a system is quantified by parameters that are continuously monitored.

Numerous condition monitoring and diagnostics methodologies are utilizing to identify machine faults to take corrective action.

Condition based maintenance is performed to serve the following two purposes:

• determining if a problem exists in the monitored item, how serious it is, and how long the item can be run before failure, and

• detecting and identify specific components in the items that are degrading and diagnose the problem [ref.1]-[ref.26].

A central part of condition based maintenance is thus monitoring, often called condition monitoring. Monitoring is defined as: "Activity, performed either manually or automatically, intended to observe the actual state of an item."

Condition monitoring can be performed using a number of various approaches and utilizing different levels of technology.

The purpose of monitoring the condition of an item is to collect condition data to make it possible to detect incipient failure so that maintenance tasks can be planned at a proper time. Another purpose of condition monitoring is to increase the knowledge of failure cause and effect and deterioration pattern; indeed, it could be very useful to have a direct connection between CBM and FMECA.

A number of different techniques exist to measure the condition of an item. Depending on the type of potential failure condition one is set out to measure, one or more techniques can be utilized.

It is possible to classify condition monitoring techniques according to the symptoms they are designed to detect:

• dynamic effects, such as vibration and sound,
• particles released into the environment,
• chemicals released into the environment,
• physical effects, such as cracks, fractures, wear, and deformation,
• temperature rises in the equipment, and
• electrical effects, such as resistance, conductivity, dielectric strength, etc.

The condition based maintenance approach thus implies utilizing the results of the monitoring activities (i.e. the potential failures found) and further analyzing them.

It is worth mentioning that, by implementing a condition based maintenance approach, there is much to gain in the form of:

• Reduced maintenance costs, less unnecessary repairs and replacements saving labor, spare parts, and unavailability.
• Damage limitation, incipient failures are easier to repair than breakdowns, also less secondary damage is at stake.
• Reduced production losses.

The comprehensive technology in condition based maintenance can be visualized as a condition based maintenance system (see Figure 8). A condition based maintenance system is defined as: "A system that uses condition based maintenance to determine and schedule predictive maintenance actions autonomously or in interaction with other systems or humans."

A condition based maintenance system can be described in the following way, as proposed by seven modules/activities: data acquisition (sensors), signal processing, condition monitoring, health assessment (diagnostics), prognostics, decision support, and presentation.

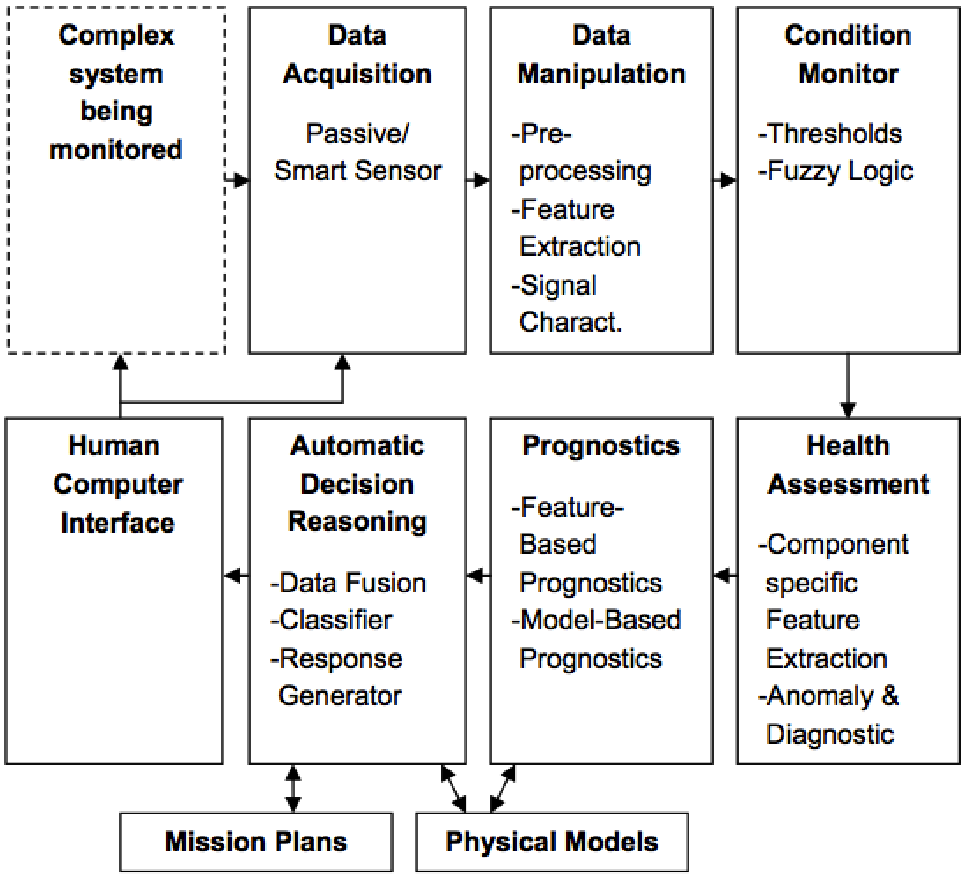


Figure 8. The seven modules in a condition based maintenance system[ref-26].

Data acquisition is thus the first component. Normally, when used in an objective context, sensors are components of the data acquisition and considered parts of a condition monitor module. It is possible to define sensors as: “a device that receives a signal and responds with an electrical signal.” It is thus the equipment that captures the dynamic effect caused by the incipient failure. The purpose of the signal processors is three-fold: (1) to remove distortions and restore the signal to its original shape, (2) to remove irrelevant sensor data for diagnostics or prognostics, and (3) to transform the signal to make relevant features more explicit. In the condition monitoring module, the measured data is compared to normal data with either threshold values or other techniques such as artificial intelligence. If normal levels are exceeded or other unnatural phenomenon occur, such as sudden increases or decreases in the level (but still not exceeding normal levels), the data needs to be diagnosed. Warning limits can be established that are either static or dynamic. Static warning limits utilize pre-determined threshold values. An example of such limits is the ISO [ref.9] , which has produced vibration severity charts for specific types of applications. Static warning limits are more easily administered than dynamic ones. Nonetheless, they lack diagnostic power for predicting when the alarm will be reached.

Diagnostics in condition based maintenance can be divided into three categories: (1) rule-based diagnostics, (2) case-based diagnostics, and (3) model-based diagnostics. Following a diagnosis, the system now has knowledge as to something being unnatural in the condition, where it is unnatural, and what is causing the unnatural measurements; it now needs to be prognosticated. Prognostics can be performed as the diagnostics module, through different techniques of artificial intelligence, such as recurrent neural networks and dynamic wavelet neural networks. The major difference in prognosis compared to diagnosis is that a number of additional parameters need to be taken into consideration. The last step in the condition based maintenance system process is to make a decision concerning what maintenance actions to perform and when. All the previous activities should of course be integrated into a decision support for the best possible solution for this particular event. Here, additional information that has been recovered through this system should be applied, such as production scheduling and labor.

Of course, condition based maintenance and condition based maintenance systems can have different levels of automation, stretching from humans performing all the tasks of the modules to, as explained above, hardware and software performing all those tasks. The table below presents nine levels of automation that can be used to explain different levels of automation in condition based maintenance. Imagining a condition based maintenance system as a computerized operation, the level of automation can, as the table depicts below, range from humans generating all the tasks, deciding one or more, and executing the option(s) to a computer suggesting and executing one option.

LoA	Computerized/Cognitive Tasks/Activities	Mechanized/Physical Tasks/Activities
1	The human generates the options, decides and executes the option without any assistance from the computer.	Entirely manual physical work; no physical tools are used, only human muscular strength.
2	The computer presents all suitable options; the human can then choose and execute one of the options.	Manual physical work supported by a static hand tool.
3	The computer suggests a number of options; the human can then choose and execute one of the options.	Manual physical work supported by a dynamic hand tool.
4	The computer generates a number of options and recommends one of them; the human can then choose to execute that option.	Manual physical work supported by an automated hand tool.
5	The computer suggests one option; the human can then decide, and the computer executes the decision.	Human control of machine/robot on site that executes the task.
6	The computer suggests one option, decides and executes the option; the human is always informed.	Supervision of machine/robot on site that executes the task.
7	The computer suggests one option, decides and executes the option; the human is always informed if the human demands information.	Supervision and control of one or many machines/robots from a central control room.
8	The computer suggests one option, decides and executes the option; the human is only informed if the computer demands that the human should be informed.	Automated physical work by machine/robot; the human is only involved when the machine needs assistance.
9	The computer suggests one option, decides and executes the option without any assistance from the human.	Entirely automated physical work; the machine/robot solves problems by itself when they emerge. The human is never involved.

Table 5. The computerized/cognitive tasks/activities can be used to explain that condition based maintenance systems as well can have different levels of automation.[ref.9]

The tool proposed in this work can be defined within level 4 (according to the classification of Table n.5). Generally a diagnostic tool can fall in level 3. The proposed tool achieves level 4 because the link to FMECA allows highlighting the severity of the failure modes, thus providing a recommendation on which failure mode is more severe.

In many cases, the implementation of a condition based maintenance approach implies that an entire company needs to be involved and old routines need to be changed into new.

4.2 Description of relevant background knowledge

Development of CBM

Traditional CBM (Condition Based Maintenance) is a maintenance program that recommends maintenance actions based on the information collected through condition monitoring. CBM attempts to avoid unnecessary maintenance tasks by taking maintenance actions only when there is evidence of abnormal behaviors of a physical asset. The previous CBM carries out maintenance task that focuses only on condition monitoring and diagnostics.

In recent years, a development of CBM called CBM plus (CBM+) [ref.5] is put forward, which is the application and integration of appropriate process, technologies, and knowledge-based capabilities to improve reliability and maintenance effectiveness . CBM+ has a broad scope. It is built on the concept of CBM, but is optimized by reliability analysis. The original policy for CBM+ was released in November 2002 by the Deputy under Secretary of Defense (DoD) and became popular since 2006.

The ‘‘plus’’ designation represents the extension of CBM with other encompassed technologies, processes, and procedures that enable improved maintenance and logistics practices. CBM+ is not a process in itself. It is a comprehensive strategy to select, integrate, and focus a number of process improvement capabilities, thereby enabling maintenance managers and their customers to attain the desired levels of system and equipment readiness in the most cost-effective manner. At its core, CBM+ is maintenance performed on evidence of need provided by RCM analysis and other enabling processes and technologies.

RCM

RCM was developed in 1970s by the Air Transport Association (ATA), the Aerospace Manufacturers' Associates (AMA), and the US Federal Aviation Administration (FAA). RCM is an industrial improvement approach focused on identifying and establishing the operational, maintenance, and capital improvement policies that will manage the risks of equipment failure most effectively. It is an engineering framework that enables the definition of a complete maintenance regime.

If RCM is correctly applied, it can reduce the amount of routine maintenance work by 40–70%. The benefits of RCM can usually be traced back to two broad categories: risk reductions and cost savings.

To compare CBM, CBM+, and RCM, on one hand, CBM is a traditional maintenance strategy or technology, while CBM+ expands the capability and reliability of CBM; CBM+ focuses on providing the support net required for performing condition based maintenance. The RCM uses CBM as one primary failure management strategy. This relationship is shown graphically in figure n.9 below.

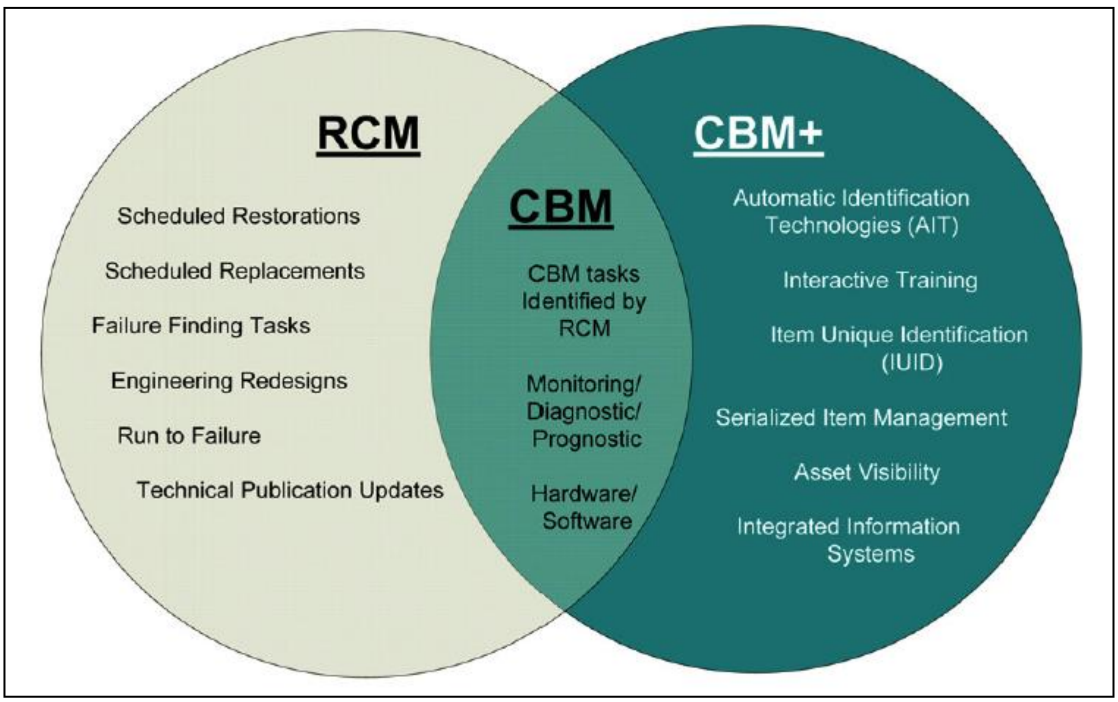


Figure 9.RCM/CBM/CBM+ relationship

Data fusion

Data fusion, which fits many examples in engineering, is identified as the process of combining data and knowledge from different sources with the aim of maximizing the useful information content, for improved reliability or discriminant capability, while minimizing the quantity of data ultimately retained. Based on the different phases, general data fusion structure can be divided into three types: signal-level (or data-level), feature-level, and decision-level.

- **Signal-level fusion**: In this fusion level, all sensor raw data from a measured object are combined directly and a feature vector is then extracted from the fused data. At this stage, a pattern recognition process is performed as shown in Fig.10(a) . Fusion of data at this level contains the maximum information and can give good results. However, sensors used in this level must be commensurate. This means the measurement has to be the same or of similar physical quantities or phenomena such as vibration signals. As a consequence, the signal-level application is limited in the real environment, where there are many physical quantities to be measured for synthesis analysis.

- **Feature-level fusion** (fig n.10b): In this level, features are extracted from each sensor according to the type of raw data. Then, these non-commensurate sensor information are combined at the phase of the feature level. All feature vectors are combined in turn to a bigger single feature vector, which is then used in a special classification model for decision making .

  The function of feature vector normalization must still be performed prior to linking the feature vectors from individual sensors to a single larger feature vector in order to limit them to a same value range.

- **Decision-level fusion** (fig n.10c): In this structure, the processes of feature extraction and pattern recognition are employed for single-source data obtained from each sensor. Then the generated decision vectors are fused using decision-level fusion techniques such as Bayesian method, behavior knowledge space (BKS).

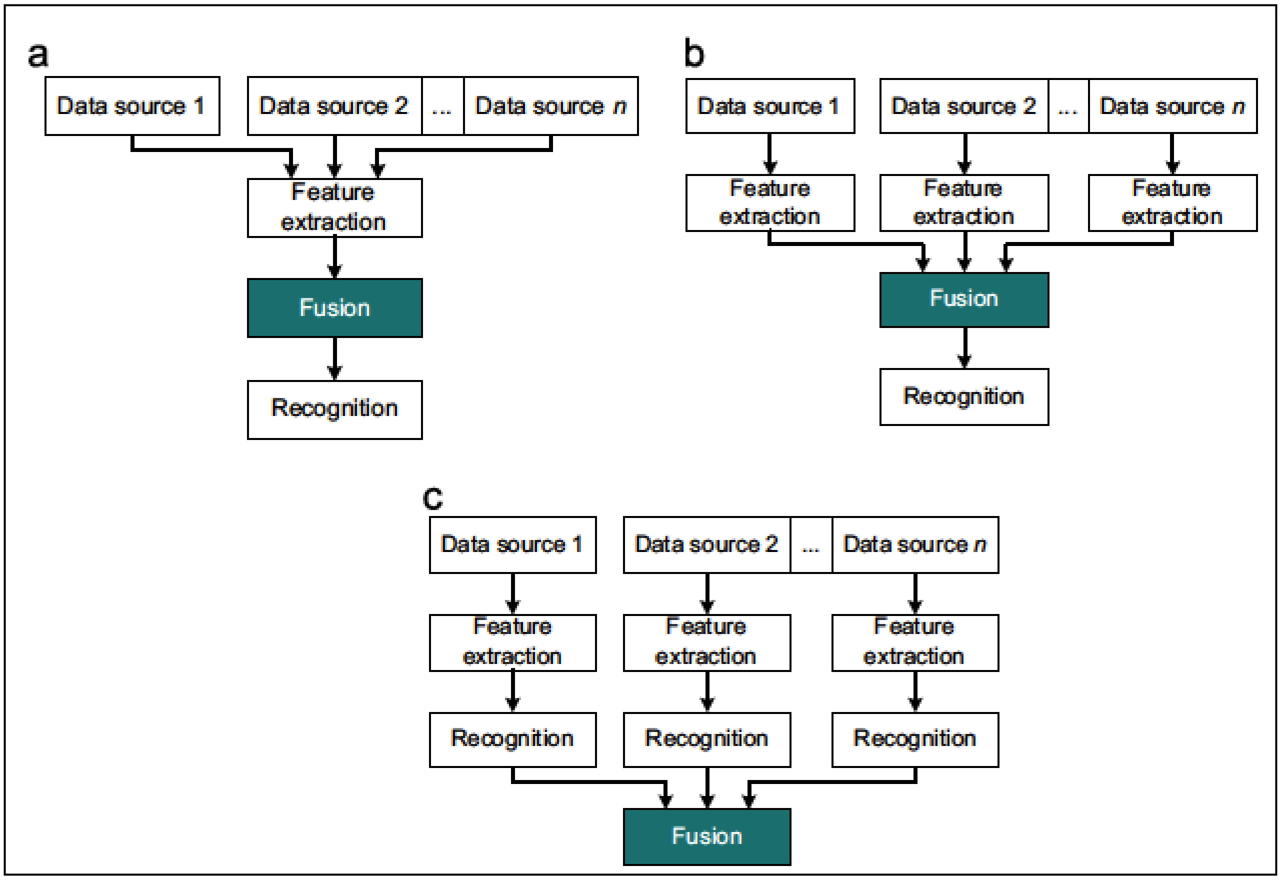


Figure 10. Data fusion: (a)signal-level(data-level) fusion; (b) feature-level(data-level) fusion and (c)decision-level(data-level)fusion.

There could be then another level that is not mentioned by [ref. 5]. This level is related with fusion of information that has a practical value for the maintenance operator that would like to take decisions upon information coming from different point of view. In fact, more than an analysis that consider data coming from different sensors, in different physical location, it is interesting to analyze information coming from the machine, namely sensors, but also coming from a management point of view, namely information related with the impact of a failure on the overall production system, where the machine is installed. This lack, leaded to the development of the present work

4.3 Implementation of CBM

Before CBM can be applied it is necessary to determine the distribution of failure modes from an accurate failure history (if one exists) or from an estimate derived from similar plant.

So, in any case a failure mode analysis is needed before carrying out CBM, thus information about FMECA to be included in an integrated tool could be almost always available.

Ideally, mature plant has a thoroughly recorded history on a computerized data base. From data such as breakdown date and time, duration of breakdown, and failure classification, it is possible to compute parameters such as mean time between failure (MTBF), mean time to repair (MTTR), failure rate, availability and reliability. Perhaps the most useful statistics are simply the total number of incidences and total down time for each failure mode [ref.13].

In the absence of primary data it is possible to use manufacturer's data, published data sources, and heuristics based on experience. It is clear that the majority of new plant has no failure history, but maintenance planning is still required. Having established the failure modes it is necessary to select techniques which can predict the major modes based on parameter detection or performance evaluation.

It is important that the technique selected is sensitive enough to give a long lead time, so that a repair can be scheduled, and parts ordered when necessary. The frequency at which measurements are made must be low enough to be practical, since this affects the operation costs.

To find the parameters to be monitored it is important also to know the main categories of waveform data analysis. In fact there are numerous signal processing techniques and algorithms based on the different data analysis for diagnostics of mechanical systems.

In literature, there are three main categories of waveform data analysis: time domain analysis, frequency domain analysis and time-frequency analysis.

The first calculates characteristic features from time waveform signals as descriptive statistics such as mean, peak, peak to-peak interval, standard deviation, crest factor, high-order statistics: root mean square, skewness, kurtosis, etc. This will be the approach used for the diagnostic analysis within the proposed tool.

More advanced approaches of time-domain analysis apply time series models to waveform data. The main idea of time series modelling is to fit the waveform data to a parametric time series model and extract features based on this parametric model. The popular models used in the

literature are the autoregressive (AR) model and the autoregressive moving average (ARMA) model.
There are many other time-domain analysis techniques to analyze waveform data for machinery fault diagnostics. Wang et al. discussed three non-linear diagnostic methods, known as pseudo-phase portrait, singular spectrum analysis and correlation dimension, based on the signal time series and time series analysis theory. Other works on application of these methods are: for pseudo-phase portrait and for correlation dimension. Zhuge and Lu proposed a modified least mean square algorithm to model the non-stationary impulse-like signals. Baydar et al. investigated the use of a multivariate statistical technique known as principal component analysis (PCA) for analysis of the time waveform signals in gear fault diagnostics.
Frequency-domain analysis is based instead on the transformed signal in frequency domain. The advantage of frequency-domain analysis over time-domain analysis is its ability to easily identify and isolate certain frequency components of interest. The most widely used conventional analysis is the spectrum analysis by means of fast Fourier transform (FFT). The main idea of spectrum analysis is to either look at the whole spectrum or look closely at certain frequency components of interest and thus extract features from the signal. The most commonly used tool in spectrum analysis is power spectrum.
The last analysis, time-frequency analysis, has been developed for non-stationary waveform signals. Traditional time–frequency analysis uses time–frequency distributions, which represent the energy or power of waveform signals in two-dimensional functions of both time and frequency to better reveal fault patterns for more accurate diagnostics. Short-time Fourier transform (STFT) or spectrogram (the power of STFT) and Wigner–Ville distribution are the most popular time–frequency distributions [ref.1].

4.4 FMECA-CBM

FMECA, an acronym for 'Failure Mode Effects and Criticality Analysis', as described in chapter 2, is a procedure for equipment reliability analysis. FMECA has been found useful primarily in the design cycle of new products and systems because it offers a way to identify design deficiencies so that corrective modifications can be made.
However, it has also been employed to a lesser extent within the maintenance planning function. Despite its many advantages FMECA is unknown among the majority of plant and maintenance engineers. FMECA requires training, organizational infrastructure, commitment and access to a

reliable database of failure data. This project aims at proposing a new approach to make FMECA more accessible during maintenance and to make it continuously updated.

There is a natural relationship between the principles of Condition Based Maintenance (CBM) and FMECA. The latter includes among its procedures the identification of a detection method as well as the consequences or effects of each failure mode. Normally, in order to be practical, CBM must include an automated diagnostic capability.

Condition Based Maintenance in general, can be well served by the use of FMECA. Using CBM without a preliminary analysis, one tends to collect data over a long period of time in the hope that trends will provide clues pointing to the failing equipment component as well as to the time remaining before the loss of that component's function.

A FMECA, if performed in correct way in a diagnostic program, will guide the user in choosing an optimal set of condition monitoring techniques and relate the condition indicators to precise failure modes. FMECA is a fundamental analysis technique that combines the analyst's intimate knowledge of each component's function, its failure modes, and the effects of each failure mode. One first collects detailed component information in a database. Useful reports are subsequently generated. Standard FMECA reports assist in the decision process regarding maintenance management policy on preventive and condition based maintenance.

This link between FMECA and CBM is then a new avenue towards the integration and unification of diagnostic techniques and knowledge about the failure modes.

Looking in scientific data-bases for papers related with the integration of RCM-CMMS-CBM, considering that FMECA is one of the main tool for RCM analysis, some papers were found, the most important are:

- **2003 – Shum, Y.S. and Gong, D.C. Design of an integrated maintenance management system. Journal of the Chinese Institute of Industrial Engineers 20 (4):337-354.**

This research addresses the overall view of an Integrated Maintenance Management System (IMMS). Its structure consists of three subsystems of the Real Time Monitoring (RTM), Reliability Centered Maintenance (RCM), and the Computerized Maintenance Management System (CMMS). These subsystems are not independent. A stand-alone RTM cannot continuously provide improving suggestions without analytical capability. On the other hand, a

RCM may offer ambiguous analyses if it is short of real-time data from equipment. In addition, a tedious CMMS creates the burden for making decisions. Therefore, these three subsystems should be considered together to form an integrated structure. This structure is expected to provide guidance for the maintenance operation flow development of an efficient factory. Modules of the IMMS will be illustrated in details. A mold injection machine is also described in this paper for the argument.

- **2006- Ranganath Kothamasu, Samuel H. Huang, William H. VerDuin. System health monitoring and prognostics - A review of current paradigms and practices. Int J Adv Manuf Technol (2006) 28:1012-1024.**

This paper reviews the philosophies and techniques that focus on improving reliability and reducing unscheduled downtime by monitoring and predicting machine health.
Hence, research opportunities include development of modeling technologies that are precise, adaptive, comprehensible, and configurable (by user). There is also an opportunity to integrate the qualitative information that can be extracted from failure mode and effects analysis (FMEA) or fault tree analysis (FTA) of a process or machine into the quantitative analysis that generates diagnostic recommendations

- **2008- Jaw, L., Merrill, W. CBM+ Research Environment – Facilitating Technology Development, Experimentation, and Maturation. Aerospace Conference 2008, IEEE.**

Since its beginning in late 1990s, prognostics and health management (PHM), or condition based maintenance plus (CBM+), has not only spawned a vital research community, but has also become a requirement for complex systems, like air and space vehicles, to achieve the goals of condition base maintenance (CBM) and reliability centered maintenance (RCM). CBM+ extends the traditional capabilities of fault detection and isolation with the capabilities of prognostics and logistics. To enable these extended capabilities, a modern CBM+ system requires two essential components: 1) an on-board monitoring unit; and 2) an off-board (or ground-based) information system. The CBM+ research environment is an implementation of the off-board information system to facilitate CBM and RCM practices. Historically, aircraft propulsion systems (or engines) have led the way in deploying CBM+ capabilities, because of their criticality on flight safety and their significance in driving maintenance cost. By using the propulsion system as the target application, this paper describes the off-board information system being designed by the authors. It also presents some examples to demonstrate the concept of the CBM+ research environment. .

4.5 Conclusion

This chapter motivates the present research and the related proposed tool through an overview of the literature concerning the scope of investigation.

Nowadays FMECA is not widely used especially dynamically, in fact, when a FMECA of a plant or of a product is made, it is used only to identify design deficiencies but it will be not update until a next review of the plant (if updated).

In addition, FMECA, up to now, is not used in an integreted way with an automated diagnostic tool (part of a CBM program), but it is usually used separately as instrument that give information for planning maintenance.

The idea of the software tool, proposed in this work, is to build a FMECA updated using fault data history got through a diagnostic tool.

In this way, an update FMECA can give significant contribution for diagnosis and practically support the maintenance of a plant or a machine.

Moreover, this study wants to take part of CBM analysis and, through fault detection based on PCA, show how and when it is possible to link the FMECA information with process control functions.

Some parameters of FMECA, such as severity of fault, can be used to give an idea about the seriousness of the failure happened, when a fault is diagnosed through a CBM system.

Linking FMECA with CBM, then, can be seen as a positive approach to give to maintenance operators more accurate and correct information to make decisions about maintenance task to carry out.

5
The proposed tool

This chapter presents the main activity that has been carried out during the project. The chapter summarizes ESA (Electric Signature Analysis) issues and discusses the development of the tool and how it has been implemented.

5.1 ESA (Electric Signature Analysis)

There are different approaches to diagnose failures on industrial machines, but all are principally based on monitoring the degrade state of a machine/system.
To identify features associate to a particular working condition, it is possible to use information acquired from electrical signals of power supply.

Electric signature is defined as a particular feature linked with each working condition. To obtain the features it is possible to analize signals of added sensors or to monitor them using sensors already integrated in the machine.
The general principle of ESA, in a diagnostic analysis, is that each change of the good working feature is a sign of presumable degradation.

Using ESA, it is possible to check in advance the precence of a failure and it allows us to understand the causes of the failure, if a proper tool for the analysis is established.
ESA allows to make this diagnosis because a failure of a electromechanical device interferes with electrical signal of power supply (see figure n. 11 as example), since it trasmits some perturbation on the electric motor that drive it.

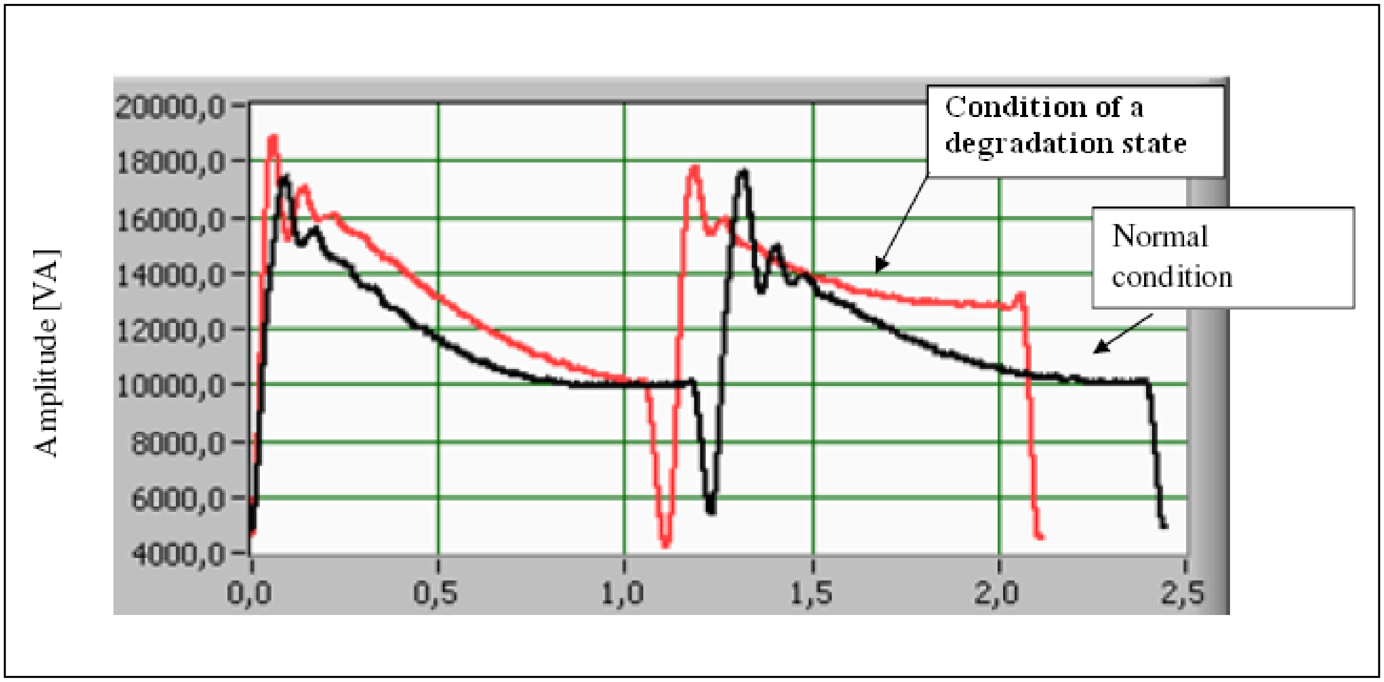


Figure 11. Example of electric signature acquired in condition of a degradation state and in normal condition.

The reasons to use ESA as diagnostic instruments are:

- A fault can be anticipated from a degradation state.
- The degradation leads the machine to a failure state before fault is visible to operator.
- A monitoring activity can point out abnormalities of working condition before they evolve in a dangerous fault.

In order to analyse the signature, the electrical signature must be acquired, considering current and voltage of the three phases. These 6 signals are converted into power signal that is displayed in a time diagram. Then the shape of the signature must be analysed and different shapes (of normal or degraded state) have to be compared. Nevertheless comparing the shape of the electric signature is not an easy task and it could be mainly related with image analysis.

To overcome this process, some indexes representing the shape of the signal can be built. Then the indexes can be used for all the comparisons and analysis needed by the diagnostic tool. In this way it is possible to use these indicators as mean to identify incoming failures, identifying them from a corresponding degradation condition.

In this project the indexes are calculated starting from the signature, without filtering the signal into the software. The indexes are:

- Kurtosis index: a measure of the "peakedness" of the probability distribution of a real-valued random variable (see fig. 11).
- Skewness index: a measure of the asymmetry of the probability distribution of a real-valued random variable. The skewness value can be positive or negative, or even undefined (see figure).
- Crest factor: a measurement of a waveform, calculated from the peak amplitude of the waveform divided by the RMS value of the waveform (see figure).
- Absorption index: integral of power signal.

In the figure 12 below, functions corresponding to some indexes values are showed in order to explain the meaning of each index.

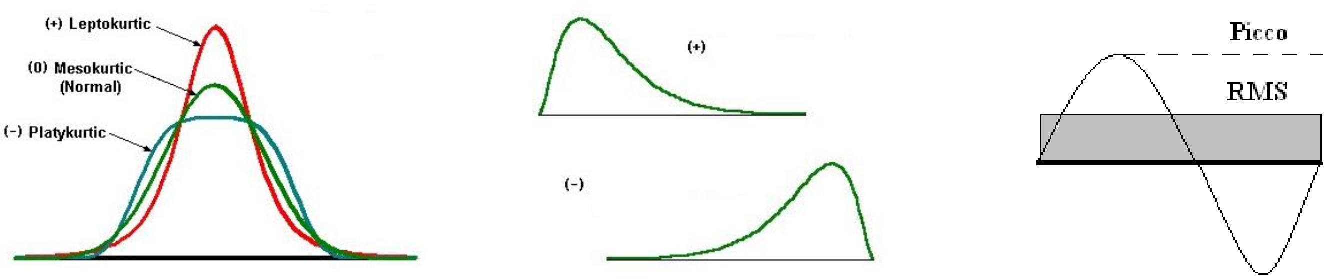


Figure 12. The first picture is Kurtosis, the second is Skewness and the last is Crest Factor

It is possible to find three important steps (according to the definition of diagnostics given by Jardine et al. (2006))[ref.41] to develop a diagnostic tool based on ESA:

- **Fault detection**: Detect malfunctions in real time, as soon and as surely as possible.
- **Fault isolation**: Find the root cause, by isolating the system component(s) whose operation mode is not nominal
- **Fault identification**: to estimate the size and type or nature of the fault.

5.2 Aim of the tool

The tool developed in this project shows how it is possible to link FMECA with a diagnostic analysis tool through an operator interface (figure n.15).
The tool is developed to consider mainly only electro-mechanical faults and not software problems, according to the capability of ESA technique that is the one that is used as background to acquire signals of the machine.
Then, it is worth noticing that some characteristics of the tool have been properly tailored on the needs of the industrial case study used for the validation of the tool (see Chapter 6).

The tool is developed to have the following modules:

1) A module to make diagnostic analysis, based on the Principal Components Analysis, discussed in Chapter 3 and that I call DIAGNOSTIC ANALYSIS module
2) A module to manage a database that contains information related with FMECA
3) A module to manage a database that contains some information related with history of maintenance action. It can be considered a little version of a CMMS, so I called it CMMS-like. CMMS means Computerized Maintenance Management System and with this acronym are generally mentioned the information systems dedicated to maintenance management.

In the figure n.13 below is shown how the tool integrates different issues. It is worth noticing that the 3 modules above mentioned are not represented separately in the Human Machine Interface (HMI), but their functionalities are mixed according to the needs and common use of maintenance personnel who can use the tool.

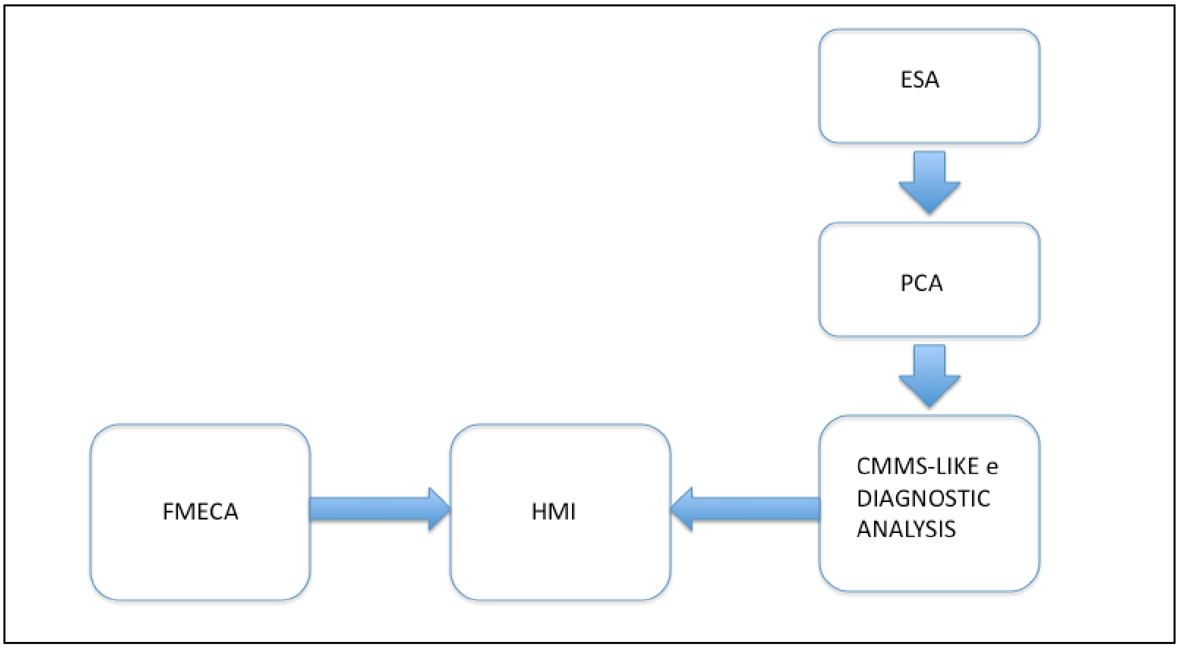


Figure 13. Integration of FMECA,HMI,CMMS-LIKE and Diagnostic analysis

It is possible to find in the HMI two parts: DIAGNOSTIC ANALYSIS and FMECA (in figure n.15 this division is shown directly in the window of the tool), while the CMMS-LIKE module works in background as support database.

The first part analyzes the presence of a fault using the PCA technique, described in chapter 3, while FMECA, described in chapter 2, is used here with the idea of giving important information about faults (for instance severity of the failure mode) during diagnosis and of getting continuously updates data for the features of failures (probability, rate of failure, etc), so to keep FMECA always updated.

5.3 Architecture of the software

The software is designed on two programs: Matlab2008a and LabView8.5.

It is possible to think this software like an ensemble of the three modules mentioned before that communicate with each other.

The figure n.14 shows this structure and it tries to explain all the possible links between these modules: CMMS-LIKE, FMECA and DIAGNOSTIC ANALYSIS.

HMI, represented stylized in the center of the picture, plays an important role to achieve the goal of the project, because it allows the maintenance operator to communicate with the tool. In fact HMI is an interface that controls input and output data inside the software.
The working station depicted in figure n.14 is a normal calculator, controlled by HMI, in which all data are elaborated and the results transfer to the right section of the software.
CMMS-LIKE and FMECA are database structured with different parameters that allow them to be identified.
CMM-LIKE is structured as a database with the following fields:

- EVENT: progressive number of failure mode
- CODE OF COMPONENT: code of component faulted
- DATE: date of analysis/fault
- TYPE OF ACTION: preventive or corrective action
- CODE of FAILURE MODE

While FMECA database is structure in this way:

- CODE OF COMPONENT: code of component analyzed
- CODE of FAILURE MODE: code that identify a failure mode
- SEVERITY: severity of the fault
- FREQUENCY: frequency of the fault
- CAUSE: cause of the fault
- EFFECT: effects generated from a fault condition
- MTBF: Mean Time Between Failure
- MDT: Mean Down Time

The data of these two structures are used by math functions to achieve the aim of the tool.

DIAGNOSTIC ANALYSIS, instead can be seen divided in two parts:

1. One collects all the information about the detection ,the isolation and the identification of the faults
2. One manages the signals acquired to make a diagnosis

The exchanging of data is shown in the figure through arrows, that can be of three kinds:

- Red arrow, that represents the information from FMECA, CMMS-LIKE or DIAGNOSTIC ANALYSIS to the working station controlled by operator through HMI.
- Black arrow, that represents the information exchanged from the working station to the components of the software.
- Blue arrows for all functions managed inside each components.

All arrows are associated with a number that represents the corresponding button of HMI (see table n.6).

About the transfer of information, you can see for example (through black arrows) the transfer of historical data from the working station to CMMS-LIKE and FMECA when the operator pushes the button number 1 or 2 of the HMI; or in the opposite direction (red arrows) the transfer of information to the database of working station through button 4 and 5.

Each buttons of HMI triggers a function, which generates as result an exchanging of data within single section or between two parts of the software.

The description of each functions of the buttons are explained below, in the next chapters where is shown the functioning of the software.

As said before HMI manage not only input but also outputs, and these can be of two kinds:

- Red light alarms, that switch on when a possible fault is diagnosed
- Parameter (such as: code of components, code of failure mode ,etc.)

In figure the outputs are described with letters (A,B,C,etc). You can see for example the letter E in the HMI, which represents a light alarm (see figure n.15 of real interface implemented) that switch on when at least fault is diagnosed.

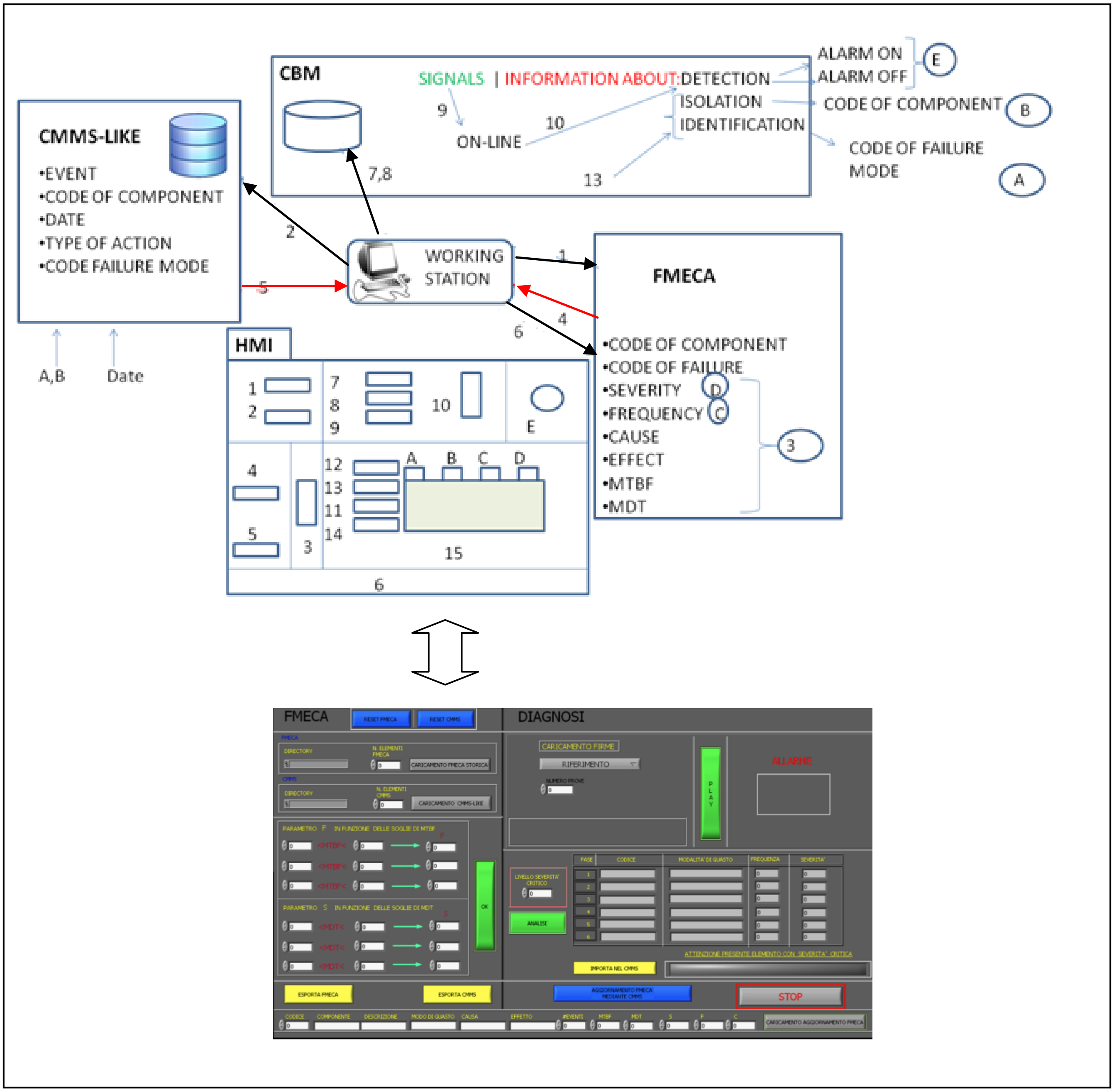


Figure 14. Structure of the software

N° Button	Description	Output	Description
1	Load historical FMECA	A	Code of component
2	Load historical CMMS-LIKE	B	Code of failure mode
3	Confirm the values about MTBF and MDT	C	Frequency
4	Export FMECA	D	Severity
5	Export CMMS-LIKE	E	Alarm :presence of fault
6	Load new element of FMECA		
7	Load fault signatures		
8	Load reference signature		
9	Load signature to be tested		
10	Confirm of loading and Start detection analysis		
11	Start Identification		
12	Limit of Severity		
13	Start Isolation		
14	Updating of FMECA		
15	Exit		

Table 6. Functions and outputs of HMI

5.4 Functions and uses of the software

When a user uses this software for a first time on a new machine, he has to load, using the right command above described, all the information needed to start the diagnosis and the identification of faults.
The necessary data are:

- Historical FMECA.
- Historical CMMS.
- Frequency and Severity of the failure mode with respect to different levels of MTBF(mean time between failure) and MDT(mean down time).
- A number of electric signatures based on good working condition.
- A number of electric signatures based on different fault working conditions.

These data are necessary because the diagnostic tool, to work and so to diagnose a probable fault, must know all the parameters that describe the reference working condition.

In fact PCA works comparing scores of good reference condition with scores of a signature to be diagnosed.

It is also important to load a number of electric signatures based on different fault working conditions to have different fault reference regions.

In fact in this way the software can identify the kind of fault occurred searching in which region the score of the fault is (Identification step).

The graphics interface (see figure 12), built in LabView8.5, allows the operator to exchange data with the software tool:

- choosing the right directory where to load data about FMECA and CMMS.
- deciding what kind of signature load through a combo box.
- defining values about parameters that are used in the tool.

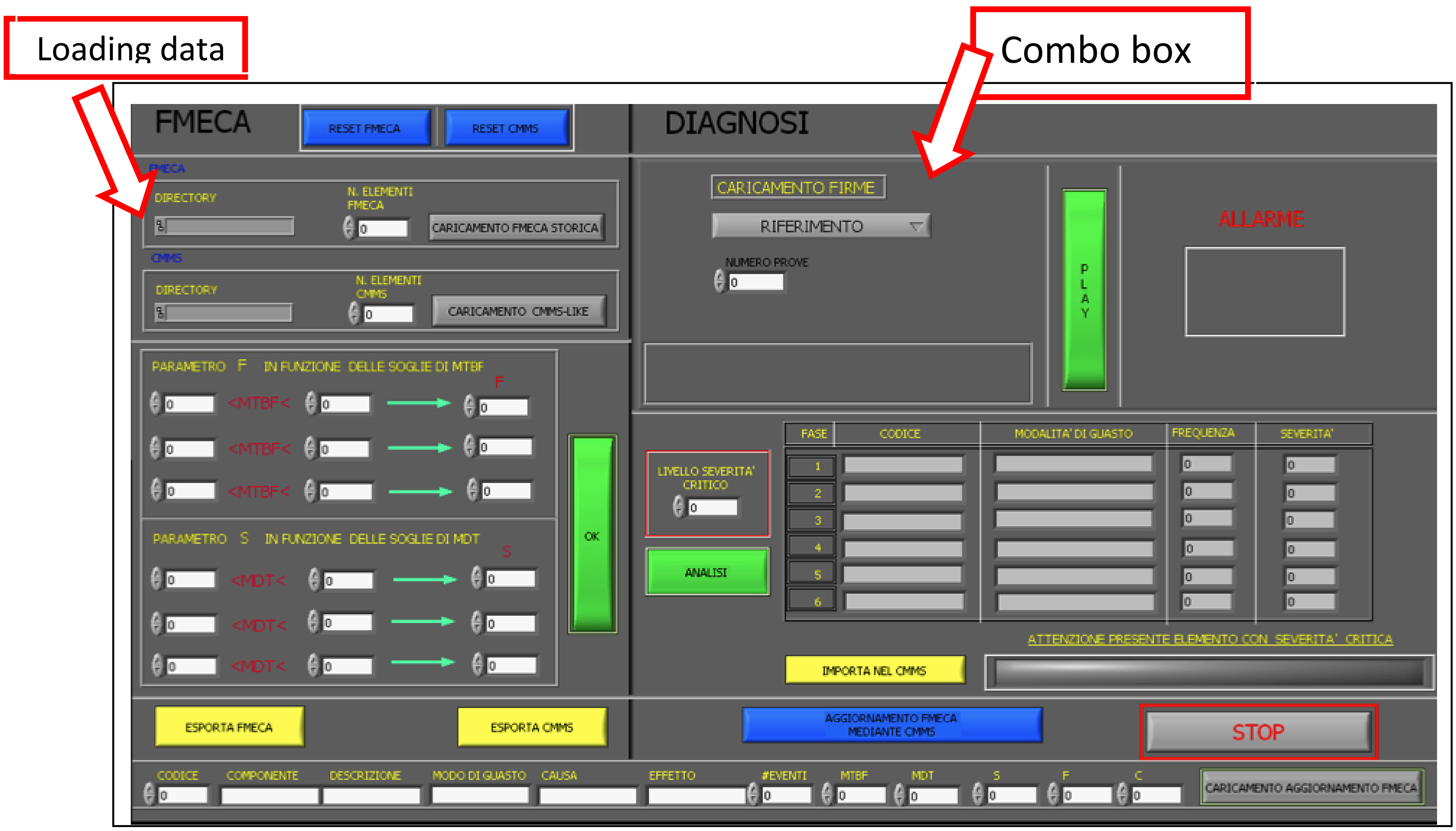


Figure 15. The graphic interface developed through LabView8.

At the same time, HMI allows the operator to know the state of the machine using the diagnostic tool. In fact HMI presents outputs in which information (frequency, probability, code of component, etc) about fault diagnosed are explained.
So HMI makes the software flexible, in fact the operator can communicate and interface with the machine easily.

A limit of this software is that it is only possible to load stored data of signature and not real time data. This aspect shows that software is based on OFF-LINE diagnosis.

The next two paragraphs show in detail the commands of the software and for which reasons it was decided to implement them in the software.

5.4.1 Functions implemented for FMECA and CMMS

First of all, the most important function about FMECA and CMMS-LIKE is the possibility to load their databases through the right command.
As said before, this software is designed with the possibility to load a whole database of FMECA or only part of it, modifying in this way the database just stored.
Also for CMMS-LIKE, that is a database of all fault events, there is the same possibility, in fact you can load or modify its values.
It was decided to implement this opportunity for both databases, to give the possibility to change some elements of FMECA or CMMS when the maintenance operator finds errors in the databases or when some parameter during tests changes its value.
There are two ways to add new values into CMMS:

1. Using the diagnostic tool inside the software: If diagnostic tool integrated in the software discovers a fault, through the right button, an operator can decide to add it automatically in CMMS-LIKE.
2. MANUALLY: Using other way to diagnose faults, when the fault is discovered you can add it manually inside the CMMS database (.xls format).

While if you want to add or modify a new element into FMECA, there is in the HMI a box in which you can insert your changes.

If the element added is already inside FMECA database, it will be update automatically; while if the element is new, because it has a new code, the software recognizes it and it adds the characteristics of the new component in right order in the database.

The figure below shows the box where it is possible insert changes of a component inside FMECA.

Figure 16. The graphic interface where it is possible to update FMECA database.

To load the whole databases the user has to respect some particular rules in fact:

- **To load the historical FMECA the user has to follow the structure below using the format .txt :**

Code	Component	Description	Type of fault	Cause	Effect	Number of event	MTBF	MDT	S	F	C

Table7. Header of FMECA implemented.

The chosen structure follows the generic description of FMECA, shown in chapter 2, that describes both the code of the component and its possible faults and its effects with statistical parameters.
To upload these data, after choosing the right directory and selecting the number of elements of FMECA you have to click ‘CARICAMENTO FMECA STORICA’ button.

This database will be stored in additional folder inside software and all elements can be used each time the software needs.

In the figure below you can see how the part of loading is structured.

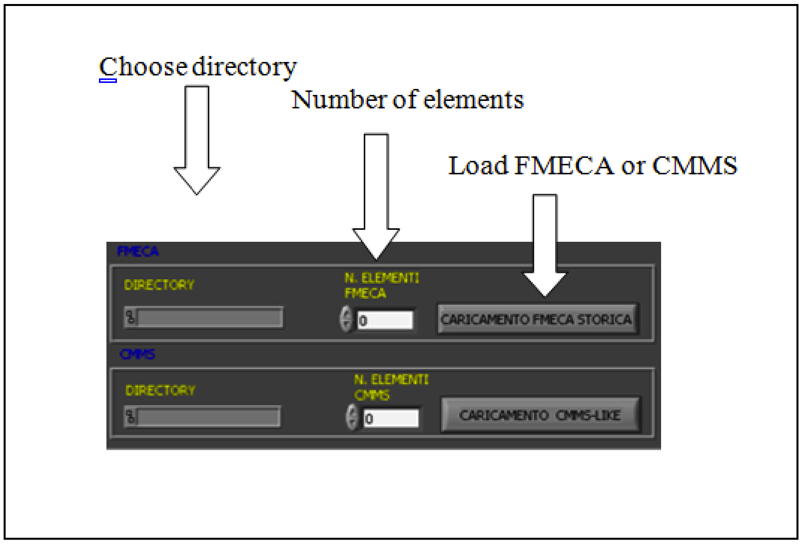


Figure 17. The graphic interface where it is possible to load FMECA and CMMS-LIKE databases

- **To load a historical CMMS the user has to follow the structure below saved with format .txt :**

Code of component with fault	Cause of fault	Day	Month	Year	Type of action

Table8. Header of CMMS-LIKE implemented.

In the same way, described above for FMECA the user can load CMMS.

Also here the structure of CMMS is projected to describe the most important parameters of a fault diagnosed.

In fact, you can see the date of the diagnostic test and other parameters such as Type of action done to diagnose the fault.

As you can see from HMI in the FMECA part, there are other buttons for other functions:

- **Choose the parameters of Frequency and Severity respect three different levels of MTBF (Mean time between failure) and MDT (Mean down time).**

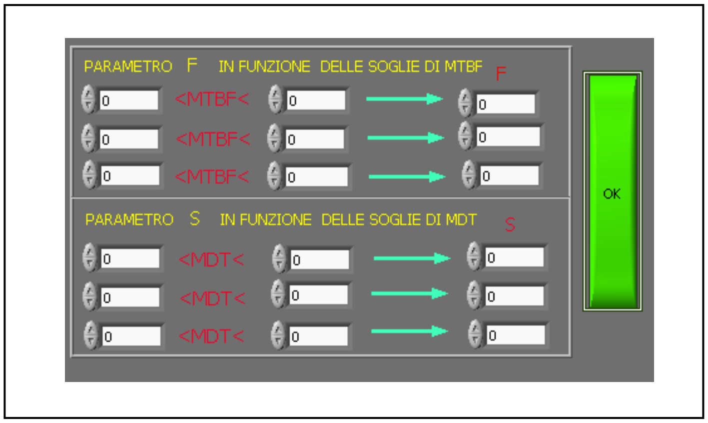


Figure 18. The graphic interface where it is possible to load values about MDT, MTBF and corresponding value of Severity and Frequncy

In this way the maintenance operator can decide personally the appropriate level of MDT and MTBF of the machine and the corresponding values of Severity and of Frequency. When you choose these levels and you click OK botton the software automatically updates FMECA with the defined values.

- **Export CMMS or FMECA (format .xls) in a appropriate folder create on Desktop**.

Figure 19. The graphic interface where it is possible to export FMECA and CMMS-LIKE databases

After creating a folder on Desktop you can save the databases of FMECA and CMMS to make off-line a qualitative analysis about it. In this way the maintenance operator can analyze updated FMECA and all faults happened and recorded through CMMS.

➢ **Reset the table of CMMS or FMECA in the folder on Desktop.**

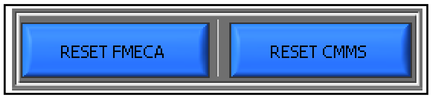


Figure 20. The graphic interface where it is possible to reset all FMECA and CMMS-LIKE data stored

These two commands allow the operator to decide to clear the databases of FMECA and CMMS (they are used usually when you install the software for the first time or when you to clear data stored until that time).

5.4.2 Buttons and Functions implemented for diagnosis

The diagnostic tool is designed to work off-line, in fact the idea is to test the signature acquired during constant intervals. In this way it is possible to understand if the machine is in a failure condition or not.

This diagnostic tool can operate only if a number of reference signatures (format .txt) are loaded.

These values allow the tool to compare, through PCA, the reference condition with the scores calculated from the signature to be tested.

The figure below shows where it is possible to load signatures and where the maintenance operator can understand if a fault occurred during the test cycle, through a warning light.

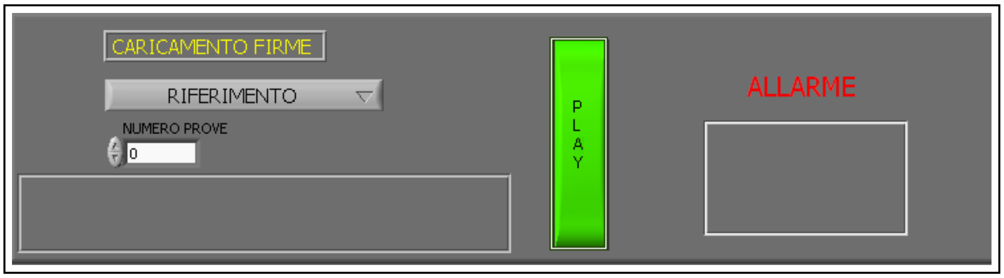


Figure 21. The graphic interface where it is possible to load signatures and where the operator can know if a fault happened through an alarm.

It is possible to choose one kind of signature that you want to load from a combo box. After loading the reference signatures and the signature to be tested you can start to diagnose it clicking PLAY bottom.

If the diagnostic tool finds at least a fault inside the signature tested a red alert switch on. This signal allows the operator to know that a fault happened (DETECTION).

To know what kind of failure happened (ISOLATION and IDENTIFICATION) the user can click on "ANALISI" button and then can see all code of occurred failure modes with their modality of fault, frequency and severity. Frequency and Severity come from updated FMECA loaded during the setting of the software.

The figure below shows how the codes of broken elements appear to the operator.

Figure 22. The graphic interface where it is shown the identification and the isolation of the fault. In this part of HMI the operator can see not only the code of the component faulted but also the failure mode, the Frequency and the Severity associated.

If you want to have an alert indicator about the severity of the faults, you can choose one severity level as maximum; when at least one fault exceeds this level a red alert switches on.

The bottom "IMPORTA CMMS" loads in CMMS-LIKE, the codes of broken components with the date of the event and type of action.

It was decided to add this botton to give the possibility to choose if store these fault data or if ignore them.

In fact the operator can decide to make maintenance on machine and store data about faults or continue to work in degraded condition without saving data about faults diagnosed.
With “AGGIORNAMENTO FMECA MEDIANTE CMMS” botton, shown below, the database of FMECA can be updated through the data of CMMS.

Figure 23. The graphic interface where it is possible to update FMECA using data of CMMS-LIKE

In this way the software, analyzing the fault codes and the date about when the fault was diagnosed, can update the parameters of Frequency and Probability, associated to the specific code, inside FMECA.

5.5 Conclusion

As it is possible to see in this chapter all commands in HMI are in Italian, this because the software is developed to be used for the case study (Chapter 6) carried out with an Italian company.
This chapter tries to explain the structure and the functioning of the tool and aims at showing the possibility to use it for different industrial applications.
The software developed during this project in fact wants to bring FMECA updated using fault data history got through a fault diagnostic tool implemented using PCA technique.
In this way, an update FMECA can give significant contribution for diagnosis, supporting the decisions of the maintenance operators about the maintenance tasks to carry out. In fact, through the severity of the failure mode, a prioritization of the interventions is possible.
The tool demonstrates also how it is practically possible to link the FMECA information with diagnostic ones and namely with information coming almost in real time form the machine.
Next chapter shows how it is possible to validate this tool using a case of study based on a balancing machine.

6

Case of study

This chapter summarizes the structure of the industrial machine used to validate the developed software tool and how it was possible to test it.

6.1 The structure of the industrial machine

This project is tested on a particular family of industrial machine: automatic balancing machine. The machine used, in particular, is designed by Balance Systems s.r.l and its name is BVK4 (see figure). It is projected to balance circular component like saw blades, disk brakes with weight up to ten Kilos and diameter up to 400 mm.
The processing of work for the reduction of imbalance is normally based on milling or drilling and thus the machine is designed to remove material to properly balance the machine.

The production cycle is generally composed by three steps:

- **Measure of unbalancing**
- **Reduction of unbalancing using mechanic processing (milling or drilling)**
- **Measure of residual unbalancing**
- **If necessary, reprocess the part.**

Figure 24. Picture of the balancing machine used during the project

The balancing machine is composed by the following main elements:

- **Measure station**

  The piece, that must be worked, is in rotation through a spindle . It can be measured using acceleration instruments. A software on the machine calculates the position and estimates the value of unbalancing.

- **Working station**

  These machines are designed to reduce the imbalance with removing of material. The station is usually a milling with three axes (see figure).

Figure 25. The working station

- **HMI**

  It is an industrial computer that interfaces the machine with operators. Using this device an operator can control the production and define the work cycle of the machine.

The balancing machine can be loaded both manual and automatically through a robotic arm.
In this case of study the machine has manual loading and it is equipped with particular instruments to get measures about electric power signature.

6.2 Mathematical validation of the Diagnostic tool

The diagnostic algorithm of Fault Detection and Isolation (FDI), included in the tool described in the previous chapter and customized for the balancing machine of the case study, is designed to analyze different behaviors of mechanical components of the machine. For the validation of the tool, the diagnosis is limited only to the belts that transmit power from electric motors to move the axes and drive the spindles of the machine (working spindle and measuring spindle).

To validate the diagnostic tool, first of all, it was decided to select a particular working cycle with which is possible to test the tool.

This working cycle (that is a test cycle) is composed by six segments, in each segment a single component works. In this way it is possible to study the behavior of a single element.

During this test cycle no real work is carried out on the parts, but the components are simple moved to verify that all the moving components of the machine move properly and that are no affected by failures or degradations.

The calculations presented here are based on data acquired during tests on the balancing machine BVK4 at disposal for this test in a laboratory of the company Balance Systems.

In these chapter, to show the validation of the tool will be described only one test based on a signature acquired in a particular known condition of failure.

The data, used for the validation, are composed by 96 signatures structured in this way:

- 50 signatures, for each segment of the test cycle, that describe the reference of optimal condition of the machine.
- 45 signatures, for each segment of the test cycle, that describe the reference of malfunction condition about belts.
- 1 signature, for each segment of working cycle, used like test to validate the diagnostic tool based on PCA.

First the tool needs to load, through the right commands described in the chapter above, the first fifty signatures to create a reference space of good condition. Then, using scores of this region it is possible to calculate the statistical indexes (T^2 and Q) described in Chapter 3.

In this way the software creates a region in the principal component space. The figure n.26 below describes a typical area obtained through PCA analysis of the signature about one step of the test cycle. This graph is obtained using Matlab, the script are shown at the end of the thesis.
It is possible to see the three principal components on axes and the scores as red points.
These create a sort of parallelepiped that delimits an area of a particular condition of the machine.

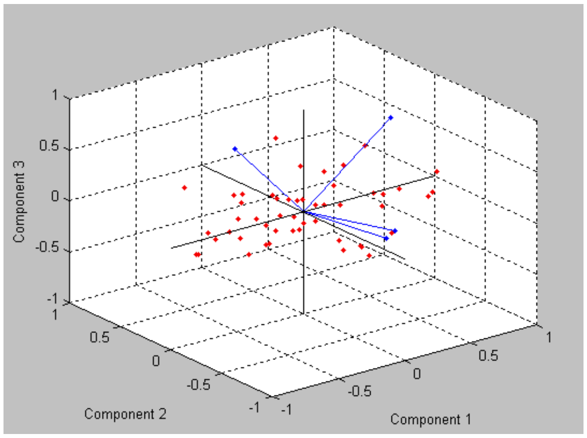


Figure 26: Example of Matlab PCA analysis

In this way it is possible, loading different signatures, to obtain more areas, that describe particular failure condition of the component analyzed.
The analysis of each boundary of these areas is used to identify and isolate a probable failure mode of a component.
To show the validation of the tool, it is used the third segment of the test cycle because it uses only two principal components (according to the PCA analysis) and in this way it is easier to show the validation phase.

The picture below shows the scores of the reference optimal condition about this step. This area, that these points described, is then compared with the new score that have to be analyzed.
It is possible to generate for each segment a similar analysis.

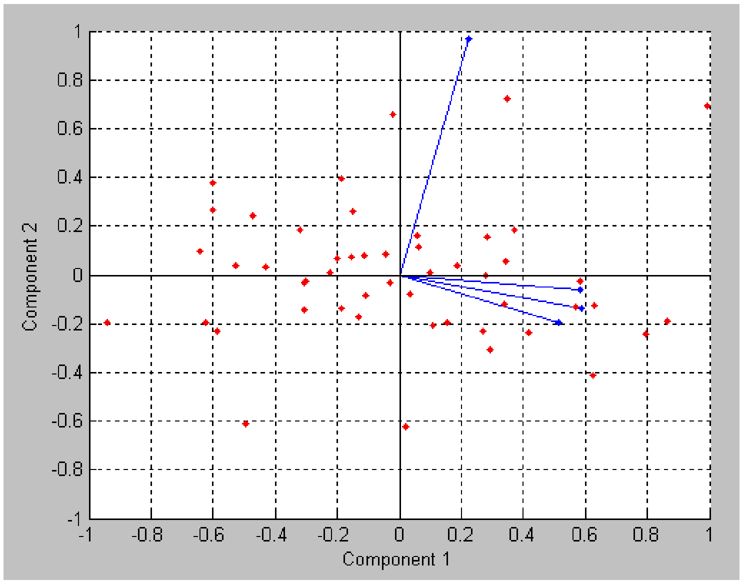


Figure 27: Plot of optimal reference scores about the third step of working cycle on associated principal component

To generate the fault area about bad condition you have to load the other forty-five signatures that have been acquired.

These scores must be plotted in the same space calculated before (you have to use the same principal components) because the diagnostic tool uses these areas as references.

In fact the software, when you load a signature to be tested, compares the positions of the new score with the reference regions obtained before.

In this way, depending where the score is placed, the tool can diagnose the presence of a fault.

In the figure below (fig. n.28) it is possible to see the points of reference conditions (optimal and fault) described before.

During the study done to develop this tool, it was understood that each type of failure defines a different region, differently oriented, in the principal component space.

In this way the software can detect not only the presence of a fault (through the comparison of statistical indices of good condition with indices of bad conditions) but also indentify the precise failure mode, namely isolating the part where the failure is occurring and then identifying the right failure mode occuring.

In fact each area would represent a particular failure mode of the component that works in the segment of the analyzed test cycle.

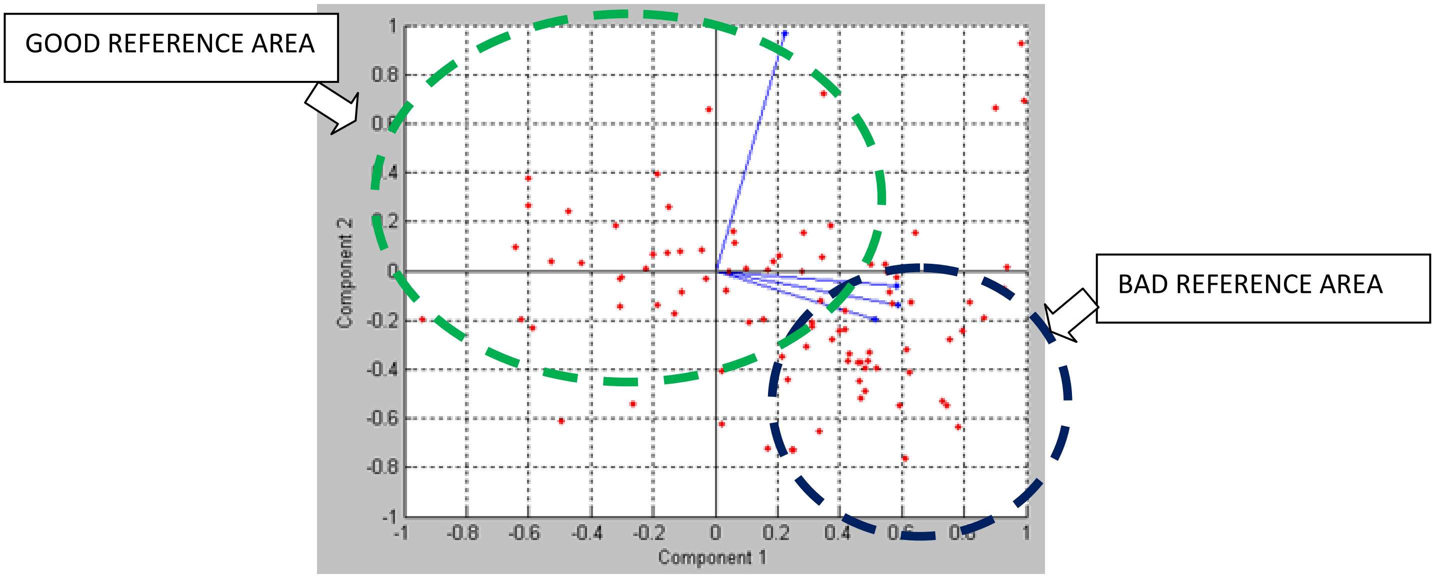


Figure 28: Plot of optimal reference scores and of bad reference condition about the third step of working cycle on associated principal component

In the figure n.29 is shown :

- The region of normal behaviour of the machine.
- The region of degradated condition.
- The score of a signature used as test to validate the software.

You can see that this score is inside the bad reference area and this means that the signature about the third segment of working cycle (used as test) is affected by a fault.
In addition to the graphic analysis it is possible to show the presence of a fault in the signature through the statistical analysis.
In fact the T^2 parameter, that described the limit beyond which there is a symptom of failure, is lower than T^2 obtained from validation signature. (T^2_{obt}=8.4 ; T^2_{test}=14.2)
Also the parameter Q of good reference condition is lower than Q obtained from validation signature.($Q_{obt=}$ 5.2 ; Q_{test}=7.8)
These values, as described in chapter 3, show the presence of failure condition in the machine for the analyzed signature.

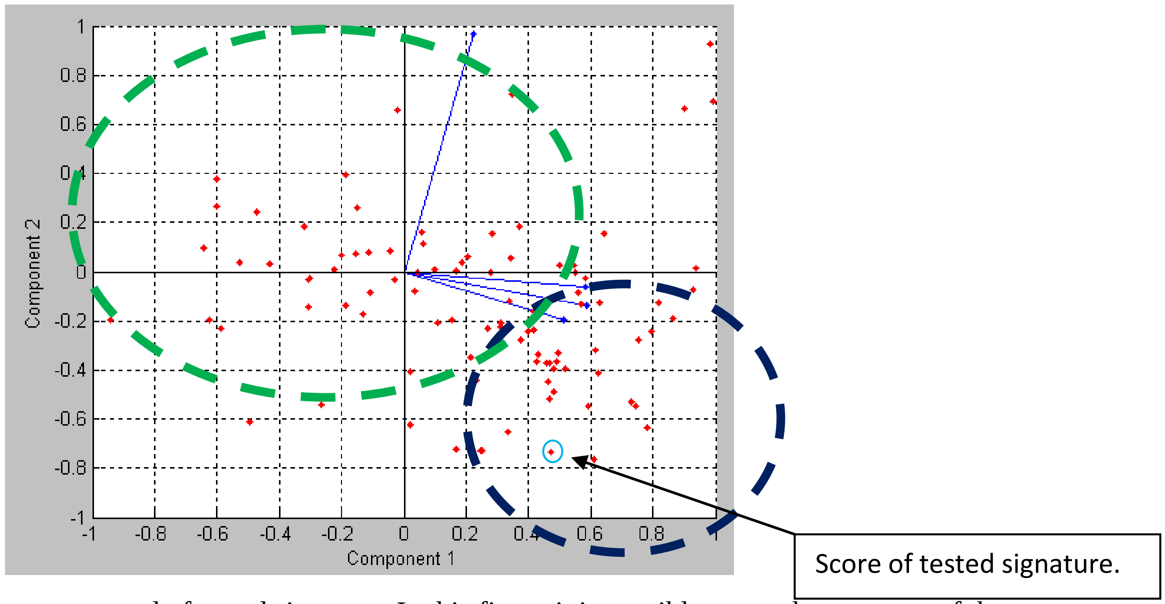


Figure 29: Plot of reference scores and of tested signature. In this figure it is possible to see the presence of the score of the signature tested inside the bad reference region

6.3 Validation of the Diagnostic analysis using the developed software tool

To validate the good functioning of the HMI of the proposed tool, it was decided to load the same data used before in the software tool.

After the loading phase, the red alarm switches on. This represents the presence of at least one degraded component or failure in the machine, for the signature used as tester. The tool is able to detect this malfunctioning, so it correctly carries out the DETECTION task.

In the figure below you can see the part of HMI where the alarm is displayed.

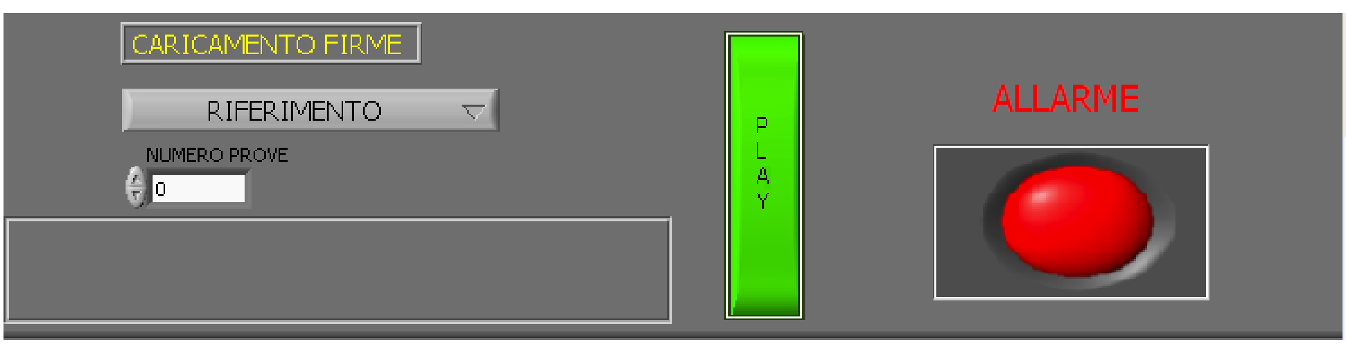


Figure 30: The graphic interface where it is shown the presence of fault in a signature tested through the red alarm switch on .

To identify each possible failure mode that is occurring and to obtain all their parameters associated, the user has to push “ANALISI” button.

The figure below shows what appears after clicking this button.

Figure 31: The graphic interface where it is shown the list of faults diagnosed and their associated parameters (failure mode, frequency and severity)

It is possible to see for each segment (called “Fase” in the Italian HMI of the tool) of the test cycle:

- The code of the components (in the figure the name of the component is displayed)
- The failure mode that is occurring
- The frequency associated to that failure mode
- The severity associated to that failure mode

This information belongs to the last two stages of the diagnostic analysis: ISOLATION and IDENTIFICATION.

6.4 Conclusion

This chapter has showed the possibility to diagnose a failure mode starting from the only electric power signature.
In fact it is explained that, once properly configured the tool, it is possible to identify the presence of a fault and indentify it through indicators.
The possibility to carry out diagnostic from the data of electric signature reduces the number of sensors that would be necessary for a traditional diagnostic analysis.

7
Conclusions

This chapter presents the conclusions of the research. It starts with a discussion of the research purposes and ends with a discussion of the advantages to use the proposed tool that integrate diagnostic analysis with FMECA and the limits found during the development of the project.

This thesis had the aim to describe the development and the application of Fault Detection and Isolation (FDI) system based on principal components analysis (PCA) integrated with FMECA.
In fact, after having briefly explained the context of maintenance management (Chapter 1), the use of FMECA (Chapter 2) and the theory of diagnostic analysis (Chapter 3), the possibility to link the two approaches (FMECA and diagnostic analysis) has been discussed (Chapter 4). Then in Chapter 5 a tool has been proposed to practically link the two approaches and, in Chapter 6, the performance of the software tool was validated, using a set of data acquired during tests carried out in an industrial case study.
A practical result of the present work is also related with the possibility to introduce the use of PCA technique in a real industrial case. In fact in literature the Principal Component Analysis seems mainly related with the field of scientific and technological research. Regarding this statement, however, a further literature review would be needed to support the assortment. This was however out of the scope of this work.
In this thesis, a diagnostic approach with PCA was used with FMECA to develop a tool for a balancing machine. Indeed, this is the main novelty that is addressed in this work, as also been demonstrated by the identification of reference publications taken from recent literature in the maintenance domain.
The next and last paragraph wants to present the benefits of the adoption of this kind of tool, based on the electric signature analysis, wilst it shows the limitations encountered during the development and the use of the software tool.

7.1 Benefits and limits of using this software

Different analyses of maintenance policies adopted in industry show that most companies use policies based on time based preventive maintenance. The use of these policies requires that the useful lifetime of the generic component or device (before its replacement) is estimated, and this is normally done in a conservative manner.
This determines, in many cases, the replacement of parts still in good health and that would be able to provide good service for other time yet.
On the other hand, it can also happen that a failure occurs before the determined time, so between two maintenance actions, creating problem in the production activity.
In recent years, several technological factors have led the development of new maintenance techniques for the diagnosis of failure events, allowing the transformation of maintenance policies from traditional approaches to more effective approaches.
The development of mechatronics, due to increasing interaction between mechanics, electronics and computer equipment, made available devices to support maintenance with less costs.

Benefits
The use of the method analyzed in this study, based on analysis of the electric signature, is a practical demonstration of this trend. The developed tool is, in fact, an instrument that will be able to decrease the number of parts replaced and, consequently, the direct costs incurred by the maintenance companies. The improvement in efficiency will ensure to companies to be more competitive in the global market thanks to wider margins and aggressive pricing policies. Moreover, the capability to anticipate failures will allow to better plan maintenance, thus avoiding some unexpected failures that can also create problems to the production plans.
Moreover, if the point of view of the producer/vendor of the machine is considered, the proposed tool can increase the attractiveness of potential customers to whom the new technology might be offered as a "plus" or together with a proper maintenance service that the company could provide thanks to the developed technology.

Another important advantage of the software implemented in this study is its capability to integrate diagnostic analysis with FMECA.
In fact, in this way, FMECA is continuously updated with historical data acquired during the analysis. Moreover, FMECA can give a lot of information about failure modes when the user

start the diagnostic module (namely information about the severity of the failure mode that is occurring).
An updated FMECA can show the real severity of failure mode of fault of all the components. This aspect is very important when the user wants to take a decision about the maintenance activity to carry out after a diagnostic activity that identified a degraded condition.
Moreover a FMECA, that is automatically linked with the diagnostic system that register the failures of the system, allow to keep FMECA always up to date and so the information related to FMECA are more reliable when used to take decisions, for example when maintenance plan are revised or when the redesign of a component is planned to improve the reliability of the machine.
So, the proposed software tool can bring both economical and technical advantages to the maintenance operators but at same time also to the companies that have designed the industrial machine and would like to have some feedback information to improve the machine..

Limits

The software developed has shown some limits that are a starting point for future work in this research area.
One of these is the use of the tool only in OFF-LINE mode. This modality does not let to diagnose a failure condition while the machine is working but it is possible to test the machine only using a special test cycle (presented in chapter n.6) and previously defined.
A future purpose is improve the tool with the introduction of a module that allows to diagnose a failure condition in ON-LINE mode, that means to discover faults when the machine is working under all operating conditions. This issue is related with the improvement of the diagnostic analysis to identify the condition of a machine while it is working.
This will not be easy to implement because each working conditions is different, depending from the kind and type of operation the machine has to do on the parts that are worked.
Moreover, the development of this ON-LINE mode is also related with the implementation in the tool of an interface with the tool developed to acquire the electric signature. This seems just a detail of the implementation since the tool for the acquisition of the signal that was used in the case study has been developed with Labview, as the one proposed in this work.

Another limit of the proposed approach is related to the way FDI has been implemented, i.e. based on PCA technique. Indeed, applying PCA there is the need to define different reference regions about operating conditions, before you run correctly the tool. This makes complicated the setting phase of the tool in which the software acquires data to learn how to do the diagnosis. It is impossible, using PCA technique to diagnose a fault without having these reference data; thus, it is impossible to overcome this negative aspect because this analysis is based on the comparison of new score with the reference regions.

The setting of the tool is also related with a proper FMECA that has to be carried out, at least in a first draft version, before running the tool, because from FMECA failure modes and other important information needed by the tool (severity, probability, etc.) has to be used.

In fact to identify a critical failure condition, due to a specific failure mode, related to a specific component, the user has to initially load an historical database of FMECA as explained in chapter 5.

References

1. Andrew K.S. Jardine, Daming Lin, Dragan Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mechanical Systems and Signal Processing, Volume 20, Issue 7, October 2006, Pages 1483-1510.

2. K.F Martin , A review by discussion of monitoring and fault diagnosis in machine tools, Int. J. Mach. Tools Manufact, Vol. 34, 1994, Pages 527-551.

3. F. Baggiani, S. Marsili-Libelli,Analisi di guasto in tempo reale per un processo di depurazione biologica, IA Ingegneria Ambientale, vol. XXXVIII n. 6 , Giugno 2009,pages 297-309.

4. JB. Bowles, Materials. Park, Failure modes and effects analysis, ASM International, 2002,pages 50-59

5. Gang Niu, Bo-Suk Yang, Michael Pecht, Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance, Reliability Engineering & System Safety, Volume 95, Issue 7, July 2010, Pages 786-796.

6. Andrea Carpignano, Claudia Vivalda, Dalla manutenibilità alla manutenzione, Manutenzione e progettazione Tecnica e Management, April *2005* , Pages 1-8.

7. Sheng-Hsien (Gary) Teng, Shin-Yann (Michael) Ho, Failure mode and effects analysis: An integrated approach for product design and process control, International Journal of Quality & Reliability Management, Vol. 13 No. 5, 1996, Pages 8-26.

8. N. Baydar, Q. Chen, A. Ball, U. Krunger, Detection of incipient tooth defect in helical gears using multivariate statistics, Mechanical Systems and Signal Processing, 2001,Pages 303-321.

9. A. G. Starr, A structured approach to the selection of condition based maintenance, 5th International Conference on FACTORY 2000, 2-4 April 1997,Pages 131-138.

10. J. Sikorska ,L. Hammond ,P. Kelly ,Identifying failure modes retrospectively using RCM data,Elsevier,Pages 1-7.

11. Kothamasu Ranganath, Huang Samuel, VerDuin William, System health monitoring and prognostics — a review of current paradigms and practices, The International Journal of Advanced Manufacturing Technology ,2006-07-01,pp.1013-1017.

12. Albert H.C. Tsang, Condition-based maintenance: tools and decision making, Journal of quality in maintenance engineering, 1995, pp. 3-17.

13. J.H Williams ,Condition-based maintenance and machine diagnostics, Kluwer academic publisher,1994,first chapter.

14. Chuenusa Cholasuke, Ramnik Bhardwa, Jiju Antony, The status of maintenance management in UK manufactoring organisations:results from a pilot survey, Journal of Quality in Maintenance Engineering, 2004 , pp. 5-15.

15. C.W. Gits, Structuring Maintenance Control systems, International Journal of Operations e production Management, 1994, pp. 5-17.

16. Somnath Deb, Sudipto Ghoshal, Amit Mathur, Roshan Shrestha and Krishna R. Pattipati, Multisignal Modeling for Diagnosis, FMECA, and Reliability,IEEE,pp. 3-17.

17. Pratesh Jayaswal, A. K.Wadhwani, K. B.Mulchandani, Machine Fault Signature Analysis, International Journal of rotating machinery, 2008,pp. 1-10.

18. Fumihiko Kimura, Tomoyuki Hata and Noritomo Kobayashi, Reliability-Centered Maintenance Planning based on Computer-Aided FMEA, The 35th CIRP-International Seminar on Manufacturing Systems, 2002,pp. 1-7.

19. Roberto Cigolini ,Francesco Turco, Total productive maintenance practices: a survey in Italy, Journal of quality in maintenance engineering, 1997,pp. 259-272.

20. Ricky Smith,Bruce Hawkins,Lean Maintenance :Reduc costs,improve quality and increase market share,Plant engineering,pp. 0-150.

21. Garg Amik, Deshmukh**,** Maintenance management:literature review and directions, Journal of quality in maintenance engineering,2006.

22. D Sherwin ,A review of overall models for maintenace management, Journal of quality in maintenance engineering,2000.

23. AW Labib, A decision analysis model for maintenance policy selection using CMMS, Journal of quality in maintenance engineering,2004.

24. Artes M., Del Castillo L., Perez J. (2003) Failure prevention and diagnosis in machine elements using cluster, in: Proceedings of the Tenth International Congress on Sound and Vibration, Stockholm, Sweden, pp. 1197–1203.

25. Artes M., Del Castillo L., Perez J. (2003) Failure prevention and diagnosis in machine elements using cluster, in: Proceedings of the Tenth International Congress on Sound and Vibration, Stockholm, Sweden, pp. 1197–1203.

26. Artes M., Del Castillo L., Perez J. (2003) Failure prevention and diagnosis in machine elements using cluster, in: Proceedings of the Tenth International Congress on Sound and Vibration, Stockholm, Sweden, pp. 1197–1203.

27. Artes M., Del Castillo L., Perez J. (2003) Failure prevention and diagnosis in machine elements using cluster, in: Proceedings of the Tenth International Congress on Sound and Vibration, Stockholm, Sweden, pp. 1197–1203.

28. Bengtsson M., Olsson E., Funk P., Jackson M. (2004) Technical design of condition based maintenance system—A case study using sound analysis and case-based reasoning, in: Maintenance and Reliability Conference—Proceedings of the Eighth Congress, Knoxville, USA, 2004.

29. Campbell J.E., Thompson B.M., Swiler L.P. (2002) Consequence analysis in predictive health monitoring systems, in: Proceedings of Probabilistic Safety Assessment and Management, vols. I and II, Amsterdam, 2002, pp. 1353–1358.

30. Iung B. (2003) From remote maintenance to MAS – based E – maintenance of an industrial process. Journal of Intelligent Manufacturing, 14(1), 59 – 82.

31. Jardine A.K.S., Lin D., Banjevic D. (2006) A review on machinery diagnostics and prognostics implementing condition based maintenance. Mechanical Systems and Signal Processing, 20, 1483–1510

32. Lee J., Ni J., Djurdjanovic D., Qiu H., Liao H. (2006) Intelligent prognostics tools and e-maintenance. Computers in Industry, 57, 476 – 489.

33. Sohn H., Worden K., Farrar C.R. (2002) Statistical damage classification under changing environmental and operational conditions, Journal of Intelligent Material Systems and Structures, 13, 561–574.

34. Wang W.Y. (2002) Towards dynamic model-based prognostics for transmission gears, in: Component and Systems Diagnostics, Prognostics, and Health Management II, 4733, 157–167.

35. Wang W.Q., Golnaraghi M.F., Ismail F., (2004) Prognosis of machine health condition using neuro-fuzzy systems, Mechanical Systems and Signal Processing, 18, 813–831.

36. Furlanetto L., Garetti M., Macchi M. (2006). Principi generali di gestione della manutenzione. Milano, Franco Angeli (Italian).

37. UNI 10366: 1994, Manutenzione – Criteri di progettazione della manutenzione,1994.

38. UNI 13306: 2003, Manutenzione – Terminologia, 2003

39. UNI 9910: 1991, Terminologia sulla fidatezza e sulla qualità del servizio, 1991

40. UNI 10147: 1993, Manutenzione – Terminologia, 1993.

41. Jardine, Andrew K. S., Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance." *Mechanical Systems and Signal Processing*, 2006: 1483–1510.

Attachments

Matlab scripts to obtain reference indices:

```
clear all
close all
clc

 N_prove=80;
 %posizione firma di riferimento FASE 1 NON FILTRATA
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento\Fase1\Non_filtrata');
for i=1:N_prove;
    filename=['F1_',int2str(i),'_NF.txt'];
    MF1{i}=load(filename,'-ascii');
end;
 %posizione firma di riferimento FASE 2 NON FILTRATA
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento\Fase2\Non_filtrata');
for i=1:N_prove;
    filename=['F2_',int2str(i),'_NF.txt'];
    MF2{i}=load(filename,'-ascii');
end;
 %posizione firma di riferimento FASE 3 NON FILTRATA
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento\Fase3\Non_filtrata');
for i=1:N_prove;
    filename=['F3_',int2str(i),'_NF.txt'];
    MF3{i}=load(filename,'-ascii');
end;
 %posizione firma di riferimento FASE 4 NON FILTRATA
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento\Fase4\Non_filtrata');
for i=1:N_prove;
    filename=['F4_',int2str(i),'_NF.txt'];
    MF4{i}=load(filename,'-ascii');
end;
 %posizione firma di riferimento FASE 5 NON FILTRATA
```

```
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento\Fase5\Non_filtrata');
for i=1:N_prove;
    filename=['F5_',int2str(i),'_NF.txt'];
    MF5{i}=load(filename,'-ascii');
end;
 %posizione firma di riferimento FASE 6 NON FILTRATA
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento\Fase6\Non_filtrata');
for i=1:N_prove;
    filename=['F6_',int2str(i),'_NF.txt'];
    MF6{i}=load(filename,'-ascii');
end;
 %posizione file di salvataggio
cd('C:\My Dropbox\Università\Programmazione\Dati new\Firme_riferimento');
 %--------------------CALCOLO AREA FIRMA DI RIFERIMENTO --------------------
%fase1
for i=1:N_prove
    AreaF1(i)=trapz(MF1{i});
end;
AreaF1=AreaF1';
%fase2
for i=1:N_prove
    AreaF2(i)=trapz(MF2{i});
end;
AreaF2=AreaF2';


%fase3
for i=1:N_prove
    AreaF3(i)=trapz(MF3{i});
end;
AreaF3=AreaF3';
 %fase4
for i=1:N_prove
    AreaF4(i)=trapz(MF4{i});
end;
```

```
AreaF4=AreaF4';
 %fase5
for i=1:N_prove
    AreaF5(i)=trapz(MF5{i});
end;
AreaF5=AreaF5';
 %fase6
for i=1:N_prove
    AreaF6(i)=trapz(MF6{i});
end;
AreaF6=AreaF6';
 xlswrite('Matrice indicatori di riferimento_NF.xlsx',AreaF1,'Area','B2:B81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',AreaF2,'Area','C2:C81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',AreaF3,'Area','D2:D81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',AreaF4,'Area','E2:E81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',AreaF5,'Area','F2:F81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',AreaF6,'Area','G2:G81');
 %---------------------------CURTOSI-------------------------------------
%Fase1
for i=1:N_prove;
    kurt_rif1(i)=kurtosis(MF1{i});
end;
kurt_rif1=kurt_rif1';
 %Fase2
for i=1:N_prove;
    kurt_rif2(i)=kurtosis(MF2{i});
end;
kurt_rif2=kurt_rif2';
 %Fase3
for i=1:N_prove;
    kurt_rif3(i)=kurtosis(MF3{i});
end;
kurt_rif3=kurt_rif3';
 %Fase4
```

```
for i=1:N_prove;
    kurt_rif4(i)=kurtosis(MF4{i});
end;
kurt_rif4=kurt_rif4';
 %Fase5
for i=1:N_prove;
    kurt_rif5(i)=kurtosis(MF5{i});
end;
kurt_rif5=kurt_rif5';
 %Fase6
for i=1:N_prove;
    kurt_rif6(i)=kurtosis(MF6{i});
end;
kurt_rif6=kurt_rif6';
xlswrite('Matrice indicatori di riferimento_NF.xlsx',kurt_rif1,'kurtosis','B2:B81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',kurt_rif2,'kurtosis','C2:C81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',kurt_rif3,'kurtosis','D2:D81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',kurt_rif4,'kurtosis','E2:E81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',kurt_rif5,'kurtosis','F2:F81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',kurt_rif6,'kurtosis','G2:G81');
 %-------------------------- SKEWNESS ------------------------------------
%Fase1
for i=1:N_prove;
    skew_rif1(i)=skewness(MF1{i});
end;
skew_rif1=skew_rif1';
 %Fase2
for i=1:N_prove;
    skew_rif2(i)=skewness(MF2{i});
end;
skew_rif2=skew_rif2';
 %Fase3
for i=1:N_prove;
    skew_rif3(i)=skewness(MF3{i});
```

```
end;
skew_rif3=skew_rif3';
 %Fase4
for i=1:N_prove;
    skew_rif4(i)=skewness(MF4{i});
end;
skew_rif4=skew_rif4';
 %Fase5
for i=1:N_prove;
    skew_rif5(i)=skewness(MF5{i});
end;
skew_rif5=skew_rif5';
 %Fase6
for i=1:N_prove;
    skew_rif6(i)=skewness(MF6{i});
end;
skew_rif6=skew_rif6';
 xlswrite('Matrice indicatori di riferimento_NF.xlsx',skew_rif1,'skewness','B2:B81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',skew_rif2,'skewness','C2:C81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',skew_rif3,'skewness','D2:D81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',skew_rif4,'skewness','E2:E81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',skew_rif5,'skewness','F2:F81');
xlswrite('Matrice indicatori di riferimento_NF.xlsx',skew_rif6,'skewness','G2:G81');
 %------------------------------ CREST -------------------------------------
%fase1
for j=1:N_prove;
    tot1=sum(MF1{j}.^2);
    msv1=tot1/length(MF1{j}); %mean square value
    rms1=sqrt(msv1); %RMS value
    cf1(j)=max(abs(MF1{j}-mean(MF1{j})))/rms1; %crest factor
end;
cf1=cf1';
 %fase2
for j=1:N_prove;
```

```
tot1=sum(MF2{j}.^2);
    msv1=tot1/length(MF2{j}); %mean square value
    rms1=sqrt(msv1); %RMS value
    cf2(j)=max(abs(MF2{j}-mean(MF2{j})))/rms1; %crest factor
end;
cf2=cf2';
 %fase3
for j=1:N_prove;
    tot1=sum(MF3{j}.^2);
    msv1=tot1/length(MF3{j}); %mean square value
    rms1=sqrt(msv1); %RMS value
    cf3(j)=max(abs(MF3{j}-mean(MF3{j})))/rms1; %crest factor
end;
cf3=cf3';
 %fase4
for j=1:N_prove;
    tot1=sum(MF4{j}.^2);
    msv1=tot1/length(MF4{j}); %mean square value
    rms1=sqrt(msv1); %RMS value
    cf4(j)=max(abs(MF4{j}-mean(MF4{j})))/rms1; %crest factor
end;
cf4=cf4';
 %fase5
for j=1:N_prove;
    tot1=sum(MF5{j}.^2);
    msv1=tot1/length(MF5{j}); %mean square value
    rms1=sqrt(msv1); %RMS value
    cf5(j)=max(abs(MF5{j}-mean(MF5{j})))/rms1; %crest factor
end;
cf5=cf5';
 %fase6
for j=1:N_prove;
    tot1=sum(MF6{j}.^2);
    msv1=tot1/length(MF6{j}); %mean square value
```

```
    rms1=sqrt(msv1); %RMS value

    cf6(j)=max(abs(MF6{j}-mean(MF6{j})))/rms1; %crest factor

end;

cf6=cf6';





 xlswrite('Matrice indicatori di riferimento_NF.xlsx',cf1,'crest','B2:B81');

xlswrite('Matrice indicatori di riferimento_NF.xlsx',cf2,'crest','C2:C81');

xlswrite('Matrice indicatori di riferimento_NF.xlsx',cf3,'crest','D2:D81');

xlswrite('Matrice indicatori di riferimento_NF.xlsx',cf4,'crest','E2:E81');

xlswrite('Matrice indicatori di riferimento_NF.xlsx',cf5,'crest','F2:F81');

xlswrite('Matrice indicatori di riferimento_NF.xlsx',cf6,'crest','G2:G81');
```

Matlab scripts to optain PCA and diagnosis analysis :

```
cd('Z:\Desktop\DIAGNOSI\file_system');
G=giorno
save saveG.mat G
M=mese
save saveM.mat M
N=anno
save saveN.mat N
N_prove=1
%posizione firma di riferimento FASE 1 NON FILTRATA
cd('E:\ESA\Dati new\Firme_degrado\Cinghie\Cinghie Non Tirate 1\Fase1\Non_filtrate')
%cd('/Volumes/LaCie/ESA/Dati new/Firme_degrado/Cinghie/Cinghie Non Tirate 1/Fase1/Non_filtrate');
for i=1:N_prove;
    filename=['F1_',int2str(i),'_DCNT1_NF.txt'];
    M1{i}=load(filename,'-ascii');
end;
cd('E:\ESA\Dati new\Firme_degrado\Cinghie\Cinghie Non Tirate 1\Fase2\Non_filtrate')
%posizione firma di riferimento FASE 2 NON FILTRATA
%cd('/Volumes/LaCie/ESA/Dati new/Firme_degrado/Cinghie/Cinghie Non Tirate 1/Fase2/Non_filtrate');
for i=1:N_prove;
    filename=['F2_',int2str(i),'_DCNT1_NF.txt'];
    M2{i}=load(filename,'-ascii');
end;

%posizione firma di riferimento FASE 3 NON FILTRATA
cd('E:\ESA\Dati new\Firme_degrado\Cinghie\Cinghie Non Tirate 1\Fase3\Non_filtrate')
%cd('/Volumes/LaCie/ESA/Dati new/Firme_degrado/Cinghie/Cinghie Non Tirate 1/Fase3/Non_filtrate');
for i=1:N_prove;
    filename=['F3_',int2str(i),'_DCNT1_NF.txt'];
    M3{i}=load(filename,'-ascii');
end;

%posizione firma di riferimento FASE 4 NON FILTRATA
cd('E:\ESA\Dati new\Firme_degrado\Cinghie\Cinghie Non Tirate 1\Fase4\Non_filtrate')
%cd('/Volumes/LaCie/ESA/Dati new/Firme_degrado/Cinghie/Cinghie Non Tirate 1/Fase4/Non_filtrate');
for i=1:N_prove;
    filename=['F4_',int2str(i),'_DCNT1_NF.txt'];
    M4{i}=load(filename,'-ascii');
end;
```

```
%posizione firma di riferimento FASE 5 NON FILTRATA
cd('E:\ESA\Dati new\Firme_degrado\Cinghie\Cinghie Non Tirate 1\Fase5\Non_filtrate')
%cd('/Volumes/LaCie/ESA/Dati new/Firme_degrado/Cinghie/Cinghie Non Tirate 1/Fase5/Non_filtrate');
for i=1:N_prove;
   filename=['F5_',int2str(i),'_DCNT1_NF.txt'];
   M5{i}=load(filename,'-ascii');
end;
%posizione firma di riferimento FASE 6 NON FILTRATA
cd('E:\ESA\Dati new\Firme_degrado\Cinghie\Cinghie Non Tirate 1\Fase6\Non_filtrate')
%cd('/Volumes/LaCie/ESA/Dati new/Firme_degrado/Cinghie/Cinghie Non Tirate 1/Fase6/Non_filtrate');
for i=1:N_prove;

   filename=['F6_',int2str(i),'_DCNT1_NF.txt'];
   M6{i}=load(filename,'-ascii');
end;


%--------------------CALCOLO AREA FIRMA DI RIFERIMENTO --------------------
%fase1
for i=1:N_prove
   AreaF11(i)=trapz(M1{i});
end;
AreaF11=AreaF11';

%fase2
for i=1:N_prove
   AreaF22(i)=trapz(M2{i});
end;
AreaF22=AreaF22';

%fase3
for i=1:N_prove
   AreaF33(i)=trapz(M3{i});
end;
AreaF33=AreaF33';

%fase4
for i=1:N_prove
   AreaF44(i)=trapz(M4{i});
end;
AreaF44=AreaF44';

%fase5
for i=1:N_prove
   AreaF55(i)=trapz(M5{i});
end;
AreaF55=AreaF55';

%fase6
for i=1:N_prove
   AreaF66(i)=trapz(M6{i});
end;
AreaF66=AreaF66';



%----------------------------CURTOSIS---------------------------------------
%Fase1
for i=1:N_prove;
   kurt_rif11(i)=kurtosis(M1{i});
end;
kurt_rif11=kurt_rif11';

%Fase2
for i=1:N_prove;
   kurt_rif22(i)=kurtosis(M2{i});
end;
kurt_rif22=kurt_rif22';

%Fase3
```

```
for i=1:N_prove;
   kurt_rif33(i)=kurtosis(M3{i});
end;
kurt_rif33=kurt_rif33';

%Fase4
for i=1:N_prove;
   kurt_rif44(i)=kurtosis(M4{i});
end;
kurt_rif44=kurt_rif44';

%Fase5
for i=1:N_prove;
   kurt_rif55(i)=kurtosis(M5{i});
end;
kurt_rif55=kurt_rif55';

%Fase6
for i=1:N_prove;
   kurt_rif66(i)=kurtosis(M6{i});
end;
kurt_rif66=kurt_rif66';



%--------------------------- SKEWNESS -------------------------------------
%Fase1
for i=1:N_prove;
   skew_rif11(i)=skewness(M1{i});
end;
skew_rif11=skew_rif11';

%Fase2
for i=1:N_prove;
   skew_rif22(i)=skewness(M2{i});
end;
skew_rif22=skew_rif22';

%Fase3
for i=1:N_prove;
   skew_rif33(i)=skewness(M3{i});
end;
skew_rif33=skew_rif33';

%Fase4
for i=1:N_prove;
   skew_rif44(i)=skewness(M4{i});
end;
skew_rif44=skew_rif44';

%Fase5
for i=1:N_prove;
   skew_rif55(i)=skewness(M5{i});
end;
skew_rif55=skew_rif55';

%Fase6
for i=1:N_prove;
   skew_rif66(i)=skewness(M6{i});
end;
skew_rif66=skew_rif66';



%------------------------------ CREST -------------------------------------
%fase1
for j=1:N_prove;
   tot1=0;
   for i=1:length(M1{j})
      tot1=tot1+(M1{j}(i))^2;
```

```
      diffr1(i)=abs(M1{j}(i)-mean(M1{j}));
   end;
   msv1=tot1/length(M1{j}); %mean square value
   rms1=sqrt(msv1); %RMS value
   cf11(j)=max(diffr1)/rms1; %crest factor
end;
cf11=cf11';

%fase2
for j=1:N_prove;
   tot1=0;
   for i=1:length(M2{j})
      tot1=tot1+(M2{j}(i))^2;
      diffr1(i)=abs(M2{j}(i)-mean(M2{j}));
   end;
   msv1=tot1/length(M2{j}); %mean square value
   rms1=sqrt(msv1); %RMS value
   cf22(j)=max(diffr1)/rms1; %crest factor
end;
cf22=cf22';

%fase3
for j=1:N_prove;
   tot1=0;
   for i=1:length(M3{j})
      tot1=tot1+(M3{j}(i))^2;
      diffr1(i)=abs(M3{j}(i)-mean(M3{j}));
   end;
   msv1=tot1/length(M3{j}); %mean square value
   rms1=sqrt(msv1); %RMS value
   cf33(j)=max(diffr1)/rms1; %crest factor
end;
cf33=cf33';

%fase4
for j=1:N_prove;
   tot1=0;
   for i=1:length(M4{j})
      tot1=tot1+(M4{j}(i))^2;
      diffr1(i)=abs(M4{j}(i)-mean(M4{j}));
   end;
   msv1=tot1/length(M4{j}); %mean square value
   rms1=sqrt(msv1); %RMS value
   cf44(j)=max(diffr1)/rms1; %crest factor
end;
cf44=cf44';

%fase5
for j=1:N_prove;
   tot1=0;
   for i=1:length(M5{j})
      tot1=tot1+(M5{j}(i))^2;
      diffr1(i)=abs(M5{j}(i)-mean(M5{j}));
   end;
   msv1=tot1/length(M5{j}); %mean square value
   rms1=sqrt(msv1); %RMS value
   cf55(j)=max(diffr1)/rms1; %crest factor
end;
cf55=cf55';

%fase6
for j=1:N_prove;
   tot1=0;
   for i=1:length(M6{j})
      tot1=tot1+(M6{j}(i))^2;
      diffr1(i)=abs(M6{j}(i)-mean(M6{j}));
   end;
   msv1=tot1/length(M6{j}); %mean square value
   rms1=sqrt(msv1); %RMS value
   cf66(j)=max(diffr1)/rms1; %crest factor
```

```
end;
cf66=cf66';


%%%%%%%%%%%%%%%%%%%%

Xnew11=[AreaF11 kurt_rif11 skew_rif11 cf11];
Xnew22=[AreaF22 kurt_rif22 skew_rif22 cf22];
Xnew33=[AreaF33 kurt_rif33 skew_rif33 cf33];
Xnew44=[AreaF44 kurt_rif44 skew_rif44 cf44];
Xnew55=[AreaF55 kurt_rif55 skew_rif55 cf55];
Xnew66=[AreaF66 kurt_rif66 skew_rif66 cf66];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%DIAGNOSI A PARTIRE DAGLI INDICI
%cd('/Users/paoloberno/Desktop/DIAGNOSI/file_system');
cd('Z:\Desktop\DIAGNOSI\file_system');
SP=[]
save saveSP.mat SP
SP2=[]
save saveSP2.mat SP2
load saveXmean.mat
load saveXstd.mat
load saveVk.mat
load savesigma.mat
load savespelim1.mat
load  savet2lim1.mat
Sxnew11=[];

for k=1:N_prove

Sxnew11=[Sxnew11,((Xnew11(k,:))'-Xmean')./Xstd'];

end


%calcolo scores
t11=[];
for k=1:N_prove
   t11=[t11,Vk'*Sxnew11(:,k)];
end
t11=t11';
%X1=max(t11)
% M1=min(t11)
 %save saveM1.mat M1
 %save saveX1.mat X1
t1n=t11
save savet1n.mat t1n
T21=[];
for k=1:N_prove

   T21=[T21,(Sxnew11(:,k))'*Vk*inv(sigma)*Vk'*Sxnew11(:,k)];

end
T21n=T21'
save savet21n.mat T21n
r1=[];
for k=1:N_prove
     r1=[r1,(eye(4,4)-Vk*Vk')*Sxnew11(:,k)];
end
SPE1=[];
for k=1:N_prove
     SPE1=[SPE1,r1(:,k)'*r1(:,k)];
end

SPE1n=SPE1'

save saveSPE1n.mat SPE1n

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
load saveXmean2.mat
load saveXstd2.mat
load saveVk2.mat
load savesigma2.mat
load  savespelim2.mat
load  savet2lim2.mat
%standardizzo fase2

Sxnew22=[];
for k=1:N_prove

Sxnew22=[Sxnew22,((Xnew22(k,:))'-Xmean2')./Xstd2'];

end


%calcolo scores
t22=[];
for k=1:N_prove
   t22=[t22,Vk2'*Sxnew22(:,k)];
end
t22=t22';

 %X2=max(t22)
 %M2=min(t22)
 %save saveM2.mat M2
 %save saveX2.mat X2
t2n=t22;
save savet2n.mat t2n
T22=[];
for k=1:N_prove

   T22=[T22,(Sxnew22(:,k))'*Vk2*inv(sigma2)*Vk2'*Sxnew22(:,k)];

end
T22n=T22'
save savet22n.mat T22n
%calcolo SPE del nuovo dato dalla condizione di buon funzionamento
r2=[];
for k=1:N_prove
      r2=[r2,(eye(4,4)-Vk2*Vk2')*Sxnew22(:,k)];
end
SPE2=[];
for k=1:N_prove
      SPE2=[SPE2,r2(:,k)'*r2(:,k)];
end

SPE2n=SPE2'
save saveSPE2n.mat SPE2n

%
% %standardizzo fase3
load saveXmean3.mat
load saveXstd3.mat
load saveVk3.mat
load savesigma3.mat
load savespelim3.mat
load  savet2lim3.mat

Sxnew33=[];
for k=1:N_prove

Sxnew33=[Sxnew33,((Xnew33(k,:))'-Xmean3')./Xstd3'];

end


%calcolo scores
t33=[];
for k=1:N_prove
```

```
   t33=[t33,Vk3'*Sxnew33(:,k)];
end
t33=t33';
t3n=t33;

 %X3=max(t33)
 %M3=min(t33)
 %save saveM3.mat M3
 %save saveX3.mat X3

save savet3n.mat t3n
T23=[];
for k=1:N_prove

   T23=[T23,(Sxnew33(:,k))'*Vk3*inv(sigma3)*Vk3'*Sxnew33(:,k)];

end
T23n=T23'
save savet23n.mat T23n
%calcolo SPE del nuovo dato dalla condizione di buon funzionamento
r3=[];
for k=1:N_prove
     r3=[r3,(eye(4,4)-Vk3*Vk3')*Sxnew33(:,k)];
end
SPE3=[];
for k=1:N_prove
     SPE3=[SPE3,r3(:,k)'*r3(:,k)];
end

SPE3n=SPE3'
save saveSPE3n.mat SPE3n
%standardizzo fase4

load saveXmean4.mat
load saveXstd4.mat
load saveVk4.mat
load savesigma4.mat
load  savespelim4.mat
load savet2lim4.mat
Sxnew44=[];
for k=1:N_prove

Sxnew44=[Sxnew44,((Xnew44(k,:))'-Xmean4')./Xstd4'];

end


%calcolo scores
t44=[];
for k=1:N_prove
   t44=[t44,Vk4'*Sxnew44(:,k)];
end
t44=t44';

 %X4=max(t44)
 %M4=min(t44)
 %save saveM4.mat M4
 %save saveX4.mat X4
t4n=t44
save savet4n.mat t4n
T24=[];
for k=1:N_prove

   T24=[T24,(Sxnew44(:,k))'*Vk4*inv(sigma4)*Vk4'*Sxnew44(:,k)];

end
T24n=T24'
save savet24n.mat T24n
%calcolo SPE del nuovo dato dalla condizione di buon funzionamento
r4=[];
```

```
for k=1:N_prove
      r4=[r4,(eye(4,4)-Vk4*Vk4')*Sxnew44(:,k)];
end
SPE4=[];
for k=1:N_prove
      SPE4=[SPE4,r4(:,k)'*r4(:,k)];
end

SPE4n=SPE4'
save saveSPE4n.mat SPE4n

%standardizzo fase5
load saveXmean5.mat
load saveXstd5.mat
load saveVk5.mat
load savesigma5.mat
load savespelim5.mat
load  savet2lim5.mat
%standardizzo fase5


Sxnew55=[];
for k=1:N_prove

Sxnew55=[Sxnew55,((Xnew55(k,:))'-Xmean5')./Xstd5'];

end
t55=[];
for k=1:N_prove
   t55=[t55,Vk5'*Sxnew55(:,k)];
end
t55=t55';
 %X5=max(t55)
 %M5=min(t55)
 %save saveM5.mat M5
 %save saveX5.mat X5


t5n=t55
save savet5n.mat t5n

T25=[];
for k=1:N_prove

   T25=[T25,(Sxnew55(:,k))'*Vk5*inv(sigma5)*Vk5'*Sxnew55(:,k)];

end
T25n=T25'
save savet25n.mat T25n

%calcolo SPE del nuovo dato dalla condizione di buon funzionamento
r5=[];
for k=1:N_prove
      r5=[r5,(eye(4,4)-Vk5*Vk5')*Sxnew55(:,k)];
end
SPE5=[];
for k=1:N_prove
      SPE5=[SPE5,r5(:,k)'*r5(:,k)];
end

SPE5n=SPE5'
save saveSPE5n.mat SPE5n



%standardizzo fase6
load saveXmean6.mat
load saveXstd6.mat
load saveVk6.mat
load savesigma6.mat
```

```
load  savespelim6.mat
load savet2lim6.mat

Sxnew66=[];
for k=1:N_prove

Sxnew66=[Sxnew66,((Xnew66(k,:))'-Xmean6')./Xstd6'];

end
t66=[];
for k=1:N_prove
    t66=[t66,Vk6'*Sxnew66(:,k)];
end
t66=t66';

% X6=max(t66)
 %M6=min(t66)
 %save saveM6.mat M6
 %save saveX6.mat X6
t6n=t66;
save savet6n.mat t6n
T26=[];
for k=1:N_prove

    T26=[T26,(Sxnew66(:,k))'*Vk6*inv(sigma6)*Vk6'*Sxnew66(:,k)];

end
T26n=T26'
save savet26n.mat T26n
%calcolo SPE del nuovo dato dalla condizione di buon funzionamento
r6=[];
for k=1:N_prove
       r6=[r6,(eye(4,4)-Vk6*Vk6')*Sxnew66(:,k)];
end
SPE6=[];
for k=1:N_prove
       SPE6=[SPE6,r6(:,k)'*r6(:,k)];
end

SPE6n=SPE6'
save saveSPE6n.mat SPE6n


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
T2=[T21n;T22n;T23n;T24n;T25n;T26n];
SPE=[SPE1n;SPE2n;SPE3n;SPE4n;SPE5n;SPE6n];
T2lim=[T2lim_1;T2lim_2;T2lim_3;T2lim_4;T2lim_5;T2lim_6];
SPElim=[SPElim1;SPElim2;SPElim3;SPElim4;SPElim5;SPElim6];
ex=[];
ex2=[];

for l=1:6
    if T2(l)>T2lim(l)
        ex(l)=[T2(l)];
    else
        if SPE(l)>SPElim(l)
            ex2(l)=[SPE(l)];
        end
    end
end

load saveSP.mat
load saveSP2.mat

A=find (ex>0);
B=find (ex2>0);
cod=[];
evento=[];
SP=sum(A);
```

```
SP2=sum(B);
save saveSP.mat SP
save saveSP2.mat SP2
load saveSP.mat
load saveSP2.mat
```

Matlab scripts to optain FMECA:

```
%carico fmeca
A=[];
%cd('Z:\Desktop\DIAGNOSI\FMECA');

[code,comp,descr,mod,causa,eff,num,mtbf,mdt,s,f,c] = textread(FMECA,'%d %s %s %s %s %s %d %d %d %d %d %d',k);

for i=1:k
   cd('Z:\Desktop\DIAGNOSI\file_system');
   load savefmeca.mat
   A=[code(i,1),comp(i,1),descr(i,1),mod(i,1),causa(i,1),eff(i,1),num(i,1),mtbf(i,1),mdt(i,1),s(i,1),f(i,1),c(i,1)];
   fmecaold=[fmecaold;A];
   save savefmeca.mat fmecaold
   save savefmecanew.mat fmecaold
end

save savecode.mat code
save savecomp.mat comp
save savedescr.mat descr
save savemod.mat mod
save savecausa.mat causa
save saveeff.mat eff
save savenum.mat num
save savemtbf.mat mtbf
save savemdt.mat mdt
save savef.mat f
save saves.mat s
save savec.mat c
```

Matlab scripts to obtain updated FMECA:

```
cd('Z:\Desktop\DIAGNOSI\file_system');

load savefmecanew.mat
load saveF.mat
load saveS.mat
load saveMin1.mat
load saveMax1.mat
load saveMin2.mat
load saveMax2.mat


cd('Z:\Desktop\DIAGNOSI\file_system');

load saveZ.mat
load saveD.mat

load savedata.mat
[w,s]=size(dataold)
R=[]
for i=1:w
    R=[R,dataold{i,1}]
end
R=R'
C=R
cd('Z:\Desktop\DIAGNOSI\file_system');
            load savedata.mat
            [w,s]=size(dataold)
            anno=[]
            for x=1:w
             anno=[anno,dataold{x,5}]
            end
             anno=anno'
              cd('Z:\Desktop\DIAGNOSI\file_system');
            load savedata.mat
            [w,s]=size(dataold)
            mese=[]
            for x=1:w
             mese=[mese,dataold{x,4}]
            end
             mese=mese'
             cd('Z:\Desktop\DIAGNOSI\file_system');
            load savedata.mat
            [w,s]=size(dataold)
            giorno=[]
            for x=1:w
             giorno=[giorno,dataold{x,3}]
            end
             giorno=giorno'

v=2
 for i=1:(length(C)-1)
    w=(length(C))
     for j=v:w

        Ax=[]
        if C(i)==C(j)
                  Ax= strcmp(dataold{i,2},dataold{j,2})
                  if  Ax==1

        %%%%%%%%%calcolo TBF=MDT

        if anno(i)==anno(j)
        if mese(i)==1
            if mese(j)==1
        MDT=giorno(j)-giorno(i)
```

```
else if mese(j)==2
      MDT=31+(giorno(j)-giorno(i))
   else if mese(j)==3
         MDT=59+(giorno(j)-giorno(i))
      else if mese(j)==4
         MDT=90+(giorno(j)-giorno(i))
         else if mese(j)==5
             MDT=120+(giorno(j)-giorno(i))
           else if mese(j)==6
             MDT=151+(giorno(j)-giorno(i))
             else if mese(j)==7
               MDT=181+(giorno(j)-giorno(i))
               else if mese(j)==8
                  MDT=212+(giorno(j)-giorno(i))
                 else if mese(j)==9
                    MDT=243+(giorno(j)-giorno(i))
                    else if mese(j)==10
                      MDT=273+(giorno(j)-giorno(i))
                      else if mese(j)==11
                        MDT=304+(giorno(j)-giorno(i))
                        else if mese(j)==12
                          MDT=334+(giorno(j)-giorno(i))
                           end
                         end
                       end
                    end
                  end
                end
              end
            end
         end
      end
   end
   end
else  if mese(i)==2
 if mese(j)==2
MDT=(giorno(j)-giorno(i))
else if mese(j)==3
      MDT=28+(giorno(j)-giorno(i))
   else if mese(j)==4
         MDT=59+(giorno(j)-giorno(i))
      else if mese(j)==5
         MDT=89+(giorno(j)-giorno(i))
         else if mese(j)==6
             MDT=120+(giorno(j)-giorno(i))
           else if mese(j)==7
             MDT=150+(giorno(j)-giorno(i))
             else if mese(j)==8
               MDT=181+(giorno(j)-giorno(i))
               else if mese(j)==9
                  MDT=212+(giorno(j)-giorno(i))
                 else if mese(j)==10
                    MDT=242+(giorno(j)-giorno(i))
                    else if mese(j)==11
                      MDT=273+(giorno(j)-giorno(i))
                      else if mese(j)==12
                            MDT=303+(giorno(j)-giorno(i))
                         end
                       end
                    end
                  end
                end
              end
            end
         end
      end
   end
 end
   else if mese(i)==3
```

```
 if mese(j)==3
MDT=giorno(j)-giorno(i)
else if mese(j)==4
      MDT=31+(giorno(j)-giorno(i))
   else if mese(j)==5
         MDT=61+(giorno(j)-giorno(i))
      else if mese(j)==6
         MDT=92+(giorno(j)-giorno(i))
         else if mese(j)==7
            MDT=122+(giorno(j)-giorno(i))
           else if mese(j)==8
             MDT=153+(giorno(j)-giorno(i))
            else if mese(j)==9
              MDT=184+(giorno(j)-giorno(i))
              else if mese(j)==10
                 MDT=214+(giorno(j)-giorno(i))
                else if mese(j)==11
                   MDT=245+(giorno(j)-giorno(i))
                   else if mese(j)==12
                         MDT=275+(giorno(j)-giorno(i))
                      end
                   end
                 end
               end
              end
            end
         end
      end
   end
 end
      else if mese(i)==4
 if mese(j)==4
MDT=giorno(j)-giorno(i)
else if mese(j)==5
      MDT=30+(giorno(j)-giorno(i))
   else if mese(j)==6
         MDT=61+(giorno(j)-giorno(i))
      else if mese(j)==7
         MDT=91+(giorno(j)-giorno(i))
         else if mese(j)==8
            MDT=122+(giorno(j)-giorno(i))
           else if mese(j)==9
             MDT=153+(giorno(j)-giorno(i))
            else if mese(j)==10
              MDT=183+(giorno(j)-giorno(i))
              else if mese(j)==11
                 MDT=214+(giorno(j)-giorno(i))
                else if mese(j)==12
                   MDT=244+(giorno(j)-giorno(i))

                      end
                   end
                 end
               end
              end
            end
         end
      end
   end

         else if mese(i)==5
 if mese(j)==5
MDT=giorno(j)-giorno(i)
else if mese(j)==6
      MDT=31+(giorno(j)-giorno(i))
   else if mese(j)==7
         MDT=61+(giorno(j)-giorno(i))
      else if mese(j)==8
         MDT=91+(giorno(j)-giorno(i))
         else if mese(j)==9
```

```
                  MDT=122+(giorno(j)-giorno(i))
                 else if mese(j)==10
                   MDT=152+(giorno(j)-giorno(i))
                  else if mese(j)==11
                    MDT=183+(giorno(j)-giorno(i))
                    else if mese(j)==12
                       MDT=213+(giorno(j)-giorno(i))
                       end
                     end
                    end
                  end
               end
            end
         end
       end
                  else if mese(i)==6
       if mese(j)==6
      MDT=(giorno(j)-giorno(i))
      else if mese(j)==7
            MDT=30+(giorno(j)-giorno(i))
         else if mese(j)==8
               MDT=61+(giorno(j)-giorno(i))
            else if mese(j)==9
               MDT=92+(giorno(j)-giorno(i))
               else if mese(j)==10
                  MDT=122+(giorno(j)-giorno(i))
                 else if mese(j)==11
                   MDT=153+(giorno(j)-giorno(i))
                  else if mese(j)==12
                    MDT=183+(giorno(j)-giorno(i))
                     end
                    end
                  end
               end
            end
         end
       end
                    else if mese(i)==7
       if mese(j)==7
      MDT=giorno(j)-giorno(i)
      else if mese(j)==8
            MDT=31+(giorno(j)-giorno(i))
            else if mese(j)==9
               MDT=62+(giorno(j)-giorno(i))
               else if mese(j)==10
                  MDT=92+(giorno(j)-giorno(i))
                 else if mese(j)==11
                   MDT=123+(giorno(j)-giorno(i))
                  else if mese(j)==12
                    MDT=153+(giorno(j)-giorno(i))
                     end
                    end
                  end
               end
         end
       end
   else if mese(i)==8
       if mese(j)==8
      MDT=giorno(j)-giorno(i)
            else if mese(j)==9
               MDT=31+(giorno(j)-giorno(i))
               else if mese(j)==10
                  MDT=61+(giorno(j)-giorno(i))
                 else if mese(j)==11
                   MDT=92+(giorno(j)-giorno(i))
                  else if mese(j)==12
                    MDT=122+(giorno(j)-giorno(i))
                     end
                    end
                  end
```

```
            end
      end
     else if mese(i)==9
      if mese(j)==9
        MDT=giorno(j)-giorno(i)
             else if mese(j)==10
                 MDT=30+(giorno(j)-giorno(i))
                 else if mese(j)==11
                      MDT=61+(giorno(j)-giorno(i))
                    else if mese(j)==12
                       MDT=91+(giorno(j)-giorno(i))
                        end
                     end
                 end
      end
 else if mese(i)==10
      if mese(j)==10
          MDT=(giorno(j)-giorno(i))
             else if mese(j)==11
                 MDT=31+(giorno(j)-giorno(i))
                 else if mese(j)==12
                      MDT=61+(giorno(j)-giorno(i))
                     end
                 end
      end

     else if  mese(i)==11
      if mese(j)==11
     MDT=(giorno(j)-giorno(i))
             else if mese(j)==12
                 MDT=30+(giorno(j)-giorno(i))
                 end
      end

     else if  mese(i)==12
      if mese(j)==12
     MDT=(giorno(j)-giorno(i))
      end

         end
         end
     end
         end
     end
                         end
                     end
                     end
                 end
             end
         end
     end

     else if anno(j)-anno(i) >0
             k=anno(j)-anno(i)
             MDT=MDT+k*(365)

         end
%
     end
     [b,q]=size(fmecaold)
     W=[]
     for p=1:b
     W=[W,fmecaold{p,1}]
     end
     W=W'
      h=find(W(:,1)==C(i,1))
      fmecaold{h,8}=((fmecaold{h,8}*(fmecaold{h,7}))+MDT)/(fmecaold{h,7}+1)
      fmecaold{h,7}=(fmecaold{h,7}+1)   %aggiorna numero eventi
    save savefmeca.mat fmecaold
   if fmecaold{h,8}>Min1(1,1)
```

```
          if fmecaold{h,8}<Max1(1,1)
             fmecaold{h,11}= F(1,1)
          end
      end
       if fmecaold{h,8}>Min1(2,1)
          if fmecaold{h,8}<Max1(2,1)
            fmecaold{h,11}= F(2,1)
          end
       end
       if fmecaold{h,8}>Min1(3,1)
          if fmecaold{h,8}<Max1(3,1)
            fmecaold{h,11}= F(3,1)
          end
          end
       if fmecaold{h,9}>Min2(1,1)
          if fmecaold{h,9}<Max2(1,1)
             fmecaold{h,10}=S(1,1)
          end
       end
        if fmecaold{h,9}>Min2(2,1)
          if fmecaold{h,9}<Max2(2,1)
            fmecaold{h,10}=S(2,1)
          end
        end
        if fmecaold{h,9}>Min2(3,1)
          if fmecaold{h,9}<Max2(3,1)
            fmecaold{h,10}=S(3,1)
          end
          end
       fmecaold{h,12}=fmecaold{h,10}*fmecaold{h,11}
       save savefmeca.mat fmecaold
      break
             end
      else
        j=j+1
      end
    end
    i=i+1
   v=v+1
 end
```

Acknowledgements

This thesis is the result of a nine months project. Obviously, several persons and organizations have been most helpful in the process.

First, I would like to send my gratitude to my supervisors, Eng. Marco Macchi and Luca Fumagalli, both from Dipartimento di Ingegneria Gestionale of Politecnico di Milano. Your guidance and inspiration have made this period very interesting.

I also want to thank all that help me for fruitful discussions and collaboration regarding the technical aspects of CBM and FMECA, in particular Diego.

I am grateful to have been given the opportunity to share this study with the staff from the Università degli studi di Bergamo.

Finally, I would like to send my appreciations and thanks to my family and friends.

Paolo Berno

Milano ,31 Marzo 2011