A framework for a predictive maintenance tool articulated with a Manufacturing Execution System

Rui Miguel Lima Medon

Master's dissertation

Supervisor: Prof. Luís Guimarães

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia e Gestão Industrial

2018-07-02

Abstract

In the current industrial context, players attempt to gain competitive advantage by manufacturing more and better products at a lower cost. A possible direction is the investment in more appropriate maintenance strategies. With the advent of Industry 4.0 and the Internet of Things industry is becoming equipped with the necessary technology to enable the application of predictive maintenance. This strategy attempts to detect and isolate impending faults and predict the time-to-failure or probability of failure within a given time span of a given asset by monitoring its health through condition measurements that do not interrupt normal machine operation.

The deployment of a predictive maintenance tool in a real-world context requires some previous work. More specifically, it is necessary to develop a conceptual framework that defines the different modules comprising the tool and establishes its workflow. Additionally, investigation of the existing possibilities regarding the different aspects that form the field of predictive maintenance is necessary so that decision makers understand which options better suit their needs. This thesis arises from this background and it attempts to provide an in-depth review of the different aspects comprising predictive maintenance and to develop such a framework that allows the predictive maintenance tool to answer the needs of the industrial plant.

In this thesis, a framework for a predictive maintenance tool is described in an iterative manner. Initially, a high level overview of the whole system is provided. At this stage, a high degree of abstraction is employed to define the number of basic elements that constitute the system, their roles and the basic workflow. Additionally, a distinction between main and support modules is introduced. Then, at an intermediate level, the workflow is further detailed by introducing a separation of the global workflow into two phases, offline and online, as well as a distinction regarding the roles of condition monitoring and event data. Additionally, the inputs and outputs of each module and their basic tasks are scrutinized. Lastly, at a lower level, the techniques used at each step are detailed in order to unveil deeper connections between the different modules.

Following this methodology it was possible to conclude that the effective operation of a predictive maintenance tool is accomplished by the cooperation between a set of knowledge-producing modules and support modules. Additionally, it became clear that there is a wide array of possibilities regarding which techniques to use for each task, but only a few possess a set of properties that are adequate for practical application. Some sort of previous process-related knowledge is always required, be it in the form of data or domain knowledge. The type of information each technique leverages influences its interpretabilty and adaptability. The incorporation of different kinds of process-related information in the distinct tasks of the predictive maintenance tool allows a better representation of the real manufacturing setting, which results in a more accurate operation of the tool. Lastly, techniques like the Hotelling T^2 and Q multivariate control charts for condition monitoring and the similarity-based approach for prognostics provided interesting results.

It is expected that this thesis promotes the development and application of a predictive maintenance tool in a real industrial context resulting in reduced maintenance costs and increased asset availability.

Resumo

No presente panorama industrial, os atores procuram ganhar vantagem competitiva através do fabrico de mais e melhores produtos a custos inferiores. Uma possível direção é o investimento em estratégias de manutenção mais apropriadas. Com o advento da Indústria 4.0 e da Internet das Coisas a indústria começa a ficar munida da tecnologia necessária para a aplicação de manutenção preditiva. Esta estratégia procura detetar e isolar falhas iminentes e prever o tempo-até-avaria ou a probabilidade de ocorrência de uma avaria num certo horizonte temporal monitorizando a condição física de equipamentos através de indicadores, sem interromper a sua operação normal.

A aplicação de uma ferramenta de manutenção preditiva em contexto real requer estudo prévio. Mais especificamente, é necessário desenvolver uma estrutura conceptual que defina as diferentes partes que constituem a ferramenta e estabeleça o fluxo da sua atividade. Adicionalmente, investigação acerca das possibilidades existentes para cada aspeto do campo da manutenção preditiva é necessária para que os responsáveis pelas tomadas de decisão conheçam as opções que melhor se adequam às suas necessidades. Esta dissertação surge deste contexto e procura fornecer uma revisão detalhada dos diferentes apetos que constituem o tema da manutenção preditiva e desenvolver uma tal estrutura que permita que a ferramenta responda às necessidades da fábrica.

Assim, a estrutura para uma ferramenta de manutenção preditiva é descrita iterativamente. Inicialmente, uma visão global do sistema é apresentada, empregando um nível elevado de abstração para definir o número de elementos básicos que o constituem, o seu papel e o fluxo básico do seu funcionamento. Adicionalmente, é introduzida uma distinção entre módulos principais e de suporte. Num nível intermédio, o fluxo da atividade é detalhado através da sua divisão em duas fases, *offline* e *online*, bem como através de uma distinção relativa aos papéis da *condition monitoring data* e da *event data*. Adicionalmente, os *inputs* e *outputs* de cada módulo e as suas tarefas básicas são escrutinados. Finalmente, num nível mais baixo, as técnicas usadas em cada etapa são detalhadas de modo a desvendar conexões mais profundas entre os diferentes módulos.

Seguindo este método foi possível concluir que o bom funcionamento de uma ferramenta de manutenção preditiva é conseguido através da cooperação entre módulos informativos e de suporte. Adicionalmente, tornou-se evidente que existe um vasto leque de técnicas utilizáveis em cada tarefa, mas apenas algumas possuem propriedades adequadas para aplicação prática. Algum tipo de conhecimento prévio sobre o processo é sempre necessário, seja sob a forma de dados ou de conhecimento especializado. O tipo de informação que cada técnica alavanca influencia a sua interpretabilidade e adaptabilidade. A incorporação de diferentes tipos de informação relativa ao processo nas diferentes tarefas da ferramenta de manutenção preditiva permite uma representação mais verosímil do contexto real de produção, resultando num funcionamento mais correto da ferramenta. Por fim, técnicas como as cartas de controlo *Hotelling* T^2 e Q para monitorização e a abordagem por semelhança para prognóstico proporcionaram resultados interessantes.

Espera-se que esta dissertação promova o desenvolvimento de uma ferramenta de manutenção preditiva em contexto real, resultando em custos de manutenção reduzidos e disponibilidade de equipamento aumentada.

Acknowledgements

This thesis marks the conclusion of five years of interesting challenges, rich experiences and incredible growth. As no man is an island, such challenges were only overcome, such experiences were only lived, such growth was only possible thanks to the ones that accompanied me throughout. As such, they deserve some words of appreciation.

To the ones that observed the development of the present thesis more closely: my mentor, Nuno Fernandes, for his restless support and never-ending patience; my supervisor, Professor Luís Guimarães, for his guidance and feedback; my colleagues at Critical Manufacturing, Ricardo Magalhães, João Cortez, Rui Alves and Ricardo Rios, for the valuable inputs and trust. It was truly a blessing to have had the opportunity to work with a group of such talented individuals.

To the ones that were present in the backstage of my academic path: Professor Vera Miguéis, for her ceaseless dedication and overflowing willingness to help, regardless the circumstances; Professor, and friend, Luís Malheiro, for his continuous concern with my academic development and making me feel proud in roaming the halls of our prestiged faculty.

To the ones who were with me in the trenches: my friends, the old and the new, for the laughs, for the learning, for the inspiration, for always being there, for all that which is impossible to put in writing.

To the ones who made it all possible: my parents, for all they have done that shaped me into who I am today, for always being on the lookout for what is best for me, for teaching me all the valuable things not taught in academia, for their patience.

Lastly, to the duo I cherish the most: my grandmother and great-aunt, for their overwhelming love and care, even from afar. If they could attend to the needs of industrial equipment the same way they attended mine for 22 years, there would be no need for maintenance at all.

"The key question to keep asking is,
Are you spending your time on the right things?
Because time is all you have."

Randy Pausch, *The Last Lecture*

Contents

1 Introduction **1**
- 1.1 Context & motivation . . . 2
- 1.2 Approach & goals . . . 3
- 1.3 Dissertation outline . . . 3

2 Literature review **5**
- 2.1 Traditional maintenance strategies . . . 5
- 2.2 Predictive maintenance . . . 6
- 2.3 Data acquisition . . . 6
  - 2.3.1 Data acquisition methods . . . 6
  - 2.3.2 Data types . . . 7
- 2.4 Data preprocessing . . . 8
- 2.5 Condition monitoring and fault detection . . . 11
- 2.6 Diagnostics . . . 12
- 2.7 Prognostics . . . 14
  - 2.7.1 Prognostics approaches according to the source of information . . . 15
  - 2.7.2 Prognostics approaches according to the strategy of estimation of time-to-failure . . . 15
  - 2.7.3 Prognostics approaches according to the degree of the system's heterogeneity 16
- 2.8 Research gaps . . . 19

3 Problem outline **21**
- 3.1 Background . . . 21
- 3.2 General concepts . . . 22
- 3.3 Requirements . . . 24
- 3.4 Solution and testing . . . 25
- 3.5 Limitations . . . 26

4 Predictive maintenance tool design **27**
- 4.1 High level overview . . . 27
  - 4.1.1 Basic elements & workflow . . . 27
  - 4.1.2 Main & supporting modules . . . 28
- 4.2 Intermediate level . . . 28
  - 4.2.1 Offline & online phases . . . 28
  - 4.2.2 Condition monitoring & event data . . . 29
  - 4.2.3 Inputs & outputs . . . 30
- 4.3 Low level . . . 33
  - 4.3.1 Condition monitoring . . . 33
  - 4.3.2 Diagnostics . . . 36

4.3.3 Prognostics . . . 37

5 Computational experiments **43**
5.1 Condition monitoring . . . 43
5.2 Diagnostics . . . 44
5.3 Prognostics . . . 44
5.3.1 Similarity-based approach . . . 45
5.3.2 Bayesian update approach . . . 47

6 Conclusion **49**
6.1 Key takeaways . . . 49
6.2 Future research . . . 50

A Full framework **57**

B Turbofan Engine Degradation Simulation Data Set **59**

C Reliability Centered Maintenance approach **61**

Acronyms and Symbols

PdM	Predictive Maintenance
IoT	Internet of Things
MES	Manufacturing Execution System
ODS	Operational Data Store
DWH	Data Warehouse
PCA	Principal Components Analysis
SPC	Statistical Process Control
MSPC	Multivariate Statistical Process Control
IoMT	Internet of Manufacturing Things
CEP	Complex Event Processing
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
RAE	Relative Absolute Error
RSE	Relative Squared Error
RCM	Reliability Centered Maintenance
LP	Late Predictions
PDF	Probability Density Functions

List of Figures

1.1 Architecture of Critical Manufacturing MES 2

2.1 Two-dimensional representation of the Principal Components Analysis technique 10
2.2 Illustration of prognostics and time-to-failure estimations 14

3.1 Real degradation curve of a manufacturing unit that has run-to-failure 22
3.2 Degradation curve of a manufacturing unit that has not experienced failure 23
3.3 Underused useful life as a result of preventive maintenance 23
3.4 Degradation curve of a manufacturing unit updated by sensor readings 24
3.5 Trade-off between accuracy/precision and slack for accommodating maintenance reaction time 25

4.1 High level overview of the predictive maintenance tool framework 28
4.2 Offline and online phases of the predictive maintenance tool 29
4.3 Sensor readings of a condition monitoring parameter with multiple operation modes 30
4.4 Intermediate level of the architecture of the predictive maintenance tool with operating context 30
4.5 Intermediate level view of the data acquisition module 31
4.6 Intermediate level view of the condition monitoring module 32
4.7 Intermediate level view of the diagnostics module 32
4.8 Intermediate level view of the prognostics module 33
4.9 Scatterplot of sensor 2 from training unit 1 34
4.10 Example of a moving window with step size of 1 and window width of 10 35
4.11 Summary of the results of the Principal Components Analysis on the training set 35
4.12 Hotelling T^2 control chart for test unit 1 36
4.13 Q control chart for test unit 1 37
4.14 Scatterplot of the Health Index associated with training unit 1 39
4.15 Health index regressed onto a curve with the form of an exponential (green), a 2^{nd} degree polynomial (blue) and a 3^{rd} degree polynomial (red) 39
4.16 Similarity assessment and time-to-failure calculation 40

5.1 Probability density functions of the same unit at different life stages 48

A.1 Full framework of the proposed Predictive Maintenance tool 57

B.1 Subset of Turbofan Engine Degradation Simulation Data Set 59

C.1 Histogram of the failure times of the training set units 61

List of Tables

5.1 Results of the computational experiments 45
5.2 Comparison of the average and median observed life of the test units associated with on-time and late predictions of the similarity-based approach 46
5.3 Mean absolute error and number of late predictions of a sample of 30 units at four different life stages 46
5.4 Comparison of the average and median observed life of the test units associated with on-time and late predictions of the Bayesian update approach 47

Chapter 1

Introduction

Operational safety, maintenance cost effectiveness and asset availability have a direct impact on the competitiveness of organizations and nations. Today's complex and advanced machines demand highly sophisticated but costly maintenance strategies. Domestic plants in the United States spent more than $600 billion to maintain their critical plant systems in 1981 and this figure doubled within 20 years (Heng et al., 2009). An even more alarming fact is that one-third to one-half of this expenditure is wasted through ineffective maintenance. This trend is similar in many other countries (Heng et al., 2009). Therefore, there is a pressing need to continuously develop and improve current maintenance programs.

This setting was the driving force behind research on the subject of predictive maintenance (PdM). According to Bloch and Geitner (1983), 99% of mechanical failures are preceded by noticeable indicators. Based on this principle, this maintenance strategy attempts to detect and isolate impending faults and predict the time-to-failure or probability of failure within a given time span of a given asset by monitoring its health through condition measurements that do not interrupt normal machine operation. Ideally, what PdM enables is the performance of maintenance actions at the latest possible minute before a failure occurring, thus simultaneously preventing equipment breakdown and seizing the full potential of the equipment's useful life. This requires the availability of insightful data that allows the performance of the aforementioned tasks.

However, industries just recently became capable of collecting and processing the volume and diversity of data required by such task. The advances in the fields of sensor technology, computer science, communication technology, big data and artificial intelligence bolstered by the revolutions of Industry 4.0 and Internet of Things (IoT) played a major part in this (Yan et al., 2017). Hence, the current setting is one where most industries can leverage the existing data for practical applications like PdM, thanks to information systems, like Manufacturing Execution Systems (MES), which are able to gather the data in the same place and process it to extract valuable process-related information, but few do.

Therefore, this dissertation aims to develop a framework for a PdM tool that can leverage not only health-related data but also other types of data regarding the asset that may exist in company-owned information systems like a MES.

1.1 Context & motivation

The multiple processes that comprise the manufacturing industry's value chain render the task of understanding the current condition of the plant floor extremely cumbersome. Hence, companies tend to rely on computerized systems that assist in this task like MES. Hence, companies have access to large collections of data concerning different aspects of the manufacturing process readily available for use in a number of applications. Thus, it is possible to form a vast body of knowledge concerning different aspects of the same process, which assists in understanding its behavior.

Most companies offering MES solutions possess a data storage architecture comprised of an Operational Data Store (ODS) and a Data Warehouse (DWH). The former stores historic records of collected raw data, whereas the latter aggregates and summarizes it. The user submits HTTP requests that are processed by the host, who queries both data repositories directly. Critical Manufacturing MES, on the other hand, has an additional Online Database preceding those two. Queries are submitted to the Online Database, who continuously replicates the data it contains to the ODS and, subsequently, disposes of it. Therefore, this structure only keeps records of recently collected data. Figure 1.1 depicts this architecture.

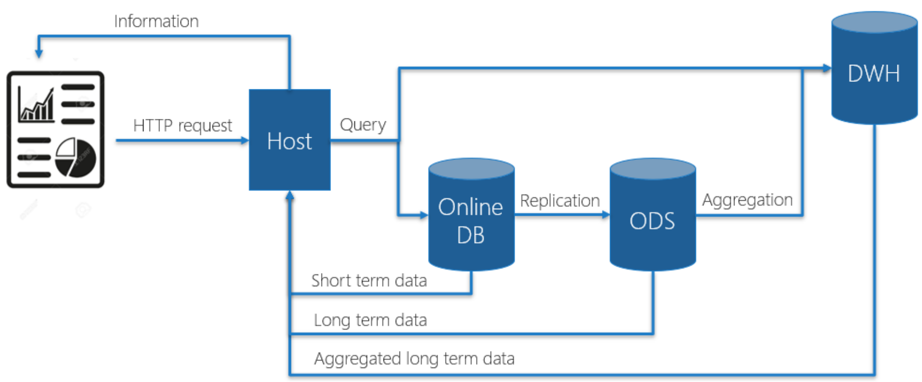

Figure 1.1: Architecture of Critical Manufacturing MES

As a result, different levels of detail can be provided, as depicted in the feedback loops of Figure 1.1. This property is very convenient for predictive tasks because the future can be anticipated in a more accurate way by simultaneously considering both macro and microtrends that the data may enclose, which can only be detected by analyzing the data at different dimensions.

Considering the aforementioned context, the access to different types of process-related data combined with the distinct levels of detail with which that data can be analyzed place the distinct architecture of Critical Manufacturing MES in a favorable position to venture in the development of a PdM tool and offer its clients an alternative to the traditional maintenance strategies that better fits their needs. As an introductory step towards this goal it is important to devise the global architecture of such a tool.

1.2 Approach & goals

In order to develop a framework for a PdM tool, an extensive literature review was conducted with the objective of understanding the current state of the art in terms of possibilities. Considering what was learned through such a review, a conceptual architecture was developed through an iterative approach starting at a higher level which was progressively detailed. The coherence of the conceptual architecture was then proved by instantiation.

It is expected that this dissertation will provide an updated review of the current state-of-the-art, giving special emphasis on its utility from an end-user perspective. Additionally, it is expected that it provides a coherent conceptual architecture that can leverage the data existing in MES and adapt to different circumstances imposed by the multiple processes occurring in the plant floor.

1.3 Dissertation outline

The remainder of this dissertation is structured in the following way. Chapter 2 provides a review of the current state of the art regarding different aspects of PdM. Additionally, it enumerates some of the major existing research gaps. Chapter 3 describes the problem at hand. For that purpose it sheds some light on the industrial background in which the monitored assets operate, introduces a number of concepts related with the problem of PdM, lists the tool requirements, describes the testing process and enumerates the existing limitations. Chapter 4 describes the proposed architecture with increasing levels of detail. Initially, a high level overview of the whole system will be provided. Then, an intermediate view will define the major existing modules and describe the general workflow. Chapter 5 provides an even lower level view of the framework, further detailing the functioning of each of those stages and instantiating the full framework. Lastly, Chapter 6 draws some final remarks on the developed work and suggests a number of directions for future efforts towards the development of a PdM tool.

Chapter 2

Literature review

PdM has been studied extensively in the past decades and some approaches to it have been reviewed by several authors (Lei et al. (2018), Javed et al. (2017), Elattar et al. (2016), Vogl et al. (2016), Jardine et al. (2006)). However, most reviews focus on specific stages of PdM, namely diagnostics and prognostics. Additionally, the categorization of different techniques applied to the different stages that comprise PdM is usually done according to technical aspects, such as being data-driven or physics-based, statistical or machine learning. From an end-user perspective, the added value in this segmentation is very little. However, very few reviews have analyzed PdM approaches from an end-to-end perspective. It is substantially more important to understand, for example, if the system is able to leverage available information regarding the asset, the working environment or the workloads, what inputs does each major stage require, what are the possibilities in terms of the outputs that can be presented to the end-user and what can be made of them and what are the limitations of the available techniques.

The following literature review will attempt to fill these gaps. Hence, it will begin by presenting the position of PdM relative to other traditional maintenance strategies and shedding some light on its basic components. Subsequently, each of those components will be thoroughly reviewed bearing in mind the aforementioned aspects. Lastly, it will briefly reflect on a number of open issues in the field of PdM.

2.1 Traditional maintenance strategies

According to British Standards Institute Staff and British Standards Institution (2001), there are two main types of maintenance strategies: reactive and preventive.

Reactive maintenance is based on the principle of allowing the equipment to operate without the intervention of any maintenance action until it reaches functional failure and consequently needs a repair or replacement. The high cost of catastrophic failures and emergency shutdowns led to the introduction of preventive maintenance.

Preventive maintenance, contrarily to reactive maintenance, does not allow assets to run to failure by submitting them to maintenance actions before they attain a state of degradation that no longer allows them to perform as intended. Preventive maintenance can be triggered by two different criteria which are the operating time or the equipment's condition.

Time-based preventive maintenance submits equipment to maintenance actions periodically. Although it reduces the total number of equipment failures, preventive maintenance is not optimal in the sense that it lends no consideration whatsoever to the equipment's condition, thus performing unnecessary maintenance actions.

Condition-based maintenance, on the other hand, overcomes this weakness by monitoring the equipment's health based on condition measurements that do not interrupt its normal operation (Heng et al., 2009).

2.2 Predictive maintenance

If the condition measurements are used to predict when the asset will fail, condition-based maintenance becomes predictive maintenance (Mobley, 2002). According to Ellis (2008), PdM should be based on asset criticality, which is given by its safety, environmental and operational impact. Hence, the targets of PdM have to be defined by prior analysis, like Failure Mode and Effects Analysis or Fault trees (Javed et al., 2017).

This maintenance strategy involves several stages. It is generally accepted that PdM systems are comprised of six major stages: Data Acquisition, Data Preprocessing, Condition Monitoring, Diagnostics, Prognostics and Decision Support (Elattar et al., 2016). The remaining sections of this chapter will address these stages. However, considering the novelty of PdM and the exploratory character of the present work, Decision Support is deemed to fall outside of its scope, hence it will not be subject to review.

2.3 Data acquisition

According to Jardine et al. (2006), data acquisition is the process of collecting and storing useful data from targeted physical assets for the purpose of PdM. Ćwikła (2013) suggests that the methods used in data acquisition are heavily influenced by the degree of automation of the production system. Thus, the following types of data acquisition methods can be found: manual, semi-automatic and automatic. Additionally, the collected data can differ on its nature and can be categorized into two main types: condition monitoring data and event data.

2.3.1 Data acquisition methods

Ćwikła (2013) provides a good description of the three types of data acquisition. Manual acquisition is based on direct communication between employees at different levels of the hierarchy of management. Due to the multitude of drawbacks associated with it, namely being inefficient in the context of modern industrial systems, error and delay prone and productivity reducing, it is mainly used when automation is insufficiently profitable. Automatic acquisition, on the other hand, relies on control systems like sensors, industrial controllers, CNC machines, robots, smart actuators, and SCADA systems to acquire data. Semi-automatic acquisition, or assisted manual acquisition, emerges in-between the aforementioned methods in the automation spectrum. If worker intervention is required, hardware and software solutions, like bar-codes, magnetic track, radio frequency (RFID) and machine vision, reduce the error rate and increase data acquisition speed. Manual input assisted by computerized information systems like MES also fall into this category. Most PdM systems rely on semi-automatic or automatic data acquisition.

2.3.2 Data types

Condition monitoring data is a set of performance assessment measurements that can be correlated to the health state of the asset from which it is collected (Jardine et al., 2006).

Condition monitoring acquisition methods usually fall into the automatic category, being mainly collected through classic transducers or sensor suites specific to the application domain (Vachtsevanos et al., 2006), implying a large variety of them, like accelerometers for vibration measurements, strain gauges, ultrasonic sensors, eddy-current proximity probes, temperature sensors, microelectromechanical system sensors and fiber-optic sensors.

Jardine et al. (2006) introduces a classification of condition monitoring data in three categories: value type, waveform type and multidimensional type. According to the authors, if the data collected at a specific time epoch for a condition monitoring variable is a single value, then it is value type. Temperature, humidity, pressure, speed and flow data, for example, fall into this category. On the other hand, if the data collected at a specific time epoch for a condition monitoring variable is a time series, like vibration data and acoustic data, then it is waveform type. Lastly, if data collected at a specific time epoch for a condition monitoring variable is multidimensional, then it is multidimension type. Infrared thermographs, x-ray images and visual images, for example, fall into this category.

Event data, on the other hand, informs about "what happened and/or what was done to the targeted physical asset" (Jardine et al., 2006). This type of data is usually collected manually.

Wang et al. (2017) introduce a classification of event data in six categories: maintenance records, system logs/messages, inventory data, utilization data, environmental data and configuration information. Maintenance records may include the date and time of creation, the identification of the support teams and the spare parts involved in the maintenance operation, the reported symptom, the resolution and call-backs information, if the resolution was not successful. System logs, on the other hand, may be provided in an unstructured or structured manner, e.g. XML format, and include information like the equipment ID, timestamp, message content, code and priority of the addressed issue. Inventory data provides information regarding the manufacturer, the machine type, its serial number, location and installation date. Wang et al. (2017) argue that these three types of event data are the most useful in terms of equipment failure prediction.

2.4 Data preprocessing

Raw data is almost never immediately fit for effective knowledge extraction. Indeed, most of the times it contains a number of missing values, outliers, errors and redundant variables, it is contaminated by noise, it is scattered among a number of databases, all of which compromise the outcomes of the knowledge extraction tasks (Jardine et al., 2006). For this reason, raw data has to be processed before it can be adequately used. The preprocessing routines to which the raw data is submitted depend on whether they are handling condition monitoring data or event data. However, in most cases where event data is available via information systems like a MES, it is already fit for use thanks to mechanisms of those systems that impede the adulteration of the data. Hence, the focus of this subsection will be on preprocessing routines of condition monitoring data.

According to Venkatasubramanian et al. (2003), the preprocessing routines to which condition monitoring data is submitted are tailored to the specific needs of the subsequent knowledge extraction tasks. Indeed, if the subsequent tasks incorporate prior process knowledge, feature extraction techniques using prior knowledge of the problem are usually required (Georgoulas et al. (2013), Zhang et al. (2013), Pecht and Gu (2009), Yan et al. (2008)). On the other hand, in situations where prior knowledge is not available, an agnostic, more generic treatment of the condition data can be applied (Wang (2016), Yongxiang et al. (2016), Gebraeel et al. (2005)). Given this distinction, both types of preprocessing techniques will be reviewed.

Preprocessing techniques incorporating prior knowledge Given that bearings are one of the most common and widely studied components in modern rotating machinery and that a number of different types of data can be collected from them, namely vibration, temperature, chemical, acoustic emission and sound pressure data (Yan et al., 2008), they will be used as an example of the possibilities regarding preprocessing techniques incorporating prior knowledge. Considering that each aforementioned data type has its own specific set of preprocessing techniques, a review of every technique would be very time-consuming and would provide little to no extra value. Given the considerable amount of techniques studied for vibration data, this review will focus on those techniques.

According to Caesarendra and Tjahjowidodo (2017), there are three main types of analysis that can be conducted, namely time-domain analysis, frequency-domain analysis or time-frequency-domain analysis. Time-domain analysis is usually conducted as a preliminary step for fault detection, because it allows the unveiling of time-invariant features of the vibration signal in which incipient faults show signs of their presence (Yan et al., 2008). Thus, time-domain feature extraction techniques encompass the calculation of descriptive statistics such as the root mean square, the peaks, the peak-to-peak intervals, the crest factor, the mean, the variance, the skewness and the kurtosis (Abu-Mahfouz, 2003), the construction of models representing the vibration signal (Kang et al., 2012) or signal processing techniques like time-synchronous averaging (Dalpiaz et al., 2000). Frequency-domain feature extraction techniques, on the other hand, are directed towards fault localization and encompass spectral analysis, envelope analysis, cepstrum and higher-order spectra (Caesarendra and Tjahjowidodo, 2017). Lastly, time-frequency-domain feature extraction techniques are used in situations where the signals change over time. Some of the most widely used techniques are Short-time Fourier Transform (Dekys et al. (2017), Lee (2015), Yu et al. (2014)) and Wavelet Transforms (Bajric et al. (2016), Kankar et al. (2011), Dalpiaz et al. (2000)).

Generic preprocessing techniques An extensive literature review revealed that there is a number of preprocessing steps which are commonly found in PdM systems. Those steps are regime partitioning, outlier detection and handling, normalization and dimensionality reduction.

Regime partitioning is responsible for distinguishing different operating regimes as well as telling healthy operation apart from faulty operation. This can be done by visually inspecting the collected data (Yongxiang et al., 2016) or by building a model of the monitored system (Deshpande and Patwardhan, 2008).

In a real-world context, data may contain outliers, which can significantly compromise the outcome of the analysis (Ekwaro-Osire et al., 2017). Hence, outlier detection and subsequent handling is vital for the effective functioning of a PdM tool. One of the simplest and most widely used outlier detection techniques is the median method (Basu and Meckesheimer, 2007). This technique computes the median of a neighborhood of points defined by a window with a predefined size and classifies any point that falls outside of a given confidence interval centered at the median as an outlier. The range of that confidence interval is usually set as the median absolute deviation multiplied by a given constant (Leys et al., 2013). This statistic does not assume any distribution of the data and it is rather robust to the presence of outliers, therefore it is preferable when compared to the standard deviation around the mean. Once outliers are identified, they can be eliminated, replaced by the mean or median or by the most adequate estimate.

Regarding normalization, there are three main procedures: min-max scaling, unit length scaling and standardization. Since a number of fault detection, diagnostic and prognostic approaches require the data to be scaled and centered around the mean, the latter procedure will be reviewed more thoroughly. This procedure transforms the variables into new ones by subtracting the sample mean and then dividing the obtained value by the sample standard deviation. The new variables are, thus, given by

$$x' = \frac{x - \bar{x}}{\sigma} \tag{2.1}$$

where x is the original feature vector, $\bar{x}$ is the mean of that feature vector and σ is its standard deviation. This procedure is based on the assumption that the data is normally distributed.

According to Cheng et al. (2010) PdM systems requires the monitoring of a large number of equipment condition parameters. Depending on the complexity of the equipment, this number may reach thousands of parameters. Given such a large number of variables, the complexity and computational expensiveness of the problem increase dramatically. Additionally, some of the variables are bound to be correlated and, consequently, provide little added value. Thus, dimensionality reduction techniques that are able to simultaneously preserve most of the data variability and handle the existing correlations between variables are used.

One of the most widely used techniques in PdM is Principal Components Analysis (PCA) (Harrou et al. (2015), Marton et al. (2013), Ahmed et al. (2012), Villegas et al. (2010)). This technique is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, that is, it accounts for as much of the variability in the data as possible, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. According to Miranda et al. (2008), this method requires the data to be previously centered around the mean and scaled so as to ensure the units in which the variables are measured in do not interfere with the variance.

Figure 2.1 depicts this transformation for a two-dimensional problem. Given a set of points as the ones depicted in the scatterplot, the covariance matrix of that dataset can be decomposed in a set of eigenvectors, which are orthogonal, and eigenvalues. Geometrically, in a two-dimensional setting, the eigenvectors determine the axis of an ellipse that surrounds the data and captures its variance and the eigenvalues determine the scale of each of these eigenvectors, that is, the scale of the variance of the data along that direction. By sorting the eigenvectors according to their respective eigenvalues, the directions of major variance are obtained. Those directions are the principal components. The transformation occurs by projecting the original datapoints onto the principal components. Given that each principal component accounts for a given proportion of the data's variance, a selection rule can then be devised so as to maintain a number of principal components that explains more than a predefined cumulative variance, thus effectively achieving dimensionality reduction.

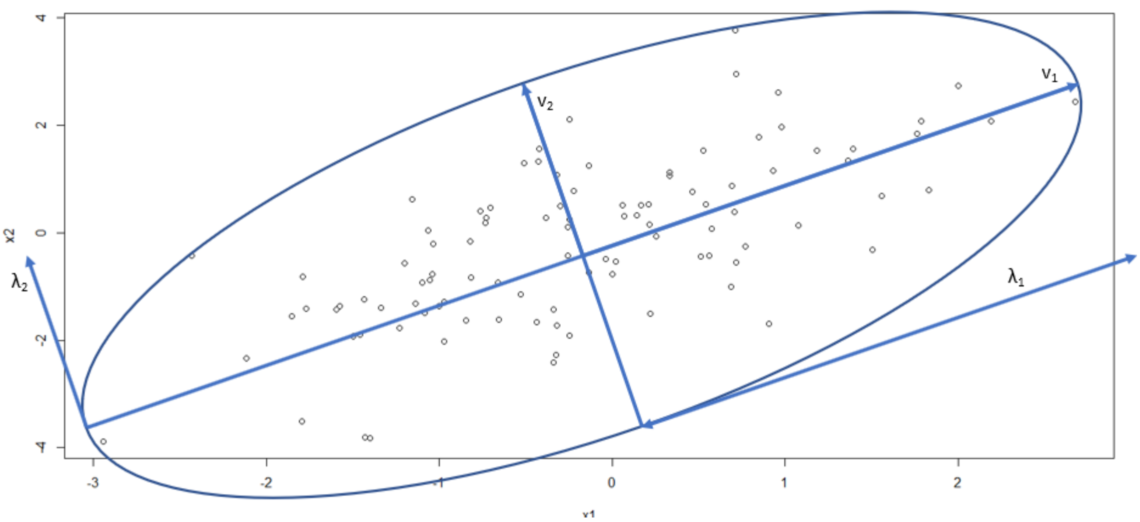

Figure 2.1: Two-dimensional representation of the Principal Components Analysis technique

2.5 Condition monitoring and fault detection

There is a number of options regarding condition monitoring and fault detection techniques spanning from prior-knowledge-based models of the controlled system, which are used when there is an availability of expert domain knowledge regarding the functioning of the process, to artificial-intelligence-based models, which are used when that knowledge is difficult to obtain (Das et al., 2012). Despite the diversity of approaches, statistical process control (SPC) has enjoyed widespread appraisal in industry for a number of years (Dhini, 2016). In fact, due to the necessity of monitoring a large number of process control variables, the original SPC approaches based on the univariate analysis of each variable gave way to multivariate statistical process control (MSPC) in order to overcome the limitations of their predecessors. Indeed, univariate analysis of a large number of variables is particularly cumbersome and fails to acknowledge the existence of correlations among the data, which oftentimes leads to erroneous conclusions (Lowry et al., 1992). MSPC, on the other hand, has enjoyed very successful applications in the industry (Westerhuis et al., 2000). The concept underlying it is the comparison of current process measurements against a model built under normal operating conditions. Control charts with empirically calculated limits monitor the current process measurements, which are signaled as abnormalities if they are located outside the chart's range. Harrou et al. (2015) states that statistical projection methods such as PCA make good candidates for MSPC because they require no prior knowledge about the process model and they can effectively deal with highly correlated process variables. Additionally, they have the ability to effectively reduce the number of process variables into a smaller set of latent variables which, combined with the visual character of control charts, enable a more immediate grasp of the process control state and its relevant process variables. PCA-based fault detection indicates the presence of a fault by the Hotelling T^2 monitoring statistic crossing its given threshold (Dhini, 2016). The T^2 statistic is an extension of the t-test for the multivariate case and it is given in Hotelling (1933) by

$$T^2 = X^T \widehat{P} \widehat{\Lambda}^{-1} \widehat{P}^T X = \sum_{i=1}^{l} \frac{t_i^2}{\lambda_i} \tag{2.2}$$

where X is the standardized data matrix, $\widehat{P}$ is a matrix containing the loading vectors associated with the l columns of $\widehat{\Lambda}$, $\widehat{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_l)$ is a diagonal matrix containing the eigenvalues associated with the l retained PCs and t_i is the i^{th} principal component. It follows the F distribution, thus, its threshold is given in Hotelling (1933) by

$$Threshold = \frac{p(n-1)}{n-p} F_{\alpha(p,n-p)} \tag{2.3}$$

where p is the number of variables, n is the sample size and α is the significance level. If all the parameters of the underlying population are known, the χ^2 distribution can be used instead.

While the Hotelling T^2 statistic measures the amount of variation captured by the PCA model, the Q statistic measures the amount of variation that the model is not able to capture and, thus, indicates if the model is a good representation of the process (Mansouri et al., 2015). Control charts based on the Q statistic are preferred over the T^2 charts because they are "more sensitive to faults with smaller magnitudes" (Harrou et al., 2015). This monitoring statistic is given by

$$Q = \left\| (I - \widehat{P}\widehat{P}^T)X \right\|^2 \tag{2.4}$$

where I is the identity matrix, $\widehat{P} \in \mathbb{R}^{m \times l}$ is a matrix containing the l retained eigenvectors and X is the standardized data matrix. The presence of a fault is indicated by the Q statistic crossing the threshold, Q_α, given in Jackson and Mudholkar (1979) by

$$Q_\alpha = \varphi_1 \left[\frac{h_0 c_\alpha \sqrt{2\varphi_2}}{\varphi_1} + 1 + \frac{\varphi_2 h_0 (h_0 - 1)}{\varphi_1^2} \right] \tag{2.5}$$

$$h_0 = 1 - \frac{2\varphi_1 \varphi_3}{3\varphi_2^2} \tag{2.6}$$

$$\varphi_i = \sum_{j=l+1}^{m} \lambda_j^i, i = 1,2,3 \tag{2.7}$$

where l is the number of retained PCs, m is the number of process variables and c_α is the value of the normal distribution with α levels of significance. This threshold value assumes that the observations are time independent and multivariate normally distributed (Harrou et al., 2015), which makes the Q statistic very susceptible to modelling errors.

2.6 Diagnostics

British Standards Institute Staff and British Standards Institution (2001) define fault diagnosis as the "actions taken for fault recognition, fault localization and identification of causes". Indeed, these three tasks provide information regarding which degradation mechanism is underway in a given asset, what is the physical location of the problem and what abnormality caused it in the first place, which better prepares the end-user in terms of how to deal with the impending fault.

Fault recognition According to Venkatasubramanian et al. (2003), there are a number of fault recognition techniques that can be distinguished based on the used resources. One of the possible approaches is based on the development of models representing the process operation. Those models incorporate existent knowledge about the physics of the process and, more specifically, about the failure mechanisms that it might observe. In situations where this knowledge is not available, but there are historic records of past machine failure, it is possible to extract knowledge that aids in the characterization of future faults. Despite the existence of different options, both approaches require domain knowledge to some extent. Model-based approaches require in-depth knowledge regarding the physical behavior of the process in order to build a mathematical description of the asset's operation (Venkatasubramanian et al., 2003), whilst data-based techniques require expert labelling of the historic records into their respective fault classes (Kerkhof et al., 2013).

Fault localization One of the pitfalls in fault recognition techniques is that no information is obtained about the cause of the disturbance (Westerhuis et al., 2000). Oppositely to fault recognition, fault localization does not require any particular knowledge regarding the process. Alternatively, techniques like control statistic decomposition can be used. By decomposing control statistics into a sum of terms representing the contributions of each process variable to the out of control signal, it is possible to learn which variables significantly contribute to the occurrence of the abnormality (Westerhuis et al., 2000). This technique assumes that variables associated with the fault exhibit large contributions. The decomposition of the Hotelling T^2 can be performed with the help of the Mason, Young and Tracy method (Mason et al., 1997) which is given by

$$T^2 = T_1^2 + T_{2\cdot 1}^2 + ... + T_{p\cdot 1,2,...,p-1}^2 \tag{2.8}$$

where the first term, T_1^2, is an unconditional Hotelling T^2 for the first variable of the observation vector X and it is given by

$$T_1^2 = \frac{(x_1 - \bar{x}_1)^2}{s_1^2} \tag{2.9}$$

and where the other terms, referred to as conditional terms, are given by

$$T_{j\cdot 1,2,...,j-1}^2 = \frac{(x_j - \bar{x}_{j\cdot 1,2,...,j-1})^2}{s_{j\cdot 1,2,...,j-1}^2}, j = 1, 2, ..., p \tag{2.10}$$

where $\bar{x}_j$ represents the sample mean of the n observations on the j^{th} variable and s_j represents their standard deviation. Equation (2.8) represents only one of the possible $p!$ different orderings of these components, which give the same overall T^2 value. Hence, it is possible to conclude that, excluding redundancies, there are $p \times 2^{(p-1)}$ possible terms that should be evaluated for potential contribution to a signal, which implies that the computational expensiveness increases dramatically with the number of variables.

According to Kerkhof et al. (2013), this is one of the most extensively used tools for fault localization due to its ease of implementation and absence of need of historic fault data. Additionally, it can be used in conjunction with PCA (Westerhuis et al., 2000). However, its effectiveness requires that only one variable is associated with the impending fault. If multiple variables are associated with the fault, its effectiveness is hindered by a phenomenon called smearing-out effect, which refers to the propagation of contributions of faulty variables to the contributions of variables that do not contribute to the fault (Westerhuis et al., 2000). In cases where this phenomenon occurs and when this technique is used in conjunction with PCA, a physical interpretation of the principal components associated with a fault is required in order to link their disturbances to the underlying process dynamics. If that interpretation is not available, univariate analysis of the original process variables associated with the fault is advisable (Kerkhof et al., 2013).

Identification of causes Very little research has been done regarding automatic identification of the causes of faults in industrial equipment. However, borrowing from studies on fault cause analysis of power distribution networks it is possible to conclude that historic fault data and domain knowledge are fundamental for such task (Xu et al. (2006), Thukaram et al. (2005), Chen-Fu Chien et al. (2002)). This is because root cause analysis can be viewed as a classification problem in which domain experts categorize fault data into previously studied fault cause classes (LANA Labs (2017), Xu et al. (2006)). However, manual root cause analysis techniques like Ishikawa diagrams, 5-Why's, cause mapping and fault tree analysis are the most common practice in industry (J. Duphily, 2014).

2.7 Prognostics

According to ISO-13381-1 (2015), prognostics is the "estimation of time-to-failure and risk for one or more existing and future failure modes". In order to perform this task, a prognostic module requires the indication of the underlying degradation process, which is borrowed from the condition monitoring and diagnostics module, of the future operating conditions and the workload (Javed et al., 2017). Based on these inputs, the current health state of the asset can be estimated and the degradation trajectory is projected into the future until it reaches the threshold for functional failure. At this moment, the asset is considered to have failed. Then, the time-to-failure of the asset is given by the difference between the failure time and the current time, as depicted in Figure 2.2, where tD represents the instant when a fault is detected, tc represents the current time and tf represents the estimated time of failure.

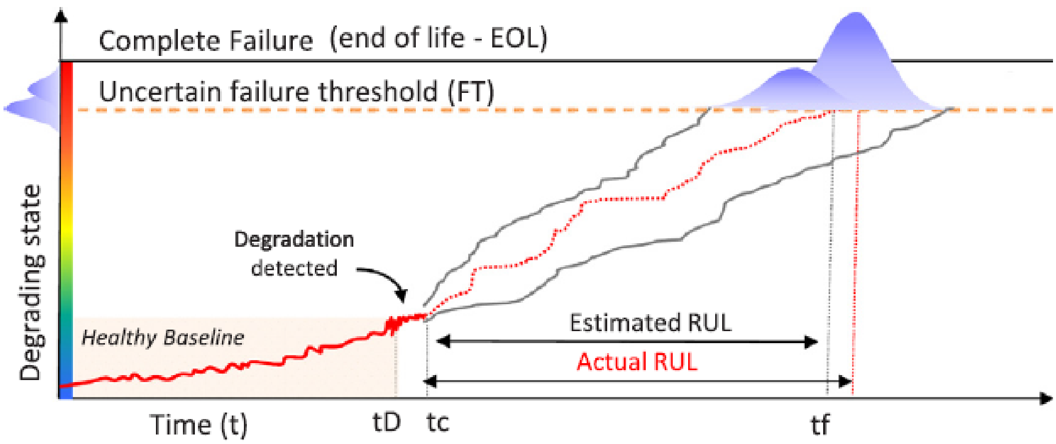

Figure 2.2: Illustration of prognostics and time-to-failure estimations (Javed et al., 2017)

Several attempts to categorize different prognostic approaches have been made over the years (Lei et al. (2018), Si et al. (2017), Javed et al. (2017), Elattar et al. (2016), An et al. (2015), Jardine et al. (2006), Vachtsevanos et al. (2006)). In order to comprehend the many aspects of existing prognostic approaches, three distinct classifications seem necessary. They categorize those approaches according to the source of information for the prognostic task, to the strategy for the estimation of time-to-failure and, lastly, to the degree of the system's heterogeneity. These classifications will be thoroughly detailed subsequently.

2.7.1 Prognostics approaches according to the source of information

Most published reviews (Lei et al. (2018), Elattar et al. (2016), An et al. (2015), Jardine et al. (2006), Vachtsevanos et al. (2006)) classify prognostic approaches into either physics-based, which rely on mathematical descriptions of the behavior of the process, or data-driven, which predict the behavior of a given asset based on the behavior of other similar assets. Some publications advance another category of prognostic approaches which encompasses hybrids between the previous two types of approaches. This classification is relevant to understand which applications are more appropriate considering the existent means.

Physics-based models The mathematical models of physics-based approaches require expert domain knowledge regarding the physical phenomena underlying the degradation process, its dynamics and principal factors influencing it (An et al., 2015). These requirements make this type of approach very accurate, but difficult to scale to larger and more complex systems, which is why it has enjoyed more success in component-specific applications (Elattar et al., 2016). However, oftentimes such expert knowledge is unavailable.

Data-driven approaches These techniques overcome the pitfalls of physics-based models by taking advantage of the progress in sensor, data storage and processing technologies and extracting prognosis-relevant data from historic degradation records of similar equipment (Lei et al., 2018). This enables system level applications (Javed et al., 2017), but large collections of degradation data are required, which carry large costs in data acquisition and storage. Additionally, allowing the assets to degrade is inherently undesirable.

Hybrid approaches Looking at how there is no one approach which does not possess any limitations, hybrids integrating different methods were built in an attempt to leverage their advantages and offset their limitations (Lei et al. (2018), Javed et al. (2017)). Not only that, they are also able to reduce the complexity of computation and improve prediction precision (Peng et al., 2010).

2.7.2 Prognostics approaches according to the strategy of estimation of time-to-failure

Given that industries are moving towards a digitalized culture with data at its core (PwC, 2016), it is with no surprise that rapid advances in data-driven prognostics approaches have been witnessed (Elattar et al., 2016). This particular type of approach can be viewed from a different perspective. There are distinct prognostics strategies to obtain an estimate of the time-to-failure. One of them relies on the estimation of the current degradation state and its extrapolation into the future until a predefined failure threshold is met. Oppositely, the time-to-failure can be directly estimated based on pattern matching between currently observed and historic degradation trajectories. Each strategy has its own strengths and weaknesses.

Degradation modelling This approach looks to estimate the current state of the equipment's health and evolve it into the future until a predetermined threshold is reached. This occurrence indicates that the equipment has failed. Hence, this method is not only able to provide an estimate of time-to-failure, but it can also provide an estimate of the asset's current and future health state. This is particularly useful for the end-user who can learn more about the degradation process of that specific asset. Moreover, the degradation model can be plotted in a chart whose visual character allows for a quick grasp of the asset's current and near future condition. Lastly, degradation modelling only requires partial degradation records instead of large collections of run-to-failure data due to its ability of projecting the degradation trajectory into the future, which is in line with the best interest of the industry to observe the minimal possible number of unexpected breakdowns. However, it does require the specification of a failure threshold, which is often very difficult (Javed et al., 2017).

Direct time-to-failure estimation This method is based on the principles of curve similarity and pattern matching. Given a set of online data points and an array of offline degradation curves, the similarity between the data points and the curves is measured by some specific metric. Then, the curve that maximizes the similarity metric is assumed to be representative of the future degradation of the online data points.

The techniques falling within this category do not require the fixing of a failure threshold, but they do need a set of smooth and monotonic features to ensure the effectiveness of the pattern matching task. Plus, the accurate estimation of time-to-failure depends on the existence of a large collection of run-to-failure data that covers all the possible variations of the degradation process.

2.7.3 Prognostics approaches according to the degree of the system's heterogeneity

According to Si et al. (2017), the degradation of a system is not only a function of its inner degradation, but also of the working environment and the workloads to which it is subjected. Hence, and considering the dynamic working conditions in which modern manufacturing systems are required to operate, it is important to address the issue of heterogeneity across those systems and how it can be incorporated into prognostics. Si et al. (2017) provide a satisfactory categorization of the existing prognostics approaches into three groups according to the sources of heterogeneity across manufacturing systems, namely the production unit, the working environment and the workloads. For this reason, the remainder of this section will be based on the work of these authors.

Unit heterogeneity The degradation processes of two distinct manufacturing units coming from the same population can be modelled by two different types of parameters. Si et al. (2017) provide a good explanation towards the motive. The fact they originated from the same population implies that there are certain characteristics of the degradation processes that remain unchanged between both units. Hence, these characteristics can be viewed as an universal effect across all units belonging to the same population and can be easily modelled by a set of constants called population parameters. However, the authors state that it is known, based on vast experimentation and engineering phenomena, that, simultaneously, the two units degrade differently, even though they originated from the same population. This is due to the existence of some degree of variability in the structures of each unit. Hence, in practice, the parameters associated with this variability are modelled as random variables governed by unit-specific distributions. What this distinction enables is for the end-user to learn more about the degradation process of the assets, more specifically which part of the degradation is due to population characteristics and which is due to individual abnormalities. Additionally, these methods provide the time-to-failure estimates in the form of distributions, from which the probability of failure within a given time span can be obtained.

Working environment heterogeneity The degradation of manufacturing systems is a complex phenomenon. For that reason, oftentimes assumptions are made regarding the working conditions of the system in order to simplify the models describing its degradation. In the literature, it is not uncommon to find studies assuming that the systems are operating in a steady state, that is, where the working conditions remain unchanged throughout the entire operation. However, if the dynamic conditions of the working environment can be modelled, the accuracy of the time-to-failure estimates will improve significantly. A number of techniques has been proposed to answer specific problems inherent to working environment heterogeneity.

Methods based on stochastic filtering assume that the degradation of a given process cannot be directly observed by condition monitoring data. Instead, this data directly influences a set of variables whose variation directly affects the degradation process. In other words, the condition monitoring data is correlated with the degradation state of the process, but does not cause it. This type of representation is known as a state-space model. Thus, these methods establish stochastic relationships between the lifetime of the process and the information regarding condition monitoring and operating environment. Furthermore, some methods build state-space models in which the state variables follow the changes in the operating environment and then update them with each new set of observed condition monitoring data. These methods have the advantage of being able to successfully provide estimates of the time-to-failure in situations where the available data is only symptomatic of the actual degradation. Also, the update of the estimates by leveraging the online measurements refines their accuracy. However, state-space models must be built, which may not always be feasible, and future operating conditions are disregarded.

The degradation of a manufacturing process is not always strictly monotonic, nor does it have always the same rate of progression. In some cases, it is able to distinguish several stages of degradation. Multi-stage degradation models can be developed for such situations. These models require that the change points are determined, which refer to the time epochs at which the degradation starts to develop and the different degradation stages begin and end. The starting point is given by the condition monitoring module, but the remaining points have to be determined. Additionally, the number of stages has to be preset and model parameters have to be estimated from large collections of data spanning all degradation stages. Lastly, disregarding future operating conditions may result in innacurate estimates.

According to Gorjian et al. (2009), the hazard rate of most assets is influenced by different risk factors. It is, therefore, desirable to isolate the effects of these factors in order to understand what is their contribution to the hazard rate (Kumar and Klefsjo, 1994). These variables, which are called covariates, can be constant or a function of time or even random variables, and may be related to the operating environment (Kumar and Klefsjo, 1994). The Proportional Hazards Model is a classical covariates-based method and assumes that the hazard rate is a function of a baseline hazard rate function, $h_0(t)$, and the covariate function, $\Psi(\beta z(t))$,

$$h(t;z(t)) = h_0(t)\Psi(\beta z(t)) \tag{2.11}$$

where $z(t)$ are the covariate variables and β are the regression coefficients which can be estimated using historical lifetime data or censored lifetime data of the system from the same category. This method offers a tremendous advantage over others due to its "strong explanatory property", as Si et al. (2017) put it. However, the estimation of the regression coefficients and the baseline hazard rate require lifetime data that sometimes is difficult to obtain and the form of the baseline hazard rate function becomes hard to determine for complex systems.

Lastly, some methods assume the degradation of a system is caused by the interaction and competition between the system's inner degradation and external random shocks, which are defined as external events that negatively impact the system's performance and degradation process. A number of models have been proposed in which the interdependency between random shocks and the degradation process is considered and in which it is not. This approach allows the end-user to learn more about how the production systems behave in the presence of external factors and how they impact its degradation process. However, given the random occurrence of external shocks, continuous inspection of the system is required in order to ensure they are identified. Additionally, some random shocks may impact the degradation process positively, which is usually ignored.

Workload heterogeneity Throughout the course of their lifetime, manufacturing systems are subject to multiple workloads due to the large number of tasks they perform. Effectively modelling such dynamics and tasks allows for better estimates of the time-to-failure. Two types of methods fall within this category, namely the ones that model the dynamic operating conditions and the ones that model maintenance actions. The former is helpful in terms of understanding how each operating condition impacts the degradation of the process. However, the relationships between the two can be quite difficult to obtain in complex systems. The latter overcomes the limitation of only considering events with negative impact on the degradation of the process. Yet, it is based on a number of assumptions regarding the duration, diversity and effect of the maintenance actions that severely limit its potential.

2.8 Research gaps

Reflecting on the content of the previous sections, it is possible to identify a number of gaps in PdM research. The most noticeable is the lack of industrial applications of PdM. Jardine et al. (2006) suggest that this is due to a lack of data that enables the successful monitoring of manufacturing equipment, poor communication between researchers and practitioners, a lack of efficient validation approaches and difficulties of implementation due to the ever-changing industrial environment. However, these are not the only reasons for the underdevelopment of practical applications of PdM. Much of it is related to other research gaps which are worth pointing out.

Firstly, very little effort was put into developing a standardized platform that fully integrates all the elements comprising PdM (Javed et al., 2017). Although some attempts have been pursued towards this goal (Lebold et al. (2002), ISO-13381-1 (2015), ISO-13374-1 (2003)), the vast majority of research specializes on specific aspects of the problem. According to Javed et al. (2017), industrial deployment would largely benefit from research in that direction, because that way industries could save costs by using already existing standard components to build their PdM systems, instead of developing such a platform from scratch. Additionally, it would allow vendors to specialize in different components of PdM systems, which would promote competition and cooperation, ultimately leading to progress in the field. In a sense, the present work is an attempt at developing such an integrated platform.

Secondly, considering the novelty of PdM research, most efforts are aimed at tackling simpler problems. Indeed, issues like the development of prognostics approaches for complex systems comprised by a number of subsystems and components, the development of algorithms robust to the uncertainty and variability inherent to a real-world manufacturing environment or the incorporation of event data in condition monitoring, diagnostics and prognostics remain largely unaddressed (Jardine et al., 2006).

Lastly, the amount of attention research in post-prognostics reasoning, or decision support, has received when compared to other components of PdM systems is staggeringly reduced. This is comprehensible given the fact that it requires the development of an information system that is able to integrate operation, maintenance, logistics, decision support and decision making, while, at the same time, providing the multiple types of end-user with the information deemed relevant for them without interrupting the system (Elattar et al., 2016). Nevertheless, it constitutes, along with the previously mentioned aspects, a research gap, that hinders the implementation of PdM systems in a real-world context.

Chapter 3

Problem outline

From the literature review of Chapter 2, a PdM tool is comprised of a number of different modules that communicate and mutually support each other with the goal of performing their tasks in the most effective way. Therefore, the challenge is to develop the architecture for a PdM tool that would enable such a goal. However, there is not a single universal architecture that can provide an effective solution to the needs of every industrial operation. Indeed, the demands and constraints to which the tool is subjected can experience considerable variability depending on the setting in which it is to be implemented. Hence, the specification of such demands and constraints devoid of context would have no meaning. For that reason, in this chapter, a brief background description of the modern industrial environment is provided. It is not the aim of this description to be extensive, but rather representative of the aspects concerning PdM. Then, some general concepts associated with equipment degradation and failure are presented. Lastly, a specification of the demands and constraints that a PdM tool is subjected to by such environment is provided.

3.1 Background

The modern industrial environment can be characterized by multiple processes that interact with each other and whose objective is performing a specific set of tasks. Drilling down on a specific process, one is able to identify a role of complex machinery with distinct degrees of criticality to the process. What this means in terms of maintenance strategy is that different assets have different levels of priority.

Each asset performs a set of subtasks which are at their core the same, but display a number of dissimilarities due to a number of demands, namely distinct product specifications and production recipes. In fact, the same manufacturing unit can observe multiple changes in both product specifications and production recipe throughout its operation. They also possess their own set of typical degradation processes and subsequent failures, which are dictated by specific physical mechanisms. However, even the same mechanism can display minor discrepancies due to dynamic operating conditions and workloads.

Given all these aspects, it is crucial that industries possess information regarding its processes. Manual acquisition of process information becomes an overwhelmingly cumbersome task. Therefore, modern industry is gravitating towards automatic or semi-automatic information gathering and storage. This way they gain access to historic records as well as real-time data from its processes and equipment. Such information is then made readily available through information systems like MES.

3.2 General concepts

Every equipment is subject to a certain degree of deterioration. It is usually assumed that an equipment that has just initiated its operation is in mint condition and it degrades throughout its lifetime until it reaches a state in which it can no longer operate as intended. The state of degradation can be expressed by a scalar value ranging between 0 and 1, in which the value 0 represents perfect health condition and 1 represents failure. This scalar is usually known as health index. The degradation of an equipment can be illustrated by a curve enclosing the evolution of its health index. Figure 3.1 depicts that degradation curve for a given equipment that has been allowed to run-to-failure.

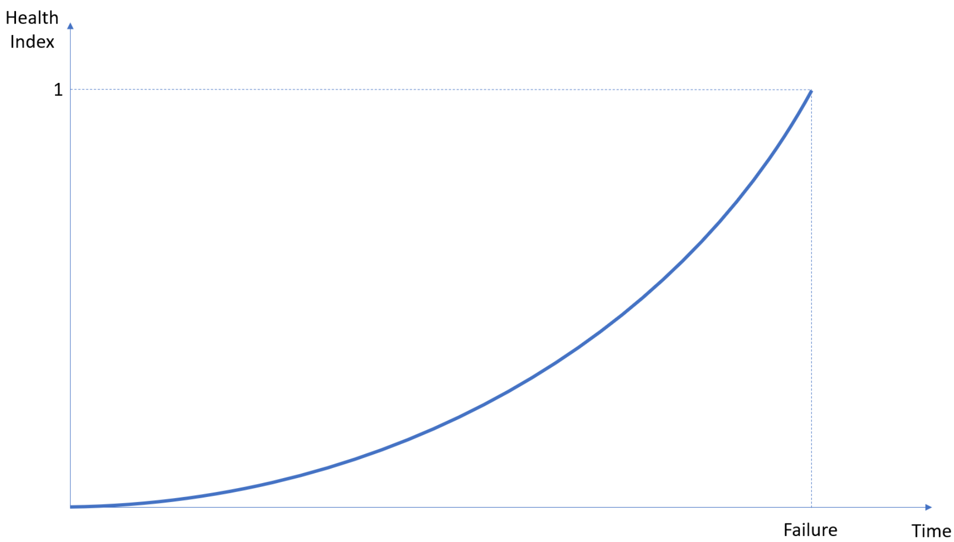

Figure 3.1: Real degradation curve of a manufacturing unit that has run-to-failure

The full shape of the degradation curve for a given equipment is only known if it is allowed to run-to-failure, because only that way the actual data that can locate the health index at any given time is available. If the equipment has not yet failed, the shape of the degradation curve can only be known to be contained within a given confidence interval. The same applies to the time of failure. Figure 3.2 depicts this situation.

Considering maintenance actions do not occur instantly, there is a time interval between the moment when maintenance is requested and when it is actually performed whose duration depends on several factors such as the speed of the maintenance decision, the deployment of the

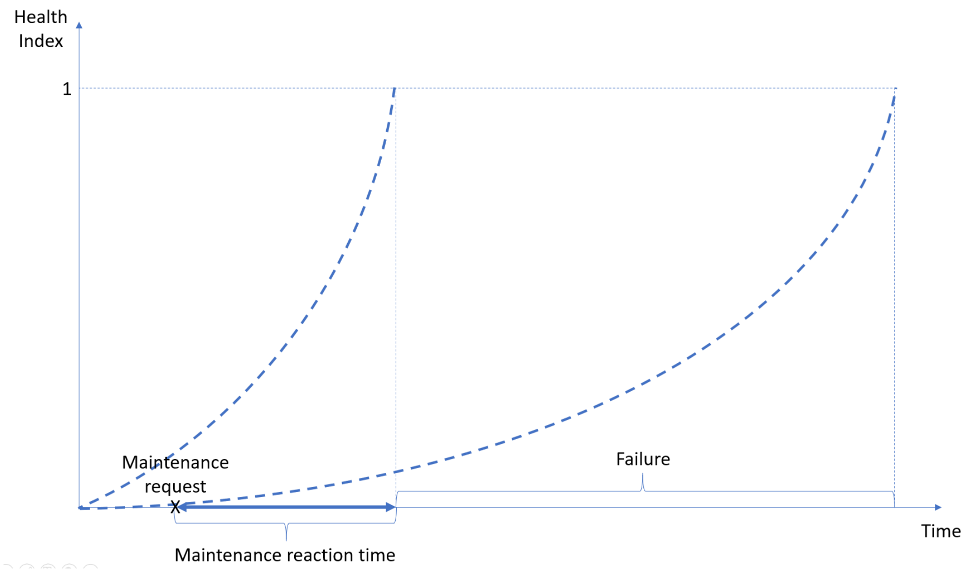

Figure 3.2: Degradation curve of a manufacturing unit that has not experienced failure

maintenance schedule and the duration of the actual maintenance tasks (Javed et al., 2017). In a worst case scenario, failure occurs the earliest, thus preventive maintenance is requested in such a way that it is performed just in time before the occurrence of failure while also accommodating the maintenance reaction time. The consequence is that if the failure would actually occur later, there will be a loss of productivity due to underused useful life. Figure 3.3 depicts this concept.

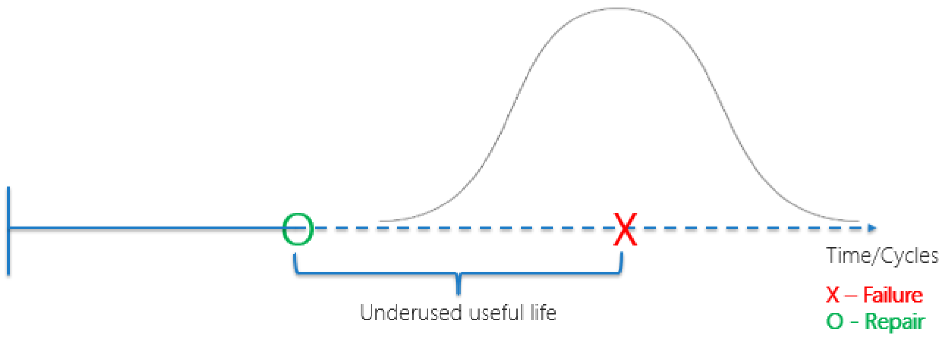

Figure 3.3: Underused useful life as a result of preventive maintenance

In PdM, sensor readings keep track of the equipment's real degradation by providing a snapshot of its condition at a given time. This allows for the degradation curve to be updated by a more accurate estimate as illustrated by Figure 3.4. Thus, a trade-off occurs: on the one hand, the longer the equipment is allowed to operate freely, the more sensor readings are obtained and, consequently, the more accurate the estimate of the real degradation curve will be; on the other hand, the shorter will be the slack allowed to accommodate maintenance reaction time as depicted in Figure 3.5 by the shaded lines.

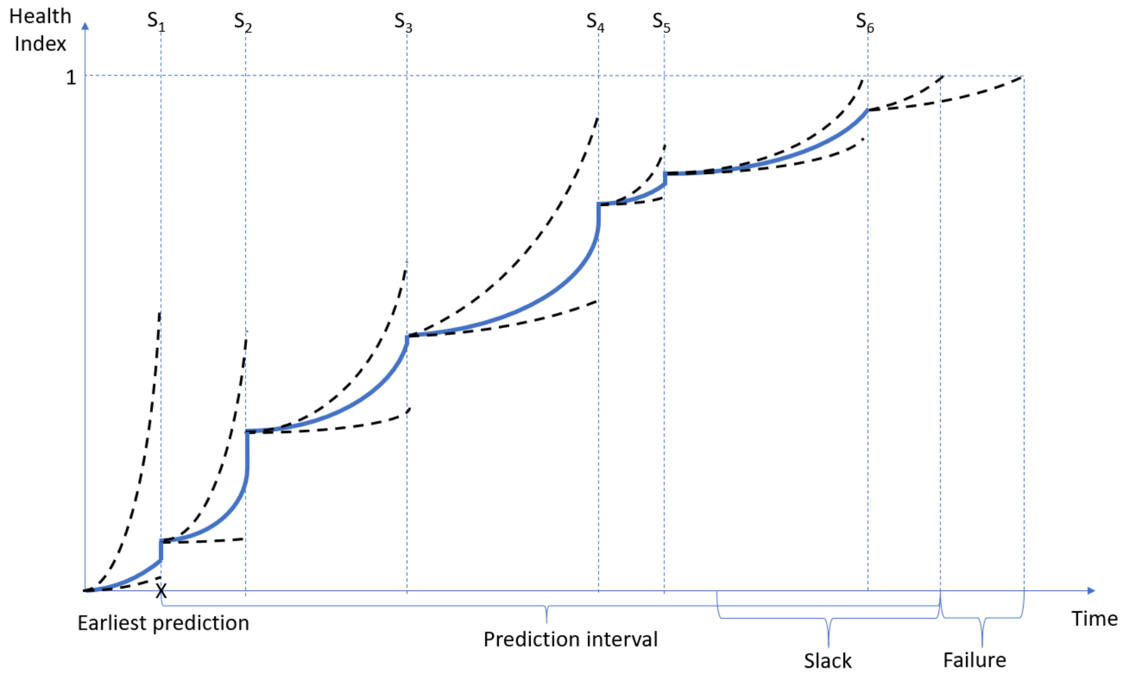

Figure 3.4: Degradation curve of a manufacturing unit updated by sensor readings

3.3 Requirements

Considering all that was presented in the previous sections, a well-designed PdM tool is required to meet a number of practical requirements, which will now be enumerated.

Adaptability As it was shown in the background section, there is considerable variability inherent to any modern manufacturing system at all levels. A well-designed PdM tool should be able to model as much of the system's variability as possible, particularly the one associated with multiple degradation processes, operating conditions and workloads.

Accuracy and precision It is desirable that a PdM tool provides consistently accurate estimates of the time-to-failure. However, there is an important caveat to the accuracy requirement. Considering that the estimates will only be able to approximate the real time-to-failure, it is preferable that the tool underestimates the real time-to-failure rather than overestimating it. The reason for this is that overestimation results in unexpected breakdowns due to the false belief that the asset is able to operate for a longer period of time, whereas underestimation only results in loss of productivity due to underused useful life. Assuming that the total cost of reparation of unexpected breakdowns and production stoppage is much larger than the total cost of preventive replacement and loss of productivity due to underused useful life, underestimation of the real time-to-failure is preferable.

Speed The existence of a reaction time of maintenance actions requires the PdM tool to provide an estimate of time-to-failure as fast as possible so as to ensure that there is enough slack to accommodate the reaction time. However, this requirement is conflicting with the accuracy of the system, due to the existing trade-off depicted in Figure 3.5.

Data integration Due to the current upward trend in data acquisition across all industries, a PdM tool would benefit from the integration of such data. Specifically, it should be able to simultaneously acquire and process both condition monitoring data and event data. Furthermore, historic records and real-time data can be looked at as sources of different types of information. Hence, PdM tools should contemplate both online and offline data.

Interpretability The ultimate goal of a PdM tool is to arm the end-user with an array of new information that allows him to learn more about the degradation process of the asset and become better prepared to provide an effective solution to the impending failure. For this purpose it is advisable that the outputs of the PdM tool are presented in such a way that the end-user can quickly grasp their meaning.

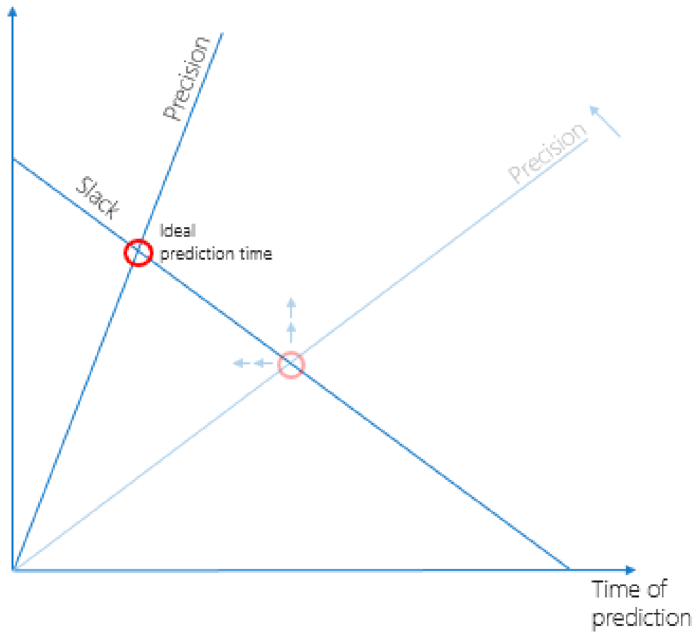

Figure 3.5: Trade-off between accuracy/precision and slack for accommodating maintenance reaction time

3.4 Solution and testing

Considering the list of requirements that was presented, the architecture for a PdM tool that attempts to enable their achievement was devised. The intent was then to test the solution in a real-world dataset so as to ensure the system's architecture would be tailored to the specifics of an industrial setting. However, real-world datasets are uncommon due to a number of reasons. For one thing, although data acquisition through sensor technology is currently gaining traction in the industrial sector, it is occurring at a slow pace, hence, the number of companies currently performing it is fairly modest. Furthermore, this type of data contains classified production-process-related information, thus companies tend to not share it. In order to overcome this lack of real data, publicly available datasets were resorted to. Among them, there is NASA's Prognostics Center Turbofan Engine Degradation Simulation Data Set, which has been extensively used in the past years for PdM research. The reader is directed to Appendix B provide for a full description of this dataset.

3.5 Limitations

It is important to mention that due to the nature of the testing dataset and the low maturity level of technology associated with PdM, mainly in the prognostics department, there are a few limitations that should be acknowledged.

Firstly, the dataset used for testing is structured as a time series, which limits the existing options in terms of which algorithms can be used. It is not a severe limitation, for most research in PdM was done based on time series data. Nevertheless, it is worthy of pointing out.

Secondly, the same dataset was built for the purpose of prognostics research. Therefore, it is not indicated for fault detection and diagnostics testing. However, for the sake of consistency, the same dataset was used to test out all modules of the PdM tool by using some simplifications and assumptions.

Thirdly, considering the maturity level of some of the technology involved in PdM, the development of a tool that incorporates and gives an answer to all the different aspects of the industrial setting, that were mentioned in the previous background section, would be an overwhelmingly ambitious endeavour. For this reason, it was assumed that the target of the developed PdM tool would be a manufacturing unit with static operating conditions and workloads and a single failure type and respective degradation mechanism.

Lastly, the way the newly found information can be conveyed to the end-user is extremely subjective. For one, assuming that the end-user can be either a maintenance engineer, a line manager or a plant manager, the information might be presented in multiple different ways and with distinct levels of detail. Additionally, optimization tasks, which fall within the scope of the decision support module, largely depend on industry-specific requirements and constraints, thus there is no "one-size-fits-all" system. Hence, the decision support module of the PdM tool will not be thoroughly explored. Nevertheless, the proposed tool architecture will consider the existence of this module.

Chapter 4

Predictive maintenance tool design

In order to ensure the requirements of a PdM tool are met, specifically the compatibility between basic elements and adaptability, a hierarchical, top-down, iterative design approach seemed to be the most appropriate. Initially, a high level overview of the whole system is provided. At this stage, a high degree of abstraction is employed to define the number of basic elements that constitute the system, their roles and the basic workflow. Additionally, a distinction between main and support modules is introduced. Then, at an intermediate level, the workflow is further detailed by introducing a separation of the global workflow into two phases, offline and online, as well as a distinction regarding the roles of condition monitoring and event data. Additionally, the inputs and outputs of each module and their basic tasks are scrutinized. Lastly, at a lower level, the techniques used at each step are detailed in order to unveil deeper connections between the different modules.

4.1 High level overview

4.1.1 Basic elements & workflow

As previously stated in the literature review chapter, it is generally accepted that PdM systems are comprised of six basic elements: data acquisition, data preprocessing, condition monitoring, diagnostics, prognostics and decision support.

Data acquisition refers to the process of collection of the different data types. The preprocessing module is responsible for taking the raw data and applying the necessary routines to make the data usable by the following modules. Condition monitoring keeps track of a specific set of parameters and triggers the diagnostic module whenever it detects signs of the development of a fault in the system. At this stage the diagnostics module identifies the detected fault. Then, with this new information, the prognostics module is able to estimate the current health state of the asset, its time-to-failure or its probability of failure within a given time span. Lastly, the decision support module reports the findings to the end-user in the way that is most convenient to him and suggests a response to the impending fault. These elements are required at different moments in time and succeed one another as Figure 4.1 depicts.

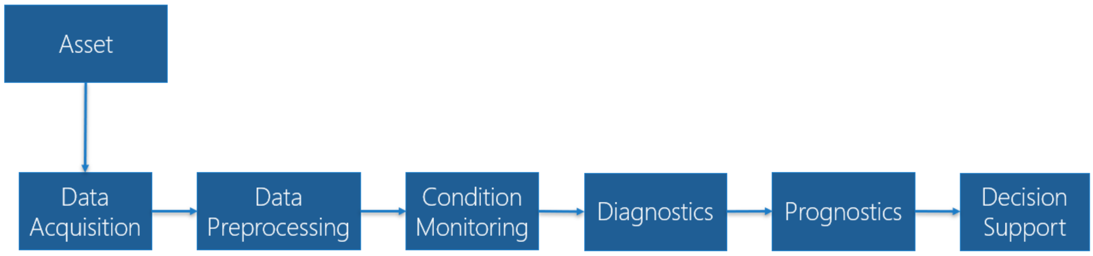

Figure 4.1: High level overview of the predictive maintenance tool framework

4.1.2 Main & supporting modules

Considering the roles of each module, it is possible to identify a distinction between them. Condition monitoring, diagnostics, prognostics are the modules that actually produce knowledge regarding the degradation process of the monitored asset upon which the end-user can act. The data acquisition, data preprocessing and decision support modules, on the other hand, do not produce new information regarding the asset, yet play supportive roles in the effective execution of the tasks of the main modules. Hence, the former will be referred to as the main modules, whereas the latter will be referred to as support modules. The cooperation between the support and the main modules is vital for the good functioning of the PdM tool.

4.2 Intermediate level

In this section, the high level framework will be further detailed. It will begin by introducing the architectural separation between an offline learning phase and an online application phase. Then, it will present the distinct workflows of condition monitoring data and event data. Lastly, the inputs and outputs of each module will be analyzed.

4.2.1 Offline & online phases

Observing the sections of Chapter 2 referring to the main modules, it is possible to conclude that each requires prior knowledge regarding the degradation process of the monitored asset to some extent. In the case of condition monitoring, a model of the normal operating conditions of the monitored asset is required, whereas, in diagnostics and prognostics, it is necessary to know how the asset behaves when a given fault is present. This leads to believe that there are two main moments in the operation of the PdM tool: a learning moment, in which a supporting body of knowledge is built, and an utilization moment, when this body of knowledge is used to effectively perform the solicited tasks.

Additionally, the degradation of a given equipment is not deterministic. It depends on a number of factors. Some of these factors are the same across all elements of a given population. However, no two units of the same population are exactly alike. Despite belonging to the same population, there are individual characteristics that introduce some variability in the population.

As a result, the successful deployment of a PdM tool requires access to insights on both population-wise and individual-wise behavior.

That said, it is possible to observe that the learning phase and the utilization phase are closely related to the extraction of population-wise and individual-wise insights, respectively. The framework must, then, reflect these aspects.

Figure 4.2 introduces a separation of the workflow into two distinct lanes. The top lane represents the offline learning phase in which the tool forms a body of knowledge that will provide support to the execution of the tasks associated with the three main modules. For that purpose it leverages the data contained in historic records of manufacturing units of the same population. The lower lane, on the other hand, represents the online phase which combines the body of knowledge formed in the offline phase with unit-specific information acquired through sensor technology, thus allowing the actual execution of the tasks of the main modules.

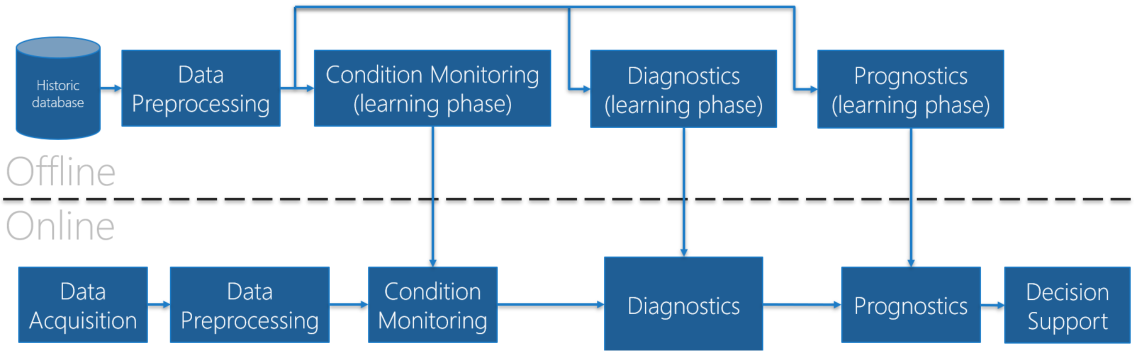

Figure 4.2: Offline and online phases of the predictive maintenance tool

The purpose of the offline phase is to assist the main modules in the successful execution of their tasks in the online phase by supplying them with all the information required. Indeed, as Figure 4.2 attempts to depict, the offline phase oversees the main modules and bridges over to the online phase, supplying it with vital information that it was able to extract from the existing historic data collections.

4.2.2 Condition monitoring & event data

Condition monitoring, diagnostics and prognostics are challenging tasks. One of the main reasons for this is that, in a real-world setting, most production systems are not static. They enjoy multiple operation modes, which complicate the aforementioned tasks. Indeed, a set of sensor readings representing a certain condition indicator might correspond to normal operation for a given operation mode, whereas for another, the same set might indicate faulty behavior. Figure 4.3 depicts a representative example of this situation, where a given manufacturing unit operates in normal conditions, but with multiple operation regimes. Devoid of any context, the condition monitoring module of the PdM tool would label the operation occurring after the 50^{th} cycle as anomalous, when, in fact, considering the operation modes, it represents perfectly normal behavior.

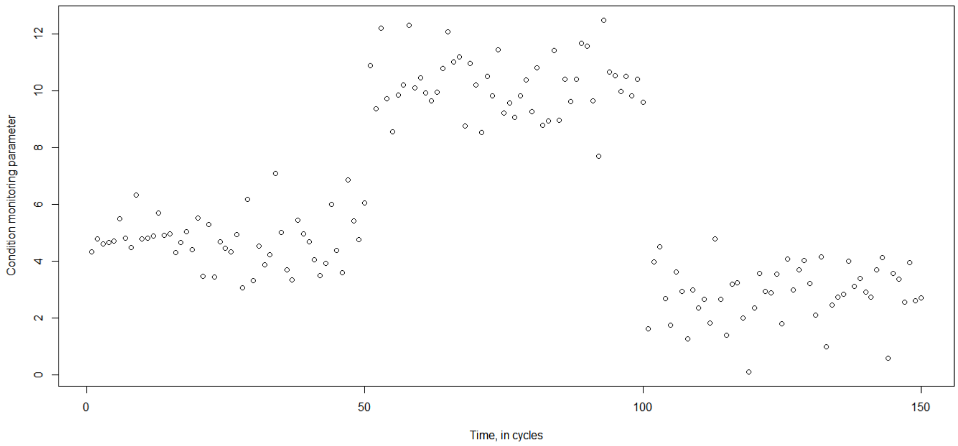

Figure 4.3: Sensor readings of a condition monitoring parameter with multiple operation modes

One is, then, able to conclude that some context has to be provided to the main modules so that the condition monitoring data they utilize has significance. That context can be given through event data, like the product that is being manufactured, the production recipe that is currently being used or the workload. Figure 4.4 presents a more detailed version of the framework in which the MES provides the PdM tool with the necessary context for the main modules to perform their tasks correctly and in a more realistic manner.

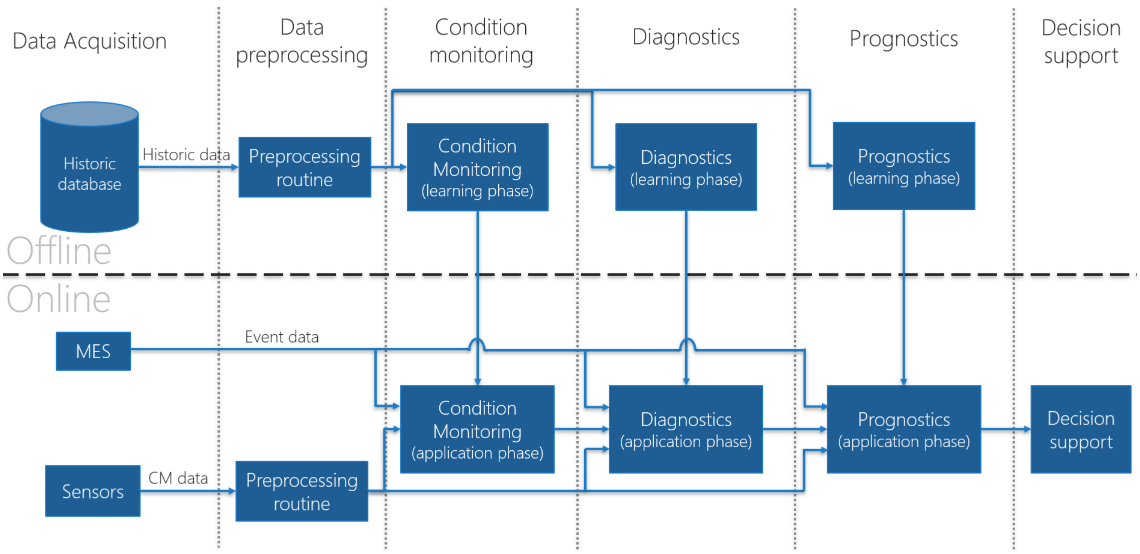

Figure 4.4: Intermediate level of the architecture of the predictive maintenance tool with operating context

4.2.3 Inputs & outputs

For a better understanding of the information flow between the different modules, this section will analyze their inputs and outputs.

Data acquisition Figure 4.5 depicts the intermediate level view of the data acquisition module. As shown, condition monitoring data and event data are acquired in different manners. Condition monitoring data is collected directly from the monitored asset via an ensemble of sensors and is made available for access in the MES thanks to two components. The first is an Internet of Manufacturing Things (IoMT) module which establishes the bridge between the asset and the information system. The second is a Complex Event Processing (CEP) module which handles the data stream. Event data, on the other hand, can be either manually introduced in the MES by the user or directly acquired through operations performed by the system. Once the data is in the system, a timeseries database is built, for, as the section respecting the low level vision of the framework will show, most techniques associated with the subsequent modules require a time series format. Data is acquired continuously through this process until the monitored unit has failed. At that moment, the data contained in the online timeseries database is a representation of the full degradation process of the monitored unit. Hence, it is replicated into the offline historic database so it can contribute with insights regarding that specific type of unit.

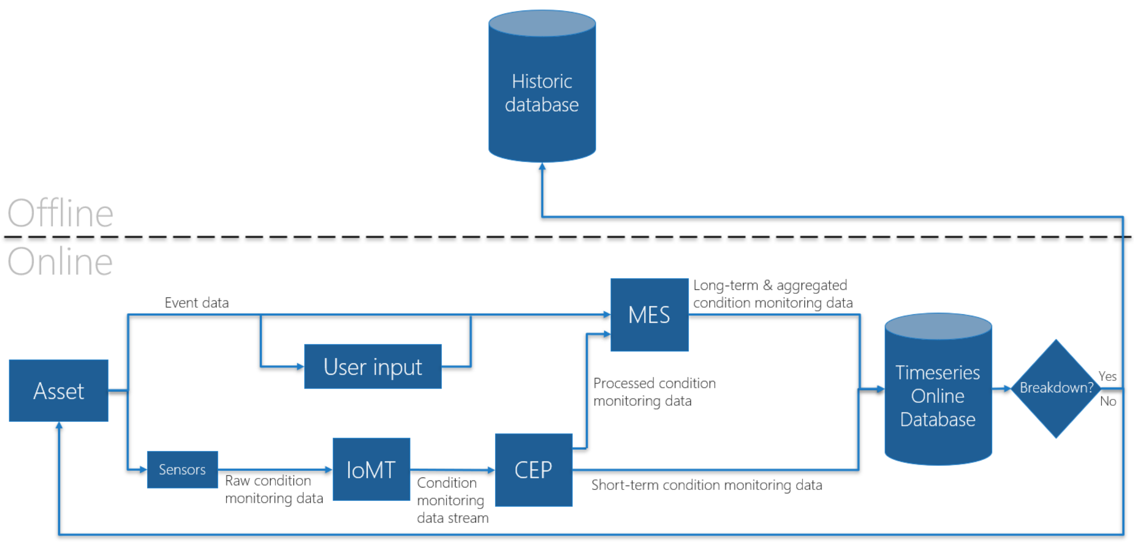

Figure 4.5: Intermediate level view of the data acquisition module

Data preprocessing Due to the differences between event and condition monitoring data, each type has its own preprocessing routine. Both routines aim at preparing the data for subsequent use, ensuring its quality. However, the one associated with condition monitoring data is more concerned with its content, whereas the one associated with event data is more concerned with the format. The tasks of each routine depend on the techniques employed in the main modules. Hence, data preprocessing will be detailed further in the next section, along with such techniques. Considering that the MES is responsible for the preprocessing of event data, the next section will focus on condition monitoring preprocessing.

Condition monitoring Process monitoring occurs online. The module continuously monitors the condition data coming from the previous module. This task is assisted by an operation mode locator, which is able to provide the operating context mentioned in the previous section, and the knowledge that could be extracted from the existent historic data. Once a fault is detected, the diagnostics module is triggered. Figure 4.6 depicts the described process.

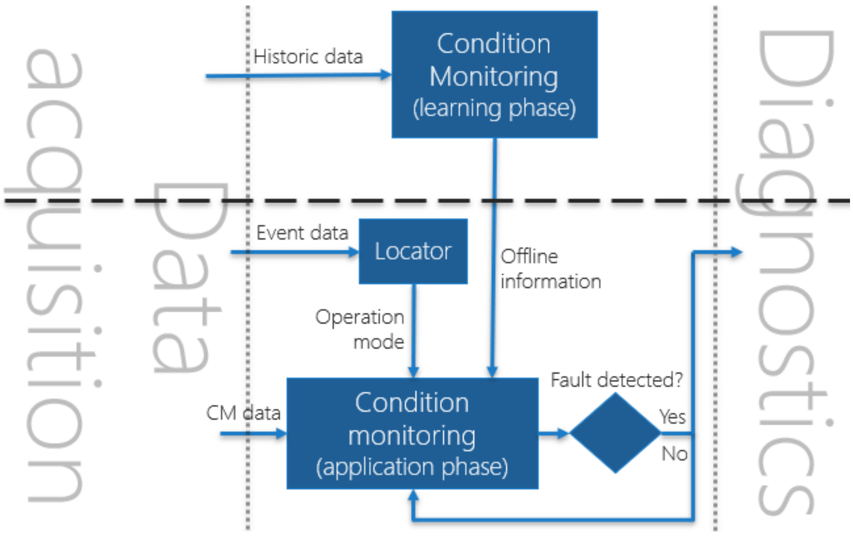

Figure 4.6: Intermediate level view of the condition monitoring module

Diagnostics Diagnostics can provide three different types of results. Firstly, by looking into the patterns described by the condition monitoring data, it is able to determine the failure mode behind the observable degradation. Secondly, it is able to identify the parameters associated with the detected fault and quantify their correlation. Lastly, knowing which parameters are related to a given fault, it can identify its root cause. Figure 4.7 depicts the described process.

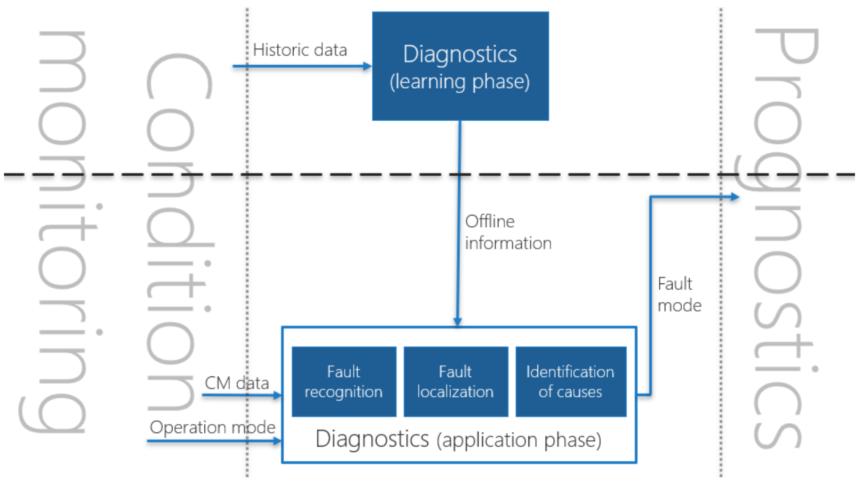

Figure 4.7: Intermediate level view of the diagnostics module

Prognostics Once the incipient fault has been identified, the prognostics module is triggered. Each failure mode has its unique underlying failure mechanism, which governs the way an equipment degrades over time. Similarly, the operation mode also has an effect on the degradation process. For instance, a more intense workload is expected to accelerate the degradation process. Hence, both, along with the knowledge extracted offline, are inputs of the prognostics module, that is able to perform several tasks, namely assess the current health state of the monitored equipment, estimate its time-to-failure or estimate the probability of failure within a given time span. Figure 4.8 depicts the described process.

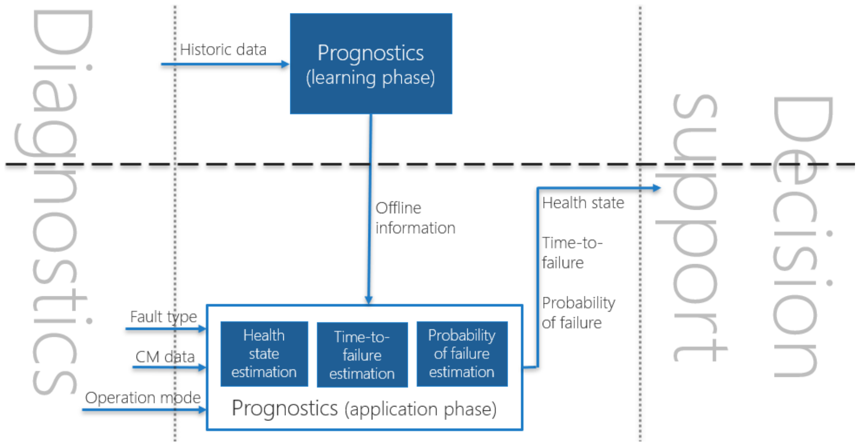

Figure 4.8: Intermediate level view of the prognostics module

Due to the previously referred motives, the decision support module will not be detailed any further. Hence, the analysis of the intermediate level of the framework is complete.

For a complete visualization of the developed architecture, it is recommended that the reader consults Appendix A, more specifically Figure A.1.

4.3 Low level

Once the general framework is conceptualized and the basic workflow defined, testing is required to assess the architecture's robustness and unveil deeper interactions between modules. In this section, a detailed explanation of how each module is expected to work is given based on the testing against the dataset presented in Chapter 3. The PdM tool modules that provide the most significant outputs are condition monitoring, diagnostic and prognostic. On that account, this section is structured in such a way that the functioning of each of these modules will be detailed thoroughly including their interactions with other modules, particularly the preprocessing module.

4.3.1 Condition monitoring

As was mentioned previously, the purpose of condition monitoring is to supervise the production process by keeping track of a given number of performance parameters. Being one of the most widely used techniques currently, a SPC chart was applied to the present case. Due to the large number of variables contained in the dataset, a multivariate version was appropriate. Thus, the PCA-based statistics, Hotelling T^2 and Q, were used. However, these techniques required some previous data preprocessing, namely regime partitioning, outlier removal, standardization and nominal model building.

Regime partitioning Firstly, considering the purpose of condition monitoring is to detect deviations from normal operating conditions, it is fundamental to possess a sample of such regime. By visually inspecting the scatterplot depicted in Figure 4.9 it is possible to conclude that during the first few cycles the sensor readings are fairly stable, fluctuating around a set value. The smoothed line allows for a better visualization. The same trend is observed in the remaining sensors and units. Based on this assumption, the first 50 cycles of each sensor from each training unit were considered representative of normal operating condition and extracted to form a dataset containing normal equipment behavior only.

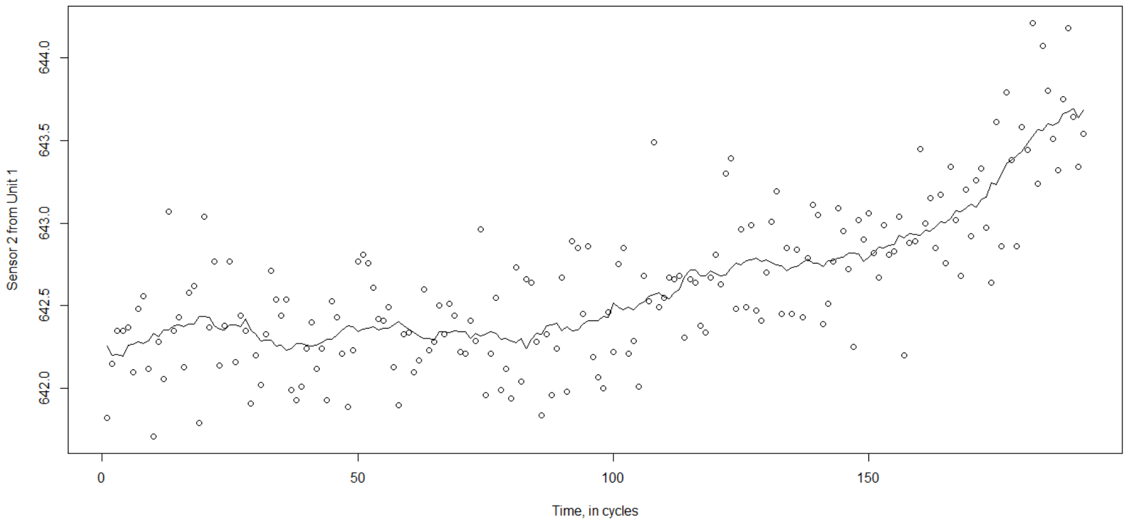

Figure 4.9: Scatterplot of sensor 2 from training unit 1

Outlier removal The generated dataset contained abnormal readings, which could represent faulty sensor behavior or microtendencies of irregular operation. These abnormal readings may affect negatively the representativeness of the healthy operating condition sample. Having no way of distinguishing between the two possible causes of such values, it was decided that outliers would be removed. The technique employed for outlier detection was a simple moving window that calculates the median and the median absolute deviation, *MAD*, for the points that fall within the window. Figure 4.10 shows an example of a moving window with step size of 1 and window width of 10, which were the chosen values in the present case. If a data point laid outside of the interval $[Median - 3MAD, Median + 3MAD]$ it was considered an outlier and it was replaced by the respective median.

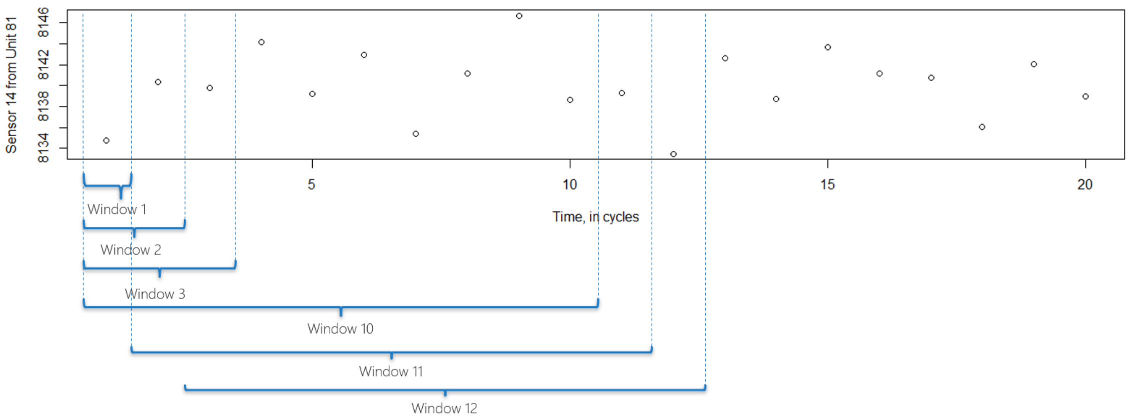

Figure 4.10: Example of a moving window with step size of 1 and window width of 10

Standardization As mentioned in Chapter 2, PCA requires the data to be centered and scaled. Due to this constraint, z-score standardization is applied to both the healthy regime dataset and the original dataset containing faulty behavior. The mean and standard deviation used for centering and scaling are estimated from the healthy regime dataset.

Nominal model The application of this technique to the problem at hand occurred in two moments. Firstly, it was applied to the healthy regime dataset. Figure 4.11 presents a summary of the results of this analysis, which shows that, by retaining 9 principal components, over 90% of the cumulative proportion of the dataset's variance could be explained. Then, having obtained in the previous step the matrix of variable loadings associated with the retained principal components, the original dataset can be rotated, thus giving the score of each data point along each principal component's axis.

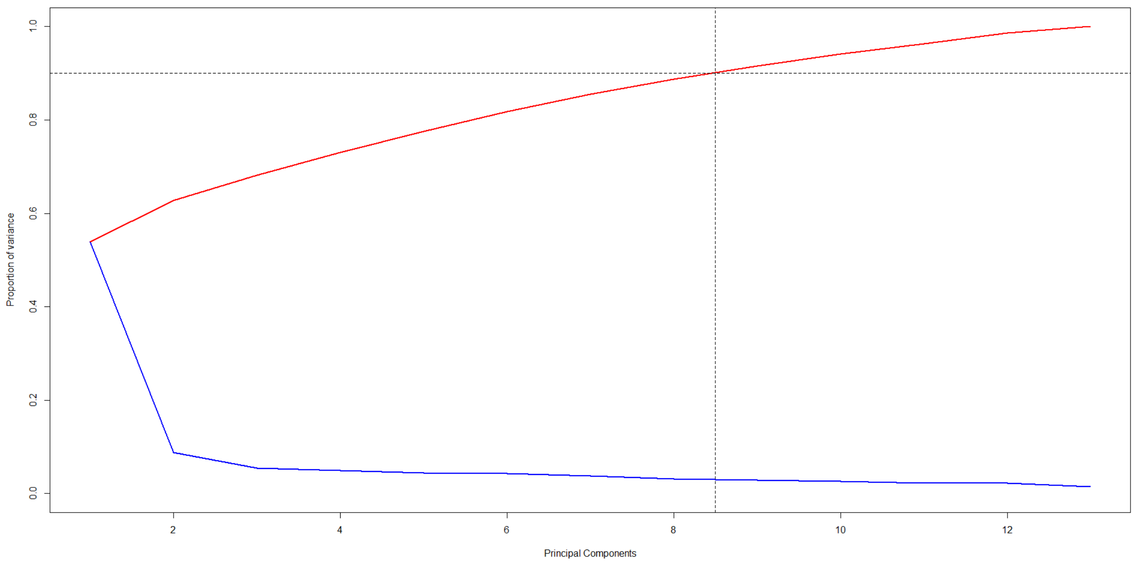

Figure 4.11: Summary of the results of the Principal Components Analysis on the training set

Hotelling T^2 control chart Finally, the necessary conditions for obtaining the Hotelling T^2 control chart were met. The upper threshold was calculated as in 2.3 for a significance level of 0.01 and degrees of freedom of 5000 and 9, which are the number of observations in the healthy regime dataset and the number of retained principal components, respectively. The meaning of a significance level of 0.01 is that it is expected to witness 1 false alarm in each set of 100 observations. Thus, the upper threshold obtained for Hotelling's T^2 statistic was of 21.73. The statistic *per se* was computed as in 2.2 for each new observation. Figure 4.12 depicts the Hotelling T^2 control chart for test unit 1, which shows that the technique is able to detect a consist drift to an out-of-control region starting at around 170 cycles.

Q control chart Granted that the model assumes normal operating conditions, a drift of the Q statistic off of its in-control region would indicate that some occurrence rendered the model no longer representative of normal operating condition. That occurrence can be interpreted as the presence of a fault in the system. Hence, the upper threshold for the Q statistic was calculated as in 2.5 for 9 retained principal components and a significance level of 0.01 for the parameter c_α which returned a value of 3.77. The Q statistic was computed as in 2.4 for each observation of each unit. Figure **??** depicts the Q control chart for test unit 1, which shows that, just like the Hotelling T^2 control chart, this technique is able to detect a consistent drift to an out-of-control region starting at around 130 cycles.

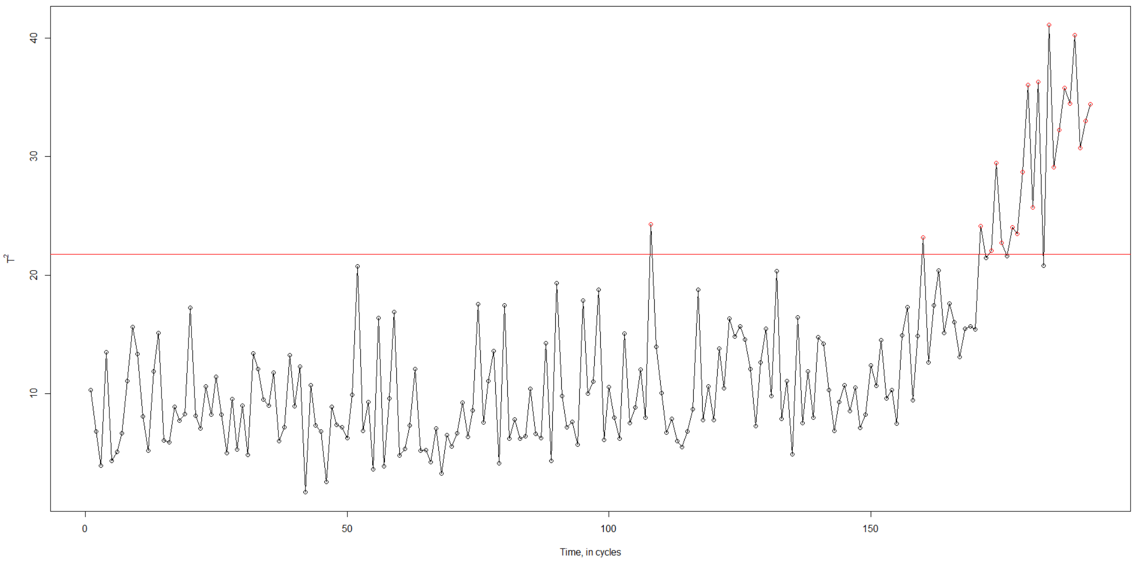

Figure 4.12: Hotelling T^2 control chart for test unit 1

4.3.2 Diagnostics

The detection of a fault is not sufficient for the effective prescription of a maintenance action. In fact, different faults have different degradation processes. An asset may degrade quicker or slower, which will lead eventually to a sooner or later failure. For that reason, it is necessary

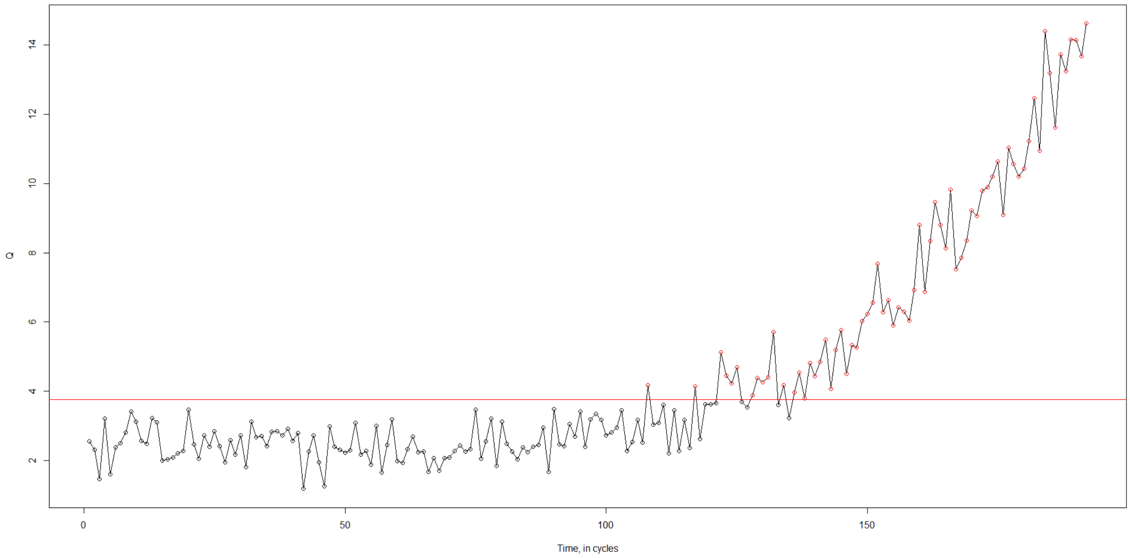

Figure 4.13: Q control chart for test unit 1

to identify the developing fault. Furthermore, additional information regarding that fault can be extracted, namely which sensors are the most related with it and what is its root cause. The ideal diagnostics module is one that tackles these three tasks at once. However, fault identification and root cause analysis require some extent of domain knowledge. Due to this restriction, it was only attempted to tackle the faulty variables problem by applying the technique for Hotelling T^2 decomposition described in Chapter 2. Once the condition monitoring module detects an out-of-control signal, it triggers the diagnostics module by lending it the out-of-control multivariate T^2 statistic. Applying T^2 decomposition to this statistic as described in Chapter 2, the module is able to ascertain which variables are the most correlated with the developing fault. This method was tested in two circumstances which were before and after performing PCA.

4.3.3 Prognostics

Two prognostic approaches were tested. The first one rests on the assumption that units with resembling degradation curves experience failure at the same time in their lives. The second one assumes that equipment belonging to the same population observe degradation curves with the same functional form, whose parameters are updated by individual observations collected throughout the unit's lifetime. These two approaches were tested against off-the-shelf machine learning solutions provided by Microsoft and against the traditional reliability centered maintenance solution.

4.3.3.1 Similarity-based approach

This approach is based on the principle that if the shape of the degradation curve witnessed up until the current moment for a given piece of equipment resembles the degradation curve of an already failed similar equipment, then it is assumed that the former will fail around the same time as the latter did. Similarly to condition monitoring, the testing of this approach requires previous data preprocessing. It shares with the condition monitoring approach the steps of outlier removal, standardization and PCA. Additionally, it requires the synthesizing of a health index, its subsequent regression and assessment of curve similarity.

Health Index There is some interest in presenting to the end-user an estimate of the current health state of the monitored equipment. One way to do so is by synthesizing a health index. This metric can be formed based on the result of the PCA. As was previously mentioned, each principal component is able to explain a given proportion of the system's variance. Those same principal components together explain a given percentage of the system's cumulative variance. It is possible to obtain the relative importance of each principal component to the explanation of the system's variance by dividing the proportion of the system's variance that it is able to explain by the cumulative variance that they explain all together. Equation (4.1) shows how to calculate this amount, where w_i is the relative weight of the i^{th} principal component, p_i is the proportion of variance it is able to explain and M is the number of retained principal components.

$$w_i = \frac{p_i}{\sum_{j=1}^{M} p_j} \tag{4.1}$$

The health index is then synthesized by taking the score of each retained principal component and multiplying it by its respective relative weight as Equation (4.2) shows.

$$HI = \sum_{i=1}^{M} w_i \times score_i \tag{4.2}$$

Hence, a PCA model is learned through the training units and health indexes are synthesized for both training and testing units. It is desirable that the health index has easy interpretation. It is not clear how negative values reflect an equipment's state of degradation, neither is it when different indexes have different upper and lower bounds. Hence, min-max normalization was used to ensure every health index was a positive scalar value bounded between 0 and 1, which represent perfect condition and failure, respectively. The result of this whole process is depicted in Figure 4.14 for training unit 1. It is shown that the unit did not start its operation in perfect health condition. Instead, it started out with a degradation level of about 20% which evolved until it reached failure around 190 cycles.

Regression Although Figure 4.14 shows a steady upward trend of the health index representing the degradation of training unit 1, that degradation would be better perceived by a smoother curve. Additionally, it would provide a better term of comparison when computing the similarity between online and offline degradation curves. For that purpose, the health index is regressed onto a curve that best fits the data. Three functional forms were considered: exponential, 2^{nd} degree polynomial and 3^{rd} degree polynomial. As Figure 4.15 shows, the 3^{rd} degree polynomial appeared to provide a better fit to the data than the other functional forms. Thus, the health index data for each training unit was regressed onto a 3^{rd} degree polynomial curve.

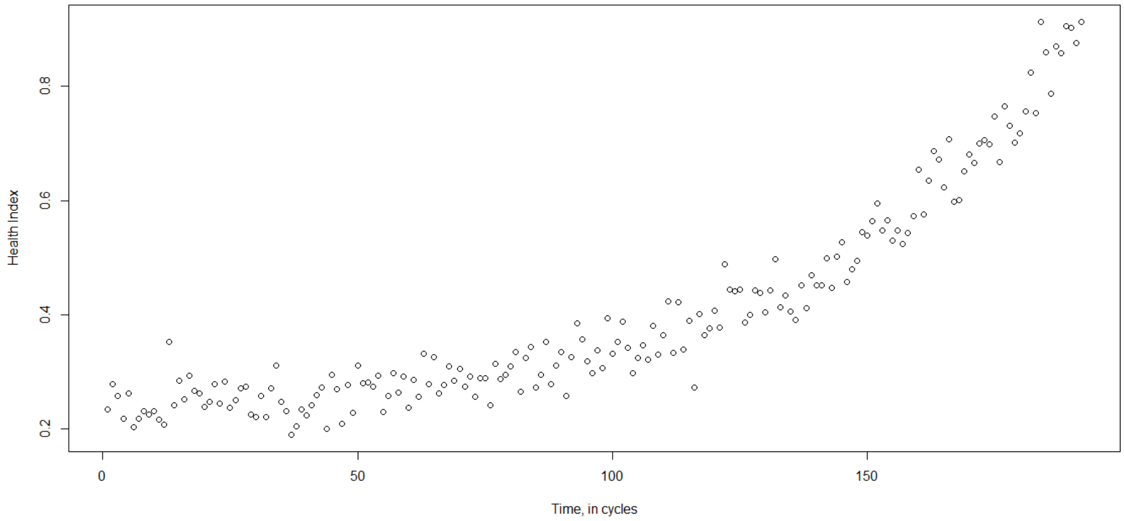

Figure 4.14: Scatterplot of the Health Index associated with training unit 1

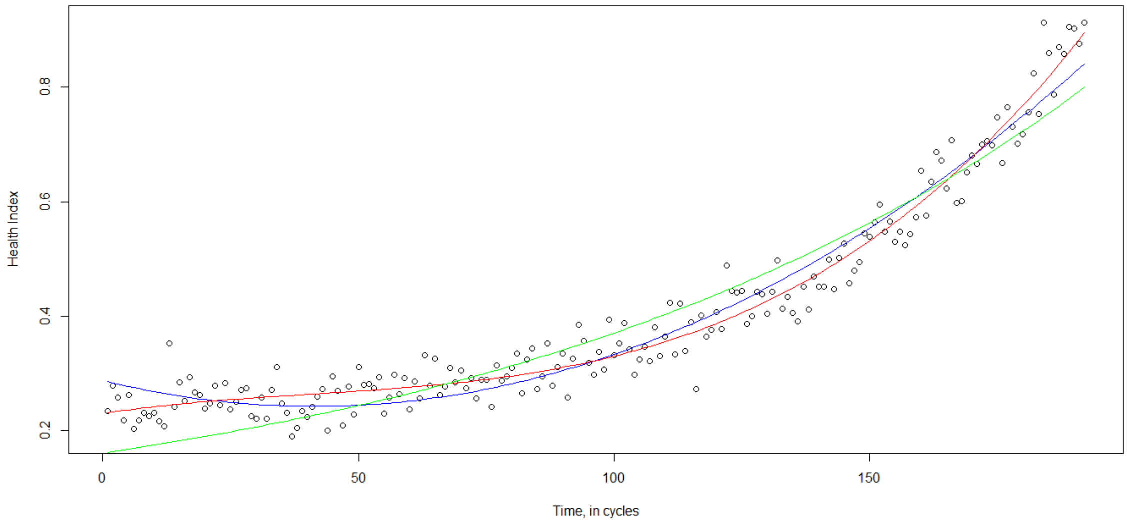

Figure 4.15: Health index regressed onto a curve with the form of an exponential (green), a 2^{nd} degree polynomial (blue) and a 3^{rd} degree polynomial (red)

Similarity assessment Once all the training degradation curves are obtained, it is possible to assess which one best fits the online degradation data. This is done by computing the mean squared error (MSE) for each one of the training degradation curves, j, which is given by

$$MSE_j = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_{ij})^2,\ j = 1,2,...,m \tag{4.3}$$

where Y_i is the i^{th} online observation, $\widehat{Y}_{ij}$ is the i^{th} observation of the j^{th} offline unit, m is the number of offline units and n is the number of online observations. The curve that provides the lowest MSE is the one that best fits the data and, so, it is assumed that the online equipment will fail at the same time as that unit did. Then, the time-to-failure of the online unit is simply computed by subtracting the number of operation cycles it lived through to the number of operation cycles the best fit offline unit has seen. Figure 4.16 depicts this procedure, where the blue points are the health index values of the test unit and the black points are the health index values of the training unit that best fits them. It should be pointed out that, in some instances, the test units outlive the training curves. When this is the case, those particular training curves with lives shorter than the current lives of the test units are deemed meaningless for the similarity assessment step and, thus, discarded.

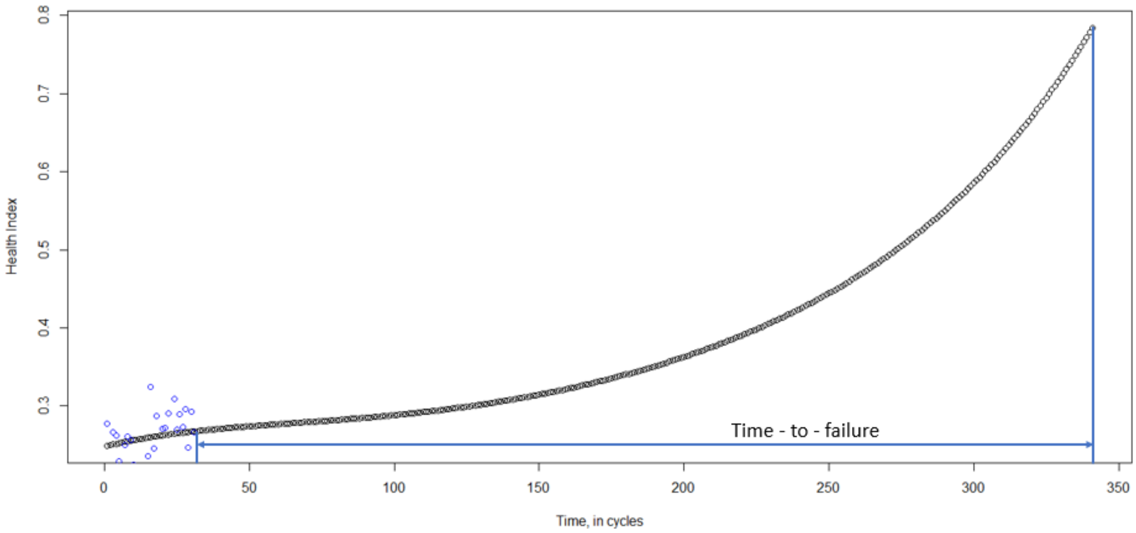

Figure 4.16: Similarity assessment and time-to-failure calculation

4.3.3.2 Bayesian update approach

This approach assumes that equipment belonging to the same population behaves similarly. Specifically, the degradation curves of all units belonging to the same population share the same functional form. However, this approach also assumes there is unit-to-unit variablility, which is why the parameters that define the shape of the degradation curve of each equipment follow specific distributions instead of being deterministic. The prior distributions of these parameters are assumed to be known and they are estimated from the training data. For a given test unit, those distributions are subsequently updated by each new set of current observations, which adapt the population distributions to that specific individual. Once this update is made, the time-to-failure distribution of that equipment can be extrapolated. Like in similarity-based approach, this approach requires previous outlier removal, standardization, PCA, health index synthesizing and regression. Additionally, it requires the estimation of the prior distributions, the update of those distributions and, lastly, the determination of the residual-life distribution. This approach is based on Gebraeel et al. (2005).

Prior distributions estimation Initially, it is assumed that the degradation signal of each training unit at any given moment, $L(t)$, can be modeled by an exponential function. Each training unit is regressed onto this model, from which it is possible to obtain a sample of estimates for the shape parameters. It is then assumed that these parameters follow a normal distribution whose mean and variance are obtained from that sample. The distributions thus obtained are referred to as prior distributions because they define the possible values for the shape parameters before considering any online observation.

Bayesian update Once the prior distributions are obtained, the joint posterior distribution of the shape parameters is given by a bivariate normal distribution which can be recursively updated in a Bayesian manner every time a new online observation is available. For a detailed explanation of the updating step, the reader is directed to Gebraeel et al. (2005).

Residual-life distribution determination In order to estimate the residual-life of each test unit, an estimate of the degradation level indicating failure is required. Letting D represent this threshold, it was assumed it was equal to the average degradation level of the training units at the time of failure. Then, letting T denote the residual life of the equipment at the current time t_k, it is possible to conclude that $L(T + t_k) = D$. Hence, the conditional cumulative distribution function representing the residual life of the equipment can be obtained by $P\{T \leq t \mid L_1, ..., L_k\} = P\{L(t + t_k) \geq D \mid L_1, ..., L_k\}$, where L_k is the degradation at time k.

Chapter 5

Computational experiments

In this chapter, the proposed framework is evaluated via its application to an example. It is important to emphasize that this is not a formal test, with the objective of proving that the proposed architecture can meet the requirements stated in Chapter 3. In fact, such a task would not be in line with the exploratory character of the present work, neither with the imposed limitations. This test must be looked at rather as a demonstration of how each of the main modules operates. Bearing this in mind, this chapter will begin by demonstrating the capabilities of the condition monitoring module, followed by the diagnostics module and, lastly, the prognostics module.

5.1 Condition monitoring

The main objective of the Condition Monitoring module is to detect the impending fault as soon as possible, in order to accommodate the maintenance reaction time. Hence, an approach that is quicker to signal the deviation of the equipment's condition to an out-of-control state will be favoured. The performance metric used for this purpose is the average number of cycles prior to failure that the module is able to detect a steady drift towards an out-of-control state.

The Hotellimg T^2 control chart and the Q control chart were tested using 10-fold cross-validation. Computing the average number of cycles prior to failure for each approach, the Hotelling T^2 control chart is able to detect the impending fault on an average of 44 cycles before failure, whereas the Q control chart detects it on an average of 103 cycles. Hence, the Q control chart is faster to signal an out-of-control state then the Hotelling T^2 control chart, which allows a longer maintenance reaction time. However, the Q chart is also very sensitive to what is used as a reference of normal operating condition. This was indicated by some occurrences where the control chart signaled an out-of-control state in early cycles. In other words, there were cases of false positives. Nevertheless, the aforementioned results were obtained from the detection of a steady drift towards an out-of-control region.

Additionally, it should be noted that both approaches are very easily interpreted due to their visual properties, as can be concluded from Figures 4.12 and 4.13. Indeed, with just a quick glance, the end-user can understand the current state of the asset, instead of having to rely on multiple univariate control charts for each of the tracked performance variables.

5.2 Diagnostics

As it was mentioned in Chapter 4 the detection of faulty variables was tested on the PCA-based and on the original-variable-based T^2 control statistic.

From what was stated in Chapter 2 regarding the computation of the contributions of each control variable, one can expect that the decomposition of PCA-based statistics, due to its dimension reduction property, is far less computationally expensive than the decomposition of a control statistic based on the original set of process variables. Indeed, the testing of both approaches lead to the confirmation of such expectations, with the PCA-based decomposition outputting the contributions of each latent variable much faster than the decomposition based on the original variables.

However, the same mechanism that allows for a reduction in dimension and makes the decomposition of PCA-based control statistics more computationally efficient, is also what renders it poor in terms of interpretability. Considering that the latent variables resulting from PCA are a linear combination of the initial variables, the signaling that a certain latent variable significantly contributes to the out-of-control state has no immediate meaning. In fact, the only possible interpretation of such an output in the absence of a physical meaning that could be attributed to the out-of-control latent variable would be that every original variable would somehow contribute to the abnormality. This could also be related to the occurrence of the smearing-out phenomenon. The decomposition of the control statistic based on the original values, on the other hand, is able to effectively pinpoint a set of variables that significantly contributed to the out-of-control signal.

In the present case, the decomposition of the T^2 control statistic indicated that the variable mix that significantly contributes to an out-of-control signal changes as the fault evolves. More specifically, there are less variables involved in the fault at earlier cycles than in later cycles. This is an expected behavior, because it is likely that the intensification of a fault tampers with more variables. Taking test unit 1 as an example, it is possible to see that, in earlier cycles, variable 13, for instance, does not contribute significantly to the abnormality, whereas in later cycles it does.

5.3 Prognostics

Regarding prognostics, the main goal is to provide the end-user with accurate estimates that enable the asset to reap the full potential of its useful production life. Hence, the similarity-based approach and the Bayesian update approach were assessed according to a set of accuracy metrics, namely the mean absolute error (MAE) and root mean squared error (RMSE). Additionally, their performance was compared to the performance of a simple predictor, which is just the average of the time-to-failure of the training set, by calculating the relative absolute error (RAE) and the relative squared error (RSE). Both were compared to a reliability centered maintenance (RCM) approach and to a number of off-the-shelf machine learning approaches developed by Microsoft, namely a decision forest regression, a boosted decision tree regression, a Poisson regression and a neural network regression. Lastly, the percentage of late predictions (LP) scored by each algorithm is also presented. Table 5.1 summarizes the findings which will be discussed throughout the remainder of this chapter.

Table 5.1: Results of the computational experiments

Algorithms	**MAE**	**RMSE**	**RAE**	**RSE**	**LP**
Decision Forest Regression	21.14	30.15	0.16	0.05	67%
Boosted Decision Tree Regression	21.28	29.62	0.16	0.05	63%
Poisson Regression	23.24	29.97	0.18	0.05	75%
Neural Network Regression	31.75	41.6	0.24	0.09	54%
SB 2.1 [a]	21.03	27.98	0.40	0.16	48%
SB 2.3 [a]	22.04	29.88	1.08	1.17	51%
SB 2.5 [a]	22.62	29.58	0.83	0.69	56%
SB 2.10 [a]	23.84	30.75	1.06	1.13	60%
SB 3.1 [a]	25.76	40.40	0.12	0.01	53%
SB 3.3 [a]	26.18	37.38	0.18	0.03	61%
SB 3.5 [a]	24.07	32.74	0.12	0.01	59%
SB 3.10 [a]	24.82	31.35	0.26	0.07	68%
BU	46.09	71.27	0.55	0.31	53%
RCM	31.81	37.02	0.24	0.07	67%

[a] The number code that follows the similarity-based approaches represent minor modifications to the original approach. SB 2.1, for instance, indicates that the similarity-based approach was used by fitting the training units to a 2^{nd} degree polynomial function and estimating the time-to-failure of the test units by comparison with 1 training unit.

5.3.1 Similarity-based approach

As the table shows, by inspecting the MAE results, SB 3.1 performed better in terms of accuracy when compared to the RCM approach. However, by inspecting the RMSE results it had a poorer performance. Considering that the RMSE metric penalizes large errors, this discrepancy evidences that the similarity-based approach is more accurate than the current reliability centered approach, but less precise.

Yet, the interest of such metrics is only partial. Indeed, MAE and RMSE entirely disregard the direction of the errors. In a prognostics context, that direction is very relevant as it is what distinguishes on-time predictions, which allow timely intervention of maintenance actions, from late predictions, which result in unexpected breakdown. Hence, the last column of the table shows the number of late predictions that resulted from the application of this algorithm. As shown, the similarity-based approach fails to provide an on-time prediction 53% of the times. Although this number is quite high, it outperforms not only the RCM approach but also Microsoft's off-the-shelf machine learning solutions.

As an attempt to explain the number of late predictions, an hypothesis was formulated. Considering that the more information is available on a given asset, the more accurate is the estimate of its time-to-failure, it is expected that the on-time predictions enjoy a longer observable life than the late predictions. Hence, this hypothesis was tested by taking the lengths of the test runs associated with on-time and late predictions, respectively, and computing their average and median. Table 5.2 shows that the expectations are met, but only slightly. Larger samples are required in order to draw significant conclusions.

Table 5.2: Comparison of the average and median observed life of the test units associated with on-time and late predictions of the similarity-based approach

	Average	Median
On-time	133	135
Late	129	132

However, further investigation into how the availability of data influences the effectiveness of this approach, it was tested on a random sample of 30 units that were extracted from the training set and whose degradation trajectories were truncated at 5 equally distributed moments in time. Ignoring the last truncation, which corresponds to the unit's full life, and computing the time-to-failure for each situation, their MAE could be calculated. As Table 5.3 shows, the MAE tends to increase with the observable life. At first, this might seem incongruent, but, in fact, the longer a unit lives, the less historic records of past units that outlive the current one exist, thus estimates may become biased. However, an extended observable life also produces less late predictions. This may be due to pure chance, given that the algorithm becomes less accurate with the extension of observable life. Hence, future investigation is recommended.

Table 5.3: Mean absolute error and number of late predictions of a sample of 30 units at four different life stages

	20% of total life	40% of total life	60% of total life	80% of total life
MAE	59.3	56.09	61.53	77.30
LP	18	18	16	14

Despite this, analyzing the metrics that compare the approach to a simple predictor, it is possible to conclude that SB 3.1 provides more accurate estimates than the simple predictor.

For the sake of experimentation, two minor modifications were made two the original approach: one was fitting the training units to a 2^{nd} degree polynomial function and the other was estimating the time-to-failure of the test units based on the average time of failure of the N training units that best fit the degradation curve of that test unit. Table 5.1 shows that, contrarily to what was expected, fitting the test units to a 2^{nd} degree polynomial function provides provides more accurate and precise estimates of the time-to-failure, thus leading to lower number of late predictions than the original approach. In fact, SB 2.1, out of all approaches, provided the lowest number of late predictions and lowest MAE and RMSE. The other modification, however, resulted in worse estimates overall.

5.3.2 Bayesian update approach

It was expected that the Bayesian update approach would achieve better performance in terms of accuracy than the similarity-based approach due to the fact that, contrarily to the latter, it does not rely on an approximation by the training data to estimate the failure time. However, by inspecting the columns of Table 5.1 respecting the MAE and RMSE it is possible to conclude this is not the case. This inaccuracy may have to do with the number of assumptions that the Bayesian update approach requires: the error terms must be independent and identically distributed with a normal distribution, the failure threshold must be previously known and defined and the degradation signal must have an exponential functional form. Additionally, the probability of an asset failing within a certain time span depends on a user-defined significance level.

Nevertheless, it is worth mentioning that this approach outperforms the RCM approach in terms of number of on-time predictions and performs just as well in this aspect as the similarity-based approach. Following the same logic as in the similarity-based approach, in an attempt to explain the number of late predictions, the lengths of the test runs associated with on-time and late predictions, respectively, were taken and their average and median computed. Table 5.4 shows the results.

Table 5.4: Comparison of the average and median observed life of the test units associated with on-time and late predictions of the Bayesian update approach

	Average	Median
On-time	138	146
Late	125	130

The outcome of such analysis is similar to the one from the similarity-based approach: on-time predictions result from units that enjoyed a slightly longer observable life. However, the difference between the observable lives of units associated with on-time and late predictions is too small to allow significant conclusions.

Although accuracy does not improve much with the length of the observed life, it is possible to observe an improvement in precision in most cases. This conclusion was drawn by extracting a random sample of 10 units from the training set, truncating their degradation trajectories at 20%, 40%, 60% and 80% of their total life and calculating their probability density functions (PDF) for each case. Figure 5.1 depicts the outcome of this process for two distinct units.

In both, it is possible to observe that the PDFs become narrower with an increase in observed life, representing more precise estimates of the time-to-failure. However, by inspecting both plots more closely, it is possible to observe that the estimated time-to-failure in the top plot tends towards zero with the progression of observed life, whereas, in the bottom plot, the time-to-failure estimate at 40% of the unit's total life is longer than that at 20%. Ultimately, the fact that this approach can compute the PDF of the time-to-failure for each unit and present in a visual manner evidence its potential.

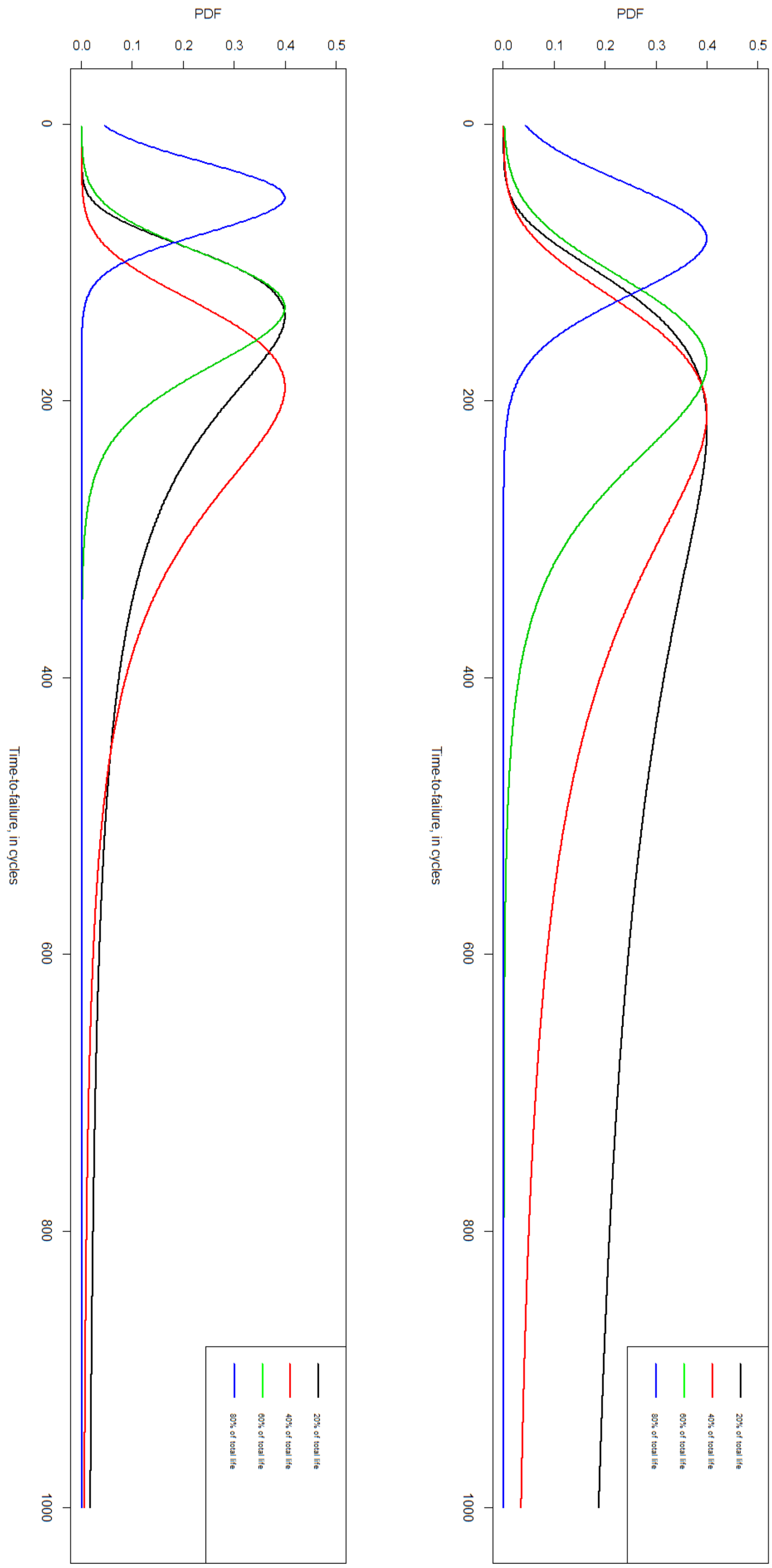

Figure 5.1: Probability density functions of the same unit at different life stages

Chapter 6

Conclusion

This chapter presents a summary of the developed work, giving special emphasis to a number of takeaways which, in retrospect, are considered to be essential. Subsequently, some suggestions regarding future work will be made based on the limitations of the present work.

Before, however, it should be mentioned that the present work did not aim to build a PdM tool ready for deployment, nor did it look to prove the dominance of some approaches over others. Rather, it should be regarded as an introductory step to the thematic of PdM within a MES, aiming to explore the existing possibilities of the field from a practical perspective. It was with that objective in mind that a framework for such a tool that is capable of leveraging the information in industrial plants was developed and subsequently tested, in an attempt to assess its coherence.

6.1 Key takeaways

PdM is a multifaceted problem. Indeed, a typical tool is comprised of six major modules, namely data acquisition, data preprocessing, condition monitoring, diagnostics, prognostics and decision support. The effective performance of each of these six major modules is vital for the overall good performance of the PdM tool.

From these major modules, only condition monitoring, diagnostics and prognostics produce, in-fact, knowledge. In reality, these are the modules that are able to detect the presence of a fault, characterize it and predict when its evolution will induce a failure. Nevertheless, the remaining modules play vital supporting roles in supplying the tool with good quality and insightful data and feeding back to the end-user the acquired knowledge in the most convenient way.

Although there is a wide array of eligible approaches that effectively execute the tasks associated with each of the aforementioned modules, only a few are appropriate for real-world applications. Indeed, depending on the prevailing circumstances, particularly the availability of process-related data or domain knowledge, some approaches may be more appropriate than others. Nevertheless, it is crucial to have access to information of whatever form from which to extract insights on the expected behavior of the monitored asset. Additionally, interpretability and adaptiveness are decisive criteria in selecting the most appropriate technique.

The knowledge-producing modules of a PdM tool rely on condition monitoring data to perform their tasks. However, relying solely on such data requires a severe simplification of the process behavior, thus making the outputs of each module less dependable. The articulation of a PdM tool with a MES can offset this limitation by granting access to other process-related information, often referred to as event data, that characterizes the setting in which the monitored asset operates.

6.2 Future research

The recommendations for future research stem out from the limitations of the present work and from the low maturity of PdM technology.

Considering that the present work was a first approach into the theme of PdM, a number of simplifications were made throughout as a way to cope with the complexity of different aspects of the problem. Albeit reasonable, such simplifications compromise the applicability of the newly found knowledge to a real-world context. Hence, a first set of limitations is associated with those simplifications, namely the existence of a single failure type and of a single set of static and known operating conditions. Future research ought to tackle these aspects. This can be achieved by testing the framework against a real world dataset that contemplates multiple failure types and operating conditions instead of testing it against a dataset obtained through simulation.

Although there was a concern with developing a general architecture that ensured adaptability to a broad number of circumstances, its testing against a single dataset limited the significance of the conclusions regarding this specific property. Additionally, the characteristcs of such dataset limited the assessment of the framework's workflow. For instance, the observable lives of the test units were too short for the condition monitoring module to detect a deviation from normal operating conditions and trigger the subsequent modules. In the future, the robustness of the framework should be assessed by testing it against multiple datasets that allow a streamlined application of all modules.

Another constraint was the unavailability of domain knowledge which obligated the present work to follow a data-driven methodology. However, its incorporation in PdM would not only allow the execution of additional tasks than the ones contemplated in the devised framework, like fault recognition and identification of causes, but it would also allow a timelier detection of incipient faults and more accurate predictions of the time-to-failure of the monitored assets.

Due to the highly subjective and industry-specific nature of the decision support module, it was opted not to explore it thoroughly. Nevertheless, considering this module is the interface between the tool and the end-user, its relevance calls for further research. A suggestion could be the development of a survey to understand the needs of the different end-users of a PdM tool regarding the level of detail of the information that is presented to them and the form in which they would rather have the data presented to them. Additionally, such a study would require the analysis of a number of operational aspects such as plant production demands and constraints, maintenance costs, spare parts availability, among others.

Bibliography

Abu-Mahfouz, I. (2003). Drilling wear detection and classification using vibration signals and artificial neural network. *International Journal of Machine Tools and Manufacture 43*(7), 707–720.

Ahmed, M., M. Baqqar, F. Gu, and A. D. Ball (2012). Faut Detection and Diagnosis using PCA of Vibration Data from a Reciprocating Compressor. (September), 3–5.

An, D., N. H. Kim, and J.-H. Choi (2015). Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering System Safety 133*, 223 – 236.

Bajric, R., N. Zuber, G. A. Skrimpas, and N. Mijatovic (2016). Feature extraction using discrete wavelet transform for gear fault diagnosis of wind turbine gearbox. *Shock and Vibration 2016*.

Basu, S. and M. Meckesheimer (2007, 02). Automatic outlier detection for time series: An application to sensor data. *Mechanical Systems and Signal Processing 11*(2), 137–154.

Bloch, H. and F. Geitner (1983). *Machinery Failure Analysis and Troubleshooting*. Practical Machinery Management for Process Plants Series. Gulf.

British Standards Institute Staff and British Standards Institution (2001). *Maintenance terminology*. Number CEN (European Committee for Standardization). British Standards Institution.

Caesarendra, W. and T. Tjahjowidodo (2017). A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing. *Machines 5*(4), 21.

Chen-Fu Chien, Shi-Lin Chen, and Yih-Shin Lin (2002). Using Bayesian network for fault location on distribution feeder. *IEEE Transactions on Power Delivery 17*(3), 785–793.

Cheng, S., M. H. Azarian, and M. G. Pecht (2010). Sensor systems for prognostics and health management. *Sensors 10*(6), 5774–5797.

Ćwikła, G. (2013). Methods of Manufacturing Data Acquisition for Production Management - A Review. *Advanced Materials Research 837*, 618–623.

Dalpiaz, G., A. Rivola, and R. Rubini (2000). Effectiveness and sensitivity of vibration processing techniques for local fault detection in gears. *Mechanical Systems and Signal Processing 14*(3), 387–412.

Das, A., J. Maiti, and R. Banerjee (2012). Process monitoring and fault detection strategies: a review. *International Journal of Quality & Reliability Management 29*(7), 720–752.

Dekys, V., P. Kalman, P. Hanak, P. Novak, and Z. Stankovicova (2017). Determination of Vibration Sources by Using STFT. *Procedia Engineering 177*, 496–501.

Deshpande, A. P. and S. C. Patwardhan (2008). Online Fault Diagnosis in Nonlinear Systems Using the Multiple Operating Regime Approach. *Industrial & Engineering Chemistry Research 47*(17), 6711–6726.

Dhini, A. (2016). Review on Some Multivariate Statistical Process Control Methods for Process monitoring. In *Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management*, Kuala Lumpur, pp. 754–759.

Ekwaro-Osire, S., A. C. Gonçalves, and F. M. Alemayehu (2017). *Probabilistic Prognostics and Health Management of Energy Systems* (1 ed.). Springer International Publishing.

Elattar, H. M., H. K. Elminir, and A. M. Riad (2016). Prognostics: a literature review. *Complex & Intelligent Systems 2*(2), 125–154.

Ellis, B. A. (2008). The challenges of condition based maintenance. *The Jethro Project*, 1–5.

Gebraeel, N. Z., M. A. Lawley, R. Li, and J. K. Ryan (2005). Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions (Institute of Industrial Engineers) 37*(6), 543–557.

Georgoulas, G., T. Loutas, C. D. Stylios, and V. Kostopoulos (2013). Bearing fault detection based on hybrid ensemble detector and empirical mode decomposition. *Mechanical Systems and Signal Processing 41*(1-2), 510–525.

Gorjian, N., L. Ma, M. Mittinty, P. Yarlagadda, and Y. Sun (2009). A review on reliability models with covariates. In *Proceedings of the 4th World Congress on Engineering Asset Management*, Number September, pp. 28–30.

Harrou, F., F. Kadri, S. Chaabane, C. Tahon, and Y. Sun (2015). Improved principal component analysis for anomaly detection: Application to an emergency department. *Computers and Industrial Engineering 88*, 63–77.

Heng, A., S. Zhang, A. C. Tan, and J. Mathew (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing 23*(3), 724–739.

Hotelling, H. (1933). Analysis of a complex of statistical variables into Principal Components. Jour. Educ. Psych., 24, 417-441, 498-520. *The Journal of Educational Psychology 24*(6), 417–441.

ISO-13374-1 (2003). Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part 1: General guidelines. Technical report.

ISO-13381-1 (2015). Condition monitoring and diagnostics of machines - Prognostics - Part 1: General guidelines. Technical report, International Organization for Standardization.

J. Duphily, R. (2014). Root Cause Investigation Best Practices Guide. Technical report.

Jackson, J. E. and G. S. Mudholkar (1979). Control procedures for residuals associted with principal component analysis. *Technometrics 21*(3), 341–349.

Jardine, A. K., D. Lin, and D. Banjevic (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing 20*(7), 1483–1510.

Javed, K., R. Gouriveau, and N. Zerhouni (2017). State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mechanical Systems and Signal Processing 94*, 214–236.

Kang, H., Y. Qi, G. Liu, J. Liu, and C. Li (2012). Application of the ARMA model in non-stationary vibration signals. *Proceedings of 2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, ICQR2MSE 2012* (2), 751–754.

Kankar, P. K., S. C. Sharma, and S. P. Harsha (2011). Fault diagnosis of ball bearings using continuous wavelet transform. *Applied Soft Computing Journal 11*(2), 2300–2312.

Kerkhof, P. V. D., J. Vanlaer, G. Gins, and J. F. M. V. Impe (2013). Contribution plots for Statistical Process Control : analysis of the smearing-out effect. *European Control Conference (ECC)*, 1–6.

Kumar, D. and B. Klefsjo (1994). Proportional hazards model: a review. *Reliability Engineering and System Safety 44*(2), 177–188.

LANA Labs (2017). Root Cause Analysis - Now automated in LANA.

Lebold, M., K. Reichard, P. Hejda, J. Bezdicek, and M. Thurston (2002). A framework for next generation machinery monitoring and diagnostics. *Proceedings of the Society for Machinery Failure Prevention Technology Meeting*, 115–126.

Lee, J. Y. (2015). Variable short-time Fourier transform for vibration signals with transients. *JVC/Journal of Vibration and Control 21*(7), 1383–1397.

Lei, Y., N. Li, L. Guo, N. Li, T. Yan, and J. Lin (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing 104*(May), 799–834.

Leys, C., C. Ley, O. Klein, P. Bernard, and L. Licata (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology 49*(4), 764–766.

Lowry, C. A., W. H. Woodall, C. W. Champ, S. E. Rigdon, C. A. Lowry, and W. H. Woodall (1992). A Multivariate Exponentially Weighted Moving Average Control Chart. *Technometrics 34*(1), 46–53.

Mansouri, M., M. Z. Sheriff, R. Baklouti, and M. N. Karim (2015). Statistical Fault Detection of Chemical Process - Comparative Studies. *Journal of Chemical Engineering & Process Technology 7*(1), 1–10.

Marton, I., A. Sánchez, S. Carlos, and S. Martorell (2013). Application of data driven methods for condition monitoring maintenance. In *Proceedings of the 4th IEEE Conference on Prognostics and System Health Management (PHM)*, Volume 33, pp. 301–306.

Mason, R., N. Tracy, and J. Young (1997). A practical approach for interpreting multivariate T2 control chart signals. *Journal of Quality Technology 29*(4), 396–406.

Miranda, A. A., Y. A. Le Borgne, and G. Bontempi (2008). New routes from minimal approximation error to principal components. *Neural Processing Letters 27*(3), 197–207.

Mobley, R. K. (2002). 18 - world-class maintenance. In R. K. Mobley (Ed.), *An Introduction to Predictive Maintenance* (Second Edition ed.)., Plant Engineering, pp. 394 – 433. Burlington: Butterworth-Heinemann.

Pecht, M. and J. Gu (2009). Physics-of-failure based prognostics for electronic products. *Transactions of the Institute of Measurement and Control 31*(3-4), 309–322.

Peng, Y., M. Dong, and M. J. Zuo (2010). Current status of machine prognostics in condition-based maintenance: A review. *International Journal of Advanced Manufacturing Technology 50*(1-4), 297–313.

PwC (2016). Industry 4 . 0 : Building the digital enterprise. Technical report.

Si, X.-S., Z.-X. Zhang, and C.-H. Hu (2017). *Data-Driven Remaining Useful Life Prognosis Techniques*.

Thukaram, D., H. Khincha, and H. Vijaynarasimha (2005). Artificial Neural Network and Support Vector Machine Approach for Locating Faults in Radial Distribution Systems. *IEEE Transactions on Power Delivery 20*(2), 710–721.

Vachtsevanos, G., F. L. Lewis, M. Roemer, A. Hess, and B. Wu (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems* (1 ed.). Hoboken: John Wiley & Sons, Inc.

Venkatasubramanian, V., R. Rengaswamy, and S. N. Ka (2003). A review of process fault detection and diagnosis Part I : Quantitative model-based methods. *Computers & Chemical Engineering 27*(3), 293–311.

Villegas, T., M. Fuente, and M. Rodríguez (2010). Principal component analysis for fault detection and diagnosis. experience with a pilot plant. *Proceedings of the 9th WSEAS International conference on computational intelligence, man-machine systems and cybernetics*, 147–152.

Vogl, G. W., B. A. Weiss, and M. Helu (2016). A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing*, 1–17.

Wang, G. (2016). Health index synthetization and remaining useful life estimation for turbofan engines based on run-to-failure datasets. *Maintenance and Reliability 18*(4), 621–631.

Wang, J., C. Li, S. Han, S. Sarkar, and X. Zhou (2017). Predictive maintenance based on event-log analysis: A case study. *IBM Journal of Research and Development 61*(1).

Westerhuis, J. A., S. P. Gurden, and A. K. Smilde (2000). Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems 51*(1), 95–114.

Xu, L. X. L., M.-Y. C. M.-Y. Chow, and L. Taylor (2006). Using the Data Mining Based Fuzzy Classification Algorithm for Power Distribution Fault Cause Identification with Imbalanced Data. *2006 IEEE PES Power Systems Conference and Exposition 22*(1), 164–171.

Yan, J., Y. Meng, L. Lu, and L. Li (2017). Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance. *IEEE Access 5*, 23484–23491.

Yan, W., H. Qiu, and N. Iyer (2008). Feature extraction for bearing prognostics and health management (PHM)-a survey. *Air Force Research Laboratory, Tech. Rep.*

Yongxiang, L., S. Jianming, W. Gong, and L. Xiaodong (2016). A data-driven prognostics approach for RUL based on principle component and instance learning. *2016 IEEE International Conference on Prognostics and Health Management, ICPHM 2016*.

Yu, Q. D., Y. L. Li, and J. Wang (2014, 7). Analysis on weak signals with short-time fourier transform method. In *Mechatronics Engineering, Computing and Information Technology*, Volume 556 of *Applied Mechanics and Materials*, pp. 5003–5005. Trans Tech Publications.

Zhang, Y., H. Zuo, and F. Bai (2013). Classification of fault location and performance degradation of a roller bearing. *Measurement: Journal of the International Measurement Confederation 46*(3), 1178–1189.

Appendix A

Full framework

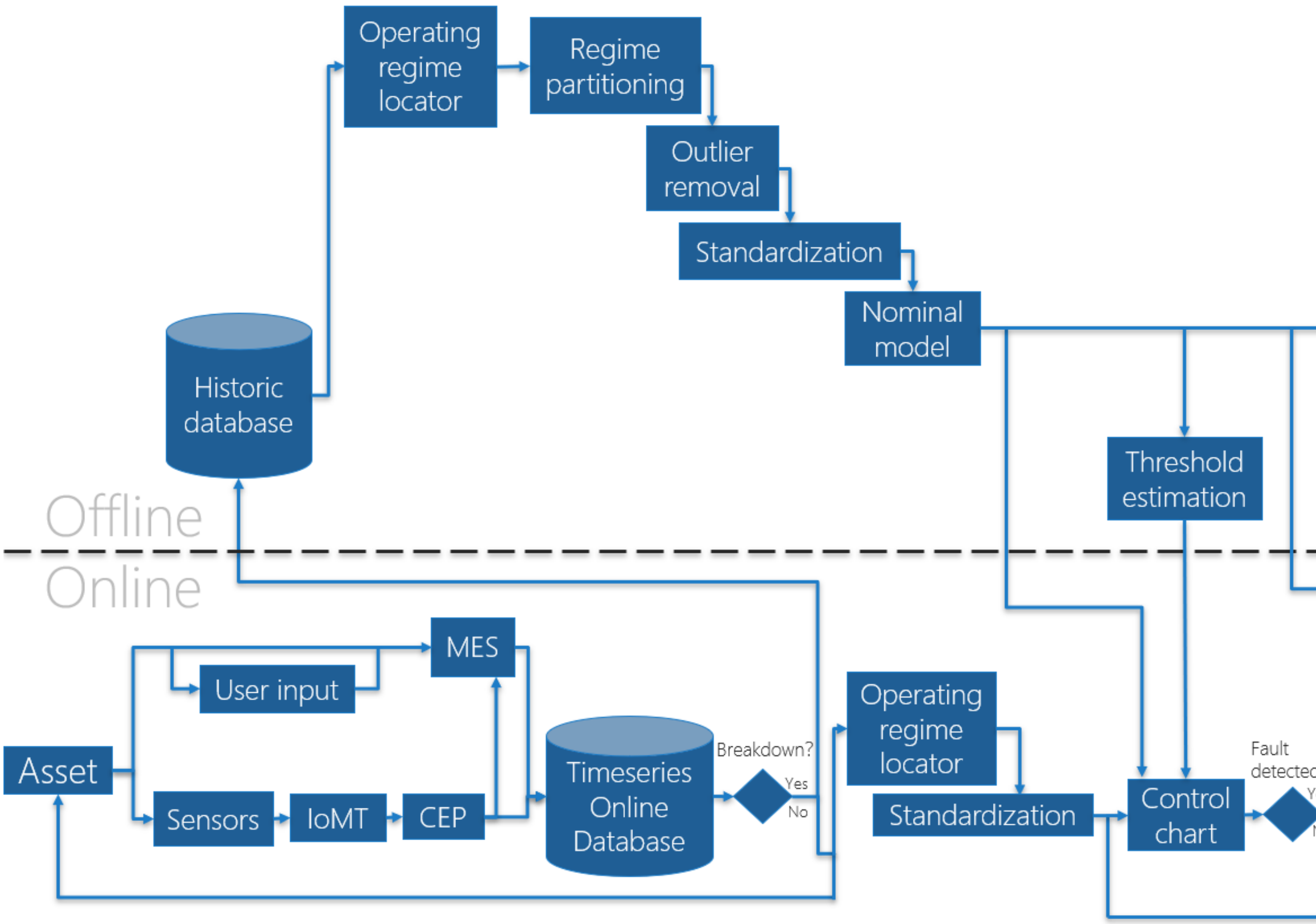

Figure A.1: Full framework of the proposed Predictive Maintenance tool

Appendix B

Turbofan Engine Degradation Simulation Data Set

This dataset is the result of a simulation study of engine degradation which was carried out by Prognostics CoE at NASA Ames using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tool. Four different sets were simulated under different combinations of operational conditions and fault modes by recording several sensor channels that characterize fault evolution. However, the present work focused on the set that contemplates a single set of operational conditions and a single fault mode.

That dataset consists of multiple multivariate time series with each representing the degradation trajectory of a different unit from a fleet of engines of the same type. The dataset is further divided into training set and test set. Both contain 100 degradation trajectories and their attributes are: the unit number, the time in cycles, 3 operational settings and 21 sensor readings. A complementary dataset containing the real time-to-failure of each test unit is also provided, which can be used for assessing the results of different prognostics approaches.

Figure B.1 depicts a subset of the entire the Turbofan Engine Degradation Simulation Data Set. Column $V1$ indicates the unit number, $V2$ indicates the time in cycles, columns $V3$ through $V5$ represent the operational settings and columns $V6$ through $V26$ the sensor readings.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26
1	1	1	-0.0007	-4e-04	100	518.67	641.82	1589.70	1400.60	14.62	21.61	554.36	2388.06	9046.19	1.3	47.47	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100	39.06	23.4190
2	1	2	0.0019	-3e-04	100	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04	9044.07	1.3	47.49	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100	39.00	23.4236
3	1	3	-0.0043	3e-04	100	518.67	642.35	1587.99	1404.20	14.62	21.61	554.26	2388.08	9052.94	1.3	47.27	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100	38.95	23.3442
4	1	4	0.0007	0e+00	100	518.67	642.35	1582.79	1401.87	14.62	21.61	554.45	2388.11	9049.48	1.3	47.13	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100	38.88	23.3739
5	1	5	-0.0019	-2e-04	100	518.67	642.37	1582.85	1406.22	14.62	21.61	554.00	2388.06	9055.15	1.3	47.28	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100	38.90	23.4044
6	1	6	-0.0043	-1e-04	100	518.67	642.10	1584.47	1398.37	14.62	21.61	554.67	2388.02	9049.68	1.3	47.16	521.68	2388.03	8132.85	8.4108	0.03	391	2388	100	38.98	23.3669
7	1	7	0.0010	1e-04	100	518.67	642.48	1592.32	1397.77	14.62	21.61	554.34	2388.02	9059.13	1.3	47.36	522.32	2388.03	8132.32	8.3974	0.03	392	2388	100	39.10	23.3774
8	1	8	-0.0034	3e-04	100	518.67	642.56	1582.96	1400.97	14.62	21.61	553.85	2388.00	9040.80	1.3	47.24	522.47	2388.03	8131.07	8.4076	0.03	391	2388	100	38.97	23.3106
9	1	9	0.0008	1e-04	100	518.67	642.12	1590.98	1394.80	14.62	21.61	553.69	2388.05	9046.46	1.3	47.29	521.79	2388.05	8125.69	8.3728	0.03	392	2388	100	39.05	23.4066
10	1	10	-0.0033	1e-04	100	518.67	641.71	1591.24	1400.46	14.62	21.61	553.59	2388.05	9051.70	1.3	47.03	521.79	2388.06	8129.38	8.4286	0.03	393	2388	100	38.95	23.4694
11	1	11	0.0018	-3e-04	100	518.67	642.28	1581.75	1400.64	14.62	21.61	554.54	2388.05	9049.61	1.3	47.15	521.40	2388.01	8140.58	8.4340	0.03	392	2388	100	38.94	23.4787
12	1	12	0.0016	2e-04	100	518.67	642.06	1583.41	1400.15	14.62	21.61	554.52	2388.09	9049.37	1.3	47.18	521.80	2388.02	8134.25	8.3938	0.03	391	2388	100	39.06	23.3660
13	1	13	-0.0019	4e-04	100	518.67	643.07	1582.19	1400.83	14.62	21.61	553.44	2388.12	9046.82	1.3	47.38	521.85	2388.08	8128.10	8.4152	0.03	393	2388	100	38.93	23.2757
14	1	14	0.0009	0e+00	100	518.67	642.35	1592.95	1399.16	14.62	21.61	554.48	2388.09	9047.37	1.3	47.44	521.67	2388.00	8134.43	8.3964	0.03	393	2388	100	39.18	23.3826
15	1	15	-0.0018	-3e-04	100	518.67	642.43	1583.82	1402.13	14.62	21.61	553.64	2388.11	9052.22	1.3	47.30	522.50	2388.08	8127.56	8.4199	0.03	391	2388	100	38.99	23.3500
16	1	16	0.0006	5e-04	100	518.67	642.13	1587.98	1404.50	14.62	21.61	553.94	2388.05	9049.34	1.3	47.24	521.49	2388.07	8136.11	8.3936	0.03	392	2388	100	38.97	23.4550
17	1	17	0.0002	2e-04	100	518.67	642.58	1584.96	1399.95	14.62	21.61	553.80	2388.06	9054.92	1.3	47.12	521.89	2388.04	8137.27	8.4542	0.03	392	2388	100	38.81	23.3319
18	1	18	-0.0031	-1e-04	100	518.67	642.62	1591.04	1396.12	14.62	21.61	554.20	2388.05	9049.55	1.3	47.21	521.76	2388.07	8132.73	8.4028	0.03	392	2388	100	38.89	23.3987
19	1	19	0.0032	-3e-04	100	518.67	641.79	1587.56	1400.35	14.62	21.61	554.18	2388.04	9053.99	1.3	47.40	521.89	2388.03	8129.13	8.4321	0.03	391	2388	100	38.80	23.3464
20	1	20	-0.0037	1e-04	100	518.67	643.04	1581.11	1405.23	14.62	21.61	554.81	2388.05	9045.90	1.3	47.22	522.07	2388.02	8129.71	8.4210	0.03	392	2388	100	39.03	23.4220
21	1	21	-0.0012	1e-04	100	518.67	642.37	1586.07	1398.13	14.62	21.61	554.08	2388.11	9048.15	1.3	47.15	522.42	2388.08	8134.02	8.4049	0.03	392	2388	100	39.09	23.3101

Figure B.1: Subset of Turbofan Engine Degradation Simulation Data Set

Appendix C

Reliability Centered Maintenance approach

From Reliability Centered Maintenance it is known that the hazard function $h(t)$ is the ratio of the probability density function $f(t)$ to the survival function $R(x)$, given by

$$h(t) = \frac{f(t)}{R(t)} = \frac{f(t)}{1 - F(t)} \tag{C.1}$$

where $F(t)$ is the cumulative distribution function.

By plotting the histogram of the times of failure of the 100 units comprising the training set, $f(t)$ can be obtained. Figure C.1 depicts this histogram. Additionally, $R(t)$ can also be obtained by simply taking the number of units that are still in operation at time instant t.

With this information it is possible to calculate the expected time-to-failure of each of the 100 test units by

$$TTF_t = \frac{\int_t^T (t \cdot f(t))\mathrm{d}t}{R(t)} - t \tag{C.2}$$

where t represents the current time instant and T represents the duration of the longest lives training unit.

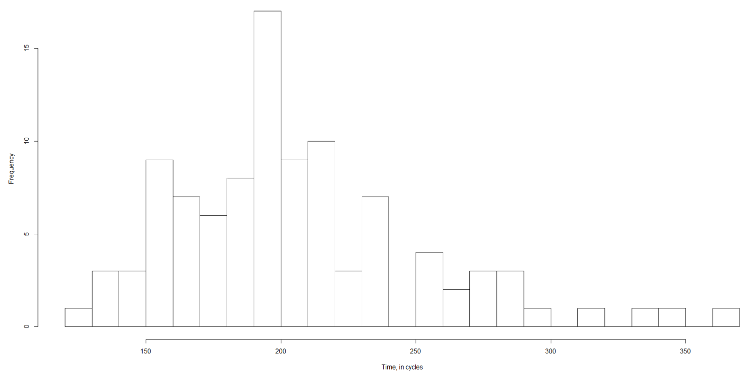

Figure C.1: Histogram of the failure times of the training set units