University of Modena and Reggio Emilia

DOCTORAL SCHOOL
INDUSTRIAL INNOVATION ENGINEERING

XXXII Period

A condition based monitoring framework for Independent Carts System

Author:
Jacopo Cavalaglio Camargo Molano

Supervisor:
Prof. Marco Cocconcelli

Coordinator of the Doctoral Course:
Prof. Franco Zanbonelli

Acknowledgements

I would like to thank my supervisor, Professor Marco Cocconcelli for his friendly guidance and expert advice, which have let me to grow as a research scientist and as a person. This PhD would not have been achievable without his guidance and constant feedback.

Many thanks to Tetra Pak Packaging Solution for the possibility given to me of working in a highly stimulating and fascinating working environment, special thanks to Mr. Davide Borghi, Mr. Stefano Rossi, Mr. Luca Capelli and Mr. Stefano Flore.

I gratefully acknowledge Siemens PLMS Software for the pleasure to work for six months in the MBS simulation group.

Thanks to my family for their constant support and to my mates for their natural joviality.

Abstract

The objective of this work is the condition monitoring of Independent Carts System with particular attention to bearings. The Independent Cart Conveyor System is a promising technology that could replace rotary driven chains and belts in the field of automatic machines. This system combines the benefits of servomotors with the advantages of linear motors. It consists of a close path made up of modular linear motors having a curved or a straight shape that control a fleet of carts independently. Each cart is placed along the motors and it is connected, through bearings, to a rail set on the motors themselves. A possible problem can rise with the use of this technology: with the demand of a high production rate, the number of movers necessary in the machine increases and consequently even the number of bearings increases. In this way the high number of rolling bearings reduces the Mean Time Before Failure (MTBF) of the whole machine, but at the same time, thanks to the independent control and the independent monitoring of each cart, it is possible to implement condition monitoring strategies for each cart. The condition monitoring of these elements is challenging for the non-stationary working conditions of variable load and speed profiles. The thesis deals with the problem of the development of a condition monitoring framework for this system from different points of view. About hardware, a new technique for the synchronization between PLCs of different vendors used for the control of this system has been developed. Moreover, bearing stiffness has been evaluated through experimental campaigns and advance computational methods. In order to get a 360-degree view of the possible solutions of this problem, data-driven and model-based condition monitoring techniques have been applied.
As regards data-driven, machine learning techniques for fault detection have been used on the basis of an experimental campaign on a specific machine application, as well as a new feature for the prediction of bearing faults has been studied.
As regards model-based, a model of the vibration signals produced by the carts with an arbitrary motion profile has been carried out. Moreover, the whole dynamics of the system has been taken into account by means of a multibody modelling of the cart, the bearings and the rail.
Both models consider the variable motion profile, the shape of the conveyor path, the mechanical design of the cart, the load variation and the type of fault on the groove ball bearings. The models are scalable and modular in order to test different configurations of the system with different working parameters and both models have been validated by means of the comparison between the simulation results and the system variables recorded during experimental campaigns.

Sommario

L'obiettivo di questo lavoro è la diagnostica di Sistemi a Carrelli Indipendenti con particolare attenzione ai cuscinetti. Il Sistema a Carrelli Indipendenti è una tecnologia innovativa e promettente nell'ambito delle macchine automatiche, che in alcuni casi può sostituire i tipici sistemi di trasporto basati su catene e cinghie guidate da motori rotativi. Il Sistema a Carrelli Indipendenti combina i vantaggi dei servomotori rotativi con quelli dei motori lineari. Esso consiste in una serie di motori lineari modulari di forma curva o rettilinea, che, combinati insieme, realizzano un circuito chiuso. I motori controllano una flotta di carrelli che sono tra loro indipendenti e questa caratteristica rende il sistema flessibile ad ogni tipo di prodotto e compito. Ognuno di questi carrelli è posizionato lungo i motori ed è collegato ad un binario fisso attraverso una serie di cuscinetti. Questo sistema può presentare un problema nel caso in cui vengano utilizzati numerosi carrelli, poiché in questo caso anche il numero dei cuscinetti aumenterebbe. L'elevato numero di cuscinetti riduce il Mean Time Before Failure (MTBF) dell'intera macchina, ma allo stesso tempo, grazie al fatto che ogni carrello è svincolato l'uno dall'altro, è possibile monitorare lo stato di salute di ogni carrello in maniera indipendente. La realizzazione di un sistema di condition monitoring per questi macchinari risulta stimolante anche se impegnativo, in quanto le condizioni di lavoro dei carrelli sono altamente non stazionarie per la variabilità dei profili di carico e velocità. Tale studio tratta il problema dello sviluppo di un sistema di condition monitoring per questa tecnologia, che viene affrontato da diversi punti di vista.
Per quanto riguarda l'hardware, è stata realizzata una nuova tecnica per la sincronizzazione dei motion task fra i PLC di differenti costruttori, che vengono usati per il controllo dei Sistemi a Carrelli Indipendenti. Inoltre, sono stati eseguiti esperimenti ed usati metodi di calcolo avanzati per la valutazione della rigidezza dei cuscinetti. Per avere una panoramica completa dei possibili metodi di monitoraggio, sono state utilizzate sia tecniche data-driven che model-based per il rilevamento di guasti nel sistema.
Per quanto riguarda i metodi data-driven, sono stati utilizzati algoritmi di machine learning per la identificazione di danni, così come sono state studiate nuove feature per la prognostica. Per quanto si riferisce al model-based, è stato sviluppato un modello che simula i segnali vibratori prodotti dai Sistemi a Carrelli Indipendenti con profilo di moto arbitrario.
Per considerare l'intera dinamica del sistema, è stato realizzato un modello multibody del carrello, dei cuscinetti e del binario. Entrambi i modelli realizzati prendono in considerazione profili di moto variabili, varie forme del circuito su cui i carrelli

possono muoversi, la meccanica dei carrelli, la variazione dei carichi agenti e differenti tipi di danno nei cuscinetti. Per poter considerare diverse configurazioni del sistema, entrambi i modelli sono scalabili e modulari. Essi sono stati validati attraverso la comparazione tra i dati simulati e i dati reali rilevati attraverso una campagna sperimentale.

Contents

Chapter 1

Motivation and objectives

1.1 Introduction

This work is focused on the development of a condition monitoring framework for a manufacturing machine called Independent Carts System. The aim is to identify the healthy conditions of the machine and the damages on the bearings placed in the machine itself.
Independent Cart Conveyor System is one of the most promising technology in automation industries. It combines the benefits of servomotors with the advantages of linear motors. It consists of a close path made up of modular linear motors having a curved or a straight shape that control a fleet of carts independently. Each cart is placed along the motors and it is connected, through rolling bearings, to a rail set on the motors themselves. A possible problem can rise with the use of this technology: with the demand of a high production rate, the number of movers necessary in the machine increases and consequently even the number of bearings increases. In this way, the high number of rolling bearings reduces the Mean Time Before Failure (MTBF) of the whole machine, but at the same time, thanks to the independent control and the independent monitoring of each cart, it is possible to implement condition monitoring strategies for each cart. Several are the producers of Independent Carts System but each of them uses a different PLC for the control of the internal tasks. This fact can reduce the flexibility in the machine architecture, which is a great limitation from the point of view of a designer. Moreover, even the possible condition monitoring architectures are limited, for example as regards the data acquisition and the integration of sensors in the machine. In order to overcome this problem, new methods for the synchronization of axes between PLCs and iPCs of different vendors has been analysed and developed.
Another problem that has been considered is the difficulty in having a large amount of historical data on Independent Carts System.
This is due to the fact that the machines can have different shapes, different cart geometry, different load profiles and, what is more, the carts of the same machine can be subject to very different working conditions; moreover, there is no data available on Independent Cart System used in manufacturing plants. In order to overcome this problem, different experimental campaigns have been carried out by

simulating possible bearing damages and physical models have been developed to simulate the behaviour of the system in different working conditions and for different configurations of the Independent Cart System.

1.2 Outline of the thesis

This paper is organized as follows:

- In Chapter 2 the condition monitoring methodologies, techniques and philosophy are investigated. A global architecture of a condition monitoring system in a packaging plant is presented considering the creation of a new condition monitoring framework for Independent Cart System. Independent Cart System is described in details focusing on the crucial aspects regarding condition monitoring.
- In Chapter 3 the interfacing problem between Independent Carts System and different industrial controllers is faced. In order to overcome the interconnectivity problem, two methods are designed, developed and verified.
- Chapter 4 presents a methodology workflow for the development of a fault detection system for Independent Cart System by means of a Data-driven approach.
- Chapter 5 deals with a Model-based approach for the development of a fault detection system. Two different models are presented, the former consists in a signal formulation of the system, while the latter is a multibody model. Both the models have been tested and validated.
- In Chapter 6 the different condition monitoring methodologies taken into account are illustrated and the prospective for future improvements is presented.

Chapter 2

Introduction

2.1 Maintenance

Good product design is a key factor for product reliability, but also correctly designed products deteriorate in time because of normal wear and use in the working environment. For this reason, maintenance strategies are essential for assuring satisfactory reliability levels and improving the productivity of machines through the reduction in financial costs [1] .
Maintenance is defined as "the combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function" [2].
In process industries, which are more vulnerable to plant downtime and to production maximization, maintenance has ever had a remarkable influence on the cost reduction and on the reliability and continuity of the process. For example, as regards the Nigerian electric-power industry, the costs of maintenance are over 70 % of the total spending [3].
In manufacturing industries, maintenance has been defined as a key factor for the reduction in the process costs only in the last decade. This is due to the rise of lean production that increases the requirement of more efficient and reliable machines with a higher productivity and a lower vulnerability to disturbance. In order to maximize the machine productivity, companies have to guarantee both the cost reduction, such as downtime, spare parts and machine waste, and the minimization of maintenance activities. To achieve the new targets, prognostic and health management (PHM) has acquired a key role in the machine design and logistic support. PHM combines the use of methods for predicting the time a device or a system no longer performs as desired with maintenance planning and decision-making support. This chapter describes the most used typologies of maintenance strategies, their differences and the advantages and disadvantages of each typology.

2.1.1 Corrective Maintenance

Corrective maintenance, also called breakdown maintenance, is the earliest maintenance technique used in industries. According to this strategy, the equipment runs

until a failure occurs, subsequently the damaged part is changed or repaired [4]. The advantages of this type of maintenance are that each component is used for its complete life, maintenance is serviced only in a period of time and not during the working of machine, the cost of labour is low. The disadvantages are that maintenance is unplanned so the continuity of the process is not reliable; moreover, if a catastrophic failure occurs, the damage of one of the components can compromise the entire machine lifetime or can provoke the failure of a second equipment. In this case, the repair costs are higher and even the cost of labour can be higher.

2.1.2 Predicted Maintenance

Predicted maintenance, also called time-based maintenance, requires that maintenance operations are performed at regular intervals of time independently of the equipment health conditions [5].
In this case, it is important to highlight that the time between failures (TBF) is a statistical variable and it can be estimated through a population of machines. In order to have great confidence in predicted maintenance, a large population of machines is necessary.
This maintenance strategy is very conservative because, when maintenance is serviced, there is no certainty about the failure of a component. The advantages of this approach are the reduction in unplanned stops and the increase of the machine lifetime. The disadvantages are that the "infant mortality failure" is not predicted, the service equipment is not used for its complete life and it is necessary to stop the production at regular intervals.

2.1.3 Condition Based Maintenance

Condition Based Maintenance (CBM), also called preventive maintenance, is based on the constant monitoring of critical functions or components; it allows to predict machine failures and to initiate maintenance before the failure occurs [6]. The key features of this methodology are diagnostics and prognostics. Diagnostics consists in the detection, isolation and identification of a fault, while prognostics consists in the prediction of a fault before its occurrence. The advantages of this strategy are the maximization of performances with the reduction of the machine downtime, the reduction of the waste due to unplanned stops and the optimization of the warehouse for the spare parts. In the last years, CBM has also focused on the prognostics of remaining useful life (RUL) of the equipment and on the optimization of maintenance strategy for the failure of each component or functional failures. This type of strategy is called CBM+ [7] and it is used in combination with maintenance methods as reliability, availability, maintainability and safety (RAMS) [8] or reliability centered maintenance (RCM) [9].
In order to use CBM/CBM+, several sensors are necessary to monitor the state of the equipment and this makes the initial investment cost increase. This strategy

uses a new decision architecture for the maintenance policy that needs an improvement in human-machine skills to carry out collaborative actions between condition monitoring system and human operators, which involves a socio-cyber-physical system (SCPS)[7].
For this reason, a high training level from the top (i.e., management) to the base (i.e., stakeholders, service engineers) and a very efficient communication network are necessary. The drawback of this strategy is the high initial investment necessary to build a CBM framework in which different company sectors are involved.
Worden and Dulieu-Barton [10] describe the identification problem as a hierarchical structure where each stage represents a higher level of the damage knowledge. The levels are the following:

1. **Detection**: the method gives a qualitative indication that a damage might be present in the structure.
2. **Localisation**: the method gives information about the probable position of the damage.
3. **Classification (or identification)**: the method gives information about the type of damage.
4. **Assessment**: the method gives an estimate of the extent of the damage.
5. **Prediction (or prognostics)**: the method offers information about the safety of the structure, for example it estimates the residual life.

The first four levels are included in CBM, while the last one is related to prognostics. Prognostics is the engineering discipline that studies the methods for the estimation of the remaining useful life (RUL) of a component.

2.1.3.1 Condition Based Maintenance in manufacturing plants

This chapter describes general condition monitoring techniques applicable for the diagnostics in industrial plants. In order to be more pragmatic, a packaging industrial plant has been taken into consideration in order to show a real architecture with more details. This work focuses on the study and the implementation of a condition monitoring framework for Independent Carts System inside the above-mentioned packaging industrial plant.

2.2 Condition Based Maintenance architecture for manufacturing plants

The purpose of the condition monitoring application to manufacturing plants is to reduce unexpected breakdowns in order to, in turn, increase machine up-time (avoiding unplanned stoppages by predicting failures), reduce waste (for the same reason) and optimize operational costs (with optimal maintenance tactics based on

prediction). This is achieved through a constant monitoring of critical functions to predict failures with the possibility of initiating maintenance before the failure occurs, through regular alerts and inspections. With preventive maintenance, the commitment to reliability is taken to the next level by predicting failures before they occur. Real-time monitoring of critical areas of the equipment is used to find deviations in machine functions that could lead to pauses in machine activity or breakdowns. In this process, the knowledge of critical functions, expert analyses, alerts and a skilled staff are instrumental and fundamental. The design of the condition monitoring system (CMS) directly depends on the plant and the functions of each system since the causes and effects of a failure are different.
Nevertheless, the high-level architecture of the condition monitoring system dealt with in this work can be extended to any manufacturing company. At the basis of an effective condition-based monitoring system there is an initial failure mode, effects and criticality analysis (FMECA). FMECA analysis is a critical and powerful tool, developed by reliability engineers in the late 1950s, to highlight failure modes with relatively high probability and severity of consequences, allowing remedial efforts to be directed to the point where they will produce the greatest benefits. These benefits are precisely quantified in terms of saved costs and they are the best incentive to introduce condition monitoring systems in industries. This thesis does not cover FMECA analysis, since it should have already been performed in order to identify the most critical components. The aim of this chapter is to give hints to the reader in order to build condition monitoring systems. This process ideally starts after FMECA analysis. All the mechanical components shown in the remaining part of the chapter are the results of an in-depth FMECA analysis that took more than one year to be completed. Several books on reliability cover the foundations of FMECA analysis. The interested reader could start, for example, with the work by Birolini [11].
At the end of the condition monitoring process, every industrial plant must have a performance management center, i.e., a team of data-scientists, managers, reliability engineers and skilled service technicians, constantly updated on the status of the fleet of monitored components. The team schedule the interventions on the basis of a preventive maintenance policy, building a database of all the service actions and computing statistics on the reliability of the components. The team are ready to intervene in the event of an alarm by the monitoring system and condition monitoring algorithms based on the new data from the field, which are constantly updated. The performance management center implements everything is necessary for the correct management of the reliability of the system. For example, the uncertainty quantification is fundamental to properly plan (preventive) maintenance policies. A list of these actions is out of the scope of the present work. The interested reader could find details, for example, in the work by O'Connor and Kleyner [12]. The processing of the collected data is divided into three parts: pre-processing that is carried out in the customer's factory, cloud-processing that is performed in the cloud, and post-processing for the management of the critical states. The flowchart of data and information throughout the process is outlined in Fig. 2.1.

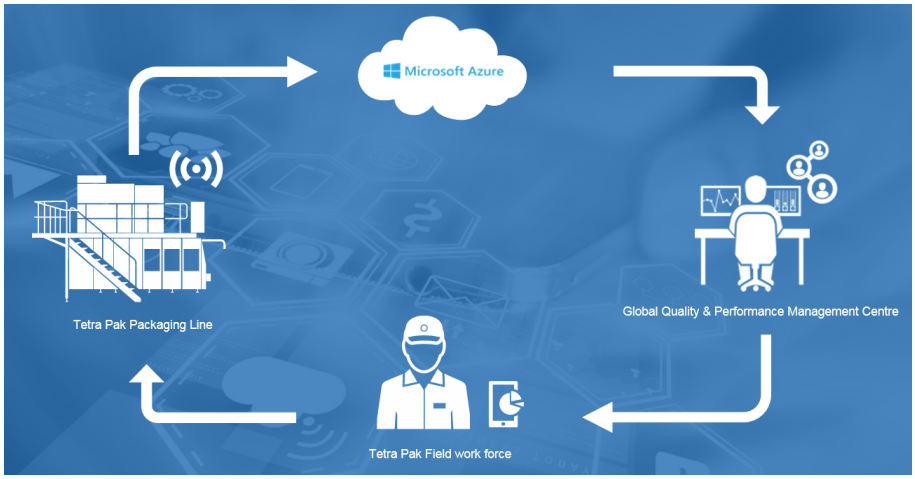


Figure 2.1: Outline of condition monitoring architecture.

2.2.1 Data acquisition

The data acquisition (DAQ) step involves the setup of the sensors on the machine, the acquisition device and a central unit that manages the data logging. The sampling of the data can be implemented in two ways:

- Continuous Condition Monitoring: sensors are recorded continuously. This sampling policy is recommended for those critical components with a high impact on the costs and a short time-to-failure.
- Periodic Condition Monitoring: sensors are recorded at scheduled time intervals. This policy is particularly suitable for components with a medium–high time-to-failure.

In a condition monitoring system with large-scale applications there are inevitable delays related to information management: the acquisition of data, the local pre-processing, the transfer to the cloud, the subsequent post-processing, the feedback from the data analysts and the service support in fixing the problem. As a consequence, a robust condition monitoring project should work mainly on periodic sampled data, keeping a margin of time for the task processing. Nevertheless, critical components can be taken into account reserving computational slots, priorities, or managing the data collection on the spot for the prompt feedback of the service engineer. The periodic condition monitoring gives a margin of time to collect data from a fleet of sensors, one-by-one, limiting the stream of data and the computational resources. This architecture of the sampling policy allows to easily update the extension of the number of sensors. The acquisition setup —i.e., the sampling frequency and the acquisition time— depends on a specific sensor and must be determined on the basis of the specific processing defined in the development stage of the condition monitoring system. For example, temperature changes at intervals of minutes, while vibrations need to be acquired thousands of times per second. The data can be divided into two main classes: on-line data and off-line data as detailed below.

2.2.1.1 On-Line Data

In this work, the term "on-line data" refers to informative data acquired in the working conditions of the machine. This is in contrast with "off-line data", which are collected during events independent of the working conditions. For our purposes, the "on-line data" are data collected by specific sensors in order to measure the state variables of the system. The choice of the type of sensors, their placement and the schedule of the data collection require the knowledge of the process, a bibliographic survey and an analysis of the impact of sensor costs on the process. Generally, the sensors can be divided into three main classes:

- Multi-purpose external sensors: they are the most used sensors for condition monitoring. They can be applied to different components (multi-purpose), measuring the effects of impacts or events in time domain and include, for example, accelerometers or external temperature sensors. These sensors are not usually present in the machine and represent an extra cost for maintenance.

- Specific external sensors: they are used for specific measurements in specific parts of the machine. Sometimes multi-purpose sensors cannot be used because of the impossibility of installation, such as environmental conditions or some possible mechanical interference with moving parts during the process. Sometimes a specific measurement is needed in a very limited but critical part of the plant, for example chemical analysis. These sensors are not usually present in the machine and represent an extra cost for maintenance. Moreover, the specificity of the measurement implies a higher cost of the sensor with respect to a multi-purpose sensor.

- Embedded sensors: they are already present in specific components of the machine, since they are used by control logics for the correct operation of the machinery. They do not represent an extra cost for maintenance. For example, in modern servo-motors there are always an encoder for position measurements, an embedded ammeter (often by means of two simple Hall sensors) for the measurement of the current absorbed by the mains and a temperature sensor (often embedded in the encoder) for the measurement of the heat inside the motor (or at least a positive temperature coefficient (PTC) thermistor in the coils for detection of over temperature).

The main sensors used in the real case taken into consideration are listed below:

- Accelerometers: they measure the vibrations of the mechanical components (e.g., rotating shafts), giving a picture of the inner health of the machine. Every month hundreds of scientific papers on the use of accelerometers for diagnostic purposes are published (multi-purpose external sensors) [13, 14, 15, 16].

- Encoders: they measure the position of rotating parts (e.g., shafts), providing a flag at each complete rotation. In particular, encoders are increasingly present

in electric motors, embedded in any servo-motor with a high angular resolution (e.g., 4096 ticks per revolution).Together with accelerometers, they allow the diagnostics of the components in the angle-domain, that is a reconstruction of the vibration signal based on the actual rotation of the component, providing immunity to speed fluctuation which can make the noise-to-signal ratio worse [17, 18, 19] (embedded sensors).

- Current/torque sensors: they are embedded sensors necessary for the correct operation of an electric motor. The current absorbed by the motor is proportional to the torque load applied to the motor shaft. It is straightforward that any change in the working conditions of the motor (e.g., an increase in wear) increases the torque load and consequently the current requested (embedded sensors).

- Pressure sensors: in order to avoid any possible interference between the moving parts of the package forming line and the cables of the sensors, it is necessary to introduce pressure sensors for the indirect measurement of the wear on cutting knives (specific external sensors).

- Temperature sensors: they measure the temperature of specific components. In particular, servo-motors can have an embedded temperature sensor to measure the heat inside the motor (embedded sensors).

The total number of the sensors depends on the size of the machine under control, the critical key points and the budget available for the condition monitoring area.

2.2.1.2 Off-Line Data

In this work, the term "off-line data" refers to informative data asynchronous to the working conditions of the machine. This is the opposite of "on-line data", which are collected during events synchronous to the working conditions. For our purposes, the "off-line data" are event data, i.e., the list of all the technical interventions performed by service engineers. These events cover scheduled service interventions, unexpected service interventions and production conditions of the machine. Examples of scheduled service interventions are firmware and software updating and preventive maintenance of specific components. Examples of unexpected service interventions are breakdowns of mechanical or electronic components. De facto, the minimization of this type of interventions is the target of every condition monitoring system. Examples of production condition events are the starting and the stopping of production and the substitution of consumables. Some off-line data can be acquired and stored automatically, for example the stopping or starting up of the machine, but most off-line data are manually inserted by the service engineers that perform the technical interventions or by the after-sales department which defines the scheduled operations. Off-line data are essential to condition-based maintenance and much more in the development step of the data-driven processing, showing the difference between supervised and unsupervised methods (for more details see [20, 21]). The collection of the off-line data can be hardly demanded from an automated system.

Consequently, proper training of the service engineers is fundamental to all the companies that want to service maintenance. By experience, not all events are declared for different reasons (the periodicity of an event may not be of concern or it is not recorded), although greater awareness of the consequences of one's work can minimize the missing events.

2.2.1.3 Data Pre-Processing

The data collected from a single machine must be pre-processed locally, before data are sent to a high-level storage structure that will be described in 2.2.1.5. The main reasons are to reduce the amount of data to be sent to the cloud platform and to decrease the latency in the decision-making process. The cloud platform costs depend on the amount of data processed, therefore aggregated data are preferable for cost reduction. Moreover, quick pre-processing can give fast alarms, since it is possible to detect the problem before the entire row of data is logged in the cloud, and send the alert to the machine promptly. The local architecture of the condition monitoring system is made up of:

- An industrial PC (iPC) for data manipulation;
- Data logger hardware for the acquisition of external sensors;
- Fieldbus (IEC 61158) network for data communication between the iPC and motor drives (or other embedded sensors).

The main functions of the pre-processing step are the following:

- Removal of empty or incomplete files: the condition monitoring system records data regularly. Only a few sensors collect data in a period of time so as to reduce computational efforts. It could be that specific parts of the system are not working during the time frame when the corresponding sensor is acquired, generating empty or incomplete files. These files must be removed to free memory space on the storage device.
- Checking of the sensors: the measurement files are checked for inconsistency of data. Especially in manufacturing machines, processes are repeated cyclically and the expected data from sensors must contain cyclic components too (e.g., at the productivity frequency of the machine). If the data recorded by a given sensor do not show cyclic components in the spectrum, it is due to a problem on the measurement chain: the sensor, the cable or the acquisition system. The inconsistency of the data must generate an alarm to the service engineer that will schedule a check of the sensor.
- Calculation of statistics: the computational capacity of modern industrial personal computers allows statistical analyses of the acquired data, such as the root mean square (RMS) value, variance, kurtosis, quartiles, etc. The main advantage is data reduction; each statistic is a single scalar value compared to the thousands of points acquired by each sensor. Statistics are the features that the data-driven diagnostic method uses to make the post-processing analysis.

- Selection of specific data: if the post-processing based on data-driven analysis reports an incipient fault, a more detailed model-based analysis is performed. The performance management center can ask the local unit for specific data useful for a targeted analysis. The local unit sends those specific raw data to the cloud.

- Storage of data: the data are locally stored for a limited period of time with a backup policy (when the storage space ends the new file overwrites the oldest one). The storage is necessary to provide selected raw data if asked.

- Sending of the data to the cloud: all relevant data, i.e., the statistics and the off-line data, are sent to the cloud for the post-processing step.

2.2.1.4 Data Cloud Processing

The data cloud processing mainly consists of cloud-computing data management. The statistics and off-line data for different machines are collected on the cloud and plotted with respect to time in order to monitor the evolution of the data constantly. Today, cloud-computation providers assure sufficient computational power to run complex algorithms and most of them already implement a Python or R-language console. De facto, these free programming languages are common languages to data scientists for statistical computing. More recently, some cloud providers have offered integration with well-known commercial software for mathematical computing. The main functions of the cloud-processing step are the following:

- Data-driven analysis: statistical data from every monitored subsystem of the machine are analysed by means of data-driven machine learning techniques, such as neural-networks, support vector machines and clustering. The machine learning system generates alarms to the performance management center, i.e., the data-scientists, who can query the local system for a more detailed analysis on specific data.

- Data transfer: the off-line data do not need further processing. In this case, the cloud acts as a simple storage device; the analysts pick up the off-line data collected from different machines for the off-line development of condition monitoring techniques.

It should be noted that the development and the training of the machine learning techniques are not performed on the cloud, but at the performance management center. The software implemented in the data cloud-processing must be ready-to-run in order to avoid interruptions of the servers.

2.2.1.5 Data Post Processing

The data post-processing mainly consists in reporting, decision support and detailed analysis of the data. In particular, the main functions of the post-processing step are the following:

- Reporting: the condition monitoring outputs are divided into several reports on the state of the sub-system components. The stakeholders of condition monitoring reports are various: service engineers, managers, consultants, external service providers etc., and each of them needs different pieces of information.
- Decision support: the reports are used by the performance management center, i.e., a structured support service, in order to update historical data for modelling upgrade development, to analyse criticality, to query advance failure analysis of specific components and to manage the technical service.
- Model-based analysis: once an alarm is received by the cloud-processing, more advanced signal processing tools can be used to provide more details on the fault, for example, if there is a fault in the inner or outer ring of a bearing.
- Service: if some problems are identified, a report of the situation is sent to the service engineers through a IoT device. In this way, the service engineers can monitor the state of the plant at any time and in case of alarm they are warned promptly. Thanks to the analysis service, the service engineers are not only warned about an incipient failure but they are also informed about the procedure necessary for the maintenance, whether it is necessary to order the broken part or it is available in the warehouse.

A web platform with a custom-driven application programming interface (API) must be developed as an infrastructure in order to satisfy different requirements of both data scientists and service engineers. They can retrieve data by using query methods or read reports through PCs or smart-phones.

2.2.1.6 Condition Monitoring Algorithms

Condition monitoring algorithms are the foundations of the maintenance policy, since they allow a reliable and fast response to incipient faults. They can create, in the customer, a feeling of confidence in condition monitoring or destroy it completely in the case of missing or false alarms. The definition of a proper algorithm requires a lot of time and its value cannot be underestimated. Several algorithms are suggested in the scientific literature every day. Each component under test has its own fault modes, i.e., a characteristic type of fault is one due to wear and based on its geometry and dynamic conditions. For example, ball bearings are one of the most common components in mechanical design and their fault modes are related to working conditions. In particular, the bearing is made up of an outer ring, an inner ring, rolling elements and a cage. Each part of the bearing can be subject to damage, which can differ in the periodicities of impacts. These differences allow the recognition of the damaged components. Despite the number of possible customized components, the most common components in mechanical design are the standard ones, such as bearings, gears, shafts and electric motors, regardless of the specific industrial field. As a consequence, an initial bibliographic survey on scientific journals is the starting point for the development of a proper condition monitoring algorithm for the data processing.

The data flow starts from the raw data acquired by sensors to the final output, usually in limited dimensionality, such as binary output or low-dimensional output. The data flow can be divided into three main classes:

- Data cleaning: it includes all the procedures activated to remove inconsistent data, for example, empty measurement files, corrupted files, disconnected sensors, broken cables etc. This is not a proper condition monitoring technique but it is a preparation process.
- Fault detection: it includes all the procedures suitable to recognize a fault in the system. It does not specify the causes of the fault, but it only specifies its presence. In most cases, anomaly detection techniques are sufficient for industrial purposes. If there is a faulty bearing in an electric motor, the motor must be completely replaced whether the fault is in the outer ring or it is in the inner one.
- Fault diagnostics: it includes all the procedures suitable to characterize the fault of a specific component and the level of the damage of the component. It is also the starting point for the estimation of the residual life of the component (prognostics) [22]. Fault diagnostic techniques are useful for redesigning a component: the detailed knowledge of the fault can suggest a better design to reduce the loads in working conditions, extending the expected life of the component.

Focusing on the fault detection and fault diagnostic techniques, the scientific literature can be divided into data-driven techniques and model-based techniques.

2.2.1.7 Data-Driven Techniques

Data-driven techniques are not related to the physical system they model, but only to the input data, independently of the type of sensors. These techniques basically provide a metric of similarity among data. Common metrics are Euclidean and Mahalanobis distances. Machine learning techniques are an example of data-driven techniques. They require a training step and a testing step. The training step defines the expected dataset for faulty and healthy components. In this step, the off-line data defined in 2.2.1.2 have great importance since they determine time instants corresponding to the breakage of a component. Data before and after that time instant can provide a good example of faulty and healthy conditions to be used in training. The testing step is the application of the machine learning techniques to the new input data. A greater similarity between the recorded data and the faulty or healthy datasets determines the actual health status of the component. The machine learning techniques that need a training step are also known as "supervised" learning techniques. Conversely, "unsupervised" learning techniques may not need a training step, depending on the method that used: for instance, one-class SVM needs a training phase [23], while artificial immune systems do not [24]. These techniques try to describe the data distribution of a healthy state (or a faulty one) in a complete way, so that any metric variation is an indicator of a faulty state (or a

healthy one). It should be noted that machine learning techniques need a sufficient amount of historical data for training, but they also need training datasets that cover all possible fault events. Hundreds of data-driven techniques have been developed in literature and an exhaustive list is out of the scope of this thesis. Relevant review papers have already been reported so far [25, 26, 27, 28], demonstrating the capability of these techniques in different fields of application. Based on the direct experience of the authors, three machine learning techniques are presented below:

- Artificial neural networks (ANN): this technique tries to mimic the biological neural networks and the way in which the pieces of information are managed by the human brain. It builds a weight matrix trying to reward or penalize input features based on the error output in the training step. One or more layers, i.e., weighting matrices, can be chosen. The key component of the ANN is the backpropagation algorithm that distributes the error term back up through the layers by modifying the weights at each node. The ANN technique has been used in several research fields [29, 30, 31, 32, 33, 34, 35].

- Support vector machine (SVM): the SVM technique [36] computes a hyperplane that divides faulty and healthy data by maximizing the distance of the hyperplane to the datasets. The dimension of the hyperplane depends on the dimension of the input data features. The key component of the SVM is the choice of kernel function, the purpose of which is to project data in a high-dimensional space where the data can be separated by the hyperplane. Once defined, the hyperplane acts as a threshold, classifying new input data into the two classes. Examples of the application of SVM to condition monitoring can be found in [16, 36, 37, 38, 39].

- Auto associative kernel regression (AAKR): this technique predicts the health status of a component thanks to the historical data deriving from a healthy dataset. New inputs are compared to the prediction of the healthy state. The difference between the two signals, i.e., the residual, is used as a metric to provide the health status of the component. Examples of AAKR applications to condition monitoring can be found in [40, 41, 42, 43, 44].

All the machine learning techniques need, as input, a subset of the acquired data. Since sampling frequencies of some sensors could exceed 10 kHz for more than 10 s, it is unthinkable to work with weighting matrices of 100.000x100.000 size. Statistics are usually computed on the input data, reducing the weighting matrices to 10x10 size (as an order of magnitude). The type of statistics and their number are the results of a trial-and-error process, depending also on the specific system under testing. Nevertheless, basic statistics, which describe the probability density function of a variable, are good attempt values, and include:

- RMS: it is defined as the square root of mean square;

- Variance: it is the second central moment of a real-valued random variable;

- Skewness: it is the third central moment of a real-valued random variable;
- Kurtosis: it is the fourth central moment of a real-valued random variable;
- Quartiles: they are the 25th, 50th and 75th percentiles of the input variable.

In some cases, even parameters linked to the dynamics of the machine are relevant, for example the hourly capacity of the machine during the acquisition of the sensors. Once trained, machine learning techniques do not require high computational efforts and return a fast classification of the new input data. For these reasons, they are particularly suitable for cloud computing and can be used for the cloud-processing described in Section 2.2.1.4.

2.2.1.8 Model-Based Techniques

In the introduction of their three-part papers on process fault detection and diagnostics, Venkatasubramanian et al. [45] give a clear and exhaustive description of model-based approaches. Model-based techniques require a priori knowledge of the set of failures and the relationship between experimental data (observations) and failures (causes). This relationship is developed by using frequency-response models or dynamic models. Venkatasubramanian et al. divide the model-based methods into two classes: qualitative or quantitative. "The model is usually developed based on some fundamental understanding of the physics of the process. In quantitative models this understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of the system. In contrast, in qualitative model equations these relationships are expressed in terms of qualitative functions centered on different units in a process [45]". In automatic control, the quantitative modelling of physical system is the core part of the so-called system identification. This research field uses statistical methods to build mathematical models of dynamical systems from measured data. De facto, the system identification determines the transfer function between input and output. By abstraction, the model of the system can be represented as a box connecting inputs (working conditions) and outputs (measured data). This box can be classified into three main classes:

- White-box model: it is a model based on first principles, e.g., Newton–Lagrange equations. It requires a deep knowledge of the system: the geometry, external loads and torques, characteristics of the materials, the type of interactions among components (e.g., friction, or impacts), masses, etc. In many cases such models will be overly complex due to the complex nature of many systems and processes. It should be noted that the development of a white-box model is not a one-shot activity but it must be continuously developed, adding more details if necessary. Examples of white-box modelling can be found in [46, 47, 48].
- Black-box model: no a priori model is available. The input/output relation of the system is statistically computed without considering the physics of the process at all. Most system identification algorithms focus on this type. The

black-box model is similar to data-driven approaches, which are not further considered in this work.

- Grey-box model: this model is between the white-box model and the black-box model. Although the peculiarities of what is going on inside the system are not entirely known, a certain model based on both insight into the system and experimental data is constructed [49]. The resulting model still has a number of unknown free parameters, which can be estimated by using the system identification. An example of a grey-box is the modelling of the expected signal produced by a faulty system (i.e., the output signal of the system). In this particular case, the grey-box model has been studied in depth in literature (e.g., a ball-bearing) and it is used to simulate the expected output signal in different working conditions. The condition monitoring analyst can use the simulated signal to develop and validate signal processing techniques. Examples of fault modelling can be found in [50, 51, 52, 53, 54, 55, 56, 57].

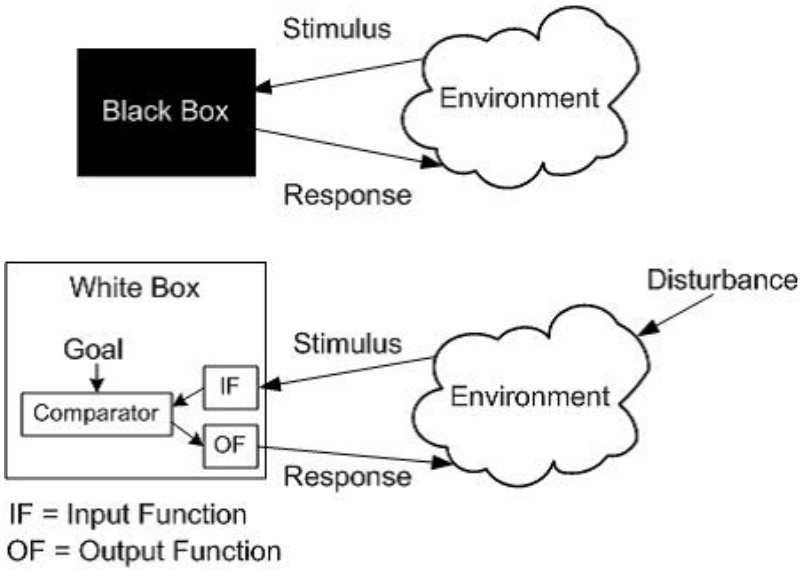


Figure 2.2: Black and white box architecture.

Based on the level of details required, the development of a model-based technique needs more time than a data-driven model. The model of a physical system depends on the characteristics of the system itself. Consequently, it is not possible to indicate a common development methodology that could be extended to a general physical system. The analysis of scientific literature is the first step of modelling. Further assistance could come from specific commercial software for the modelling of physical systems, but the analysis of the physical process that takes place is unavoidable. Due to the complexity and the demanding computational time, model-based techniques are particularly suitable for off-line computation of specific subsets of data. The results are generally better than the ones obtained by means of data-driven techniques, since the description of the fault causes is identified better. As mentioned in Section 2.2.1.5, the analysis of data in advance is useful for

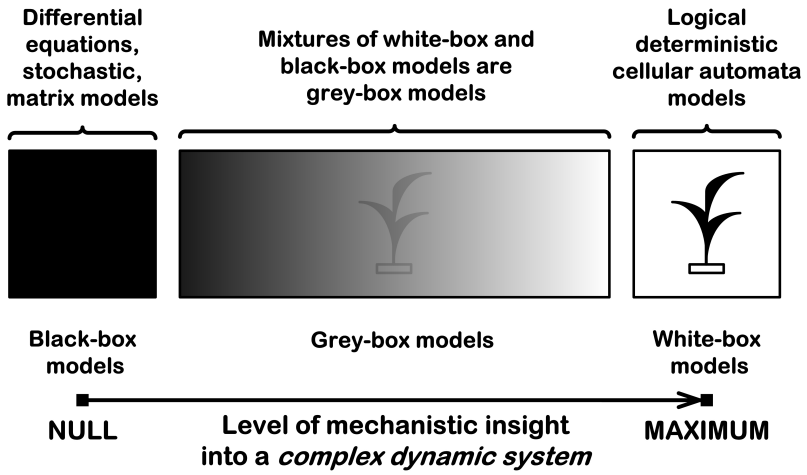


Figure 2.3: Grey box architecture.

the technical development of the redesign of components, in order to optimize the geometry and to maximize the expected life of the component.

2.3 Independent Carts System

This chapter describes the techniques used in automation industries to produce and transmit mechanical power in order to contextualize the Independent Carts Systems in their field. The first part of this chapter focuses on the most commonly used methods, chain and belt drives, while the second part explains the technology of linear motors focusing on the Independent Carts Systems.

2.3.1 Progress report on electrical motor technology in automation field

In automation industries, electrical rotary motors are used to produce mechanical motion necessary for production. In order to produce useful work, it is necessary to have a method to transmit the power of the motors to other components and mechanisms. The most used methods in automation industries are chain and belt drives.
These systems are used for transmission of power, conveyance of materials and purposes of timing.
Belt Drive is used when high speed and low torque are requested.
The advantages of this system are:

- Low costs,
- Quiet.

The disadvantages of this system are:

Figure 2.4: Belt drive.

- It can not work in a corrosive environment,
- It can not work with high temperatures,
- It can have slipping problems.

Chain drive is used when low speed and high torque are requested.

Figure 2.5: Chain drive.

The advantages of this system are:

- Strength,
- It can be used with high temperatures,
- It can be used under adverse atmospheric conditions.

The disadvantages of this system are:

- It needs lubrification
- Noise
- Weight
- Vibration
- High costs.

Belt and Chain drives have low flexibility because their characteristics directly depend on the work they have to execute. De facto, these systems are designed for a specific work and they cannot be used for different tasks. For this reason, it is difficult to build flexible machines using these systems.
In order to produce different motions with the use of Chain or Belt drives, it is necessary to have different chains with different rotary motors. Consequently, more complex the desired motion, more complex the mechanical design of the machines.
In order to overcome these problems, some new products have been put on the market. These products use the Independence Carts System, which is based on electric linear motors.
There are many differences between the Independent Carts System and the chain and belt drives. The most relevant difference lies on the type of the motor used. Chain and belt drives use rotary motors and gearboxes in order to transmit power. The Independent Carts System uses linear motors, which do not need any gearbox; in this way the contact among the mechanical parts is reduced. In order to explain how this difference changes the characteristics and the performances of the machine, the inner features of the two types of motors will be explained.

2.3.2 Linear motors

Linear motors are different from rotary motors because the stator and the rotor are "unrolled". This difference in the construction design also determines a difference in the generated force: rotary motors generate torque and linear motors generate linear force. Linear motors consist of two parts, a coil assembly and a magnet assembly.
The coil assembly consists of copper windings within a core, which can be made up of different materials. The magnet assembly consists of a set of rare-earth magnets mounted on a steel plate with alternating polarity.
The magnetic field produced by the magnets and the electrical field produced by current generate a force that is called Lorentz force:

$$F = q(E + VXB) \tag{2.1}$$

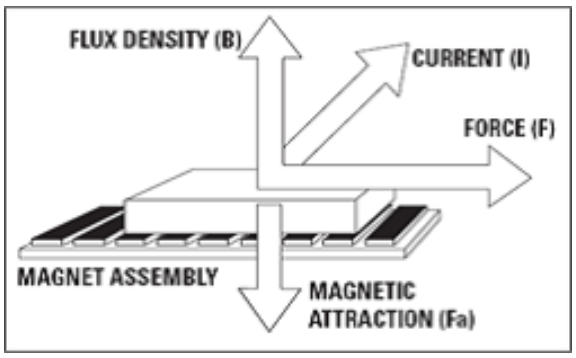


Figure 2.6: Direction of electrical and magnetic fields.

The current crossing the coil windings is controlled by a driver, which imposes

a magnitude and a direction such that the resultant Lorentz force drives the rotor to the desired position.

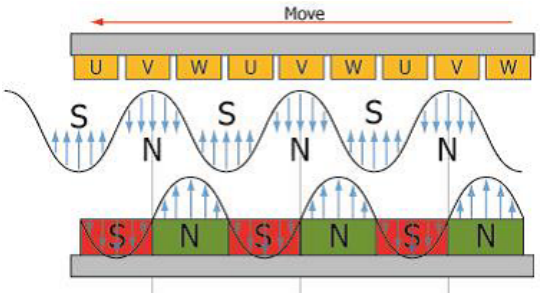


Figure 2.7: Representation of the electromagnetic behaviour of the system.

In this case, the rotor as well as the stator can be made up either of coils or of magnets. Linear motors can be divided into different families, the most important of which are:

1. Iron core
2. Ironless core
3. Slotless

2.3.2.1 Iron core

Coils are wounded by foils made up of steel. Laminations are insulated from each other in the same way as in rotary motors. In this way eddy currents, which reduce efficiency, decrease.

Advantages

1. Lower costs
2. More efficient cooling as cooling tubes can be set around laminations
3. Higher force available per unit volume as laminations concentrate flux field

Disadvantages

1. Cogging torque effect
2. High attractive forces between iron foils and magnets

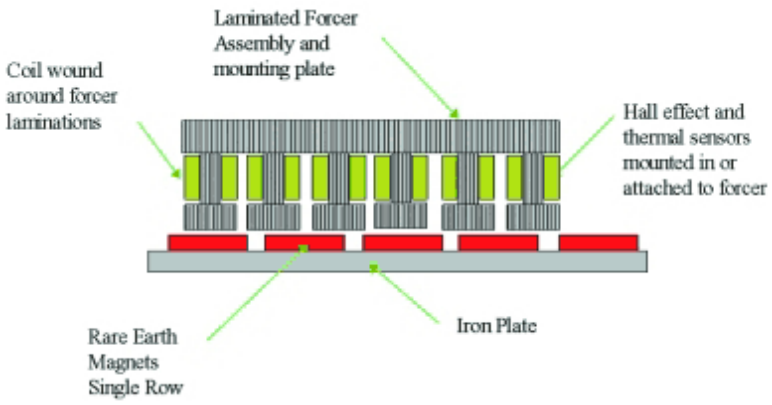


Figure 2.8: Iron core.

2.3.2.2 Ironless Core

Coils are not wounded by laminar iron, but they are inserted into a plate made up of epoxy. Magnets are generally set on two tracks having a typical "U" shape. The mover contains the coil windings, which are exposed to the magnets. The ratio of peak power to continuous force is generally high and these motors are typically used in highly dynamic applications.

Advantages

1. No cogging effect due to the lack of iron foils
2. No attractive forces thanks to the lack of iron foils
3. Higher acceleration

Disadvantages

1. Cooling problems due to poor heat dissipation,
2. Peak power range is limited to a few thousand Newton.

2.3.2.3 Slotless Core

It is a hybrid between an iron core design and an ironless linear motor design. It consists in coils with back iron contained within aluminium housing over a single magnet rail. By means of this structure, it is possible to cool the coils.

Advantages

1. Good heat dissipation
2. Structurally stronger forces

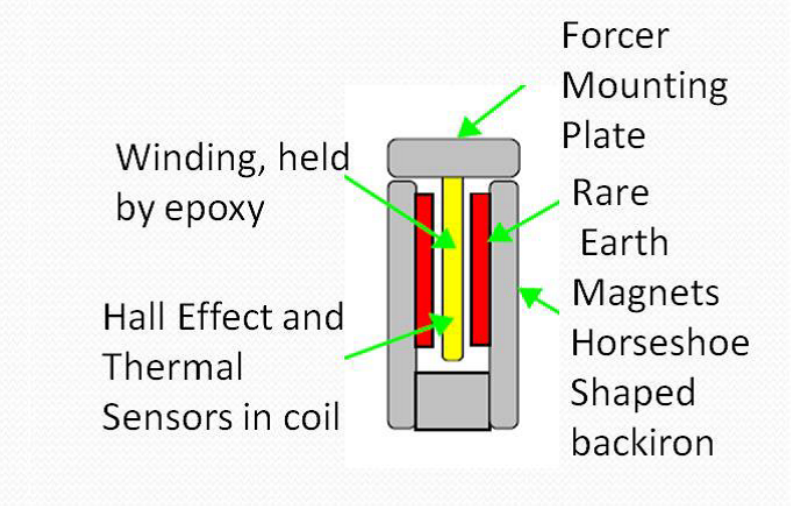


Figure 2.9: Ironless core.

3. Low weight with respect to the iron core architecture

4. Lower cogging with respect to the iron core architecture

Disadvantages

1. Some cogging effects

2. Some attractive forces

3. Lower efficiency

The improvement of technology, especially as regards logic controllers, has enabled linear motors to perform tasks, which were not possible before. In this way, mechanical complexity is reduced while software complexity is improved. In order to control the movers and to produce relative motions among them, it is necessary to improve control complexity.
The Independent Carts System uses linear motors in order to control one or more movers that are constrained by rollers to follow a track. The track can have different shapes with curved and straight parts and it has a flexible architecture in order to build modular configurations. In this way, a high performance flexible system can be produced, with this technology each mover can be controlled independently. The movers can accelerate, decelerate, take an absolute position and produce forces. The velocity of the movers can be very high with respect to rotary motors, each mover can move with a velocity of 4m/s.
Thanks to the reduction of moving parts, the maintenance of the components is reduced with respect to the technology of Chain and Belt drives. Actually only the rollers are in contact with the track and the wear of these parts can be easily monitored. The architecture of the machine is modular and for this reason it is possible to change the shape of the track and the numbers of the movers. Another

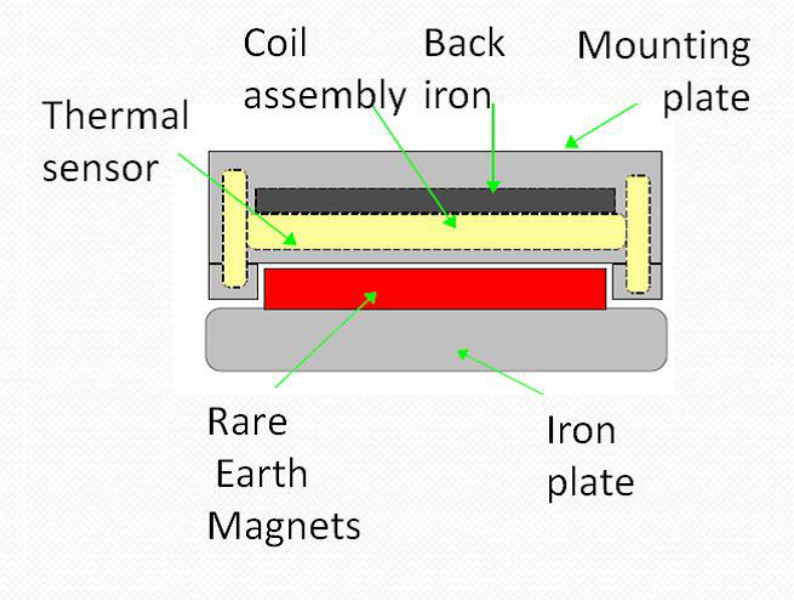


Figure 2.10: Slotless core.

advantage is that the system is very compact with respect to Chain and Belt drives and it is integrated. Different types of carts and motors exist for the Independent Carts System, the bigger difference among them lies on the possibility of having a "Moving Coil" or a "Dynamic Moving Magnet".

2.3.2.4 Dynamic Moving Magnet

Coils and drivers are set in the track and magnetic plates are placed on the carriers. In this way, the weight of the mover is reduced. There are no moving cables and it is possible to operate inside a vacuum, where there is no convection to effectively dissipate heat, but it is not necessary because the coil units can be set outside the vacuum.
The most relevant disadvantage of this system is the possibility of the heating of coils especially when the machine has to produce high forces on the same motors. Consequently, it is necessary to improve the cooling of the machine.

2.3.2.5 Moving Coil

Coil windings and a driver are set in each mover and magnets are placed on the track. The sensors are set on the rail (absolute encoders are usually used). With this architecture, the weight of the moving parts is high.
Most Independent Carts Systems are of the dynamic moving magnet type. On the market there are different companies that produce this technology and each company differs from the other one in mover design, feedback methods, coil position, control protocols etc. The most important Independent Carts Systems are:

- ACOPOStrak of B and R [58]

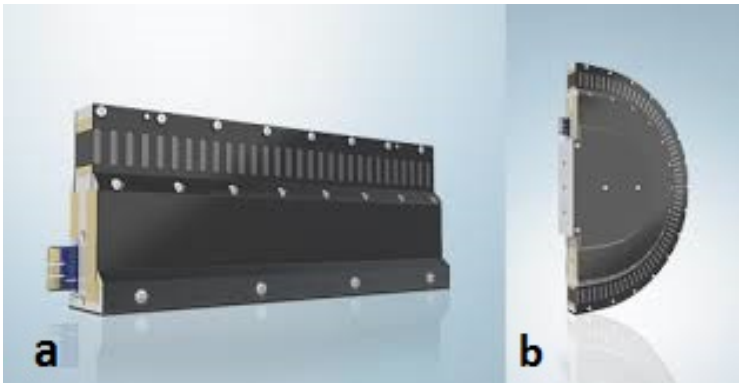


Figure 2.11: Beckhoff XTS: a) linear motors with straight shape stator b) linear motors with curved shape stator.

- Itrak of Rockwell Automation [59]
- XTS of Beckhoff [60]
- Multi carrier system of Siemens [61]

Each of them uses a different PLC for the control of the internal tasks. This fact can reduce the flexibility in the machine architecture, which is a great limitation from the point of view of a designer. In this way, even the possible condition monitoring architectures are limited, for example as regards the data acquisition and the integration of sensors in the machine. In order to overcome this problem, new methods for the synchronization of axes between PLCs and iPCs of different vendors will be introduced in Chapter 3.

The most important advantages of the system are basically its high flexibility and dynamics. As a matter of fact, this system is based on modular linear motors that can have either a curved or a straight shape as shown in Fig. 2.11. The different stators can be assembled in different configurations in order to have the best path shape with respect to the function. They contain coils that produce a controlled magnetic field which moves the carts.

The encoder is placed on each cart and it can be a Hall sensor or a radio flag which allows the position feedback and the velocity feedback. The carts, which can have different geometry, are connected to the rail (the stator) through rolling bearings (Fig. 4.8 (a) (b)), the number of the bearings can vary according to the function. The carts can be freely moved along the rail since they are controlled independently, it is possible to change the distance and the velocity among the carts during the execution of the task and they can be moved back and forth. The independent control of each cart allows a high flexibility of the system with respect to the traditional automatic machines, since the use of the Independent Cart System makes possible to dynamically adapt the functionality of the system to the process only by means of the software variation and without any mechanical changes. For example, the Independent Cart System can work with products that have different shapes, since it is possible to define a different motion profile for each cart. Thanks to this

system, it is also possible to build a machine with a variable rate of production as the Independent Cart System can change its velocity dynamically. These activities are not possible with the traditional motor-driven chains, belts and gears because, in order to change the functionality of the machine, it is necessary to design different mechanical links for every type of product and every rate of production. Another important advantage of the use of the Independent Cart System is the reduction in the downtime of the machine. De facto, when there is a variation in the product, it is not necessary to stop the machine for mechanical changes as the end-effector is fixed to each cart and so it is only necessary to change the motion profiles of the carts. On the contrary, with the chain, belt and gear system it is necessary to stop the machine for the change of mechanical parts and consequently the downtime of the system increases.

Chapter 3

Condition Monitoring for Independent Carts System

3.1 Introduction

This chapter will deal with the integration of a Condition Monitoring Framework for Independent Carts Systems in a pre-established condition monitoring architecture. The integration is quite complex because so far only some companies produce these systems and each of them has its own control system. As regards Condition Monitoring, the problem is quite serious because the diagnostics of the system could be impracticable without any integration of the working parameters of the Independent Carts System with the process variables of the machine. For this reason, the first part of this chapter will illustrate two new methods for the synchronization of axes between PLCs and iPCs of different vendors. The most important and basic part of integration is the synchronization of the motion tasks among multivendors programmable controllers. This is so important because in automation applications the motion task synchronizes all the motions of the machine, therefore with its synchronization and control, it is possible to manage the system and to have all the operational data of the machine.
In order to find a global solution to this problem, particular attention has been focused on programmable controllers, which can manage axes by means of point-by-point control or motion instructions.
Two synchronization algorithms have been developed and validated for real and virtual axes; they differ in computational load so that they can be used with programmable controllers having high or low computational performances.

3.2 Distributed real-time system

Distributed solutions for real-time system architectures are more and more used for the improvement of the industrial process and the development of smart factories in order to follow the new wave of Industry 4.0 [62]. In comparison with the centralized architecture, these systems improve the dependability, compensability,

scalability and extensibility of the products [63]. They are also called networked motion control systems (NMCS) and they consist of a set of different nodes such as controllers, sensors, drive controllers, regulators, HMIs and actuators, spatially distributed and interconnected by a communication network [64, 65]. In the automation field [66], there are several communication networks and protocols, which answer different requirements for different applications. Real-time Ethernet (RTE), which is a fieldbus technology, is commonly used for the communication of data among nodes of distributed real-time systems thanks to its security and reliability [67, 68, 69]. Several Ethernet applications exist such as ControlNet [70], PROFIBUS [71], EtherCAT [72] and Ethernet/IP [71].
The new challenges of distributed control system have been presented by Dripke et al. [73]. The most important challenges, which have been identified by them, are communication among multi-agent systems, time and synchronization due to the parallel computing nature of distributed systems, control tasks, capacity and scalability. In this direction, a lot of new communication protocols for IoT and time synchronization systems, such as OPC UA [74, 75] and TSN [76, 77, 78], are being developed and standardized in order to overcome the existing problems. While in the industrial field there are some examples of the use of the OPC UA protocol for communication [79], in the case of motion synchronization, this protocol is used together with another protocol (EtherCAT) in order to achieve the synchronization of multiple axes [80]. As regards Time-Sensitive Networking (TSN), it is a new communication protocol, which is focused on the simultaneous motion synchronization of axes and data exchange [81]. Several companies are developing new devices in order to implement this new protocol [82] and to increase its reliability, security and performances for industrial applications. A milestone of motion control is synchronization among multiple axes [83, 84, 85, 86] that is fundamental in different fields such as Computer Numerical Control (CNC), Surface Mounting Technology (SMT) [87, 88] and automated factories [89]. It is acknowledged that a new step towards the development of distributed control systems is represented by integration and interoperability among industrial programmable controllers of different producers [89]. Some applications already use communication between PLCs or iPCs of different vendors [90] but only for the exchange of information and not for the motion control of motors. Several studies have been carried out to implement distributed controls for motion axes [91, 92, 93]. In all these cases, the master control directly commands drivers of the same vendor or drivers directly compatible with the communication network of the master control. This is due to the fact that different vendors use different methods for the control of the dynamics of motors; moreover, motion control is a hard real-time task, which requires a high level of accuracy and reliability. The advantages of a distributed motion control system, which can be used with drives and motors of different vendors, are the increase of re-usability, re-configurability and extension of distributed systems. The novel approach of this work consists in the implementation of a distributed motion control in which a master iPC commands incompatible drivers and motors through a slave PLC. This method differs from the actual ones in which the motion control between different PLCs/iPCs of different vendors is possible only if the two systems use the same protocol for communication

and synchronization. An example of accurate time synchronization among multiple devices is presented by Shiying et al. [94]. They use the IEEE 1588 protocol for the design of a time synchronization network of experimental advanced superconducting tokamak (EAST) poloidal field (PF) power supply control system. In this case, they synchronize the time among a control system of PFPS on EAST, local controllers, coordination operations of several sets of power supply and DAQ systems (DAS) with the maximum time offset from the master node that never exceeds 50 ns. This chapter is focused on the development of a general master-slave real-time interface between different programmable controllers for the synchronization of motion control by the use of real-time Ethernet communication networks with the IEEE 1588 V2 Time Synchronization protocol [95, 96] and the Ethernet/IP protocol [97]. The hardware architecture, the used communication network and the algorithms, which will be subsequently shown, represent a global solution for motion control among industrial programmable controllers of different vendors. This solution can be used for any type of distributed real-time control systems that include both PLC and iPC devices. The tests have been performed with different PLCs and iPCs. The general solution illustrated in this chapter will be also used with the latest protocols, such as TSN, when they are introduced into the automation field as standard protocols. Following the method shown in the chapter, it is possible to overcome the problem of connectivity between PLCs and iPCs of different vendors without the reduction of the function of one of the two systems. De facto, it is also possible to control functions that require high accuracy and safety such as the motion control of different motors.

3.3 Notations

Cl_M	Clock Master
Cl_S	Clock Slave
Mot_M	Motion Master CPU
Mot_S	Motion Slave CPU
p_i	i-th Set position
v_i	i-th Set velocity
a_i	i-th Set acceleration
j_i	i-th Set jerk
T_{ad}	Actuation delay
T_{stt}	Motion task time of the Mot_S
CUP	Coarse update
D_i	i-th Time difference
T_{mc_i}	i-th Time instant in which Mot_M imposes the set values on its axes
T_{sc_i}	i-th Time instant in which Mot_S imposes the set values on its axes
T_t	Total time
p_{f_i}	i-th extrapolated future position
v_{f_i}	i-th extrapolated future velocity
a_{f_i}	i-th extrapolated future acceleration

T_a	Acceleration time
T_r	Time at constant velocity
p_0	Actual position
v_0	Actual velocity
v_{jog}	The input velocity given to the jog
a_{jog}	The input acceleration given to the jog
e_p	Position error
e_v	Velocity error

3.3.1 Hardware architecture

In order to generalize the architecture of the system, the hardware components used are typical of every programmable controller vendor. The only constraints are:

- the Ethernet communication network for the exchange of the kinematic data of motors with a velocity of at least 100Mbps. For some vendors it is directly integrated in the controller card while other vendors need an appropriate card
- an Ethernet card that allows the use of the IEEE 1588 V2 protocol. In some cases it is the Ethernet communication card itself if the system uses a CIP Sync protocol, or in other cases an appropriate card is necessary.

Furthermore, the programmable controller that moves the main motors is called Motion Master CPU (Mot_M), while the one that synchronizes its motors with the master is called Slave Motion CPU (Mot_S). With this method it is possible to synchronize both real and virtual axes. This is very important because in the automation field the main axis of the machine is often virtual and all the other axes, both real and virtual, move synchronously with it.
If the system needs to synchronize real motors, the hardware architecture includes drivers and motors.
If the system synchronizes only virtual axes, drivers and motors are not included in the hardware architecture.
Other studies relating to motor synchronization have been carried out by means of the use of the IEEE 1588 synchronization protocol and Ethernet connections [91]. Even they consider a master-slave architecture, but the main difference lies on the fact that they use only a master controller and some slave drivers of the same vendor as in Fig. 3.1.
On the contrary, this study aims to develop an interface between programmable controllers of different vendors, which control drivers and motors. In this case a master CPU controls different slave CPUs that send the set values to the drive in order to control several motors as represented in Fig. 3.2.

3.4 Problem definition

The synchronization of several axes between programmable controllers of different vendors can be divided into four parts:

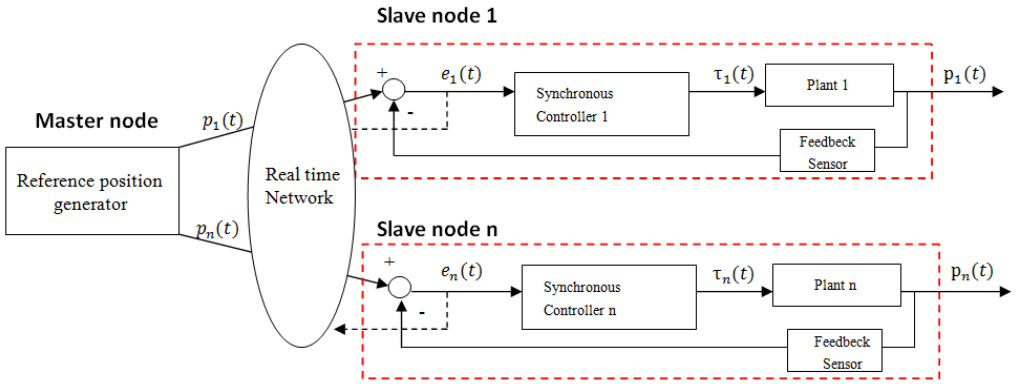


Figure 3.1: The system architecture analysed in literature

.

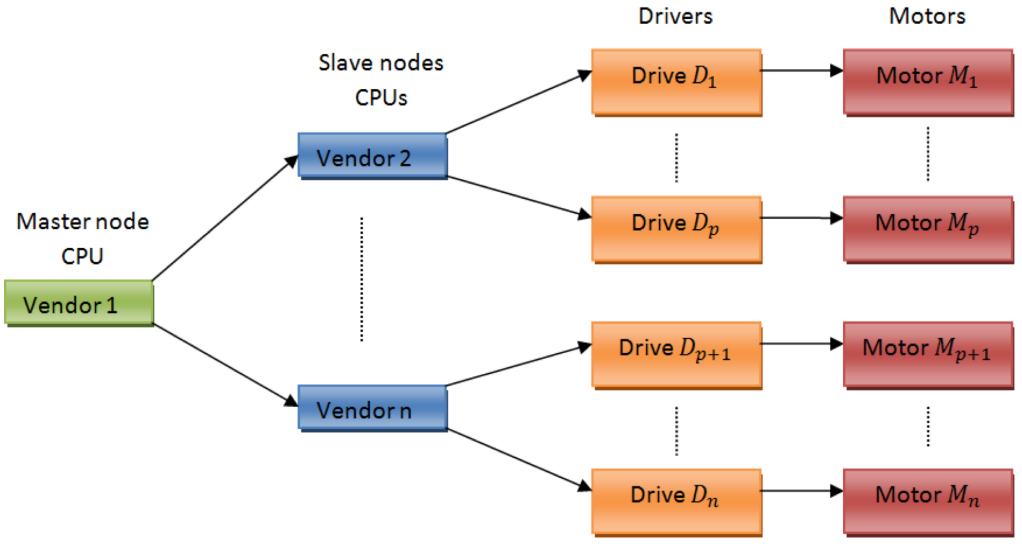


Figure 3.2: System architecture.

1. time synchronization of the system
2. exchange of kinematic data
3. computation of trajectory
4. driver actuation.

3.4.1 Time synchronization

First of all programmable controllers need to have the same time reference so that they can evaluate the kinematic values of the system properly. The clocks of programmable controllers can be very different in accuracy and reliability, consequently it is necessary to use a method which is oriented to find a common clock between the controllers. In order to obtain a high accuracy in motion synchronization, a high

accuracy of the global time of CPUs is required. Therefore, the synchronization protocol used is the IEEE 1588 V2 (Precision Time Protocol), because it is one of the most used, it is based on Ethernet communication network and it achieves a clock accuracy < 100 ns [95].
This protocol uses a Clock master(Cl_M) that is the source of the synchronization reference and a Clock slave(Cl_S) that is the destination of the synchronization message. In order to synchronize the Cl_S with the Cl_M, the former computes the time difference between its clock and the synchronization message. If the delay is higher than 100 ns, the Cl_S overwrites its time in order to ensure synchronization, this means that there are some gaps in the time reference of Cl_S. Therefore, it is very important to understand which CPU between Mot_M and Mot_S will be the Cl_M or the Cl_S. The Cl_M chosen between the two different programmable controllers has to be the Mot_S, because it is not possible to control the dynamics of an axis in an accurate way if the reference time is overwritten during the computation of motion.

3.4.2 Exchange of kinematic data (Communication configuration)

The communication network between the two programmable controllers is based on Ethernet/IP with a Produced/Consumed tag method. This type of communication allows to broadcast and to receive system-shared tags. The choice of this method for the exchange of data is due to the fact that it can be used with typical control networks, it is reliable and it is real-time communication.
The Mot_M can send the synchronization data for multiple axes within a data path. For each axis the shortest data path necessary for the synchronization of the i-th axis (Fig. 3.3) consists of: Time of master commands ($Tmc_i(i)$) and Set Position, Velocity, Acceleration, Jerk ($p_i(i + T_{stt})$, $v_i(i + T_{stt})$, $a_i(i + T_{stt})$, $j_i(i + T_{stt})$) of the axis with a look ahead equal to the Motion Task Time of the Mot_S (T_{stt}).

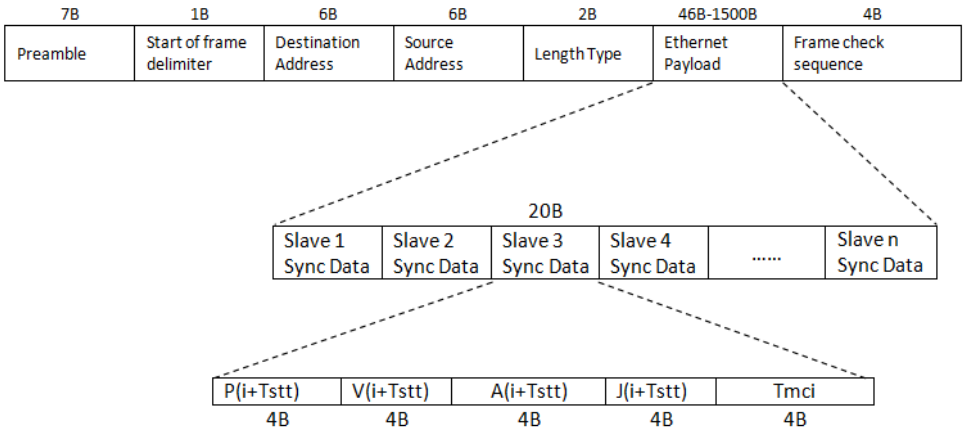


Figure 3.3: Data path sent to the Mot_S by the Mot_M

3.4.3 Computation of trajectory

Different vendors use different methods for the motion of axes. Some of them can directly move an axis point-by-point defining its p_i, v_i and a_i at each motion task. Other vendors, who cannot control a motor point-by-point, use motion instructions in order to impose the desired trajectory. This case is the most interesting because it requires to impose the desired trajectory by using a predefined motion profile. If the axes to be moved are real, the programmable controller computes their p_i, v_i and a_i and sends them to the drivers. In order to synchronize two or more virtual or real axes between two programmable controllers, the Mot_S needs to take into account the p_i, v_i, a_i and j_i of the Mot_M with respect to the instant of time in which the Mot_M will actuate these commands. Thanks to the time synchronization, the two CPUs have the same clock reference and so it is possible to correlate the time instant in both CPUs. Both in the case of the point-by-point control and in the case of the control obtained by means of motion instructions, the coarse update (CUP) of the motion task and the coarse update of the data sending task of the Mot_M must be shorter than the ones of the Mot_S. This choice reduces the possibility of receiving the same data of motion twice because of the jitter of the tasks that different CPUs have. Moreover, the task in which the motion algorithm is computed must be synchronous with the motion task of the CPU to be sure that the actuation delay (T_{ad}) with which the CPU sends the p_i, v_i and a_i of the system is constant and equal to one CUP of the motion task. To compensate the actuation delay that is equal to one CUP of the motion task of the Mot_S ($T_{ad} = T_{stt}$), the kinematic values sent by the Mot_M have a look ahead in time equal to T_{stt} and they are $(p_i(i + T_{stt}), v_i(i + T_{stt}), a_i(i + T_{stt}), j_i(i + T_{stt}))$.
The main idea for the synchronization of the two systems is based on the determination of the time difference (D_i) between the time instant in which the Mot_M imposes the set values on its axes (T_{mc_i}) and the time instant in which the Mot_S will impose the set values on its axes (Tsc_i).
With the computation of this delay and thanks to the kinematic data of the Mot_M, it is possible to extrapolate the future set position (p_{f_i}), future set velocity (v_{f_i}) and future set acceleration (a_{f_i}) that the master axes will achieve after the total time ($T_{t_i} = (D_i + T_{ad})$). The computation of T_{t_i} is afflicted by the jitter of the motion task because the Mot_S can only compute the previous instant of time in which it imposes the commands and not the next one. However, the motion task of the Mot_S is scheduled and so it is almost constant except for the jitter of the task ($T_{t_i} = T_{ad_i} + D_i + jitter_i$). PLCs and iPCs define a maximum level of the jitter of the task that is different among different vendors and different controllers. The communication jitter does not influence the computation because it is included in the D_i value and so it is determined for each calculation accurately.
In order to synchronize the axes of Mot_S with the axes of Mot_M, the Mot_S has to impose on its axes a trajectory congruent to their actual kinematic values to achieve, at the future time instant, the same p_{f_i}, v_{f_i} and a_{f_i} as the Mot_M reaches.
The extrapolated formulas are the following:

$$p_{f_i} = p_i(i + T_{stt}) + v_i(i + T_{stt})D_i + \frac{1}{2}a_i(i + T_{stt})D_i^2 + \frac{1}{6}j_i(i + T_{stt})D_i^3 \qquad (3.1)$$

$$v_{f_i} = v_i(i + T_{stt}) + a_i(i + T_{stt})D_i + \frac{1}{2}j_i(i + T_{stt})D_i^2 \tag{3.2}$$

$$a_{f_i} = a_i(i + T_{stt}) + \frac{1}{2}j_i(i + T_{stt})D_i \tag{3.3}$$

For each motion task the Mot_S has to move its axes satisfying the constraints $p_{f_i}, v_{f_i}, a_{f_i}$ with respect to their actual positions (p_{a_i}), actual velocities (v_{a_i}) and actual accelerations (a_{a_i}).

3.4.4 Drive actuation

The driver can have different control strategies to perform the best control of motors, but it generally uses only the set positions and the set velocities given by the controller. The driver interpolates the set values with a frequency of about 8 kHz and computes the current necessary to impose the defined motion. For this reason all the synchronization algorithms that have been developed have the aim of generating the correct set positions and set velocities required to synchronize two or more axes.

3.5 Simulations

Several simulations have been performed in order to demonstrate the correctness of this architecture. The tests consist in the evaluation of the synchronization error between a Mot_M axis and a Mot_S axis with the trajectories shown in Fig. 3.13 and Fig. 3.17. The method used to synchronize the two axes is the aforementioned one. The task time chosen for the motion task of the Mot_M is 2ms and the one of Mot_S is 4ms. The Mot_M sends the data path to the Mot_S every 1ms, while the Mot_S reads the data path every 2ms. A random jitter has been imposed on different tasks as follows:

- $\pm 5\mu s$ Master Send Data Task
- $\pm 40\mu s$ Slave Read Data Task
- $\pm 40\mu s$ Slave Read Data Task
- $\pm 40\mu s$ Slave Actuation Task

The timing model is shown in Fig. 3.4. The simulation results are shown in Fig. 3.5 and Fig. 3.6.

The simulations lead to the following deductions:

- synchronization errors are bounded
- jitter causes noise on the synchronization error
- it is possible to have few points with a high synchronization error of velocity and acceleration in the case in which the motion task of the Mot_M does not sample the punctual variations of the dynamic variables.

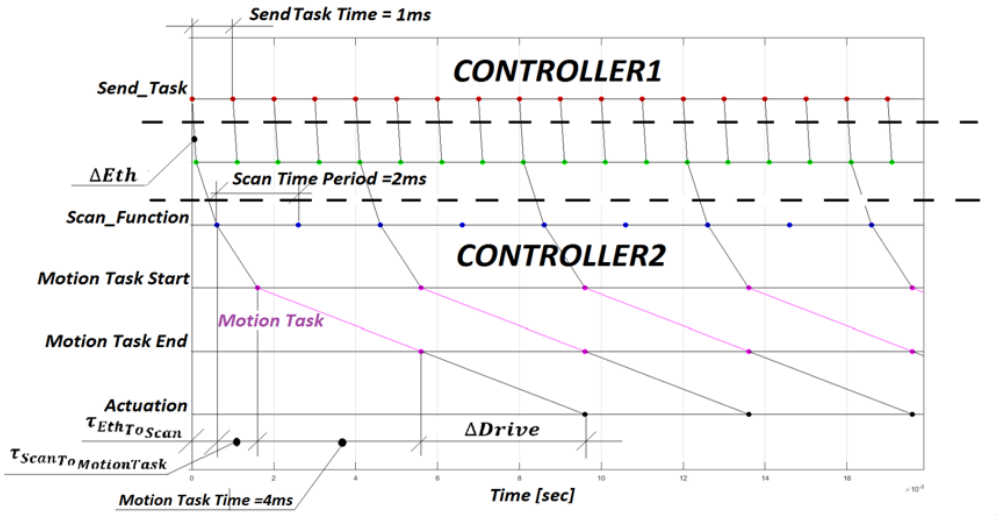


Figure 3.4: Timing model of the system

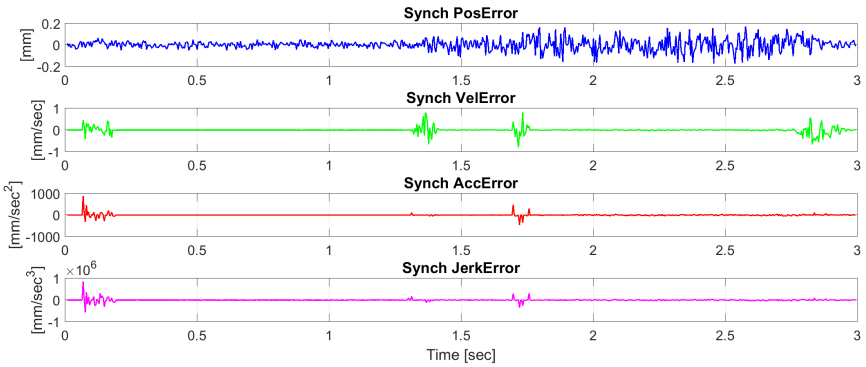


Figure 3.5: Synchronization error with the first trajectory

3.6 Point-by-point control

In case in which the Mot_S uses a point-by-point control, it is possible to directly impose p_{f_i}, v_{f_i} and a_{f_i} on the axes so synchronization is insured except for the jitter of the task.

3.7 Motion instruction control

In case in which the programmable controller can not directly impose the kinematic values on the axes by means of point-to-point control but it can only use motion instructions, it is necessary to use different algorithms to respect the kinematic constraints. In order to develop a global method for the motion synchronization between two different controllers, the motion instruction used to move the axes of the Mot_S is a jog.

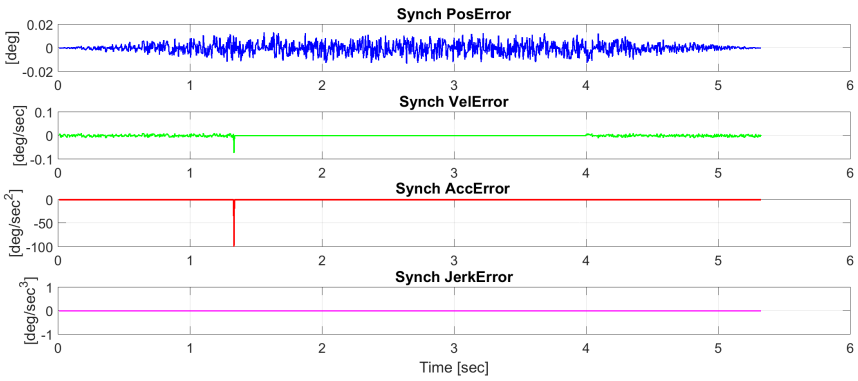


Figure 3.6: Synchronization error with the second trajectory

The choice of this type of motion instruction depends on the fact that the jog is a trapezoidal velocity profile and it is used in any type of automation PLCs or iPCs. Different vendors give different names to this instruction but, independently from the name, its behaviour is the same. The jog instruction allows to carry out a trapezoidal velocity profile (Fig. 3.7) by defining the velocity that the axis has to reach and the acceleration with which it has to obtain the desired velocity.

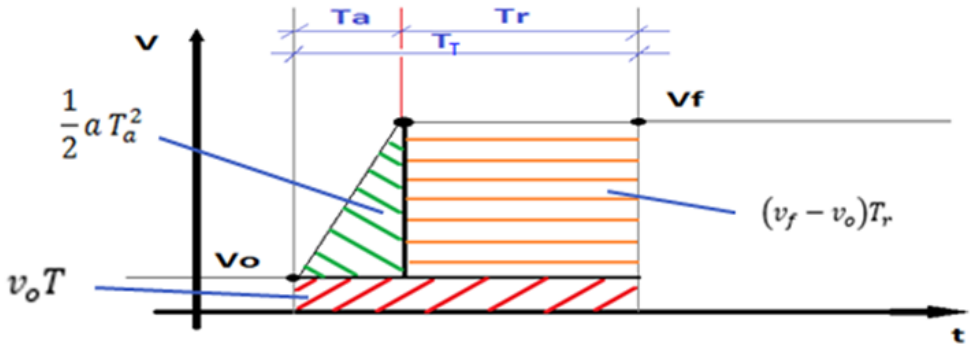


Figure 3.7: Velocity shape of a jog in which the final velocity v_f of the axis is higher than the actual velocity v_0.

In this case it is necessary to divide the problem into three different scenarios that depend on the value of the actual velocity(v_0) of the Mot_S axis with respect to the extrapolated velocity(v_f) which the axis has to reach at the next motion task time: $(v_f < v_0)$, $(v_f = v_0)$, $(v_f > v_0)$.

In case $v_f > v_0$, the equations describing the problem are:

$$p_f = p_0 + v_0 T_t + \frac{1}{2} a T_a^2 + (v_f - v_0) T_r \tag{3.4}$$

$$v_f = v_0 + a T_a \tag{3.5}$$

$$T_t = T_a + T_r \tag{3.6}$$

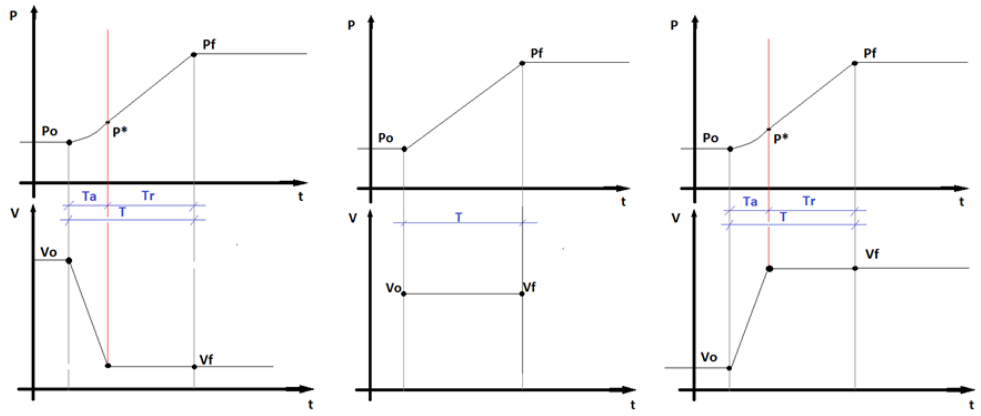


Figure 3.8: Problem scenarios $(v_f < v_0)$, $(v_f = v_0)$, $(v_f > v_0)$.

where T_a is the time of acceleration, T_r is the time in which the axis has the desired velocity and T_t is the total time. So the solution is the following:

$$T_a = 2\frac{(p_f - p_0) - v_f T_t}{v_0 - v_f} \tag{3.7}$$

$$a = \frac{(v_f - v_0)}{T_t} \tag{3.8}$$

In case $v_f < v_0$, the solution is similar to the previous one. In the case in which $v_f = v_0$, the unique solution is the following:

$$p_f = p_0 + v_0 T_t \tag{3.9}$$

$$a = \frac{v_f - v_0}{T_t} \tag{3.10}$$

The solutions, which also depend on the values of T_a and T_t, lead to four different cases:

- $0 < T_a < T_t$, the solution always exsists;
- $T_a = 0$, one and only one solution exists and it is not implementable:

$$(p_f - p_0) = v_f T_t \tag{3.11}$$

$$\frac{\Delta p}{T_t} = v_f \tag{3.12}$$

- $T_a < 0$, it is impossible to satisfy the position and the velocity constraints at the same time:

$$(p_f - p_0) - v_f T_t < 0 \tag{3.13}$$

$$\frac{\Delta p}{T_t} < v_f \tag{3.14}$$

- $T_a > T_t$, it is impossible to satisfy the position and the velocity constraints at the same time:

$$(p_f - p_0) > v_0 T_t \tag{3.15}$$

$$\frac{\Delta p}{T_t} > v_0 \tag{3.16}$$

Only in the first case there is always a solution. In the second case there are no implementable solutions because it is necessary to give a new velocity to the axis without any variation of acceleration. In the third case and in the fourth case it is not possible to respect all the constraints so it is necessary to develop an approximated solution. In order to achieve the best synchronization performances with respect to the computational effort, two different algorithms have been developed:

- Discrete approximation
- Linear approximation

3.7.1 Discrete approximation

With this algorithm, in case $T_a = 0$, the input velocity and acceleration given to the jog are the same as in the previous task. In this way the Mot_M and the Mot_S axes increase their synchronization difference and in the next task T_a will be different from 0. The synchronization error is not high if the time of the motion task is shorter than the dynamics of the Mot_M axes. In the third case ($T_a < 0$), the acceleration time is negative. In the fourth case ($T_a > T$) the acceleration time is higher than the motion task of the system, the inputs of the jog instruction are defined as follows:

$$v_{jog} = \frac{p_f - p_0}{T_t} \tag{3.17}$$

$$a_{jog} = \frac{v_f - v_0}{T_t} \tag{3.18}$$

This approximation involves a position error and a velocity error (Fig. 3.9).

To understand if the approximation error is bounded and small enough to allow the application of the algorithm, the position error and the velocity error are computed in the following three cases:

- $v_f < v_0$

$$e_p = \Delta p - \frac{v_f + v_0}{2} T \tag{3.19}$$

$$e_v = 0 \tag{3.20}$$

- $v_f = v_0$

$$e_p = \Delta p - \frac{v_f + v_0}{2} T \tag{3.21}$$

$$e_v = 0 \tag{3.22}$$

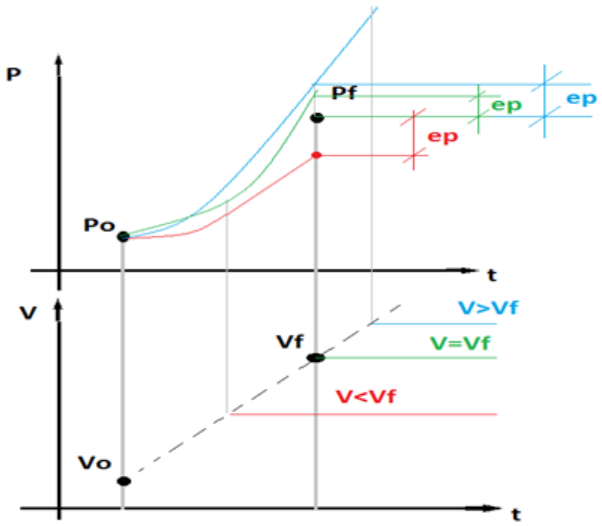


Figure 3.9: Position error due to the approximation in the cases ($v_f < v_0$), ($v_f = v_0$), ($v_f > v_0$).

- $v_f > v_0$

$$e_p = \frac{v_0^2}{2\Delta v}T + \frac{\Delta p^2}{2T\Delta v} - v_0\frac{\Delta p}{\Delta v} \tag{3.23}$$

$$e_v = v_f - \frac{\Delta p}{T} \tag{3.24}$$

Different simulations have been carried out in order to evaluate the position error with respect to Delta position ($p_f - p_0$) and the final velocity v_f.
The function is computed in the neighbourhood of the work point with a velocity of about 133,33 deg/sec (Fig. 3.10).

It presents a saddle point in the working point with a position error of about zero, in the other parts the position error is close to zero. The highest position error can be achieved only if Delta velocity and Delta position are not congruent to each other, consequently only if the kinematic values of the Mot_M axes received by the Mot_S are corrupted.
Another possible problem can be present if the extrapolated velocity is perfectly equal to the actual velocity. In this case the algorithm does not correct any velocity error and the two axes have the same velocity but a very high position error.
In order to solve this problem, a very little reduction in velocity is imposed with an acceleration equal to 1% of the actual acceleration of the system. In this way a difference between the extrapolated velocity and the actual velocity of the axes appears in the next motion task. This solution does not produce a high position error if the system has a short motion task time. Fig. 3.11 represents the complete algorithm.

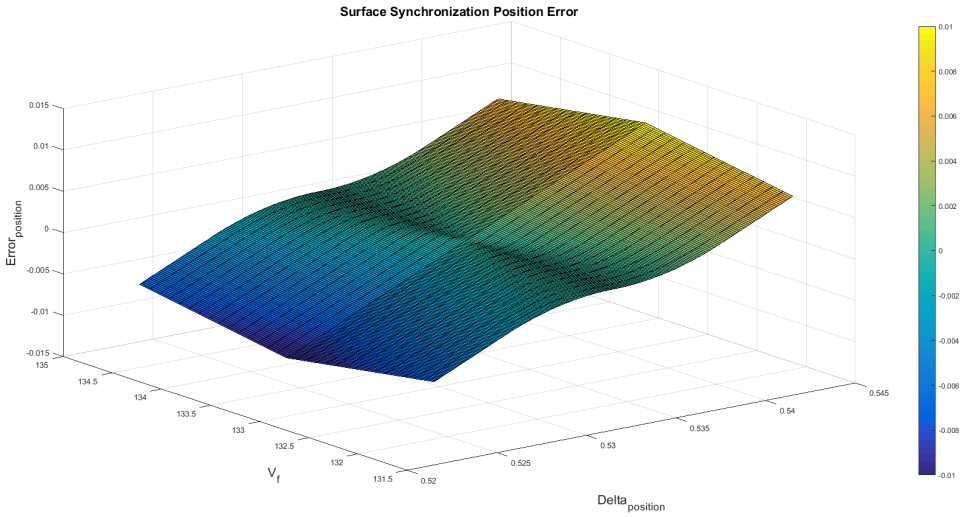


Figure 3.10: Surface of the synchronization error.

This algorithm has to be computed at each motion task time for each axis to be synchronized. Not all PLCs have a good performance with complex calculations, therefore a simplified algorithm has been developed.

3.7.2 Linear approximation

The target of this algorithm is to reduce the complexity of computation maintaining good synchronization performances. For each task time this algorithm computes the time difference (D_i), it extrapolates p_f,v_f and a_f and it calculates, by means of Eq. 3.17 and Eq. 3.18, the input velocity and acceleration to be given to the jog instruction.
The algorithm corrects the position error and the velocity error existing between the actual kinematic values and the extrapolated kinematic values in order to synchronize with the master axis.
However, the algorithm does not distinguish the different kinematic scenarios consequently there is always an approximation error. But, as seen before, the position error is very small if Delta position and Delta velocity are congruent and the computational effort is reduced. This algorithm can be a good trade off between the synchronization performances and the computational efforts if the computational power of the CPU is not too high.

3.8 Experimental results

The tests have been performed only with PLCs and iPCs that cannot control the kinematics of the axes point-by-point. This choice is due to the fact that in the case of point-by-point control it is possible to synchronize the axes by directly imposing the extrapolated position and velocity of the Mot_M axes. In the case taken into

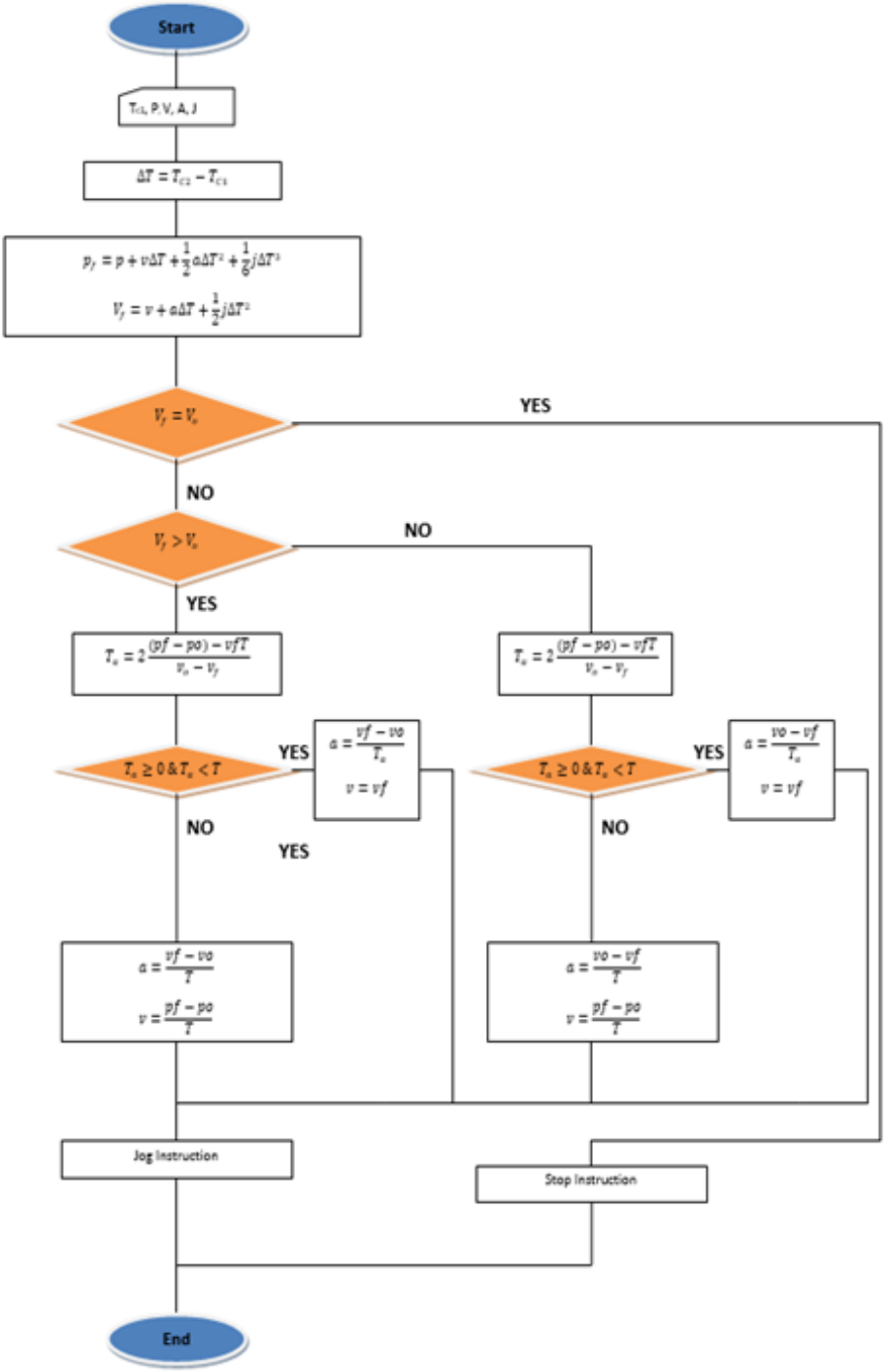


Figure 3.11: Discrete approximation algorithm flow chart.

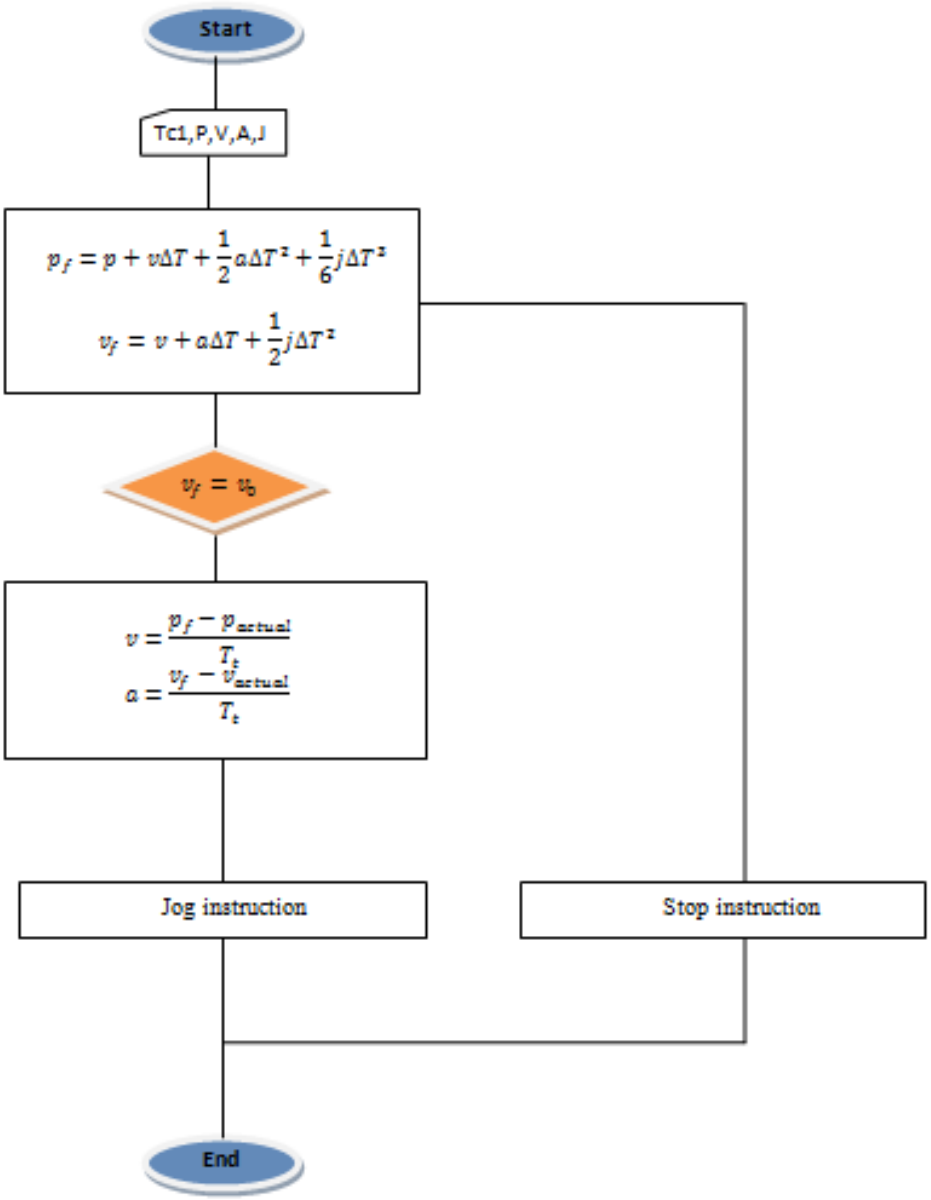


Figure 3.12: Linear approximation algorithm flow chart.

consideration by this study, the time difference between the two systems has to be computed in order to extrapolate p_f,v_f and a_f and to use the Discrete Approximation Algorithm or the Linear Approximation Algorithm required to compute the input velocity (v_{jog}) and the input acceleration (a_{jog}) for the jog instruction and to synchronize the axes.
The tests have been run on virtual and real axes.
The tests on real axes have been conducted in order to understand if the synchronization error can influence the position error or produce vibrations on real motors.
The tests on virtual axes have been performed in order to evaluate the synchronization position error and the velocity position error with respect to different motion profiles. Other tests on virtual axes have been run in order to determine how the variation of the task time of the two CPUs can influence the performances of the algorithms.

3.8.1 Tests on real motors

In order to understand if the algorithms can generate noise on real motors, the position error and the velocity error of the distributed hybrid system have been evaluated by comparing them with the position error and the velocity error of the same motor commanded only by one CPU. The motion profile chosen for the test is shown in Fig. 3.13; it consists of a hard dynamics with several parts of accelerating zones and constant velocity zones. The position error and the velocity error are the same both in the normal case and in the hybrid case.

3.8.2 Tests on position and velocity synchronization errors

In order to evaluate the synchronization performances of the two algorithms at different velocities and with virtual axes, two motion profiles have been used: the first one is represented in Fig. 3.13, while the second one is represented in Fig. 3.17.
For the sake of brevity, Fig. 3.15 and Fig. 3.16 show only one test; the variables represent the position synchronization error and the velocity synchronization error. However, the data of all the tests are listed in Tab 3.1.

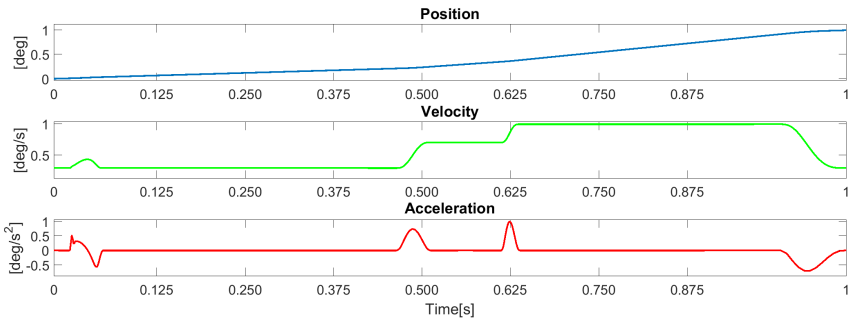


Figure 3.13: First motion profile.

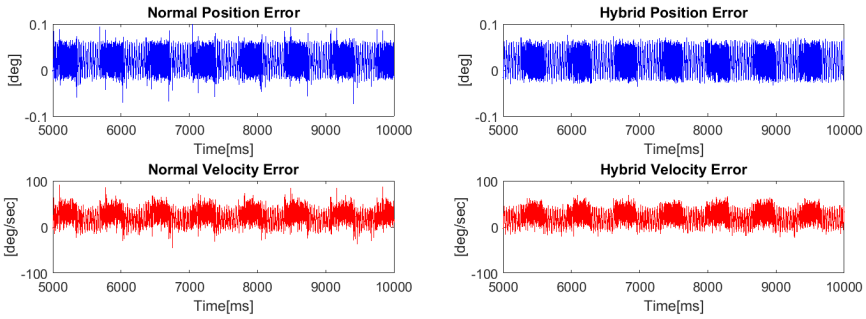


Figure 3.14: Position error and velocity error in the normal and hybrid configurations.

- First motion profile

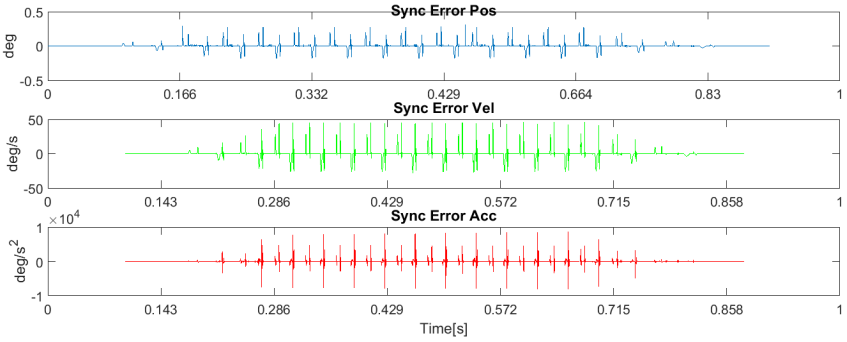


Figure 3.15: Synchronization error with the Discrete Approximation Algorithm for the first motion profile.

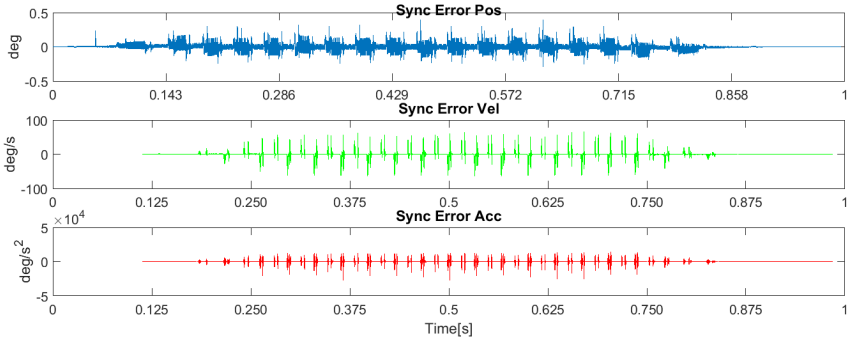


Figure 3.16: Synchronization error with the Linear Approximation Algorithm for the first motion profile.

- Second motion profile

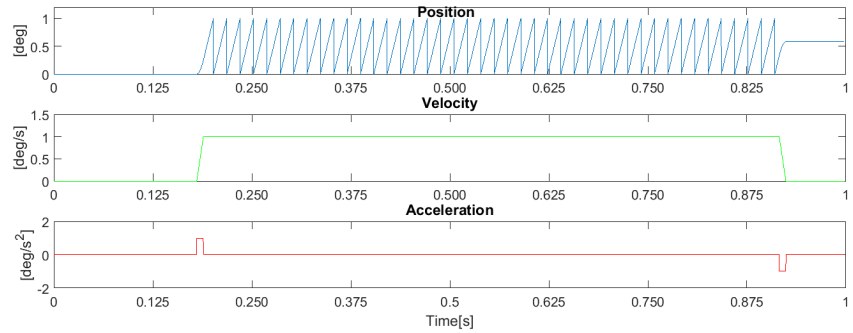


Figure 3.17: Second motion profile.

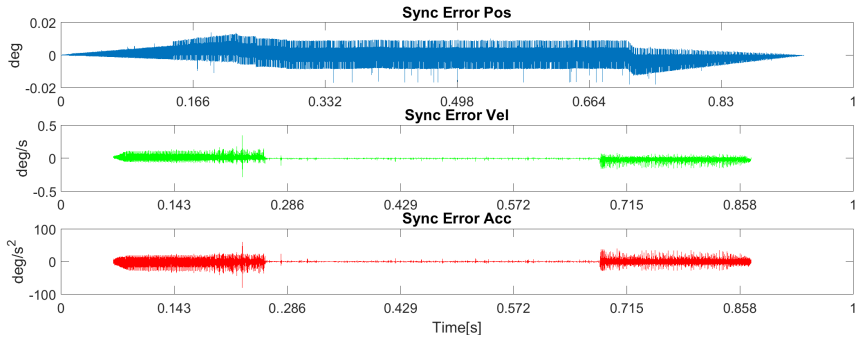


Figure 3.18: Synchronization error with the Discrete Approximation Algorithm for the second motion profile.

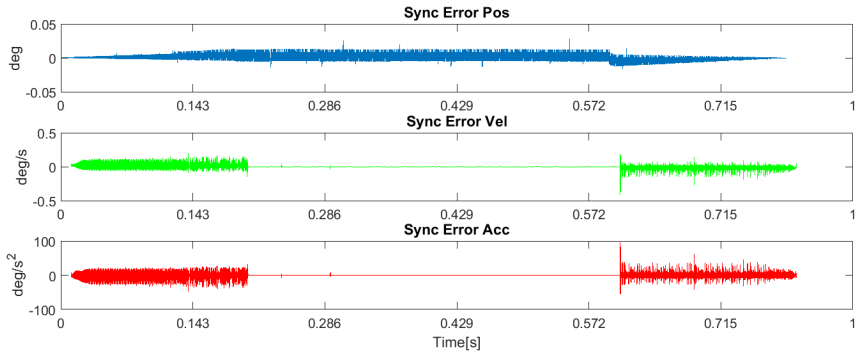


Figure 3.19: Synchronization error with the Linear Approximation Algorithm for the second motion profile.

3.8.3 Influence of the task time

An important factor for the motion control is the task time of the motion planner of CPUs. The following experiments have been conducted in order to understand if the variation of the task time influences the performances of the algorithms. The

tests have been performed for three different task times of the motion planner of the Mot_S:

- 8 ms
- 4 ms
- 2 ms

The task time of the Mot_M is always half of the task time of the Mot_S. The profile used is the same as in Fig. 3.13 at three different velocities:

- 67 deg/sec
- 133.33 deg/sec
- 200 deg/sec

The data of all the tests are shown in Tab. 3.2.

3.9 General Results

To solve the synchronization problem two possible cases have been taken into account:

- point-by-point control
- axis control by means of motion instructions

In the first case it is possible to synchronize the axes by means of a simple compensation of the time delay between the time in which the Mot_M commands its axes and the time in which the Mot_S performs the same operation. In the second case a trapezoidal velocity profile called jog instruction has been chosen in order to control the Mot_S axes and to synchronize them with the Mot_M axes. This choice has been made because in any PLC or iPC used for the motion control it is possible to define a trapezoidal velocity trajectory by imposing on the axis the velocity and the acceleration to be reached.
Therefore, two different algorithms have been developed for the solution of the synchronization problem. The main difference between the two algorithms depends on the fact that the Discrete Approximation Algorithm takes into account all the possible kinematic cases of the axes, but it involves a higher computational effort.
On the contrary, even if the Linear Approximation Algorithm has the same behavior in any kinematic case, it requires a lower computational effort.
In order to optimize the algorithms, some tests have been conducted with different task times of the motion planner as the task time is an important factor for the synchronization accuracy. De facto, the shorter the motion task time is, the faster the generation of the command values sent to the axes is. Consequently, the system is more reactive to the correction of the synchronization error and the decrease of

the task time reduces the mean and especially the standard deviation of the synchronization error.
These tests have brought to light another parameter that influences the performances. It is the jitter of the motion task of the Mot_S; in order to evaluate its influence, several tests have been conducted on virtual axes following the profile of Fig. 3.13. The data are shown in Fig. 3.20.

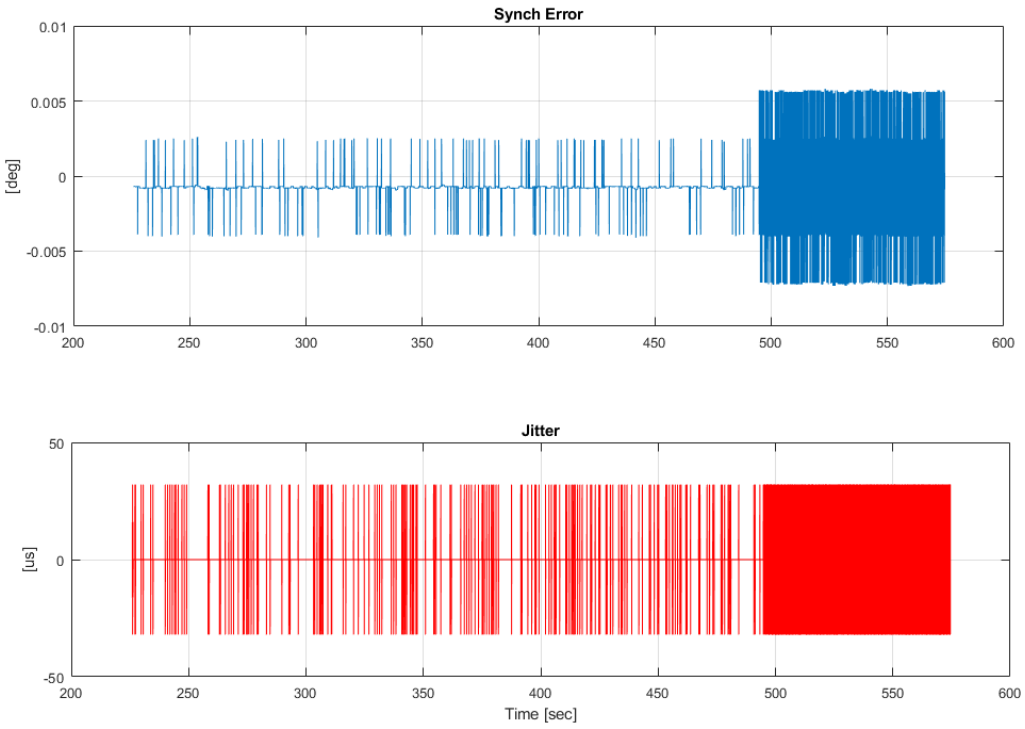


Figure 3.20: Comparison between the synchronization error and the jitter of the Mot_S motion task.

Actually a high variation of the jitter involves a higher synchronization error. Even the magnitude of the jitter influences the system: for a high magnitude of the jitter there is an increase in the synchronization error. The problem is that it is not possible to forecast the jitter and to compensate it because it has a randomic behaviour. Therefore, in order to reduce its effects, there are two possibilities:

- to choose a CPU with a very low jitter
- to reduce the task time of the motion planner of the Mot_s so that the magnitude of the jitter can be reduced.

3.10 Conlusions

The first step of this work has involved a wide research on the most used systems of real-time communication and synchronization. For the time synchronization of

different CPUs it has been necessary to individualize the hardware and software tools adopted by most vendors of programmable controllers. The second step has regarded a research aimed to define the communication method to be used for the exchange of the kinematic data of the axes between two CPUs.
Two synchronization algorithms have been developed in order to solve the synchronization problem. Several simulations have been carried out with a defined trajectory of the axes in order to evaluate the synchronization performances of the system. They have shown a position error close to zero along the working point. Subsequently, the two algorithms have been tested with real and virtual axes. As regards real axes, it emerges that the use of synchronization algorithms does not influence the position error and the velocity error of real motors. As regards virtual axes, several synchronization tests have been run with different motion profiles and at different velocities in order to evaluate the flexibility of the algorithms. They have shown that the synchronization error is bounded; it depends on the velocity of the axes and on the motion profile; magnitude is one or two orders lower than a common position error of a real motor and it has a tight standard deviation. It means that the algorithms reach good performances and in case of real motors, the synchronization error is neglectable.
In the tests conducted with the second trajectory, the synchronization position error of the axes is close to the synchronization position error of the simulations performed with the same profile. Furthermore, a good model of the system has been developed.

Table 3.1: Synchronization performances

		Discrete approximation				**Linear approximation**			
		P		**V**		**P**		**V**	
		Mean	**Std. Dev.**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
Profile 1	**50 deg/s**	0.001	0.0023	-5E-06	0.016	0.0003	0.0021	9E-06	0.0126
	133 deg/s	8E-05	0.0065	9E-06	0.03	0.0019	0.0073	2E-06	0.03
	200 deg/s	2E-05	0.0083	3E-06	0.0332	0.0023	0.0097	1E-06	0.0324
Profile 2	**50 deg/s**	0.0003	0.0249	2E-04	1.374	0.0019	0.023	5E-05	1.121
	133 deg/s	4E-04	0.0652	0.006	7.4421	5E-04	0.0769	-6E-04	7.6379
	200 deg/s	0.0015	0.0716	4E-04	15.641	-0.003	0.277	0.0053	16.658

Table 3.2: Influence of the task time on the motion synchronization

		Discrete approx.		Linear approx.	
		Mean	**Std. Dev.**	**Mean**	**Std. Dev.**
2 ms	**67 deg/s**	4.329E-2	4.140E-2	4.885E-2	4.939E-2
	133 deg/s	-2.00E-3	4.068E-2	9.372E-2	1.001E-1
	200 deg/s	-9.905E-4	6.273E-2	7.839E-2	1.255E-1
4 ms	**67 deg/s**	1.289E-2	4.029E-2	1.860E-2	4.353E-2
	133 deg/s	2.437E-2	8.316E-2	3.337E-2	9.599E-2
	200 deg/s	3.397E-2	1.318E-1	4.562E-2	1.644E-1
8 ms	**67 deg/s**	-1.057E-4	3.135E-2	-6.308E-3	5.928E-2
	133 deg/s	-4.168E-2	2.342E-1	-3.671E-2	2.321E-1
	200 deg/s	-1.064E-2	5.615E-1	3.634E-2	6.542E-1

Chapter 4

Data-driven approach for Independent Carts System

4.1 Introduction

This chapter deals with a methodology for the training of two Machine Learning algorithms for fault detection and identification in Independent Carts System. This study verifies the feasibility of using a CBM data-driven approach instead of a time-based maintenance method. De facto, the current method of maintenance used in the Independent Carts System under examination is based on the replacement of all the bearings of all the carts every 10.000 hours of production and, in case of unexpected problems, the machine stops and the faulty cart has to be changed. The system taken into account is an XTS Beckoff System and the use of Machine Learning algorithms aims to predict the time when a single cart has to be replaced because of a damage on one of the three bearings mounted on it. Therefore, the aim of the algorithms used is to classify the healthy and the faulty state of the system and in some cases to classify also the type of damages in the bearings. The training is supervised, in fact all the training data are labelled. The Machine Learning models used are:

1. Random Forest;
2. Support Vector Machine.

The metrics used for the evaluation of the two models are accuracy, precision and recall, they are the most used criteria for the comparison of machine/deep learning algorithms. The classical nomenclature is described in the confusion matrix here below.

The metrics used are the following:

1. Accuracy: it is the total number of the examples of the entire confusion matrix that are classified as correct, divided by the total number of examples. It identifies the response of the algorithm both to healthy cases and to faulty cases.

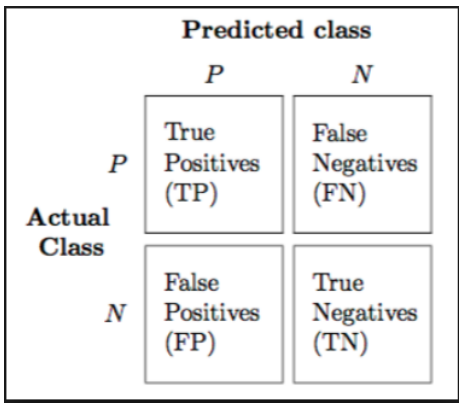


Figure 4.1: Confusion Matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

2. Faulty Recall: it is the number of the fault examples that are classified as correct, divided by the total number of fault examples. It identifies how good the algorithm is to detect damages in its presence.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

3. Faulty Precision: it is the number of the fault examples that are classified as correct, divided by the total number of fault examples, which are classified as faulty. It identifies how good the algorithm is to identify damages without giving false alarms.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

4. Receiver Operating Characteristic Curve (ROC Curve Fig. 4.2): it is a graph showing the performances of a classification model at all classification thresholds. It is based on two parameters:

   True Positive Rate (TPR)

$$TPR = \frac{TP}{TP + FN} \tag{4.4}$$

   False Positive Rate (FPR)

$$FPR = \frac{FP}{FP + TN} \tag{4.5}$$

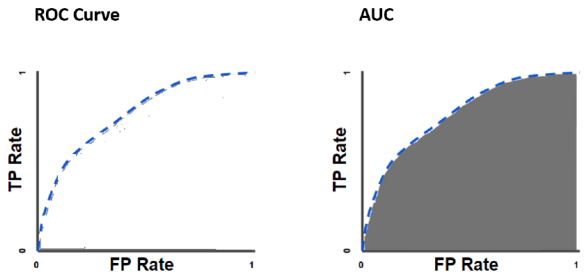


Figure 4.2: ROC and AUC score graphs.

		True Health Condition:	
		Healthy (H_0)	*Damaged*
CBM Actions	*accept H_0: Healthy*	No Alarm	**Missed Alarm** ***type II error***
	reject H_0: Damaged	**False Alarm** ***type I error***	Alarm

Figure 4.3: Confusion Matrix in Condition Monitoring analysis .

In fault detection, the nomenclature can change as described in Fig. 4.3.

In this case, H_0 corresponds to the healthy state of the machine that the algorithm identifies as healthy; Alarm corresponds to the case in which there is a damage in the machine that is detected; Missed Alarm (MA) corresponds to the case in which there is a damage in the machine but it is not detected and False Alarm (FA) corresponds to the case in which the machine is healthy but the algorithm detects a damage.

This type of nomenclature is used in order to identify the operational cases better. Moreover, an operational cost is defined for each class in industrial plants. For example, in the case taken into consideration in this chapter the operational cost of a missed alarm and a downtime of the machine is very high, while the cost of a false alarm is lower due to the possibility of maintenance control during the normal stop of the machine.

4.2 Random Forest

The Random Forest (RF) algorithm was developed by Breiman [98] in 2001. The Random Forest is a machine learning model based on tree bagging. Bagging is a machine learning ensemble meta-algorithm designed to improve stability and accuracy; it is used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. The Random Forest consists of an ensemble of simple decision-tree predictors, each of which gives a class prediction as output and the class that has the largest number of votes becomes the prediction of the model (see

Fig. 4.4).

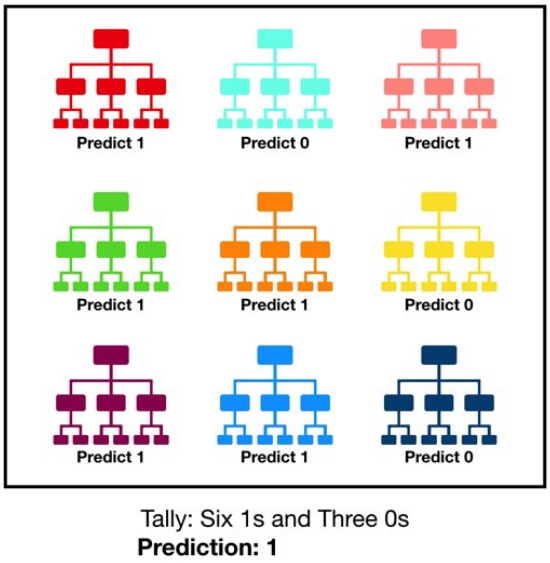


Figure 4.4: Graphical representation of Random forest from [99].

In the nodes of the trees there are thresholds based on one or more features that decide if the data must proceed to the left or to the right of the tree. On the contrary, as regards the leaves, the probabilities are calculated on the basis of the elements of each class that end up in a given leaf. As regards classification problems, the ensemble of simple trees vote for the most popular class. As regards regression problems, the responses of the trees are averaged to obtain an estimate of the dependent variables. Using tree ensembles can lead to a significant improvement in prediction accuracy.

Pros:

- The use of multiple decision trees reduces the problem of overfitting by averaging or combining the results,
- It has a smaller variance than a single decision tree,
- Flexibility and accuracy,
- It maintains accuracy even with missing data.

Cons:

- The construction of Random Forest is harder and more time-consuming than the construction of a decision tree,
- More computational resources are required and they are even less intuitive. With a large collection of decision trees it is hard to have an intuitive grasp of the relationship existing in the input data,

- The prediction process using Random Forests is more time-consuming than other algorithms.

4.3 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for regression or classification. If it is used as a classifier, it finds a separation line in the feature domain that has the maximum distance from the predefined classes. Fig. 4.5 illustrates an example of how SVM works for a linear classification. It shows the case in which a dot has to be classified into two different classes (black or white) on the basis of two input features (X_1, X_2). SVM computes the separation line by maximizing the distance of the closest elements of the two classes and by minimizing the possibility of choosing a separation line that is more favourable to one of the two classes.

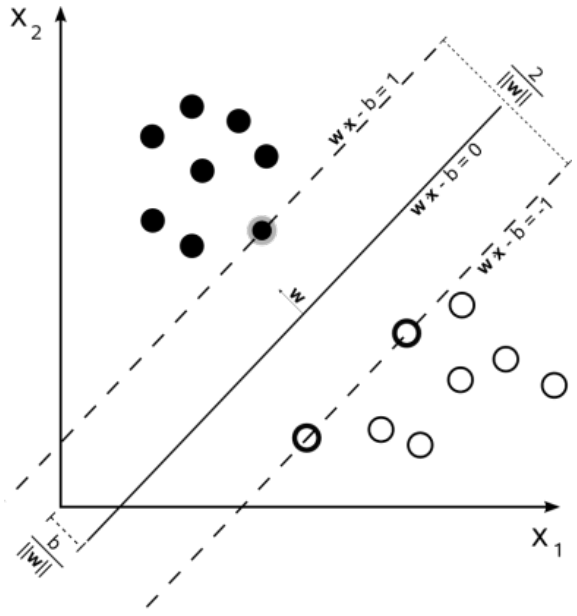


Figure 4.5: Support Vector Machine from [100].

SVM classifier can be extended to a non-linear separation hyperplane by means of kernel functions, multi-dimensional feature arrays and multi-class classification. Detailed information can be found in several books on SVM, for example in [101, 102, 103]. The core part of a machine learning tool is the definition of the input feature array. The number of features has to be reduced in order to avoid computational burden, but it has to be significant to characterize the different classes which the data should be divided into. In literature, several parameters have been proposed for the condition monitoring of ball bearings [104, 27, 105].

Pros:

- Good accuracy

- It works on smaller and cleaner data sets well
- It can be more efficient by using a subset of training points
- It can be visually interpreted.

Cons:

- It finds difficulties in working with large datasets
- It is less effective on noisier datasets with overlapping classes.

The Random Forest and Support Vector Machine algorithms have been trained for the fault monitoring of two different test rigs that are:

1. Rotary motor test rig
2. Independent Carts XTS Beckoff rig.

4.4 Rotary motor test rig

This test rig consists in a Hepco bearing, with the dimensions showed in Tab. 5.1, mounted on a rotary Beckoff servomotor AM8022-0D20-0000. The bearing is preloaded with a pneumatic jack Festo ADVC-25-10-I-P-A. All the tests executed with this test rig have been run at constant velocity and with three different preloads (0.2 bar 0.4 bar 0.6 bar). Fig. 4.6 shows the system. The system has two accelerometers IFM VSP001 placed on a flange of the test rig at 90deg from each other with a sampling frequency of 20 kHz. The velocity and the current of the servomotor are recorded.

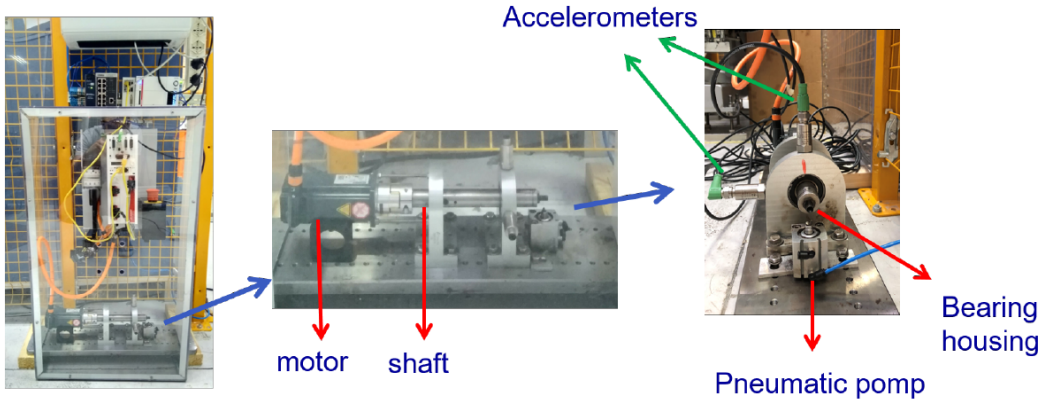


Figure 4.6: Rotary motor test rig.

4.5 XTS Beckhoff test rig

The test rig is made up of Beckhoff XTS system (Fig. 4.9) with GXF Hepco guidance and the cart shown in Fig. 4.8. The system taken into account consists of two circular paths that are 0.500 m long and an upper linear path and a lower linear path, each of which is 1m long. For the length of the machine, 8 straight motors and 2 curved motors are necessary. Twelve carts have been mounted on the track and in every test they follow the motion profile shown in Fig. 4.7.

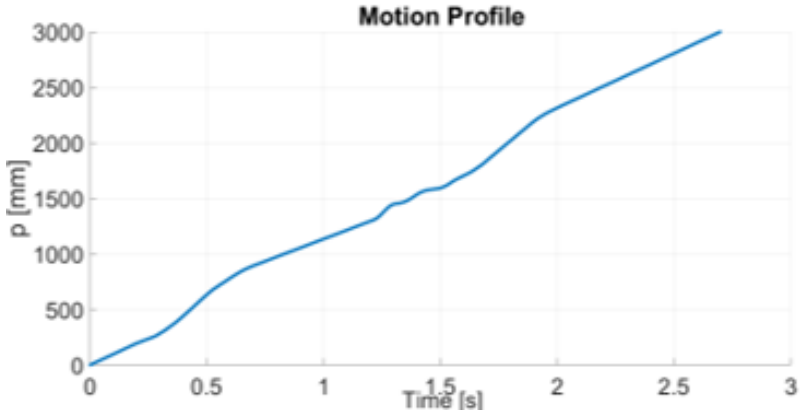


Figure 4.7: Cart motion profile

Their motion profile highlights that the carts vary their velocities along the whole path but they move from 1500 mm to 2000 mm at a constant velocity.

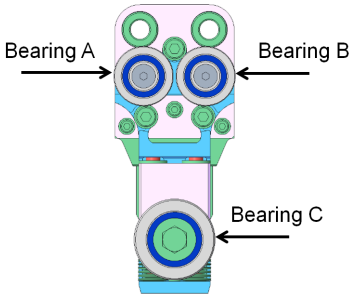
4.5.1 Method workflow

The machine under test is not yet in the field, consequently no case of faults is known at the moment. As regards Machine Learning, this could be a big obstacle because these algorithms need a lot of data referring to health and fault states to be trained. On the other hand, the market requires a native CMB system in the machine. The idea is to train these algorithms with artificial fault data in order to have a basic CMB system that will be improved in time thanks to the new data produced by the machines in the field. For the development of the CBM system for Independent Carts Systems, a standard procedure has been designed respecting the previous hypotheses and restrictions. It consists of the following steps:

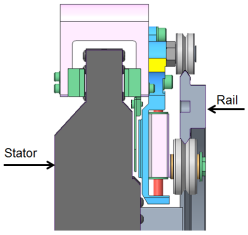
1. Define the most probable damages that can affect the bearings under consideration and replicate the damages artificially.

2. Define the system variables to be recorded for fault detection and identification.

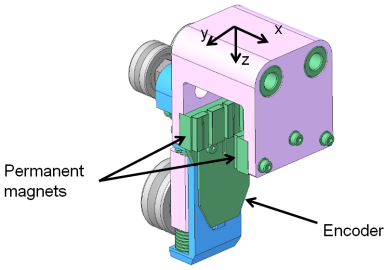
3. Verify the observability of the artificially damaged bearings by using a rotary motor test rig highly monitored and try to understand if the artificial damages are hazardous for the Independent Carts Beckhoff test rig.



(a) Bearings of the mover



(b) Mover connected to stator and rail



(c) Mover with coordinates

Figure 4.8: XTS system with mover rail connection.

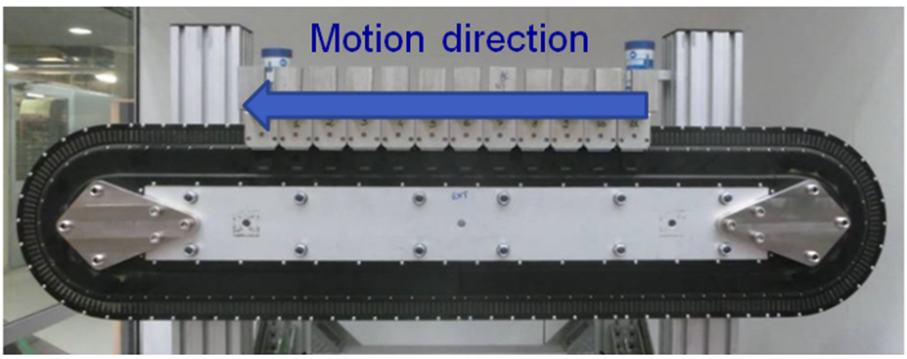


Figure 4.9: XTS Beckhoff test rig.

4. Use the artificially damaged bearings in the Independent Carts System in order to create a data training set.
5. Build Machine Learning models by using the data training set.
6. Create a test set on Independent Carts System with the artificially damaged bearings.
7. Validation Loop 1: validate the Machine Learning models with the test set.
8. Validation Loop 2: re-train and validate the Machine Learning models with the know-how extracted from Validation Loop 1.

4.6 Definition and creation of damages

Because of the newness of the Independent Carts Systems and the absence of these machines in a real production plant, there is not yet any knowledge of real damages of the bearings for the XTS system.
For this reason, in order to develop a data-driven monitoring system, fictitious damages have been created. They are as similar as possible to real damages that can occur in the field. The damages created are the following:

- Rusty damage: it is created by immersing the bearing into a solution of water and salt for one week. It is considered as a distributed damage.
- Inner race damage: it is created by drilling the inner ring with a tip of 0.2 mm. It is the lightest damage and it is punctual.
- Outer race damage: it is created by cutting the outer ring of the bearing. It is a serious damage and it is punctual.
- Blockage damage: it is created by blocking the sphere of the bearing with rust and metals. It is the most serious damage; it is also very dangerous for the rail, if it is not recognized quickly.

Figure 4.10: Damaged bearing images, from the left to the right: (1) Rusty bearing, (2) Bearing with an inner ring damage, (3) Bearing with an outer ring damage, (4) Blocked bearing.

After the creation of these four types of damages, it has been observed that the rusty bearing could not rotate. For this reason, rust has been removed and the bearing has been lubricated in order to give it the possibility of spinning. In the image below, it is possible to see the transformation.

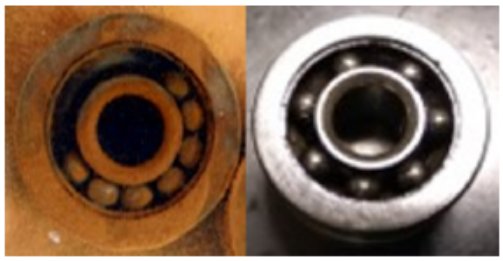


Figure 4.11: From the left to the right: (1) Bearing during water treatment (2) Rusty bearing after the removal of the rust.

4.6.1 Definition of the system variables under control

This chapter defines which variables have been taken into account for the two test rigs where the artificially damaged bearings have been tested. In the Rotary Test Rig the variables recorded are:

1. The vibrations of the lateral and upper accelerometers recorded with a sampling frequency of 20 kHz each.
2. The current of the servomotors recorded with a sampling frequency of 4 kHz.

In the XTS Beckhoff Test Rig the variables recorded are:

1. The vibrations given by the two accelerometers, one of which is placed on the top part of the frame and the other one on the bottom part. The piece of information is not linked to each mover and the sampling frequency for each sensor is 20 kHz.
2. The current referred to each cart with a sampling frequency of 4 kHz.
3. The velocity referred to each cart with a sampling frequency of 4 kHz.
4. The position error referred to each cart with a sampling frequency of 4 kHz.
5. The actual position referred to each cart with a sampling frequency of 4 kHz.
6. Beckhoff state parameters Q1, Q2 Q3 referred to each cart. They are black-box parameters given by Beckhoff for the monitoring of the healthy carts, each parameter is generated every seven minutes of the machine running.

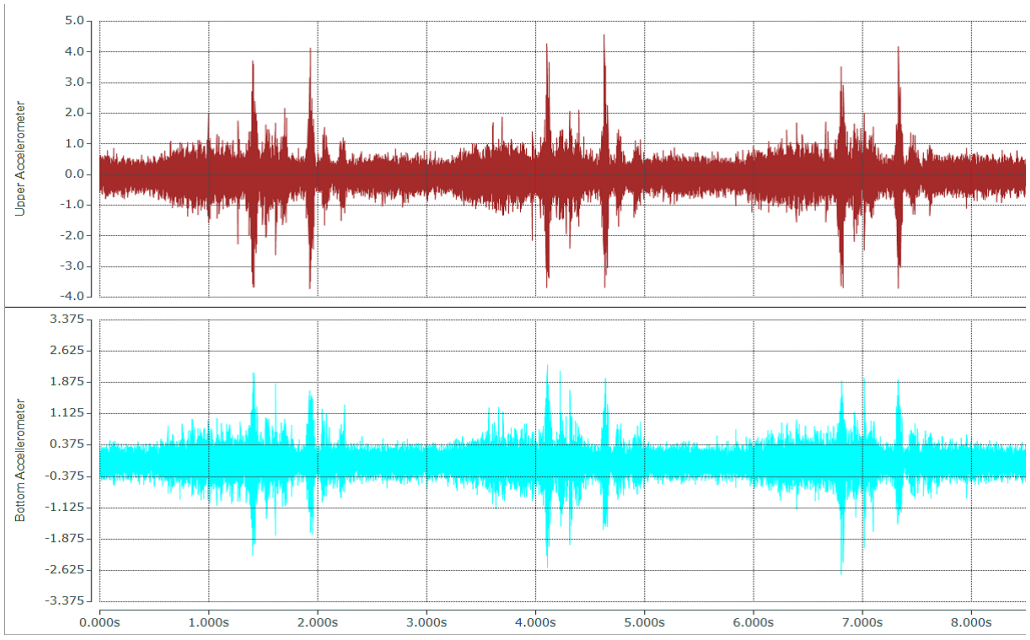


Figure 4.12: Signal of vibrations from upper and bottom accelerometers.

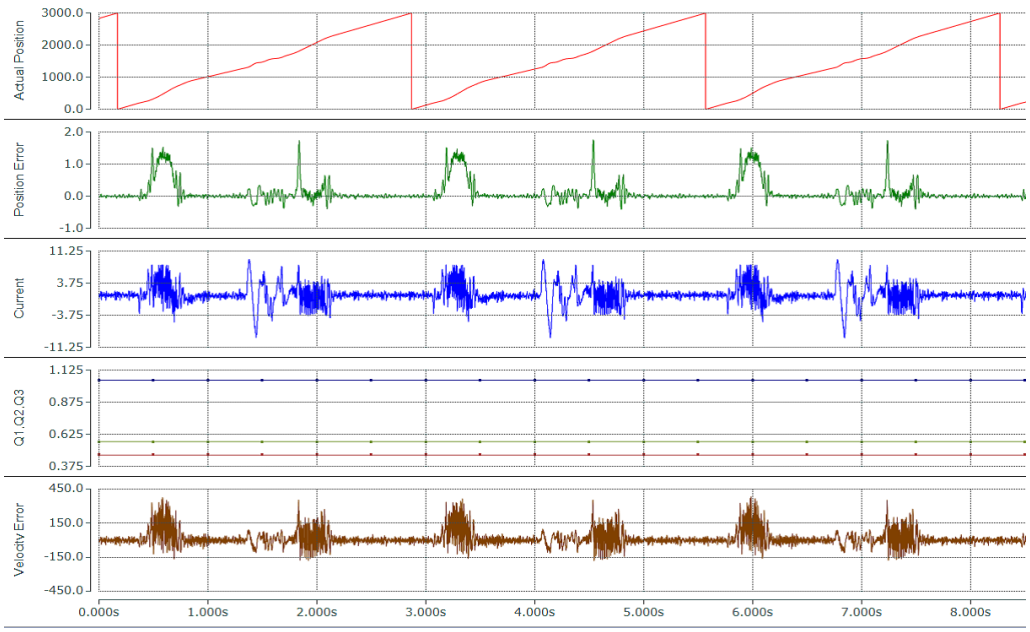


Figure 4.13: Signals of actual position, position error, current, Q1, Q2, Q3 and velocity error.

4.6.2 Verification of the observability of damages by using Rotary Motor Test Rig

The aim of this phase is to guarantee the correctness of the artificial damages in a simplified test rig. The term correctness means the possibility of observing the damages with machine learning algorithms in a simpler case. This stage also helps to check if the artificially damaged bearings are too dangerous for the entire Independent Carts System. In this case, the simplified tests consist in the analysis of the inner and outer damaged bearings and of the healthy one in the aforementioned rotary motor test rig. The tests carried out are the following:

- Healthy bearing with a load imposed of 2 bar.
- Healthy bearing with a load imposed of 4 bar.
- Damage on the outer ring with a load of 2 bar imposed on the bearing.
- Damage on the outer ring with a load of 4 bar imposed on the bearing.
- Damage on the inner ring with a load of 2 bar imposed on the bearing.
- Damage on the inner ring with a load of 4 bar imposed on the bearing.

For each type of damage, three data records have been made. Each test has followed the implementation procedure explained below:

1. Setting of the damaged bearing in the test rig.
2. Warming-up of the test rig for twenty minutes without any logging.
3. Forty-second recording of vibrations and current values for three times.

All data obtained from the tests have been combined to create two datasets, they have the same records but they are labelled in two different ways:

- Binary dataset: the records of the tests have been labelled with 0, if a healthy bearing has been used and they have been labelled with 1, if a faulty bearing has been used.
- Ternary dataset: the records of the tests have been labelled with 0, if the test has been carried out on a healthy bearing; they have been labelled with 1, if the test has been performed on the inner damaged bearing; and they have been labelled with 2, if the test has been conducted on the outer damaged bearing.

The use of the binary dataset aims to develop algorithms for fault detection in order to determine if the bearings are damaged or they are healthy. The use of the ternary dataset aims to create algorithms for fault identification. The two datasets have been utilized to create the training set and the test set, binary and ternary

respectively, by using 70% of data for the training set and 30% of data for the test set. In the training set, the same number of records has been made from each class, but in a random way (healthy and faulty for the binary dataset; healthy, inner and outer ring faulted for the ternary dataset), while the remaining data have been used for the test set. Several features in time domain and frequency domain have been computed for each record, they have been used for the development of Random Forest. The computed features are the following:

Time-domain features (Appendix A)

- RMS
- Kurtosis
- Mean
- Standard Deviation
- Variance
- Max Amplitude
- Min Amplitude
- Peak to Peak
- Square Root of Amplitude
- Skewness
- Kurtosis Factor
- Skewness Factor
- Clearance Factor
- Impulse Factor
- Crest Factor
- Total Sum
- Entropy
- Mobility
- Complexity
- Histogram Upper Bound
- Histogram Lower Bound

- Envelope
- Negentropy

Frequency-domain features (Appendix A)

- Max Power Spectrum
- Frequency Center
- Root Variance Frequency
- Spectral Kurtosis
- Spectral Skewness
- Shape Factor.

The two algorithms, which have been taken into consideration, need input features in order to produce a prediction as output. For the Random Forest it is more convenient to use a lot of features as input because, thanks to the random combination of decision trees, the algorithm can directly separate the most important features from the other ones in order to have the best prediction. As regards the scarcely meaningful features, the algorithm gives them a very low weight, so they are directly discarded. As far as the SVM is concerned, the use of a set of meaningful and scarcely meaningful features can reduce the precision of the prediction. It is due to the fact that SVM gives the same weight to all the input features without any distinction. Therefore the three variables for fault detection, which are the most used in literature, have been chosen as input variables of SVM: RMS, Kurtosis and Skewness.

4.6.3 Random Forest model for rotary test rig

In this case, the Random Forest model has been created by using all the pre-processed features, calculated on the basis of all the vibration and current values. In order to improve the knowledge about the different importance of the features used, several random forest models have been trained by selecting different features. By using all the pre-processed features created by the values of vibration and current, the following results on the importance of the features for fault detection have been obtained:

In Fig. 4.14, Fig. 4.15, Fig. 4.16, Fig. 4.17 it is possible to observe the importance given to the features by the models. The Random Forest gives a lot of importance to pre-processed features of the vibration, in fact only one feature of current is used and its importance is very low. The results of these Random Forest models can be summarized by using the confusion matrix and accuracy, precision and recall. To confirm this observation, a Random Forest model has been trained

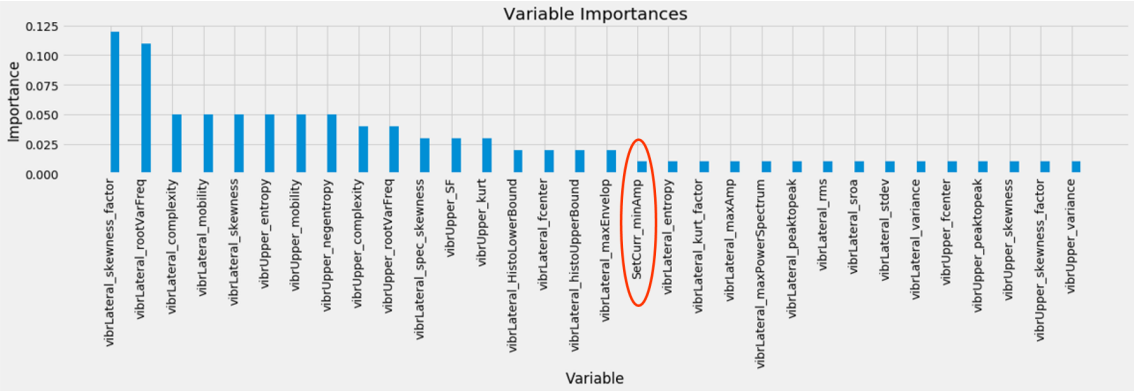


Figure 4.14: Importance of features evaluated through the Binary Random Forest algorithm.

Fault state	0	1
state		
0	8	0
1	0	56

Accuracy: 100.0 %

Precision: 100.0 %

Recall: 100.0 %

Figure 4.15: Confusion Matrix of the Binary Random Forest algorithm trained with all the features.

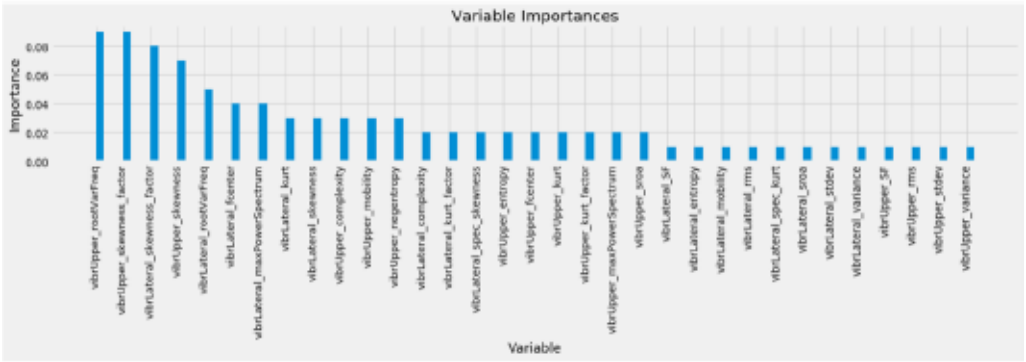


Figure 4.16: Importance of all the features evaluated by the Ternary Random Forest algorithm.

Fault state	0	1	2
state			
0	8	0	0
1	0	8	0
2	0	0	8

Accuracy: 100.0 %

Precision: 100.0 %

Recall: 100.0 %

Figure 4.17: Matrix of the Ternary Random Forest algorithm trained with all the features.

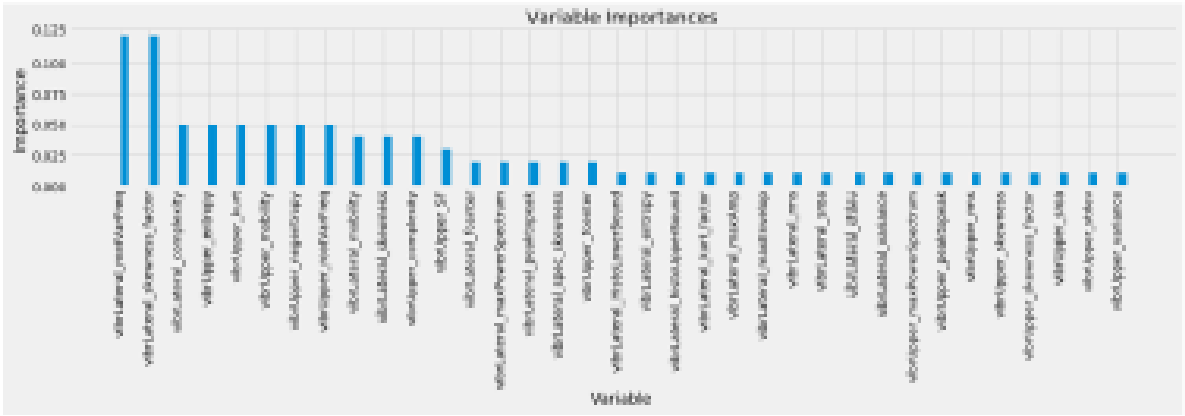


Figure 4.18: Importance evaluated by the Binary Random Forest algorithm trained only with vibration features.

Fault state	0	1
state		
0	8	0
1	0	56

Accuracy: 100.0 %

Precision: 100.0 %

Recall: 100.0 %

Figure 4.19: Confusion Matrix of the Binary Random Forest algorithm trained only with vibration features.

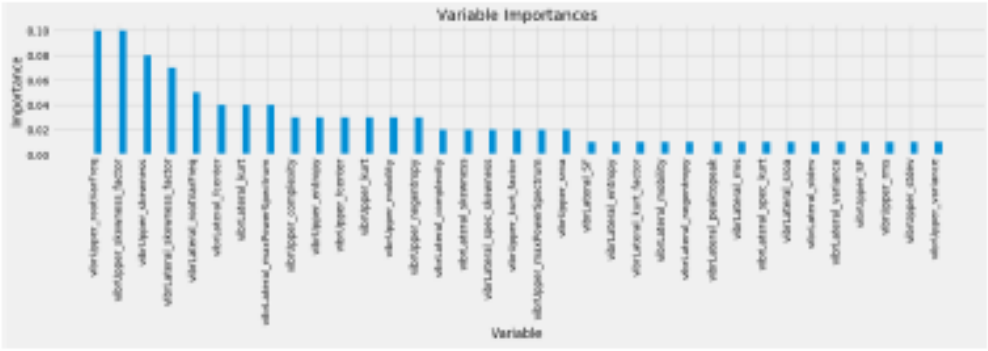


Figure 4.20: Importance evaluated by the Ternary Random Forest algorithm trained with only vibration features.

Fault state	0	1	2
state			
0	8	0	0
1	0	8	0
2	0	0	8

Accuracy: 100.0 %

Precision: 100.0 %

Recall: 100.0 %

Figure 4.21: Confusion Matrix of the Ternary Random Forest algorithm trained only with vibration features.

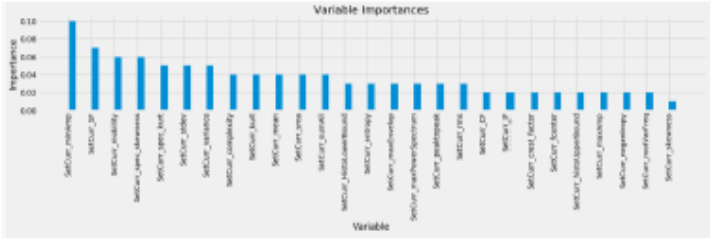


Figure 4.22: Importance evaluated by the Binary Random Forest algorithm trained only with current features.

Fault state	0	1
state		
0	8	0
1	2	54

Accuracy: 96.88 %

Precision_0: 80.0 %

Recall_0: 100 %

Precision_1: 100 %

Recall_1: 96.47 %

Figure 4.23: Confusion Matrix of the Binary Random Forest algorithm trained only with current features.

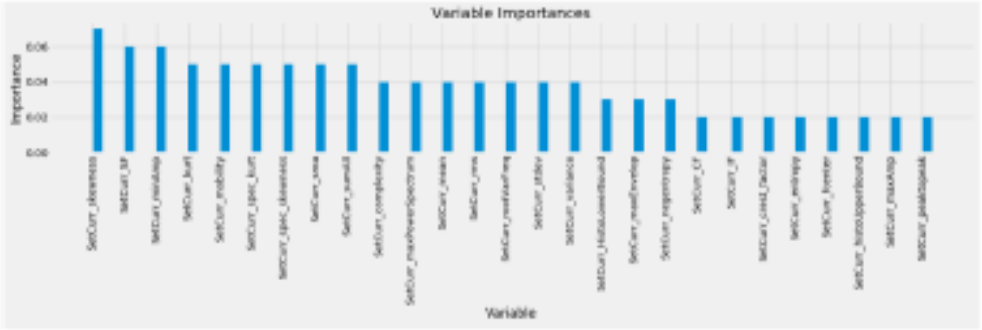


Figure 4.24: Importance evaluated by the Ternary Random Forest algorithm trained only with current features.

Fault state	0	1	2
state			
0	8	0	0
1	0	4	4
2	0	3	5

Accuracy: 70.83 %

Precision_0: 100.0 %

Recall_0: 100.0 %

Precision_1: 57.14 %

Recall_1: 50.0%

Precision_2: 55.55 %

Recall_2: 62.5 %

Figure 4.25: Confusion Matrix of the Ternary Random Forest algorithm trained only with current features.

only with the features of vibration and another one has been trained only with the features of current.

From these results, it is possible to deduce that the artificially damaged bearings are detectable by means of the Random Forest model trained with the vibration variables or with the current variables. The tests show that the current signal gives less information about the state of the bearing, most probably for its sampling rate that is lower in comparison with the sampling rate of vibrations, but it allows to identify, even if with less accuracy, the state of the bearings. Furthermore, it is possible to observe that, regardless of the pre-processed signals, all the trained Random Forest models give more importance to Skewness, RMS, Kurtosis in comparison with the other features. These three features are already used in the literature on condition monitoring and this is a further proof of the correct implementation of the Random Forest models.

4.6.3.1 Support Vector Machine for rotary test rig

In this case, four different SVMs have been trained and tested:

1. Binary SVM with the features of Current RMS, Current Kurtosis and Current Skewness.

2. Ternary SVM with the features of Current RMS, Current Kurtosis and Current Skewness.

3. Binary SVM with the features of Vibration RMS, Vibration Kurtosis and Vibration Skewness.

4. Ternary SVM with the features of Vibration RMS, Vibration Kurtosis and Vibration Skewness.

The vibration signals taken into consideration are the ones of the accelerometer placed in the same direction as the bearing preload. This choice has been made because the information on the load direction is the most meaningful. Even in this case, a balanced training set consisting in 70% of the data and a test set consisting in 30% of the data have been used. The SVM has been trained and tested with several kernels: Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian and Course Gaussian. Cubic and Medium Gaussian kernels have given the best performances, Fig.4.26, Fig.4.27 show the results of each one.

As regards prediction, these figures show a precision of 100% in case of binary and ternary predictors. The data highlight that the use of Cubic kernel or Medium Gaussian kernel is quite the same. In this specific case, even the use of current data or of vibration data is the same In conclusion, it is possible to see that the state of the damage tested is actually visible both using vibration and current. It is also possible to notice that the damaged bearings have not given any problem to the rotary motor, consequently the deduction is that it is not dangerous to mount them on the XTS test rig.

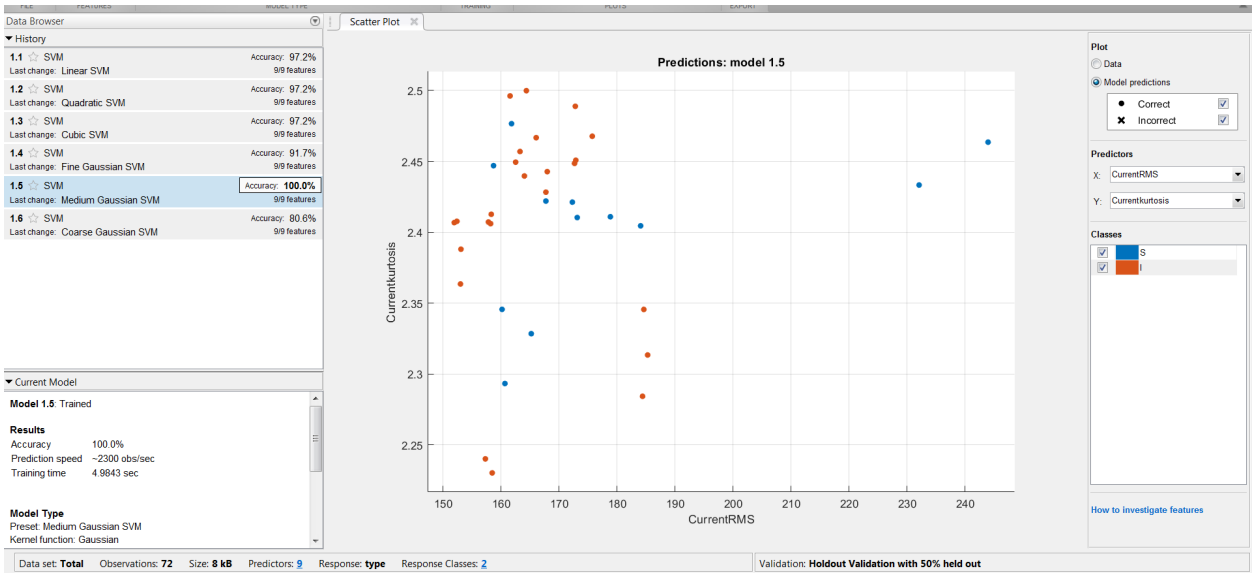


Figure 4.26: RMS-Kurtosis plot of the SVM prediction.

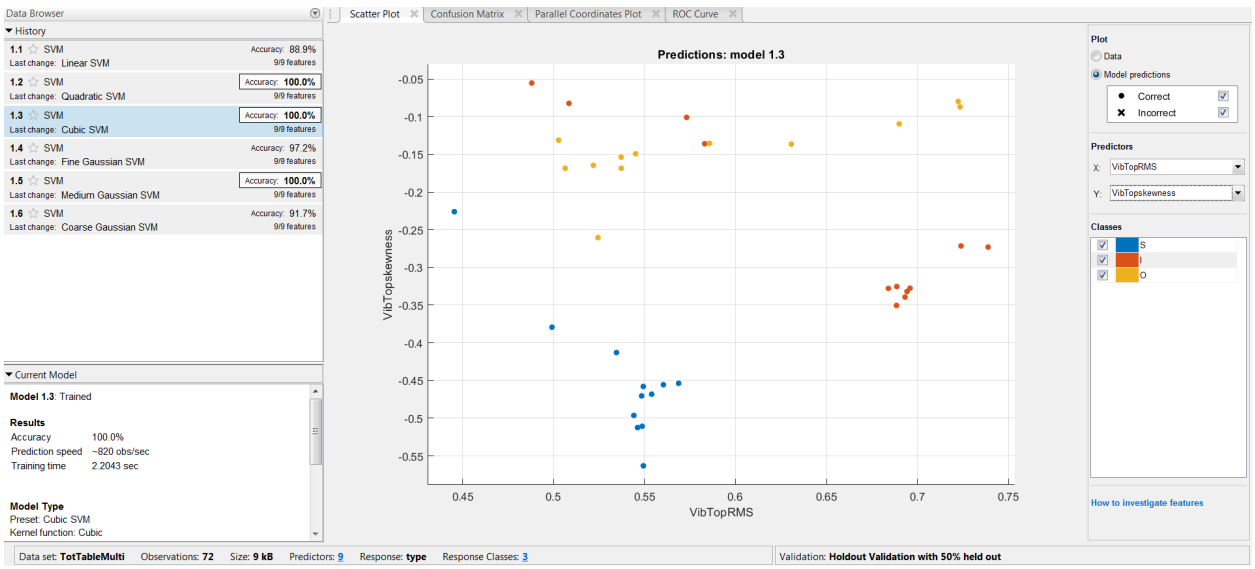


Figure 4.27: Skewness-RMS plot of the SVM prediction.

4.6.4 Use of the artificially damaged bearings in XTS Beckhoff test rig for the creation of a dataset

With the certainty that the artificial damages are visible and cannot cause damages on the Independent Carts System, a test procedure has been defined in order to develop a dataset with the aforementioned damaged bearings in the XTS test rig. The number of the movers used to test the faulty bearings and the sequence of tests have been decided randomly in order to reduce the possible environmental variations in the tests and to improve the repeatability of the tests. The total number of tests is 12 and for each test the system variables have been recorded for four times. The test table is shown in Fig. 4.28.

Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
T1	H	H	H	H	H	H	H	H	H	H	H	R
T2	H	H	H	H	H	H	H	O	H	H	H	H
T3	H	H	H	H	H	H	H	H	H	I	H	H
T4	H	H	H	H	H	R	H	H	H	H	H	H
T5	H	H	H	H	H	O	H	H	H	H	H	H
T6	H	H	H	I	H	H	H	H	H	H	H	H
T7	H	R	H	H	H	H	H	H	H	H	H	H
T8	H	O	H	H	H	H	H	H	H	H	H	H
T9	H	H	I	H	H	H	H	H	H	H	H	H
T10	H	H	H	H	R	H	H	H	H	H	H	H
T11	O	H	H	H	H	H	H	H	H	H	H	H
T12	H	H	H	H	H	H	H	H	H	H	R	H

H = Healthy State I = Inner Damage O = Outer Damage R = Rusty Damage

Figure 4.28: Training tables with the different tests for all the types of damages.

Each test has been run following this standard procedure:

1. Setting of the faulty bearing on the mover indicated by the test.
2. Twenty-minute warming-up of the test rig without any data record.
3. Start of calculation of Q1, Q2, Q3 for each data sample.
4. Forty-second recording of all variables taken into account.
5. Repetition of the procedure from point three to point five for six times.

4.6.5 Building of machine learning models by using the data training set

After the development of the training dataset, several operations have been performed to create different machine learning models.

First of all, the data have been pre-processed. The dataset has been divided into 12 sets, each of which represents a test with six records. Two different pre-processing methods have been tested in order to find the best approach. The two approaches are the following:

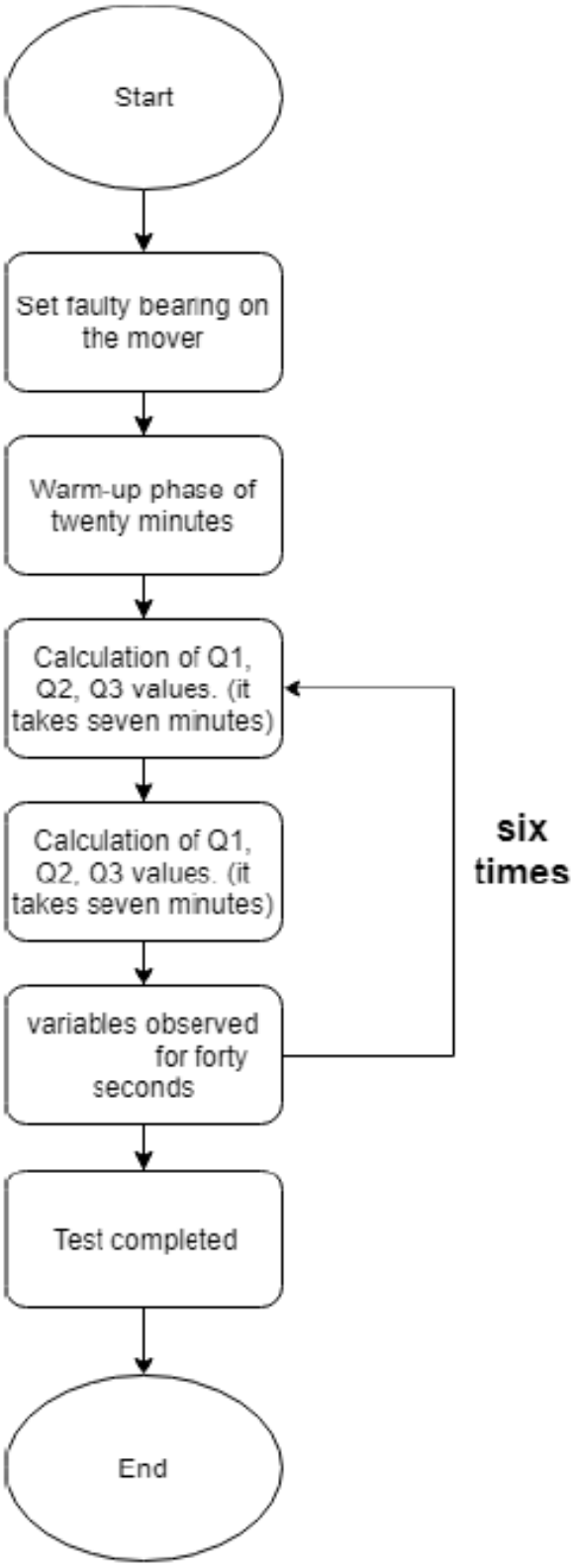


Figure 4.29: Test procedure diagram.

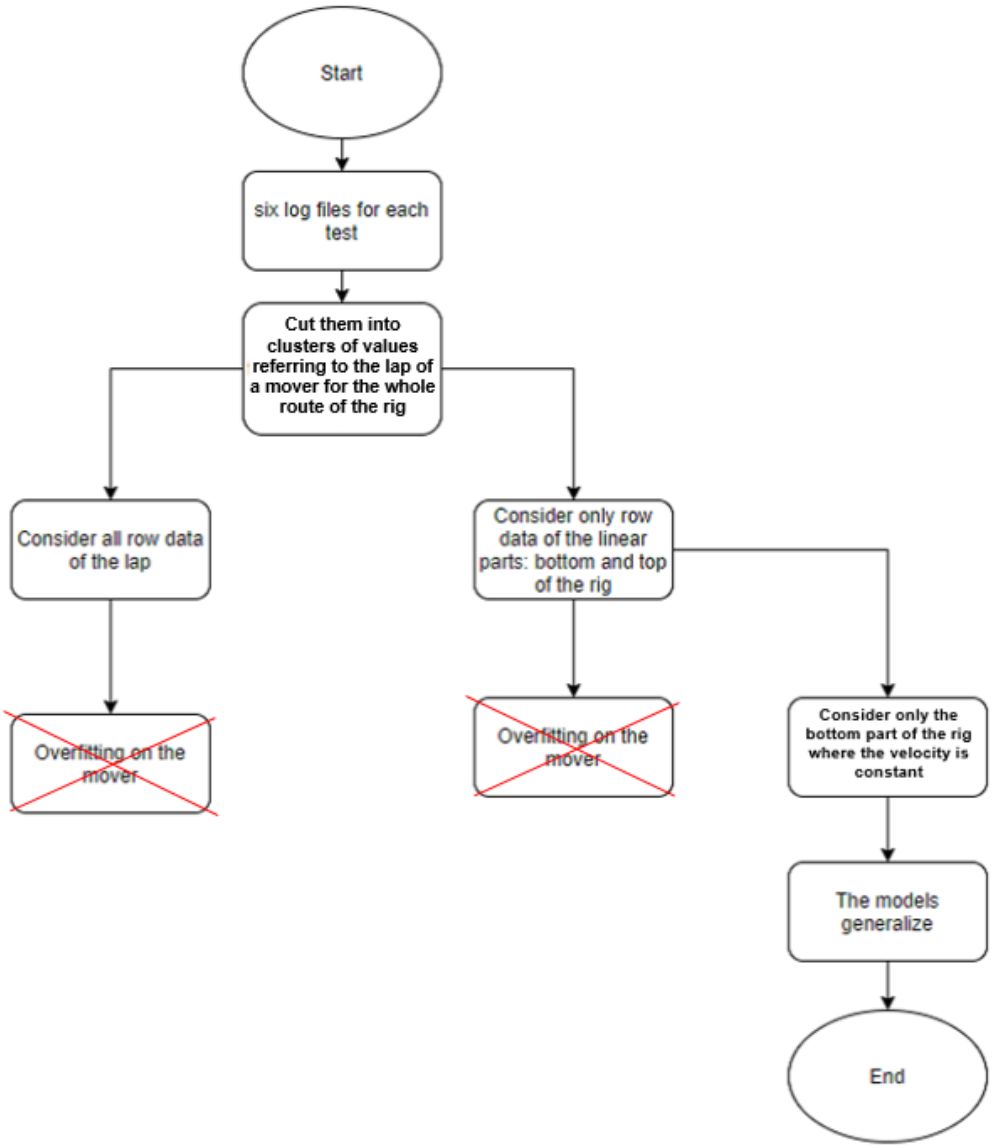


Figure 4.30: Test procedure diagram.

1. Cut the row signals into laps considering mover 1 as the main mover. The cut is made whenever mover 1 has an actual position equal to zero.(Fig. 4.31)

2. Cut the row signals into laps considering the actual position of each mover and eliminating the signals recorded along the curved parts. In this case, only the signals recorded along the straight parts of the rig are taken into account.(Fig. 4.31)

In the second case, the signals recorded in the curved parts are eliminated because the signals show a very high level of noise along these parts. In both cases, the vibrations, which are not referable to the actual position since the sensors are placed on the frame and not on each cart, are cut into different laps considering position 0 of mover 1 as starting point and ending point.

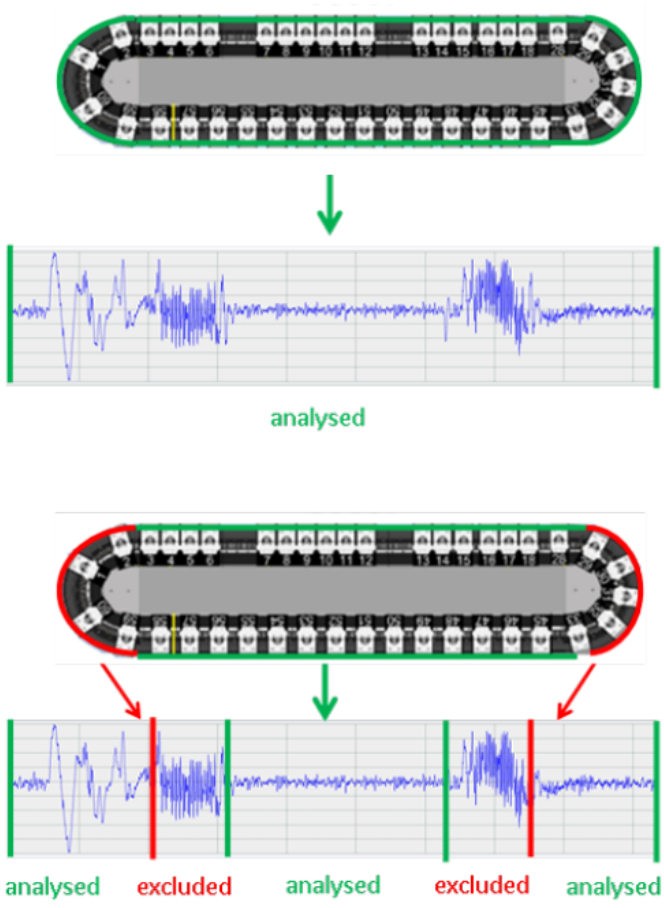


Figure 4.31: Signal cut with respect to the mover position.

Carrying out different tests on these data, it has been noticed that the features of vibrations are not considered by the random-forest algorithm because they are not referred to each mover but to the general system.
It has also been noticed that the algorithm, trained with signals recorded in the

upper part of the rig, tend to overfit the prediction on each mover. This can be explained considering that the carts have variable velocities in the upper part of the rig, while they have a constant velocity in the bottom part.
To avoid the aforementioned problems, the vibration data are not taken into account, but only the other signals recorded in the bottom zone of the test rig are considered. Fig.4.32 shows the part of the recorded signals considered after the pre-processing.

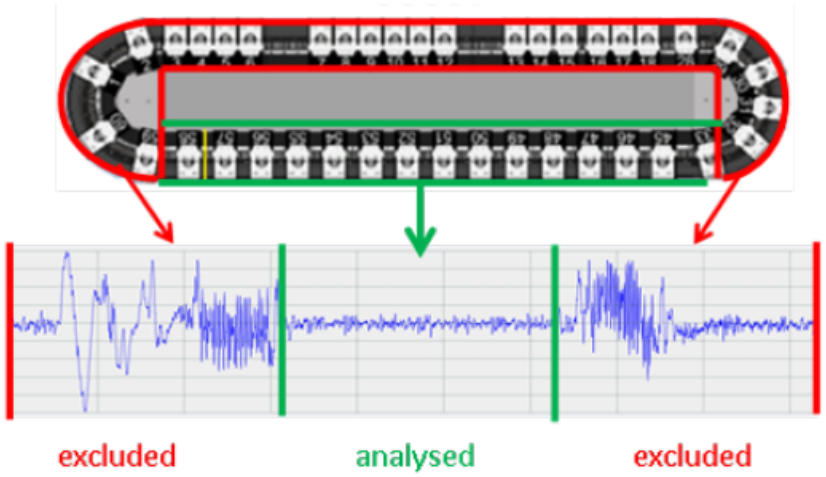


Figure 4.32: Signal cut in the constant velocity part with respect to the mover position.

4.6.6 Machine Learning pre-processing

For the two machine learning algorithms, the routine followed for the data pre-processing is very similar and is explained in the flowcharts (Fig. **??**).

After the general pre-processing part highlighted in the image with a dashed red line (Fig. 4.33), all the time-domain and frequency-domain features have been computed for each lap. Consequently the number of records for each test are in Fig. 4.34.

After the creation of the new pre-processed datasets, the data have been z-normalized, subtracting the mean value from each value and dividing the result by the standard deviation of the feature considered. In order to contain the values of the approximate thresholds in the same order of magnitude, the number of the examples given for the two classes to be predicted have also been balanced. The balance is based on the random extraction of the same number of movers in healthy and faulty conditions from the general dataset. At this point the two machine learning models have been trained.

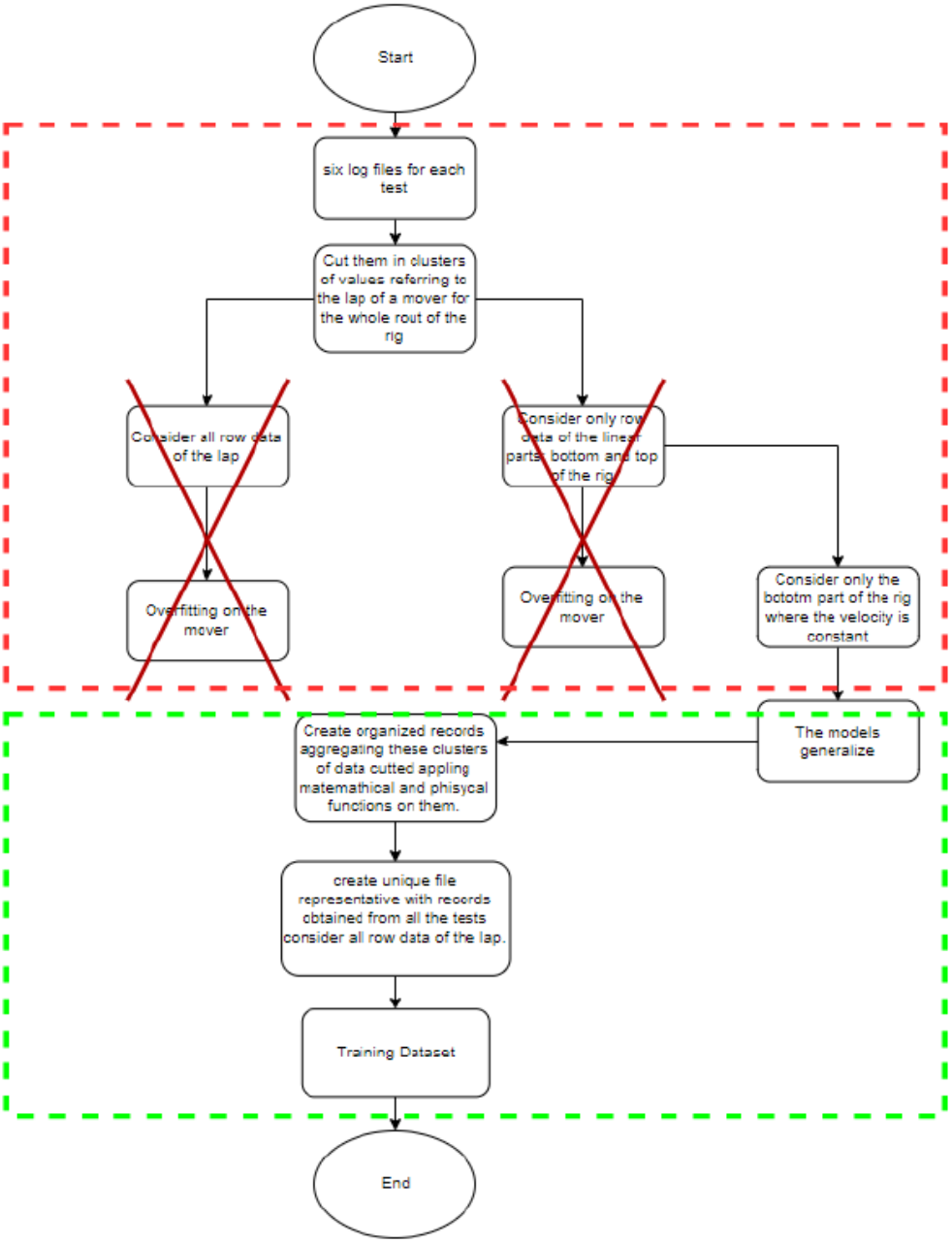


Figure 4.33: Signal pre-processing procedure for the dataset creation.

$$NRT = NLFT * [NMOV * NL * (TL/TLAP)]$$

NRT	Number of record for each test
NMOV	Number of movers
NL	number of laps
TL	time of logging
TLAP	time of lap for mover

Figure 4.34: Number of records.

4.6.6.1 Random Forest

Different types of Random Forest models have been created by training them with different sets of pre-processed features: binary (able to classify healthy and faulty states) and quaternary (able to classify also the different types of damages: healthy, inner, outer, rusty).

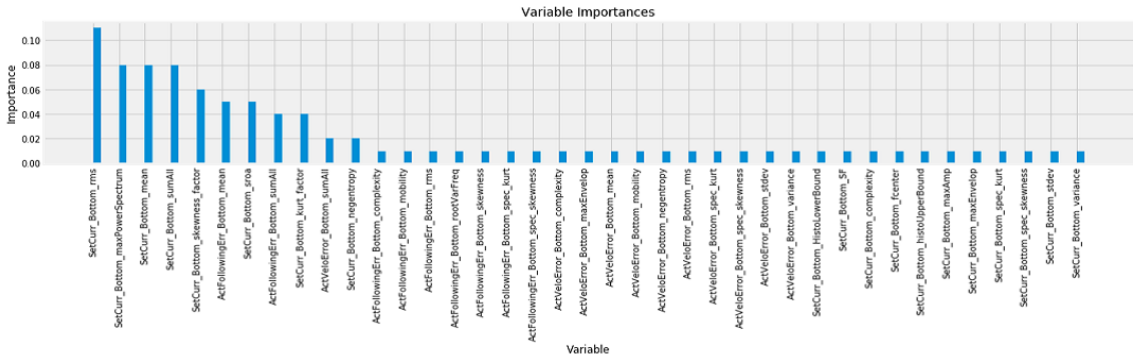


Figure 4.35: Importance of features.

Fig. 4.35 shows of the importance features evaluated by the Random Forest. It is possible to notice that the features referred to vibrations are not considered, while the ones referred to current are the most considered for prediction.
A Random Forest model has also been trained by using only the ten most important features chosen among the ones provided. As regards accuracy, the comparison between the Random Forest model based on the ten most important features and the Random Forest model that uses all the features, except vibrations, shows better outcomes for the second model.
Furthermore, after several evaluations and the creation of different models, the following two models of Random Forest have been taken into consideration:

- Binary Random Forest trained with all the features, except vibrations.
- Quaternary Random Forest trained with all the features, except vibrations.

4.6.6.2 Support Vector Machine

The features used in the training of SVMs are: Current RMS, Current Skewness, Current Kurtosis. Even in this case a balanced training dataset is used. In the following figures, it is possible to see the distribution of the points in two dimensions.

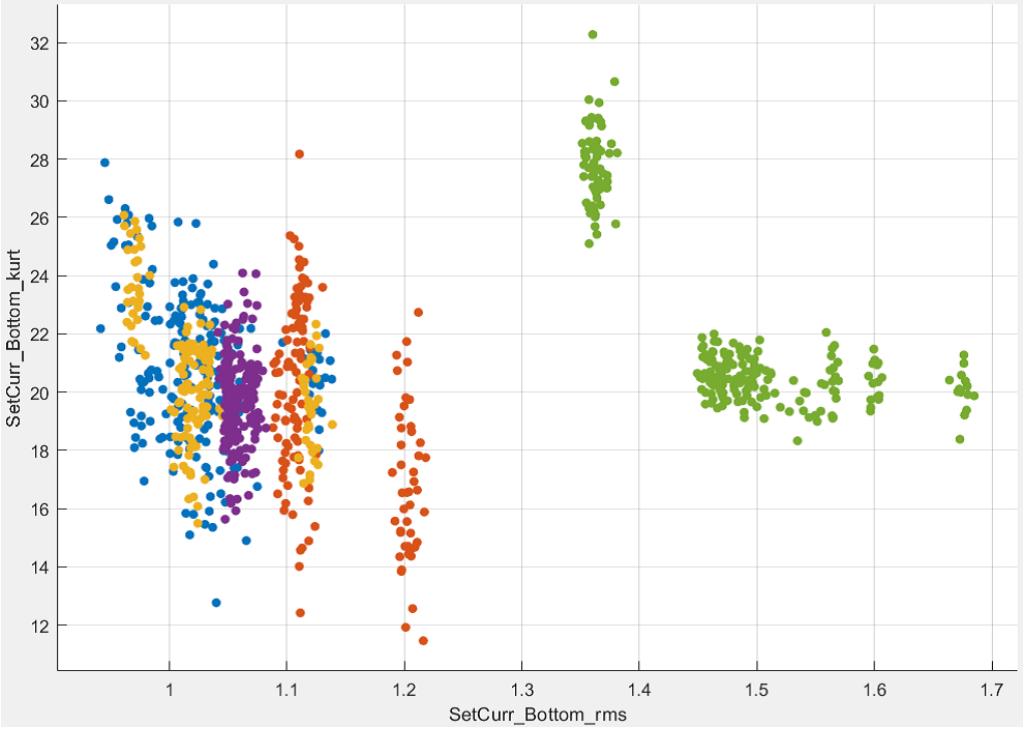


Figure 4.36: RMS and Kurtosis values of the dataset. Green: Blocked bearing, Blue: Healthy bearing, Orange: Outer ring damage, Violet: Inner ring damage, Yellow: Rusty damage.

As you can see in Fig. 4.36 and Fig. 4.37, the points that identify the rusty bearing are superimposed on the points that identify the healthy one, since the rusty bearing has been cleared before the beginning of the test, otherwise the bearing could not roll. The cleaning of the bearing makes it very similar to the healthy ones. For the outer ring damaged bearing and the blocked one, the separation is evident, while, for the inner ring damaged bearing, the figures show some points overlapping the healthy ones and other points separated.

4.6.6.3 Creation of a validation dataset referring to XTS Beckhoff test rig

The procedure for creating the validation dataset is the same procedure used for the training set as shown in Fig. 4.28, the table of the validation dataset is in Fig. 4.38.

During the training phase, the tests with the blocked bearings have not been carried out because this type of damage is so dangerous that it can irreversibly damage the system. Unlike the training set table (Fig. 4.28), it can be observed that the blocked bearings have been used in the tests for the validation dataset.

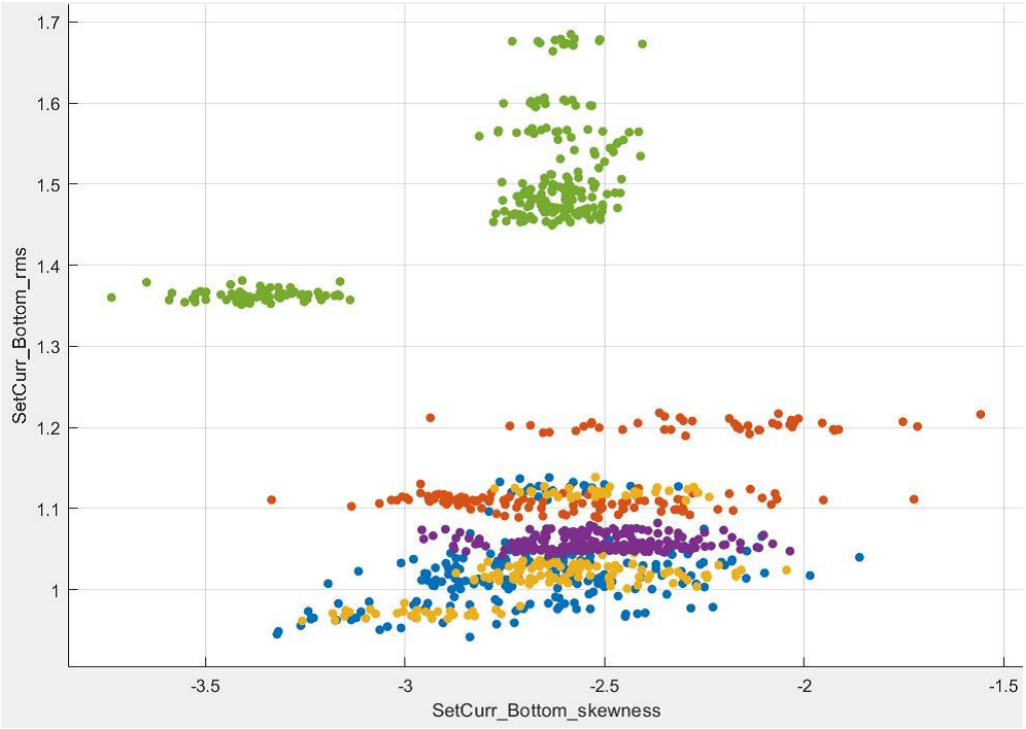


Figure 4.37: Skewness and RMS values of the dataset. Green: Blocked bearing, Blue: Healthy bearing, Orange: Outer ring damage, Violet: Inner ring damage, Yellow: Rusty damage.

		Number of movers											
Test number		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
	T1	H	H	H	H	H	H	H	H	H	H	H	I
	T2	H	H	H	H	R	H	H	H	H	H	H	H
	T3	H	B	H	H	H	H	H	H	H	H	H	H
	T4	H	H	H	H	H	H	H	H	H	O	H	H
	T5	H	H	H	H	I	H	H	H	H	H	H	H
	T6	H	H	R	H	H	H	H	H	H	H	H	H
	T7	H	H	H	H	B	H	H	H	H	H	H	H
	T8	H	O	H	H	H	H	H	H	H	H	H	H
	T9	H	H	H	H	H	H	H	H	H	H	H	H
	T10	H	I	H	H	H	H	H	H	H	H	H	H
	T11	H	H	H	H	H	H	H	H	H	H	H	R
	T12	H	H	H	H	H	H	H	H	H	H	H	B
	T13	H	H	H	H	H	H	O	H	H	H	H	H
	T14	I	H	H	H	H	H	H	H	H	H	H	H
	T15	H	H	R	H	H	H	H	H	H	H	H	H
	T16	H	H	H	H	B	H	H	H	H	H	H	H
	T17	H	H	H	H	H	H	H	H	H	O	H	H
	T18	H	H	H	H	H	H	H	H	H	H	H	H

H = Healthy State I = Inner Damage O = Outer Damage R = Rusty Damage B = Blocked Damage

Figure 4.38: Test table for the creation of the validation dataset.

4.6.7 Validation Loop 1

In this phase, the test set has been used to validate the models created. The models have been evaluated by using the metrics specified at the beginning of the chapter.

4.6.7.1 Validation Loop 1 for machine learning models

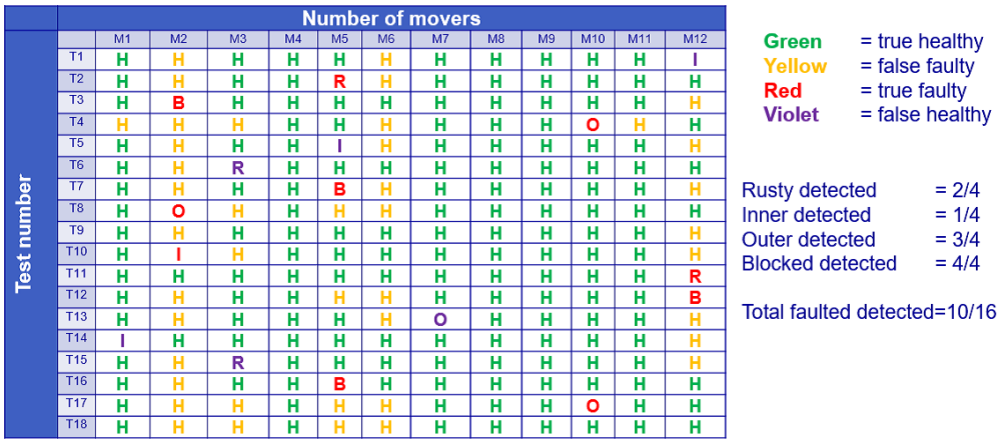
Before the validation, a rule was established in order to define when the models identify a fault or a healthy bearing. The rule can be explained in this way:

- In the binary classification, if the model classifies as faulty more than 25% of the records regarding a mover in a given test, the global prediction is considered faulty.
- In the case of quaternary classifiers, the damages have been considered to be detected, even if the predicted type of damage given in the classification is not the same as the real one.

This rule was defined because fault detection is the basic target and it allows to use even quaternary models. The tables below show the prediction of the binary and quaternary Random Forest and SVM models. Fig. 4.39, Fig. 4.40, Fig. 4.41 and Fig. 4.42 describe the outcomes of each mover for each test and the corresponding Confusion Matrices (Fig. 4.43 and Fig.4.44). Observing the results of Random Forest models, it can be noticed that the binary models are more stable than the quaternary ones. De facto, as regards the Random Forest quaternary model, it is possible to observe an increase in false positives, while the SVM quaternary model overfits the healthy state. Furthermore, it is possible to notice that mover 2 has some anomalies with respect to the other movers; in fact it is often classified as faulty although it is healthy in many cases. By means of the observation of the confusion matrices, it is possible to notice that the recall is lower for quaternary models. Actually, the recall expresses the number of times a damage is detected; in manufacturing fields it is preferable to have false alarms than missed alarms, because the miss of a true alarm involves the risk of damaging other parts of the machine.

Of course, the results are not excellent because of the high number of false positives, which involves high costs of maintenance for the high number of controls of healthy components. A good indication is that the blocked bearings, which could irreversibly damage the machine, are correctly classified at least by the binary models. In Fig. 4.45, it is possible to see the test classification divided into different diagrams referred to each type of damages.

As regards prediction, accuracy is high for block damages and outer ring damages, while for inner ring damages, which are very light, the classifiers find it hard to give a definite prediction, confusing some records of the mover as healthy. In the case of rusty damages, it is possible to notice a high error in their classification because their performances are very similar to the healthy cases.
To increase the performance of the models created, a logical disjunction (logical OR) between the output predictions of the Random Forest models and the ones of the



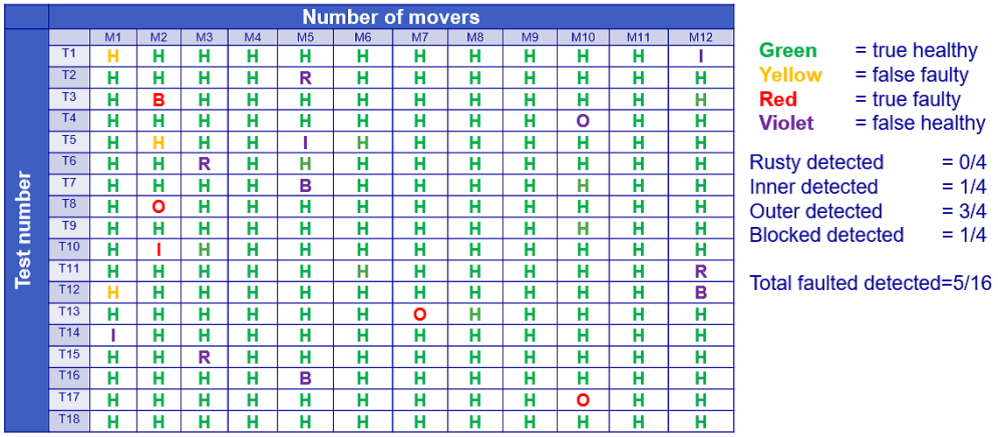
Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
T1	H	H	H	H	H	H	H	H	H	H	H	I
T2	H	H	H	H	R	H	H	H	H	H	H	H
T3	H	B	H	H	H	H	H	H	H	H	H	H
T4	H	H	H	H	H	H	H	H	H	O	H	H
T5	H	H	H	H	I	H	H	H	H	H	H	H
T6	H	H	R	H	H	H	H	H	H	H	H	H
T7	H	H	H	H	B	H	H	H	H	H	H	H
T8	H	O	H	H	H	H	H	H	H	H	H	H
T9	H	H	H	H	H	H	H	H	H	H	H	H
T10	H	I	H	H	H	H	H	H	H	H	H	H
T11	H	H	H	H	H	H	H	H	H	H	H	R
T12	H	H	H	H	H	H	H	H	H	H	H	B
T13	H	H	H	H	H	H	O	H	H	H	H	H
T14	I	H	H	H	H	H	H	H	H	H	H	H
T15	H	H	R	H	H	H	H	H	H	H	H	H
T16	H	H	H	H	B	H	H	H	H	H	H	H
T17	H	H	H	H	H	H	H	H	H	O	H	H
T18	H	H	H	H	H	H	H	H	H	H	H	H

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Rusty detected = 2/4
Inner detected = 1/4
Outer detected = 3/4
Blocked detected = 4/4

Total faulted detected=10/16

Figure 4.39: Test table with the results of classification of the model Random Forest Binary Classifier.



Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
T1	H	H	H	H	H	H	H	H	H	H	H	I
T2	H	H	H	H	R	H	H	H	H	H	H	H
T3	H	B	H	H	H	H	H	H	H	H	H	H
T4	H	H	H	H	H	H	H	H	H	O	H	H
T5	H	H	H	H	I	H	H	H	H	H	H	H
T6	H	H	R	H	H	H	H	H	H	H	H	H
T7	H	H	H	H	B	H	H	H	H	H	H	H
T8	H	O	H	H	H	H	H	H	H	H	H	H
T9	H	H	H	H	H	H	H	H	H	H	H	H
T10	H	I	H	H	H	H	H	H	H	H	H	H
T11	H	H	H	H	H	H	H	H	H	H	H	R
T12	H	H	H	H	H	H	H	H	H	H	H	B
T13	H	H	H	H	H	H	O	H	H	H	H	H
T14	I	H	H	H	H	H	H	H	H	H	H	H
T15	H	H	R	H	H	H	H	H	H	H	H	H
T16	H	H	H	H	B	H	H	H	H	H	H	H
T17	H	H	H	H	H	H	H	H	H	O	H	H
T18	H	H	H	H	H	H	H	H	H	H	H	H

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Rusty detected = 0/4
Inner detected = 1/4
Outer detected = 3/4
Blocked detected = 1/4

Total faulted detected=5/16

Figure 4.40: Test table with the results of classification of the model SVM Binary Classifier.

Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
T1	H	H	H	H	H	H	H	H	H	H	H	I
T2	H	H	H	H	R	H	H	H	H	H	H	H
T3	H	B	H	H	H	H	H	H	H	H	H	H
T4	H	H	H	H	H	H	H	H	H	O	H	H
T5	H	H	H	H	I	H	H	H	H	H	H	H
T6	H	H	R	H	H	H	H	H	H	H	H	H
T7	H	H	H	H	B	H	H	H	H	H	H	H
T8	H	O	H	H	H	H	H	H	H	H	H	H
T9	H	H	H	H	H	H	H	H	H	H	H	H
T10	H	I	H	H	H	H	H	H	H	H	H	H
T11	H	H	H	H	H	H	H	H	H	H	H	R
T12	H	H	H	H	H	H	H	H	H	H	H	B
T13	H	H	H	H	H	H	O	H	H	H	H	H
T14	I	H	H	H	H	H	H	H	H	H	H	H
T15	H	H	R	H	H	H	H	H	H	H	H	H
T16	H	H	H	H	B	H	H	H	H	H	H	H
T17	H	H	H	H	H	H	H	H	H	O	H	H
T18	H	H	H	H	H	H	H	H	H	H	H	H

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Rusty detected = 2/4
Inner detected = 1/4
Outer detected = 3/4
Blocked detected = 4/4

Total faulted detected=10/16

Figure 4.41: Test table with the results of classification of the model Random Forest Quaternary Classifier.

Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
T1	H	H	H	H	H	H	H	H	H	H	H	I
T2	H	H	H	H	R	H	H	H	H	H	H	H
T3	H	B	H	H	H	H	H	H	H	H	H	H
T4	H	H	H	H	H	H	H	H	H	O	H	H
T5	H	H	H	H	I	H	H	H	H	H	H	H
T6	H	H	R	H	H	H	H	H	H	H	H	H
T7	H	H	H	H	B	H	H	H	H	H	H	H
T8	H	O	H	H	H	H	H	H	H	H	H	H
T9	H	H	H	H	H	H	H	H	H	H	H	H
T10	H	I	H	H	H	H	H	H	H	H	H	H
T11	H	H	H	H	H	H	H	H	H	H	H	R
T12	H	H	H	H	H	H	H	H	H	H	H	B
T13	H	H	H	H	H	H	O	H	H	H	H	H
T14	I	H	H	H	H	H	H	H	H	H	H	H
T15	H	H	R	H	H	H	H	H	H	H	H	H
T16	H	H	H	H	B	H	H	H	H	H	H	H
T17	H	H	H	H	H	H	H	H	H	O	H	H
T18	H	H	H	H	H	H	H	H	H	H	H	H

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Rusty detected = 0/4
Inner detected = 1/4
Outer detected = 3/4
Blocked detected = 1/4

Total faulted detected=5/16

Figure 4.42: Test table with the results of classification of the model SVM Quaternary Classifier.

Confusion matrix Random Forest Binary Classifier

Prediction

State

	Faulty	Healthy
Faulty	10	6
Healthy	22	178

Accuracy:	87.04%
Faulty recall:	62.5%
Faulty precision:	31.25%

Prediction

State

	Faulty	Healthy
Faulty	12	4
Healthy	43	157

Accuracy:	78.24%
Faulty recall:	75%
Faulty precision:	21.81%

Figure 4.43: Confusion Matrix Binary Random Forest and SVM.

Confusion matrix Random Forest Quaternary Classifier

Prediction

State

	Faulty	Healthy
Faulty	10	6
Healthy	42	158

Accuracy:	77.78%
Faulty recall:	62.5%
Faulty precision:	19.23%

Confusion matrix SVM Quaternary Classifier

Prediction

State

	Faulty	Healthy
Faulty	5	11
Healthy	3	197

Accuracy:	93.51%
Faulty recall:	31.25%
Faulty precision:	62.5%

Figure 4.44: Confusion Matrix Quaternary Random Forest and SVM

.

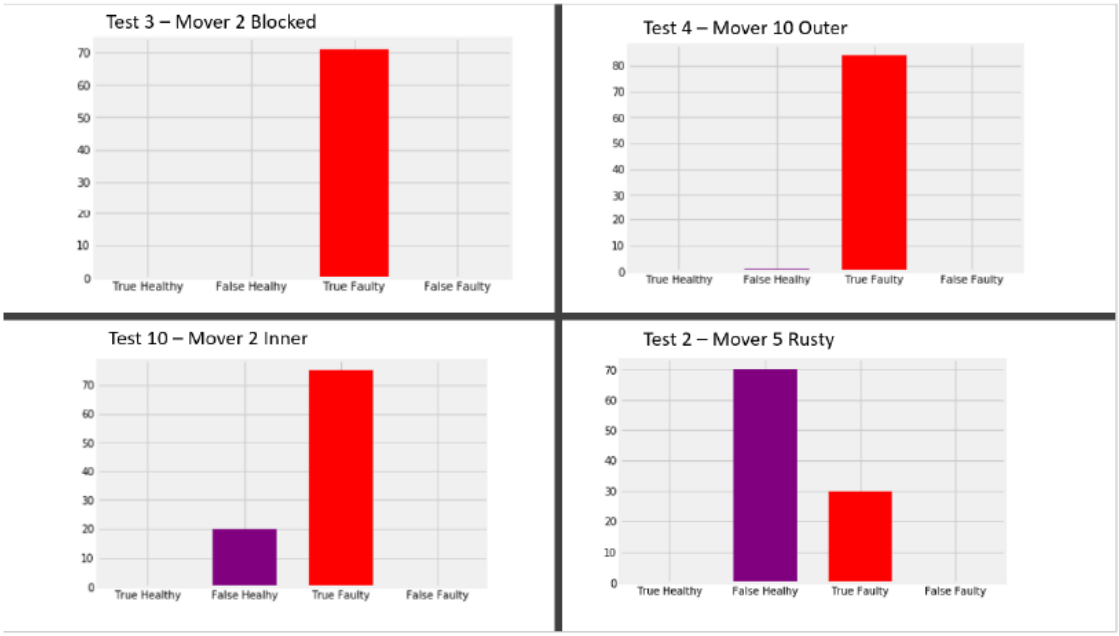


Figure 4.45: Tests with lower prediction accuracy in details.

SVM models has been implemented. In this case, if at least one of the four machine learning models identifies a mover as damaged, the output will predict a fault. This approach has been carried out to improve the faulty recall.

Observing Fig. 4.46 carefully, it is possible to see that the results of SVM quaternary and Random Forest quaternary models overlap the ones of the binary models; therefore, since the binary classifiers have given better performances, the study has been carried out only with binary models. Fig. 4.47describes the outcomes of each mover for each test and the corresponding Confusion Matrices (Fig. 4.48) in the case of the use of the logical disjunction.

The confusion matrix confirms the forecasts; in fact the faulty recall has increased while precision and accuracy have decreased because of the increase of false faulty.

At the end of this first validation cycle, the conclusions are the following:

1. The rusty bearing behaviour is very similar to the healthy one because it has been cleaned before the test.

2. The SVM and RF quaternary algorithms present worse performances than the binary ones.

3. The data training set of the two algorithms does not include blocked bearings.

4. The threshold for fault detection is equal to 25% of the mover data. It introduces several false negatives (false alarms).

Test number	Damage type	SVM binary	SVM quaternary	Random forest binary	Random forest quaternary
	Algorithms				
T1	Inner	Fail	Fail	Fail	Fail
T2	Rusty	Pass	Fail	Pass	Pass
T3	Blocked	Pass	Pass	Pass	Pass
T4	Outer	Pass	Fail	Pass	Pass
T5	Inner	Pass	Fail	Fail	Fail
T6	Rusty	Fail	Fail	Fail	Fail
T7	Blocked	Pass	Fail	Pass	Pass
T8	Outer	Pass	Pass	Pass	Pass
T9	Healthy	Pass	Fail	Pass	Pass
T10	Inner	Pass	Pass	Pass	Pass
T11	Rusty	Fail	Fail	Pass	Pass
T12	Blocked	Pass	Fail	Pass	Pass
T13	Outer	Pass	Pass	Fail	Fail
T14	Inner	Pass	Fail	Fail	Fail
T15	Rusty	Fail	Fail	Fail	Fail
T16	Blocked	Pass	Fail	Pass	Pass
T17	Outer	Pass	Pass	Pass	Pass
T18	Healthy	Pass	Fail	Pass	Pass
Total fault detected	-------	12/16	5/16	10/16	10/16

Figure 4.46: Results of the four Machine Learning algorithms.

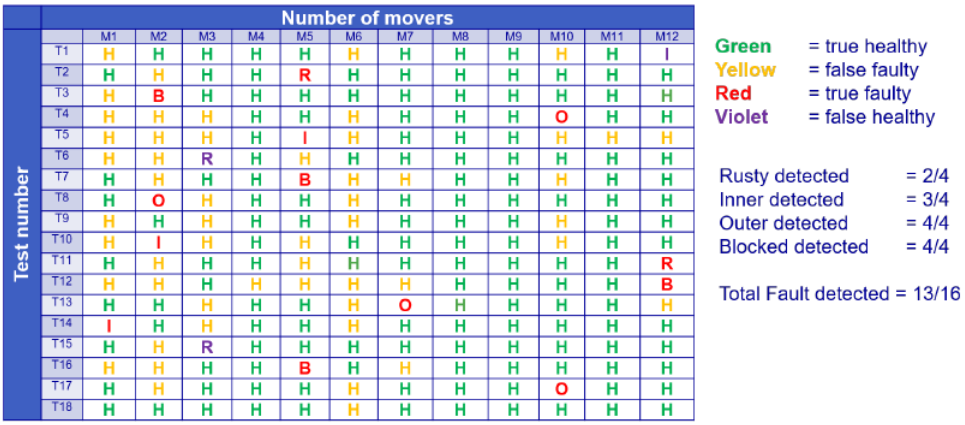


Figure 4.47: Results of the "OR " prediction between Random Forest Binary Classifier and SVM Binary Classifier.

State \ Prediction	Faulty	Healthy
Faulty	13	3
Healthy	53	147

Accuracy:	74.04%
Faulty recall:	81.25%
Faulty precision:	19.70%

Figure 4.48: Confusion Matrix of "OR" prediction between Random Forest Binary Classifier and SVM Binary Classifier.

5. The merge (through an OR function) of the SVM and Random Forest algorithms improves the faulty recall and reduces accuracy.

4.6.7.2 Validation Loop 2

In this validation cycle, different test specifications are used in order to increase the accuracy of the machine learning models. They derive from the results of Validation Loop 1.

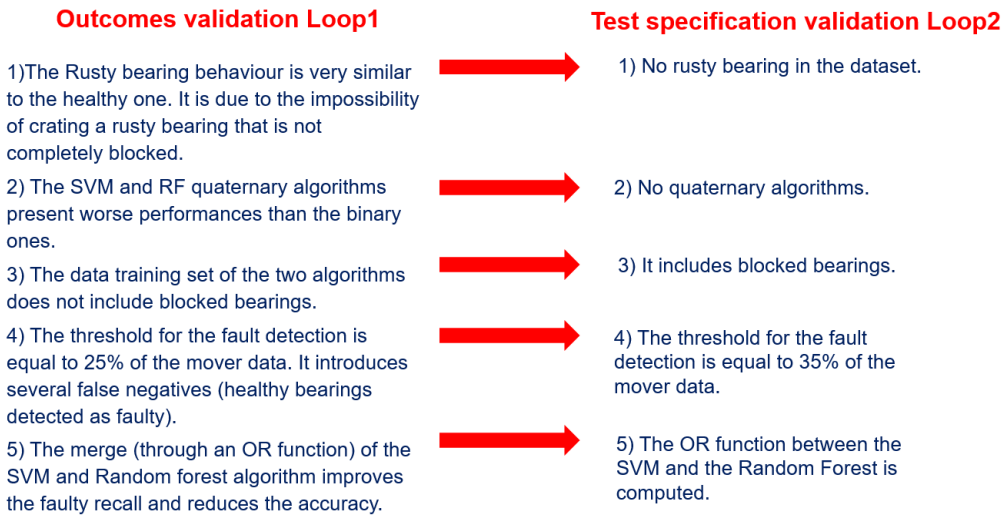


Figure 4.49: Outcomes of Validation Loop 1 and test specification of Validation Loop 2

For Validation Loop 2, the training set and the validation set have been created following some rules, which have been established with the aim of using the previous datasets without any model bias:

1. The rusty tests have been eliminated. Without any rusty test, 14 validation tests (they will be indicated with the prefix TVAL) and 7 training tests (they will be indicated with the prefix TTR) are present. Consequently, the total number of tests is equal to 21.

2. 7 random tests, including 3 outer ring faults, 3 inner ring faults and 1 blocked bearing, are chosen among the 21 tests.

3. The randomized training dataset consists in: TVAL8, TVAL10, TVAL14, TVAL17, TTR2, TTR3, TTR8.

4. The randomized validation dataset consists in: TVAL1, TVAL3, TVAL4, TVAL5, TVAL7, TVAL9, TVAL12, TVAL13, TVAL16, TVAL18, TTR5, TTR6, TTR9, TTR11.

Three cases of inner and outer damages have been used for keeping the training set balanced. Fig. 4.50 and Fig. 4.51 show the results of the tests conducted with the two binary machine learning classifiers.

Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
TVAL1	H	H	H	H	H	H	H	H	H	H	H	I
TVAL3	H	B	H	H	H	H	H	H	H	H	H	H
TVAL4	H	H	H	H	H	H	H	H	H	O	H	H
TVAL5	H	H	H	H	I	H	H	H	H	H	H	H
TVAL7	H	H	H	H	B	H	H	H	H	H	H	H
TVAL9	H	H	H	H	H	H	H	H	H	H	H	H
TVAL12	H	H	H	H	H	H	H	H	H	H	H	B
TVAL13	H	H	H	H	H	H	O	H	H	H	H	H
TVAL16	H	H	H	H	B	H	H	H	H	H	H	H
TVAL18	H	H	H	H	H	H	H	H	H	H	H	H
TTR5	H	H	H	H	H	O	H	H	H	H	H	H
TTR6	H	H	H	I	H	H	H	H	H	H	H	H
TTR9	H	H	I	H	H	H	H	H	H	H	H	H
TTR11	O	H	H	H	H	H	H	H	H	H	H	H

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Inner detected = 1/4
Outer detected = 3/4
Blocked detected = 4/4

Total of fault detected= 8/12

Figure 4.50: Test table with the results of classification of the Random Forest Binary Classifier in Validation Loop 2.

Test number	Number of movers											
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
TVAL1	H	H	H	H	H	H	H	H	H	H	H	I
TVAL3	H	B	H	H	H	H	H	H	H	H	H	H
TVAL4	H	H	H	H	H	H	H	H	H	O	H	H
TVAL5	H	H	H	H	I	H	H	H	H	H	H	H
TVAL7	H	H	H	H	B	H	H	H	H	H	H	H
TVAL9	H	H	H	H	H	H	H	H	H	H	H	H
TVAL12	H	H	H	H	H	H	H	H	H	H	H	B
TVAL13	H	H	H	H	H	H	O	H	H	H	H	H
TVAL16	H	H	H	H	B	H	H	H	H	H	H	H
TVAL18	H	H	H	H	H	H	H	H	H	H	H	H
TTR5	H	H	H	H	H	O	H	H	H	H	H	H
TTR6	H	H	H	I	H	H	H	H	H	H	H	H
TTR9	H	H	I	H	H	H	H	H	H	H	H	H
TTR11	O	H	H	H	H	H	H	H	H	H	H	H

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Inner detected = 3/4
Outer detected = 3/4
Blocked detected = 4/4

Total of fault detected= 10/12

Figure 4.51: Test table with the results of classification of the SVM Binary Classifier in Validation Loop 2.

It is possible to observe (Fig. 4.52) a decrease in false faulty, this is due to the elimination of rusty damages and an increase of the threshold from 25% to 35% for

the prediction of faulty bearings. Rusty bearings confuse the classifiers because they are very similar to the healthy state.

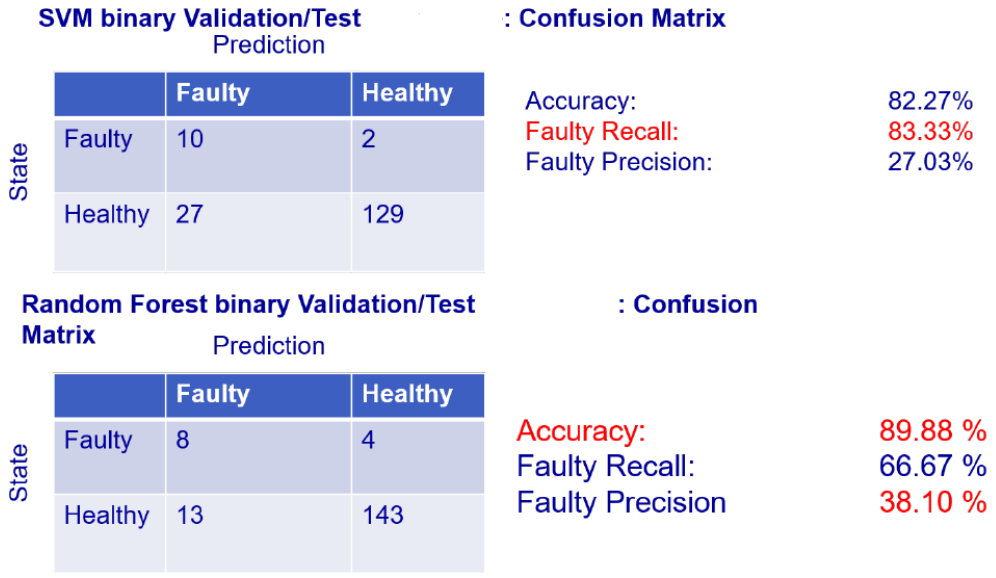


Figure 4.52: Confusion Matrix of Random Forest and SVM in Validation Loop 2.

In the confusion matrix it is possible to see an increase in the performance of the algorithms, thanks to the new conditions of Validation Loop 2 deduced from the outcomes of Validation Loop 1. Moreover, the metrics highlight the improvement of the two models with the expedients of this validation phase. Even in this case, as in Validation Loop 1, the two models are combined with an "OR" policy in order to maximize the faulty recall with a little detriment of accuracy and faulty precision. This occurs because it is essential to classify the greatest number of damages correctly.

Fig. 4.53 and Fig. 4.54 describe the outcomes of each mover in each test while Fig. 4.55 is the confusion matrix.

In this case, as in Validation Loop 1, the confusion matrix confirms the forecasts, in fact faulty recall has increased, while precision and accuracy have decreased because of the increase of false faulty. False faulty remains concentrated mainly on mover 1 and 2, this could be due to a peculiar behaviour of these two movers. To solve this problem, it is enough to increase the number of training tests in such a way as to cover all the possible varieties of the movers with the healthy state and to normalize each mover with respect to its mean and standard deviation. In Validation Loop 2, more than 90% of the damages are detected and all the most invasive damages are correctly classified. This is a good result: it is actually presumable that in future these models could increase their performances in a still more considerable way, by adding a less rudimentary algorithm of post-processing than "OR" policy, such as a soft-voting or a time variable.

Test number	Damage type	SVM binary	Random forest binary	Results
TVAL1	Inner	Fail	Fail	Fail
TVAL3	Blocked	Pass	Pass	Pass
TVAL4	Outer	Fail	Pass	Pass
TVAL5	Inner	Pass	Fail	Pass
TVAL7	Blocked	Pass	Pass	Pass
TVAL9	Healthy	Pass	Pass	Pass
TVAL12	Blocked	Pass	Pass	Pass
TVAL13	Outer	Pass	Fail	Pass
TVAL16	Blocked	Pass	Pass	Pass
TVAL18	Healthy	Pass	Pass	Pass
TTR5	Outer	Pass	Pass	Pass
TTR6	Inner	Pass	Pass	Pass
TTR9	Inner	Pass	Fail	Pass
TTR11	Outer	Pass	Pass	Pass
Total fault detected	-------	12/14	10/14	13/14

Number of movers

Inner detected = 3/4
Outer detected = 4/4
Blocked detected = 4/4

Total faulted detected=11/12

Figure 4.53: Results of the two Machine Learning Classifiers with the "OR" disjunction in Validation Loop 2.

Test number	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
TVAL1	H	H	H	H	H	H	H	H	H	H	H	I
TVAL3	H	B	H	H	H	H	H	H	H	H	H	H
TVAL4	H	H	H	H	H	H	H	H	H	O	H	H
TVAL5	H	H	H	H	I	H	H	H	H	H	H	H
TVAL7	H	H	H	H	B	H	H	H	H	H	H	H
TVAL9	H	H	H	H	H	H	H	H	H	H	H	H
TVAL12	H	H	H	H	H	H	H	H	H	H	H	B
TVAL13	H	H	H	H	H	H	O	H	H	H	H	H
TVAL16	H	H	H	H	B	H	H	H	H	H	H	H
TVAL18	H	H	H	H	H	H	H	H	H	H	H	H
TTR5	H	H	H	H	H	O	H	H	H	H	H	H
TTR6	H	H	H	I	H	H	H	H	H	H	H	H
TTR9	H	H	I	H	H	H	H	H	H	H	H	H
TTR11	O	H	H	H	H	H	H	H	H	H	H	H

Number of movers

Green = true healthy
Yellow = false faulty
Red = true faulty
Violet = false healthy

Inner detected = 3/4
Outer detected = 4/4
Blocked detected = 4/4

Total of fault detected= 11/12

Figure 4.54: Results of the Machine Learning Classifiers with the "OR" disjunction in Validation Loop 2.

Prediction

State	Faulty	Healthy
Faulty	11	1
Healthy	28	128

Accuracy: 82.27%
Faulty recall: 91.67%
Faulty precision: 28.21%

Figure 4.55: Confusion Matrix of the two Machine Learning Classifiers with the "OR" disjunction in Validation Loop 2.

4.7 Conclusions

By means of this data-driven approach based on the use of two machine learning models, the following conclusions are reached.
Granted that the system taken into account in this specific case is technologically recent, there is not yet any case of real damaged bearings. In order to overcome the problem, a methodology, which uses artificial damages, was created; it allows to explore data-driven solutions even in the early stages of the development of condition monitoring infrastructures. The machine learning models can classify the different types of damages and the healthy state of the bearings correctly, with good precision, recall and accuracy. It is possible to improve the performances of the single algorithm by means of a scoring method of the forecasts.
Even if literature often shows that deep learning algorithms have better performances than normal machine learning algorithms, they were not taken into consideration in this thesis because they need very large datasets for high performances. Since the test datasets used in this study are not very big, the machine learning algorithms show better performances than deep learning algorithms. The figure below shows a comparison between the performances of deep learning models and the ones of machine learning models on the basis of the size of the datasets.

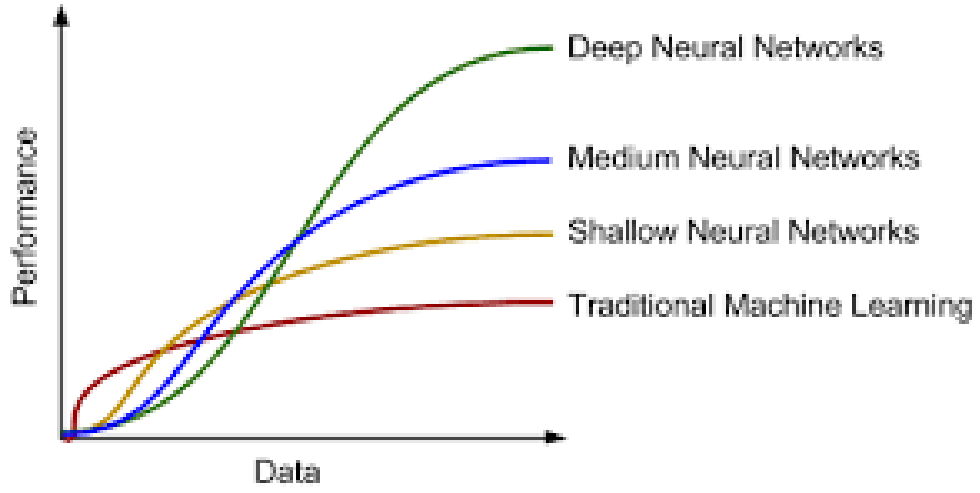


Figure 4.56: Performance comparison among neural networks and machine learning (from [106]).

Chapter 5

Model-based approach for Independent Carts System

This chapter deals with a model-based approach for Condition Monitoring System. In order to use this approach for an Independent Carts System, it is necessary to highlight the complexity of this mechatronic system, which merges mechanics, electronics and control system. Two models have been developed; both of them take into account the geometry of the carts, their variable motion profiles, their variable load profiles and different shapes of the track and of the bearings. The first model uses a signal formulation for the simulation of the vibration signals produced by the carts in case of healthy and faulty bearings, while the second one uses a multibody model. The simulated data help to reduce the complexity of the system by simulating only the most important phenomena; moreover, they can be used to develop condition monitoring thresholds and to train machine learning and deep learning algorithms [107]. Many studies are focused on dynamic models of bearing faults, especially for rotary motors that use different modelling techniques [108].

5.1 Vibration signal model

As regards the vibration signal model, McFadden and Smith [109, 110] proposed to model a bearing as an epicyclic gear where the outer ring is the annular gear, the cage is the planet carrier, the rolling elements are the planet gears and the inner ring is the sun gear. Sawalhi and Randall [111] simulated the bearing fault as a train of impulsive signals. The amplitude of these fault impulses is related to different factors such as the contact point, the load and the angular velocity of the bearing [112]. Malhi [113], Kiral and Karaglle [114] modelled the damage as a force function in FEM models. The vibration signal model proposed in this chapter is based on the work by D'Elia [115] and it is the extension of the theoretic one[116]. The model has been extended to multiple bearings with localized faults. Furthermore, a more accurate load modulation based on Tomovic [117] and lubricant and slipping effects have been taken into consideration together with the dynamic load variation of the Independent Cart System [118]. The models have been tuned and validated by means of a campaign of experimental tests performed in different operational

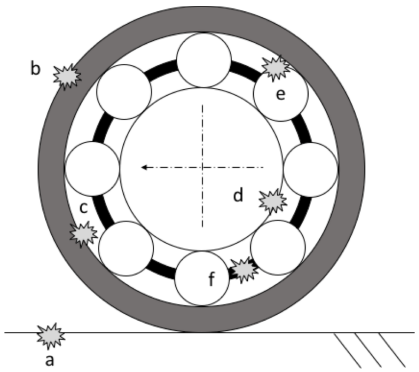


Figure 5.1: Ball bearing structure and possible damage locations: a) rail, b) external bearing, c) outer ring, d) inner ring, e) rolling element and f) cage.

conditions.

5.1.0.1 Vibration signal formulation

The incipient fault of a rolling bearing is commonly caused by fatigue and wear. In this case, the elements of the bearing are characterized by a local loss of material (spalling, pitting, etc...). Whenever an element of the bearing impacts on a damaged part, it is possible to detect a load fluctuation that excites the structural resonance of the bearing and of the accelerometer sensor. The repetition of the impacts produces a series of impulses that depends on the rotational velocity and the geometry of the bearing [112]. The amplitude of the vibration signals depends on the wear state of the damaged part, on the load applied to the bearing and the position of the internal components of the bearing itself. As shown below, the vibration signal of a localized fault is modelled on the basis of the reading of a sensor placed on the bearing.

$$x(t) = \sum_{i=-\infty}^{\infty} h(t - iT - \tau_i)q(iT) + n(t) \tag{5.1}$$

Let h(t) be the impulse response to the impact on the damage point, T the interval between two consecutive impacts due to a specific type of damage, q(iT) the modulation of the signal produced by the load distribution, τ the uncertainties due to the random slipping of the roller on the track, n(t) the background noise and i is the integration time step.
Fig. 5.1 shows the areas of the possible faults of rolling bearings, namely on the outer ring (c), the inner ring (d), the rolling element (e) and the cage (f). Furthermore, the bearing rolls on a rail and consequently two other faults may take place: a fault on the rail (a) and a fault on the external surface of the bearing (b).

Each type of fault has a unique fault frequency that depends on the rotational speed and the geometry of the bearing. By means of the fault frequency it is possible to define the T interval between two consecutive impulses. Typically, in the case of a fixed inner race and in the presence of a radial load, an inner race fault (d) will produce a uniform amplitude modulation; in the case of an outer race (c), a periodic

amplitude modulation at the period of the outer race rotation; finally, in the case of a rolling-element fault (e), a periodic amplitude modulation at the period of the cage rotation. As regards a fixed inner race, the fault frequencies are the following:

$$f_{outer} = \frac{Z}{2} f_r (1 - \lambda cos\beta) \tag{5.2}$$

$$f_{inner} = \frac{Z}{2} f_r (1 + \lambda cos\beta) \tag{5.3}$$

$$f_{ball} = \frac{f_r}{2} \frac{1 - (\lambda cos\beta)^2}{\lambda} \tag{5.4}$$

$$f_{cage} = \frac{f_r}{2} (1 + \lambda cos\beta) \tag{5.5}$$

where $\lambda = \frac{d}{D}$ is the ratio between the ball diameter d and the pitch diameter D of the bearing, Z is the number of rolling elements, β is the ball contact angle and f_r is the rolling frequency of the outer ring.

The resulting modulation frequencies [119] are:

$$Modulation_{outer} = f_r \tag{5.6}$$

$$Modulation_{cage} = \frac{f_r}{2} (1 + \lambda cos\beta) \tag{5.7}$$

As regards linear motors, two other types of fault frequencies are to be taken into consideration: the damage on the external surface of the bearing and the damage on the rail:

- Damage on the external surface of the bearing, its characteristic frequency is equal to:

$$f_{surface} = \frac{v}{\pi D_{out}} \tag{5.8}$$

  where D_{out} is the external surface diameter and v is the linear velocity, this is the governing parameter of the cyclic properties of the excitation due to faults.

- Damage on the rail, its characteristic frequency is equal to:

$$f_{rail} = \frac{v}{l} \tag{5.9}$$

  where l is the distance between the damages along the rail and v is the linear velocity of the bearing. If multiple bearings are installed on the same rail, the signal will be repeated according to the number of moving components and the distance among them. The model simulates the bearing vibration as an accelerometer is placed on the top of the mover. In this thesis, the variability of transfer path from one mover to another is not taken into consideration.

5.1.1 Vibration model implementation

The input parameters of the model are:

- Motion profile of the cart moving along the track (constant or variable velocity profiles)
- Characteristics of the cart, for example the centre of gravity (COG) and the distance among the bearings
- Characteristics of the track such as length, shape, mechanical cams and mechanical misalignment
- Load profiles depending on the machine functions
- Geometry of bearings

The implementation of the model is based on Eq. 5.1, where the T interval between consecutive faults is considered as a distance of positions along the motion profile.

This formulation allows to transpose the fault frequency as a position variation of the bearing. In this way the damage of an element of the bearing is directly correlated with the position of the bearing itself along the motion profile. In order to do that, all the fault frequencies have been rewritten as a projection of two consecutive faults along the external surface of the bearing. All the aforementioned fault frequencies can be rewritten as shown below:

$$\Delta p_{outer} = \frac{2\pi}{Z} D_{out} \frac{1}{1 - \frac{dcos\beta}{D}} \tag{5.10}$$

$$\Delta p_{inner} = \frac{2\pi}{Z} D_{out} \frac{1}{1 + \frac{dcos\beta}{D}} \tag{5.11}$$

$$\Delta p_{ball} = 2\pi\lambda D_{out} \frac{1}{1 - (\lambda cos\beta)^2} \tag{5.12}$$

$$\Delta p_{cage} = 2\pi D_{out} \frac{1}{1 + \lambda\beta} \tag{5.13}$$

where Δp is the distance between two damages of the same type, projected on the external surface of the bearing. Defining T interval according to the position and not to time has the benefit of making simpler the correlation between the position distances along the motion profiles at variable velocities with respect to the correlation between the time intervals along the motion profiles at variable velocities.

With this formulation, the fault vibration signal is represented by a series of equispaced impulses. Actually, the fault signals are not equispaced because of different phenomena: for example, the rolling elements can slightly move with respect to the cage because of the clearance between them. As a consequence, the contact angle of the rolling elements varies according to the fact that they are inside or outside the load zone [120]. Another phenomenon is the slipping of the rollers on the lubricated rail.

In order to model these uncertainties of the system for each Δp, a random slipping factor *r(t)* with a mean equal to zero and with a standard deviation equal to 1% [119] of Δp is summed. This factor is represented by τ, which is the element of uncertainty, in Eq. 5.1. The load modulation *q(t)* of the bearings consists of two elements. The first element is the variation of load on the rolling elements of the bearing because of the rotation of the bearing itself. It depends on the azimuth position of the rolling element and it can be computed by analyzing the contact deflection of the bearing and its geometry. Sjovall [121] modelled this phenomenon as a modulating factor multiplied by the radial load on the bearing. The modulating factor is computed by means of Sjovall integrals $J_r(t)$ and $J_a(t)$; the computation of the load zone parameter ϵ, the axial load F_a and radial load F_r are computed as follows:

$$W(\Psi) = W_{max}\left[1 - \frac{1}{2\epsilon}(1 - \cos(\Psi))\right]^p \tag{5.14}$$

where $W(\Psi)$ is the load distribution, ϵ is the load distribution factor that takes into consideration the influence of the ball stiffness on the total bearing stiffness, $p = 3/2$ for ball bearings ($p = 40/37$ or $10/9$ for roll bearings) and Ψ is the half load-zone angle and it is equal to:

$$\Psi(\epsilon)\begin{cases} \cos^{-1}(1-2\epsilon) & \epsilon \leq 1 \\ \pi & \epsilon \geq 1 \end{cases} \tag{5.15}$$

While the maximum load of a rolling element W_{max} is equal to:

$$W_{max} = \frac{1}{Z}\sqrt{\left(\frac{F_r}{J_r}\right)^2 + \left(\frac{F_a}{J_a}\right)^2} \tag{5.16}$$

where Z is the number of spheres. The exact formulation of J_r and J_a could be find in [121]. In this way for each angle it is possible to compute the load on the rolling element. The second element is the variation of load on the whole cart computed at the contact point of each bearing. The modulation is due to the mechanical cams, which are used in order to move the end-effector placed on the cart, to the variation of acceleration during the process and to the inertia of the cart. This contribution is calculated for each bearing with respect to the position of the cart along the motion profile.

The implementation of the model can be divided into 5 steps:

1. Define a vector S filled with zeroes that represents the expected vibration signal recorded by a sensor placed on the moving cart. The length of the vector is equal to L. Each cell of the vector corresponds to a period of time equal to $t = \frac{1}{f_s}$ where f_s is the sampling frequency of the sensor in Hertz.

2. Resample the defined motion profile *M(t)* and the load vector *q(t)* at the same sampling frequency f_s as the one of the sensor.

3. Place 1 in the cells of vector S with index values defined by dividing *M(t)* by $\Delta p_e + r(t_i)$ where $r(t_i)$ is a random factor. In this way the phenomenon of slipping is taken into account along the whole motion profile. The factor $\Delta p_e + r(t_i)$ is a rational number and its value may be between two samples of the motion profile. Therefore an error is introduced and it depends on the sampling frequency f_s (the greater f_s, the lower the error), but it is possible to compute for each division the correlate error as below:

$$E(t_i) = |M(t_i) - \Delta p_{e,i}| \tag{5.17}$$

4. Weight vector S by means of the vector of the load $q(t)$ computed at the contact point between the bearing and the rail.

5. Filter the generated vector S by means of the FFT-based model with overlap-add method, which has a filter coefficient equal to the acceleration of the impulse response of a SDOF system. This type of filtering is based on the principle that the multiplication in the frequency domain is equal to the convolution in the time domain. It consists in the following operations:

   - transformation of the input signal (in this case vector S) into the frequency domain through the FFT,
   - multiplication of the result obtained from the previous operation by the frequency response of the filter (in this case the impulse response of the SDOF),
   - transformation of the result into the time domain by using the inverse FFT.

   The generated response of the SDOF system to a unit impulse in time domain is:

$$x_{SDOF}(t) = h(t) = \frac{1}{m\omega_d} e^{-\zeta\omega_n t} sin(\omega_d t) \tag{5.18}$$

   where m is the system mass, ζ is the damping coefficient, ω_n is the natural frequency in rad/s and $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

In the case of simulation of multiple faults on the same bearing, step 3 has to be repeated for each fault. When the model simulates multiple bearings, the five steps have to be repeated for each bearing.

5.1.2 Mathematical validation

In this section the proposed model is used for the simulation of the expected vibration signal for a faulted bearing. The geometry of the bearing and the direction of the axes are shown in Fig. 4.8 (c), while the complete system, where the length of the stators is indicated, is illustrated in Fig. 5.6. In this simulation, the cart is

Table 5.1: Model data for localized outer ring fault in bearings A & B

Bearings A & B parameters	Constant velocity
Dout (mm)	21.2
d (mm)	2.75
D (mm)	14.48
$\beta(^\circ)$	9
Number of spheres	7
Velocity (m/s)	1
Rotational frequency f_r (Hz)	15.05
Sampling frequency f_s (Hz)	$20E3$
Outer-ring fault frequency (Hz)	42.79
SDOF spring stiffness k (N/m) [122]	$6E7$
SDOF damping coefficient ζ	5%
SDOF natural frequency f_n (Hz)	333

connected to the elliptic rail by means of three bearings. The system is made up of two circular stators 0.5 m long, an upper stator and a lower stator, each of which is 1 m long. The rail has the same shape of the stators. The model simulates the expected vibration signal of the right upper bearing (called bearing B) for a single cart moving along the rail. Table 5.1 lists the characteristics of the bearing taken into account and the fault frequency computed by the use of Equations 5.2-5.9 in the case of constant speed.

The radial load profile of $q(t)$, which acts on the bearing, has been computed by means of the knowledge of the motion profile $M(t)$ and the equations of dynamics. The radial load is the load orthogonal to the point of contact between the roller and the rail. The direction of the load is taken with reference to the local reference system on the cart as in Fig. 4.8 (c)). The forces taken into account are the following:

- Gravity force
- Fictitious forces (e.g. centrifugal forces)
- Preload of the bearing

The sum of the above-listed forces with respect to the position of the cart is drawn in Fig. 5.2. In the simulation, the noise is equal to zero to make the output data more readable. But it is possible to include also Gaussian noise ($n(t)$) to output signal as in Eq. 5.1. All the aforementioned possible damages have been taken into account and validated, but, for sake of brevity, only the results of the outer ring damage with the modulation of the radial load are shown. The next section describes a simulation of an outer ring fault on a single bearing of the cart with a constant velocity motion profile.

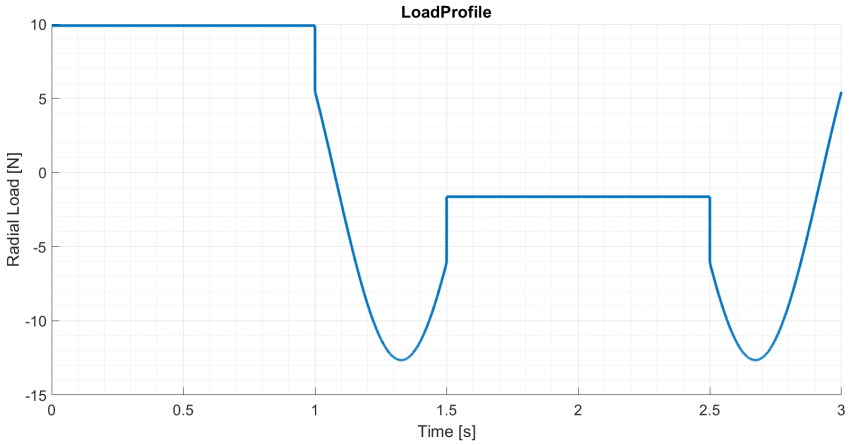


Figure 5.2: Load along the path.

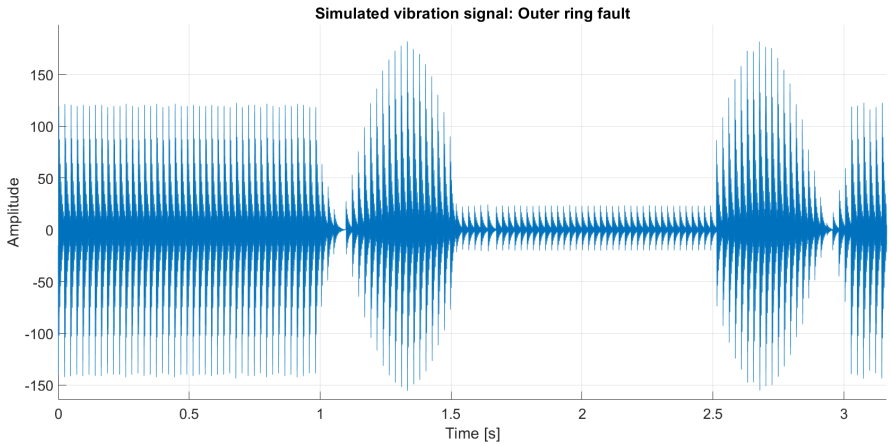


Figure 5.3: Simulated vibration signal.

5.1.3 Constant velocity

In the simulation the aforementioned data of the track and of the bearing are used, the cart has a constant velocity of 1 m/s and it executes a single round of the path starting from 0 m point to the end point, as shown in Fig. 5.6.

Fig. 5.4 illustrates the FFT of the vibration signal in the constant load zone, it shows the resonance of the system at 6 kHz clearly. The resulting vibration signal (Fig. 5.3) shows a variation of the load profile due to the change of orientation of the mover along the track and the two curves of the rail. Fig. 5.5 illustrates the envelope of the vibration signal after a filtering between 5500 Hz and 6500 Hz when the rotational frequency of the bearing (f_r) and the outer ring damage frequency (f_{outer}) are clear.

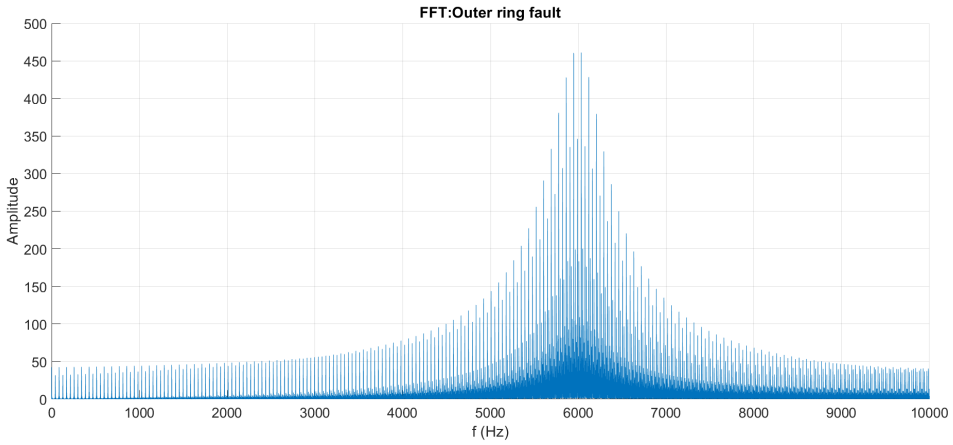


Figure 5.4: FFT of the vibration signal.

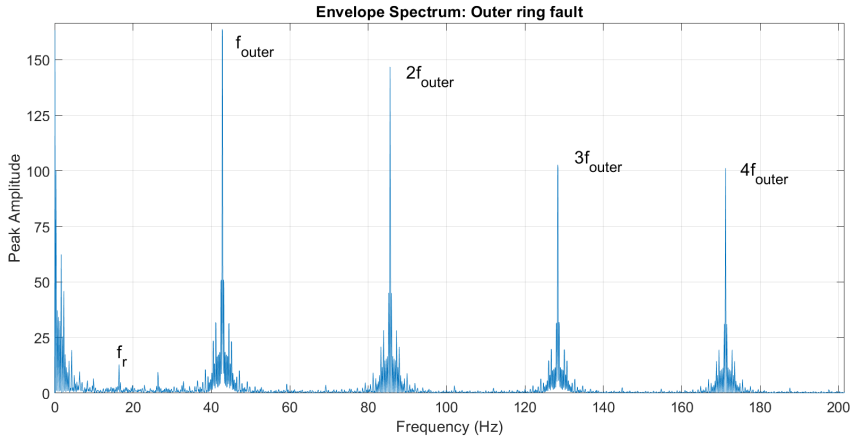


Figure 5.5: Envelope of the simulated signal.

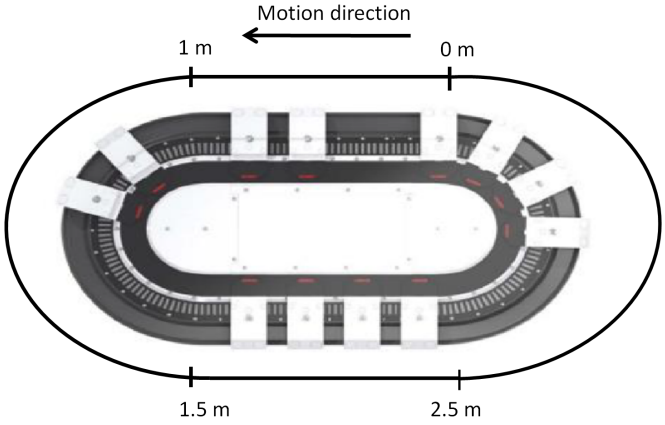


Figure 5.6: Beckhoff XTS Independent Cart System with 12 carts.

5.2 Experimental validation

In this chapter the experimental validation of the model is explained. The first part describes the experimental setup, the second section shows the experimental and simulated data in the case of healthy bearings and misalignment of the rail and the third section illustrates the experiments with a bearing damaged on the outer ring.

5.2.1 Experimental setup

The experimental activity has been carried out on a specific test rig made up of Beckhoff XTS system (Fig. 5.6) with GXF Hepco guidance and the cart shown in Fig. 4.8 in order to verify and validate the proposed model. The system taken into account consists of two circular paths that are 0.500 m long and an upper and a lower linear paths, each of which is 1.750 m long. For the length of the machine 14 straight motors and 2 curved motors are necessary. The length of the track is different from the theoretical one used in the mathematical validation, but the geometry of the mover is the same. A cart, connected to a lubricated straight rail by means of three rolling bearings (A-B-C), as in Fig. 5.8 and Fig. 4.8, moves along the motors. Bearings B and A have the geometrical dimensions described in Table 5.1; the dimensions of bearing C are listed in Table 5.2. The SDOF system, which represents the response of the bearings to a damage, has a spring stiffness k = 6E7 N/m, a damping coefficient $\zeta = 5\%$ and a natural frequency f_n=333 Hz for the bearings A, B and C. An accelerometer 356A02 - placed on the top of the cart - measures all the vibrations along the vertical direction with the sampling frequency of 20 kHz. The tests have been performed with two motion profiles: in the former the cart follows a trapezoidal motion profile consisting of three zones: an acceleration zone, a constant velocity zone and a deceleration zone, along the straight upper part of the track; in the latter the cart executes a complete round along the ellipsoid track at constant velocity.

In the first case the motion profile is produced by the cart moving from the right

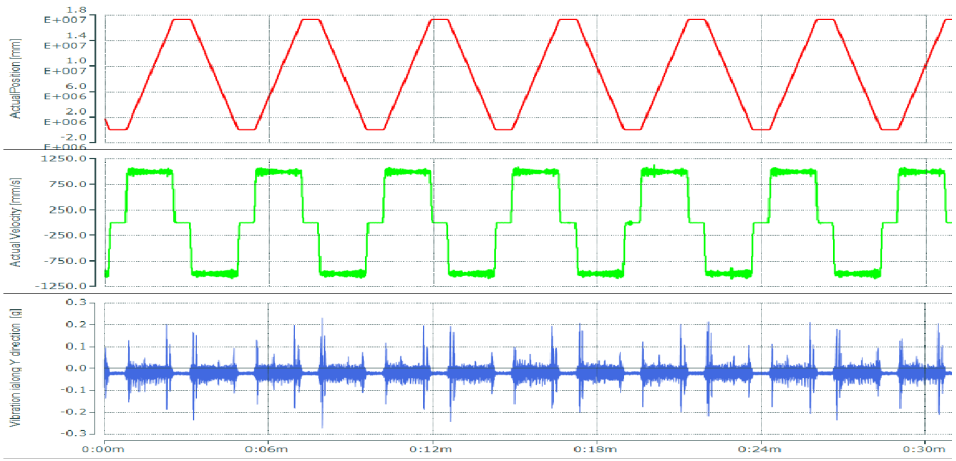


Figure 5.7: Actual position profile (red), actual velocity profile (green) and vibration signal (blue) of the cart.

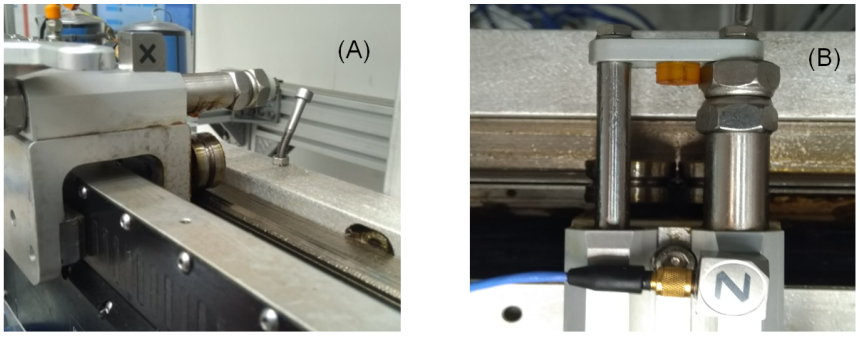


Figure 5.8: Test rig: A) picture of the side of the mover along the straight motor B) picture of the upper part of the mover on the rail. The axes of the accelerometer correspond to the reference system considered on the cart: X-axis along the moving direction, Z-axis orthogonal to the rail (load axis), Y-axis in accordance with the other axes.

to the left side and then from the left to the right side (Fig. 5.7).

In this case the test has been run at four different velocities of the cart along the constant velocity zone:

- 500 mm/s,
- 1000 mm/s,
- 1500 mm/s,
- 2000 mm/s.

The target of the first case is to validate the model at a constant velocity. For this reason the vibration signal is windowed in the constant velocity zone through the cutting and the reassemblage of the signal in the region where the cart is moving at the constant velocity. In the second case, the mover executes three rounds along

Table 5.2: Vibration signal model data of bearing C

Bearing C parameters	Constant velocity
Dout (mm)	31.1
d (mm)	4.3
D (mm)	20.06
$\beta(^\circ)$	9
Number of spheres	7
Velocity (m/s)	1
Sampling frequency f_s (Hz)	20000
Rotational frequency f_r(Hz)	10.24
SDOF spring stiffness k (N/m) [122]	$6E7$
SDOF damping coefficient ζ	5%
SDOF natural frequency f_n (Hz)	333

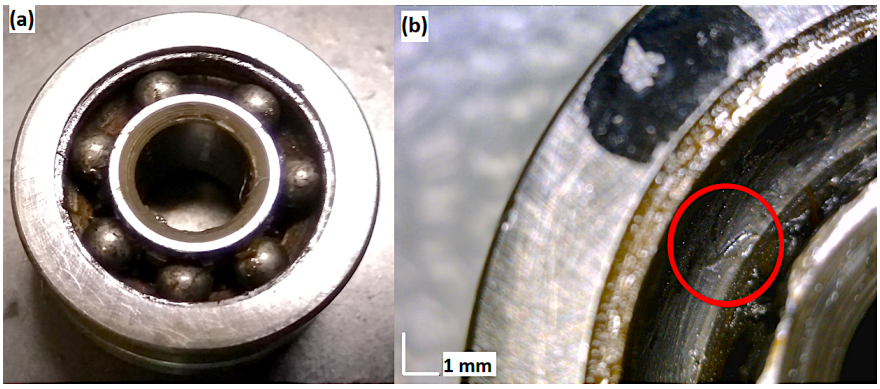


Figure 5.9: Bearing with the artificial damage on the outer ring a) bearing B b) zoom on the outer ring fault of bearing B.

the ellipsoid track at the velocity of 500 mm/s. In this case the velocity is not perfectly constant since it varies along the curves because of the dynamics of the mover and the control system of the motors.

The tests have been carried out at the aforementioned velocities with two different configurations of the mover (Fig. 4.8 (c)). In the first configuration all the bearings of the mover are healthy, in the second configuration bearing B is damaged on the outer ring surface. The outer ring of the bearing has been artificially damaged by hand with a drill (Dremel 3000) having an engraving cutter (Dremel 106, 1.6 mm head). Fig. 5.9 shows the damage of the bearing. With reference to the bearing main axis, the damage results in a longitudinal engraving. The width in circumferential direction is about 1 mm, the length in axial direction is about 2 mm.

The actual position of the mover and the vibration signal recorded along the vertical direction have been used for the verification and validation of the model.

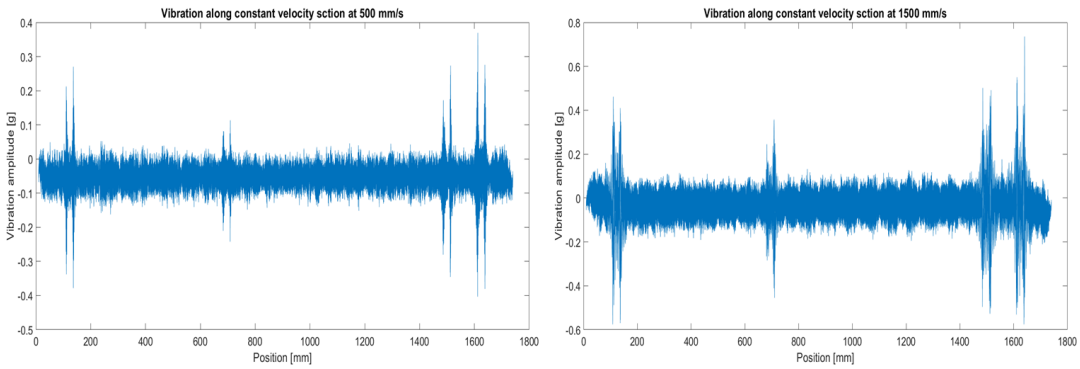


Figure 5.10: Vibration signal along constant velocity section at 500mm/s and 1500 mm/s.

5.2.2 Healthy bearings

Fig. 5.10 shows the raw vibration data in the case of a cart endowed with healthy bearings and moving along the straight upper part of the rail at the speed of 500 mm/s with respect to the absolute position of the cart. The vibrations recorded show that there are some vibration hits along the rail. The vibration hits are in the same positions even at different velocities. Some of them depend on the transition between two different motors, while the other ones depend on the configuration of the rail that presents some discontinuities due to the assembly of the mechanical system. The positions correlated to the vibration hits are shown in Fig. 5.11 and the following ones are the most important:

- Rail discontinuity: $122mm, 1500mm, 1624mm$
- Motor-to-motor transition: $700mm$.

For each position of the rail discontinuity there are three vibration peaks: the highest two are due to the two bearings (A and B) placed on the upper part of the cart, while the other one is due to the bearing (C) placed on the lower part of the mover. The time distance between the highest peaks is correlated to the distance between bearing A and bearing B that is equal to 26 mm, while the time distance between the first high peak and the low peak is correlated to the distance between bearing A and bearing C that is equal to 13 mm. The different amplitude depends on the position of the accelerometer placed on the top of the cart.
Fig. 5.12 shows the simulated signal produced by the model with the computation of the three rail discontinuities and the motor-to-motor transition in the same positions as the real ones. The amplitude of the real signal and the amplitude of the simulated signals are different, this study focuses on frequency response more than on magnitude.

5.2.2.1 Faulty bearing

The first case takes into account the cart endowed with the faulted bearing B; the cart follows the aforementioned trapezoidal motion profile.

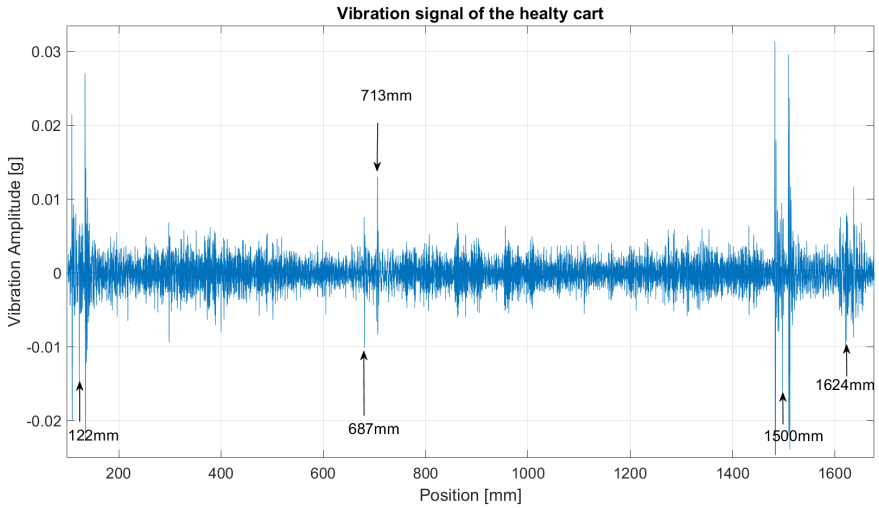


Figure 5.11: Vibration correlated to rail discontinuity and motor-to-motor transition.

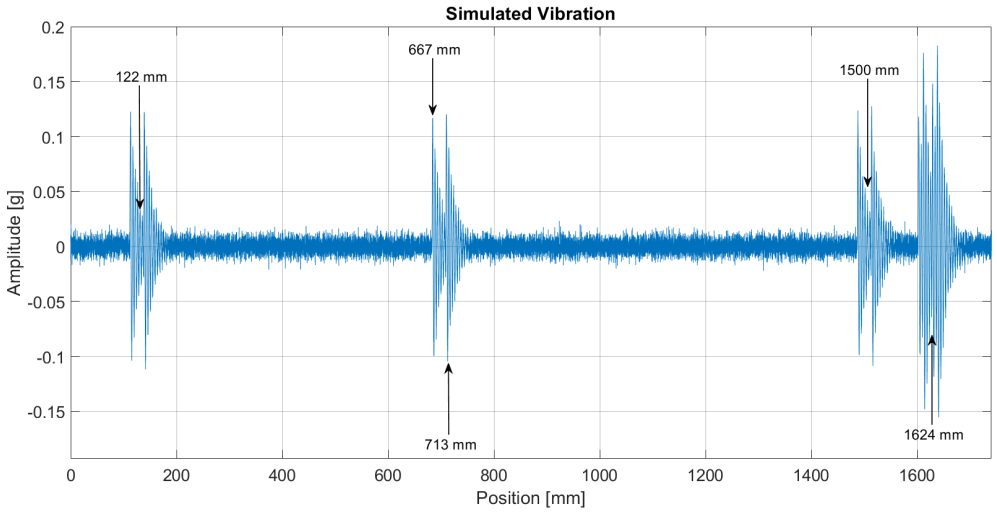


Figure 5.12: Simulated vibration signal with rail discontinuity and motor-to-motor transition.

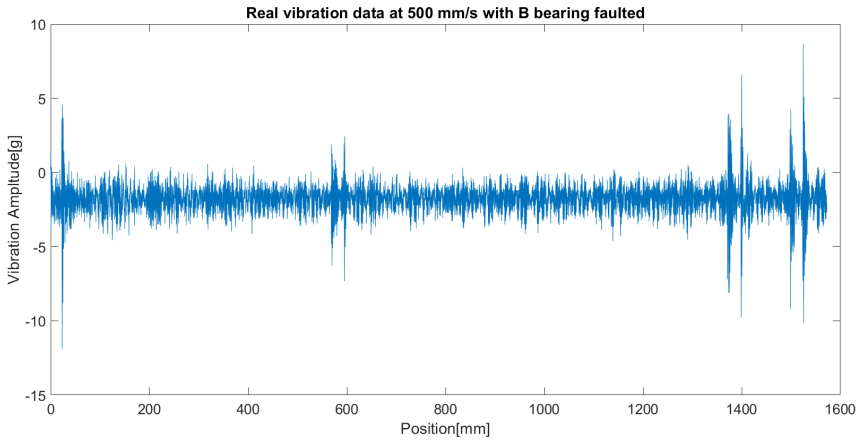


Figure 5.13: Real vibration signal at 1500 mm/s with the faulted bearing B.

Fig. 5.13 shows the raw vibration data when the cart has a trapezoidal motion profile with a peak velocity of 1500 mm/s and the bearing B has a fault on the outer ring. Even in this case the vibrations due to the rail variation and the motor-to-motor transition are evident. Through FFT transformation of the signal along the constant velocity zone of the motion profile (Fig. 5.14) it is possible to find the frequency of the fault on the outer ring of the bearing.

The model takes into consideration the rail variation, the motor-to-motor transition, the fault frequency of bearing B, the random slipping and it uses the real motion profile and the pre-computed load profile of the cart as inputs. In the time domain, the real signal and the simulated signal have a very similar periodicity, while there is a difference in the amplitude. However, the principal goal of the model is to simulate the frequency components of the experimental signal correctly.
The FFT of the simulated vibration without the random slipping is compared with the FFT of the real one in Fig. 5.14.The acquired signal consists of 20 runs of the cart, lasting 1 second each. In order to increase the frequency resolution, the 20 runs are windowed and concatenated together to get a unique vibration signal of 20 seconds (Δ f= 0.05 Hz). In Fig. 5.14, the theoretical fault frequency of the outer ring is equal to 64.18 Hz, but it is equal to 60.4 Hz in the real data. The frequency shift is caused by the slipping between the roller and the lubricated rail. As a matter of fact, if the random slipping is enabled in the simulation, the spectrum of the simulated signal is not anymore discrete and the fault frequency is reduced to 60.76 Hz as in Fig. 5.15. The FFT of the real vibration signal presents a component at 1Hz and its harmonics due to the windowing of the signal in the constant velocity zone.

Fig. 5.16 illustrates the second case: the cart, which has the faulty bearing, performs three complete rounds along the ellipsoid track at the constant velocity of 500 mm/s. It also shows the comparison between the raw and the simulated vibration data. It could be recognizable the presence of synchronous impacts in both experimental and simulated data. In the experimental case a high level of noise along all the signal is present, while in the simulated case the impacts are

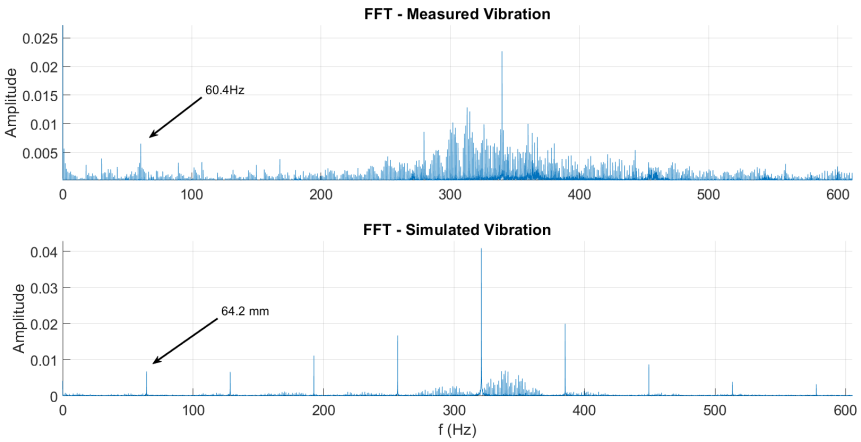


Figure 5.14: Comparison between the FFT of the real data and the FFT of the simulated data.

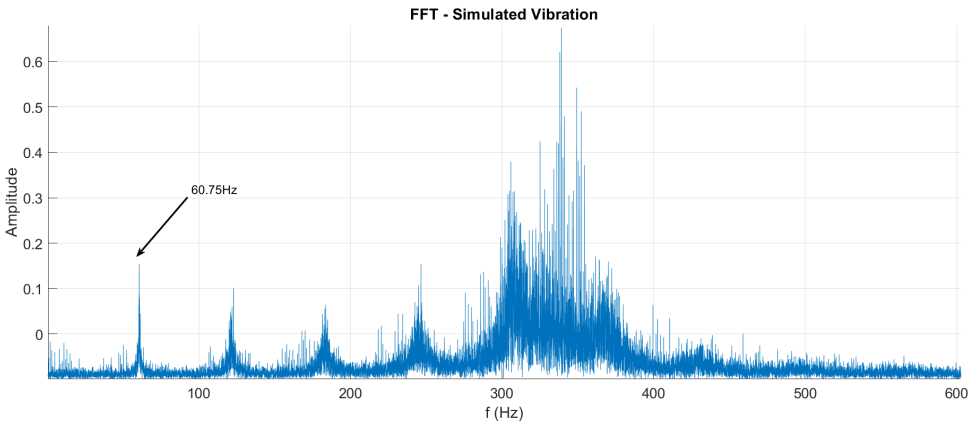


Figure 5.15: FFT of the simulated data with random slipping.

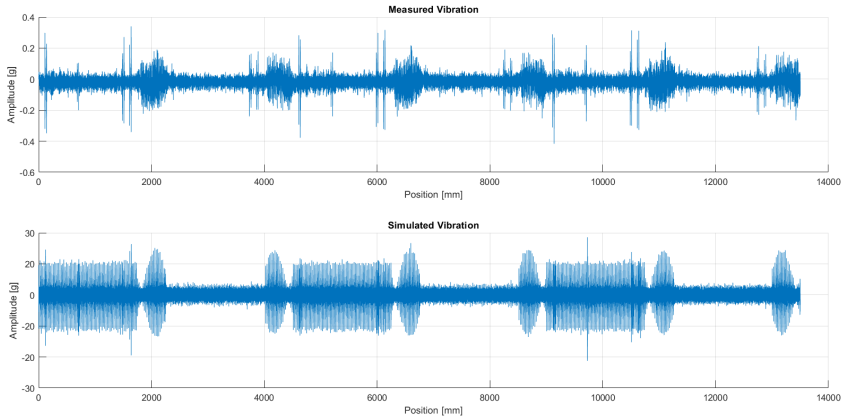


Figure 5.16: Experimental and simulated vibration signals for three rounds of the cart along the path.

more evident and clearly distinguishable from noise.

Fig. 5.17 shows the spectra of the real and simulated vibration signal and the comparison between them in the range below 150 Hz through which it is possible to appreciate the presence of numerous components. Indeed, the proposed model can only predict a small number of the numerous components present in the real spectrum. Each spectrum has been normalized by the energy in the band 0-150 Hz to foster the comparison.

With reference to the Tab. 5.1, the value of outer race fault frequency (42.79 Hz) is related to a speed of the cart of 1 m/s. In this case, the speed is 0.5 m/s, that is the outer race fault frequency decreases to 21.36 Hz. At low frequency the fault-related components are evident in the simulated data, while the effect of slipping overtakes the fault components at high frequency. In the experimental data, there are also components related to the fault frequency, in particular the first four harmonics. Both signals show a local resonance close to 330 Hz, that is the natural frequency of the bearing. The experimental data contains different high amplitude components that are probably related to the non-stationary application studied. The non-stationarity increases dynamic effects due to clearance, magnetic effects between cart and rail, inertia contribution of the cart, etc. These effects are not taken into consideration in the actual model but they will be studied and introduced in future works.

5.2.2.2 Conclusions referring vibration signal model

This section details a flexible model for the simulation of the expected fault vibration signal for Independent Cart Conveyor System.

The model allows to:

- Choose any shape of the rail track
- Use variable velocity and constant velocity profiles

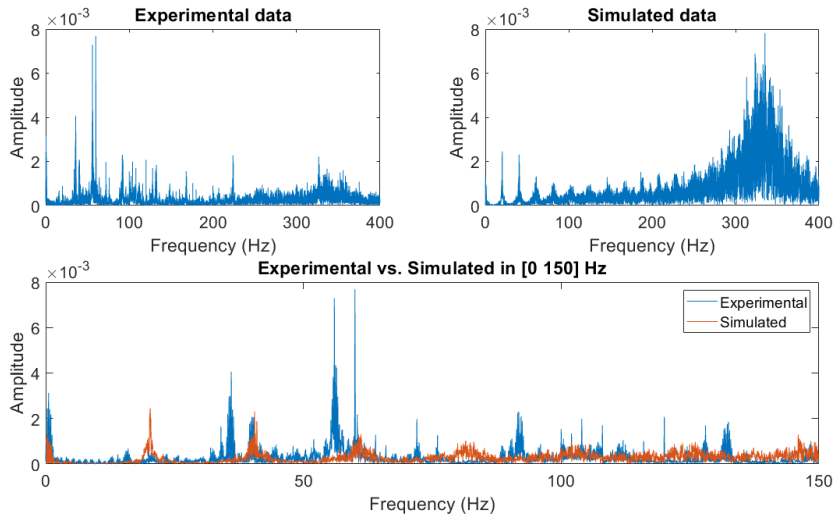


Figure 5.17: Spectra of the experimental and simulated vibration signal at 500 mm/s speed.

- Use variable load profiles
- Simulate the most common damage of a rolling bearing along a linear rail
- Consider the resonance of the system
- Consider random contributions
- Consider the modulation of the load on the bearings.

The resulting model has been used to simulate the expected vibration signal of the faulted cart moving along a short track. This track is a loop made up of two linear tracks 1.750 m long and two circular tracks 0.500 m long. The speed of the cart is kept constant along the test and a fault is made on one of the three bearings of the cart. The resulting signal has been compared with experimental data on hand-made faulted bearing. So far, the model can only foresee bearing fault components and resonances of the mechanical systems. The model could be used to simulate the vibration signal produced by a definite fault for a definite system configuration. The spectrum of experimental data reveals several high-energy frequency components that are not present in the proposed model. The characterization of these components will be investigated in future works, extending the functionality of the present model that is a preliminary but promising foundation.

5.3 Multibody model

This section will show another approach for modelling a faulty bearing of an Independent Carts System. In this case, a multibody model of a cart with healthy and faulty bearings has been developed. The first part of the study is focused on the

bearing modelling, while the second part presents the model of the entire cart. The last part illustrates the comparison between the simulation data and the real data.

5.3.1 Multibody dynamics

Multibody Dynamics (MBD) studies the dynamic behaviour of interconnected flexible/rigid bodies. The bodies are interconnected by using joints that allow large translational or rotational displacements. The joints can have some properties such as friction, stiffness and damping. The motion of the bodies is computed by means of the boundary conditions and the applied loads. There are several formulations to generate and assembly the equations of motion, which can be classified according to the type of the coordinates adopted (relative, natural or reference point Cartesian) and to the use of the redundant or minimum number of coordinates. In these models it is possible to consider the flexibility of the bodies, consequently it is possible to simulate the body deformations that can have a significant effect on the problem. Multibody models are used in different engineering areas from automotive to aerospace industries. They are principally used for:

- Parametric design to optimize complex mechanical systems.
- Simulation of the kinematic and dynamic behaviour of mechanical systems.
- Simulation of control loops that includes mechanisms.
- Analysis of coupled system frequencies and time responses.

The multibody dynamic models are used both in the first stages of the mechanical system design and in the optimization phase. The advantages of this technique are the short computational times, the very high efficiency of the simulation for parametric studies and the optimization of complex mechanical systems. The disadvantage is the lower accuracy of solution with respect to Finite Element Methods (FEM). Furthermore, some simulation software can also include FEM models with an increase of the simulation time and of the solution accuracy.

5.3.2 Multibody bearing simulation

As regards multibody bearing modelling, Harris and Kotzalas [123, 124] gave a whole description of the most relevant phenomena concerning bearings such as stiffness distribution, skewing, load etc. Xiangyang and Wanqiang [125] used a two-degree-of-freedom model that considers horizontal and vertical motions of the inner ring. With this model they developed a one-class SVM for the fault detection of bearing damages. As regards the modelling of the defects of bearings, Sopanen and Mikkola [126, 127], formulated a complete model that considers ElastoHydrodynamic Lubrication (EHL), localized defects and waviness of the bearing elements. In Modelica Leturiondo et al. [128] implemented a generic model for ball and roller bearings that considers revolute elements, cages, rings and damages as a geometric change of the different elements. The multibody model of the Independent Cart System proposed

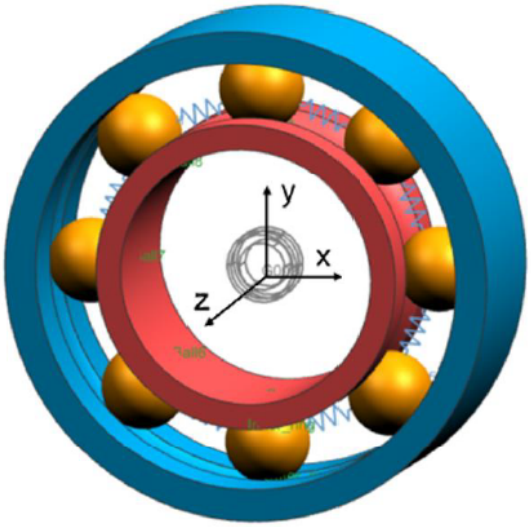


Figure 5.18: DGB6202 Bearing Model.

in this chapter consists in 96 DOF, while the bearing model that has been validated with the work by Scurria at al. [129] has 54 DOF.

5.3.2.1 Ball bearing model

The bearing model has been developed by means of Simcenter 3D Motion, a multibody software by Siemens Digital Industry Software. The model is made up of rigid bodies that are outer rings, inner rings and rolling elements. For the healthy bearing model, all the bodies are assumed to have the theoretical shape. For the cage modelling, a simplification has been used in order to reduce the simulation time. In this case, the cage is modelled as a series of springs with high stiffness, which connect the centre of each rolling element to the follower. This method was used also by Jain and Hunt [130] in order to simplify the model and, at the same time, to determine the cage force by the computation of the compression/elongation of the single spring.

In the bearing simulation, one of the most important elements is the contact model. In the model, which has been developed, only dry contact has been taken into consideration, therefore no Elastohydrodynamic Lubrication (EHL) formulation has been considered. The possible contact formulations allowed by Simcenter 3D Motion are 3D Contact Formulation and Analytical Contact Formulation. 3D Contact Formulation uses a mesh for the determination of the contact points and the direction of the Normal and Tangential Hertzian forces. This algorithm is very accurate in the computation of contact force, but it requires a very big computational effort due to the need for a fine mesh to obtain a good accuracy. This type of contact formulation is suited to conformal contact. The conformal contact takes place when two surfaces come into contact along a line, a curve or an area. On the contrary, the Analytical Contact formulation directly considers the CAD geometry of the bodies, it uses an analytical formulation that it is suited to noncoformal con-

tact. With this formulation, if the contact becomes nearly conformal, the solution can become slow. In the case of the bearing modelling, the choice has fallen upon the use of the Analytical Contact formulation because of its velocity in comparison with the one of 3D Contact and its accuracy that is suitable for the application. With Simcenter 3D Motion it is possible to impose an Analytical Contact between each rolling element and each ring. This formulation uses a penalty method to generate both impact forces perpendicular to the plane of contact and friction forces lying on this plane [131]. All the parameters are list in Fig. 5.20

The normal force perpendicular to the plane is defined as follows:

$$F_{normal}(\delta, v_{pen}) = F_{Hertzian}(\delta, v_{pen}) + F_{linear} + F_{variable}(\delta, v_{pen}) \tag{5.19}$$

F_{linear} and $F_{variable}$ can be enable in the case there are more information about the contact behaviour, in this case their value as been define as zero. For the two bodies that come into contact, it is necessary to define Young Modulus and Poisson ratio.

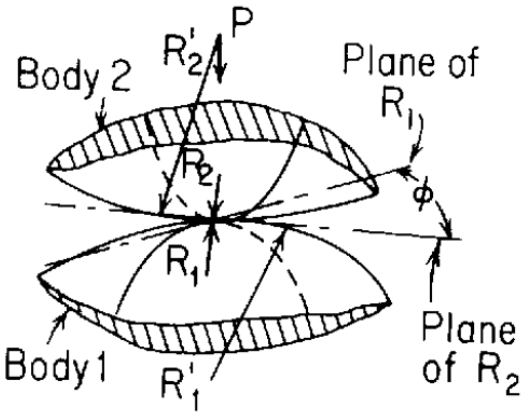


Figure 5.19: Surface contact (from [132]) .

At the point of contact the algorithm computes the minimum and the maximum radii of curvature for Body 1 and Body 2 that are: $R1, R1'$ for Body 1 and $R2, R2'$ for Body 2 (Fig. 5.19).The principal curvatures in each body ($1/R1$, $1/R1'$ for Body 1 and $1/R2$ and $1/R2'$ for Body 2) are mutually perpendicular. For the formalism, the radii are positive, if they lie on the given body. The plane containing the curvature $1/R1$ makes with the plane that contains the curvature $1/R2$ the angle ϕ.

The normal contact force is defined as follows:

$$F_{Hertian}(\delta, v_{pen}) = \mid \delta \mid^{1.5} K_{st} sgn(\delta) \tag{5.20}$$

where

$$K_{st} = K(1 - \frac{1 - e^2}{1 + e^2} \tanh \frac{2.5 v_{pen}}{v_e}) \tag{5.21}$$

$$K = \frac{\sqrt{\mid K_D \mid}}{\lambda^{1}.5C_E} \tag{5.22}$$

$$K_D = 1.5(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_1'} + \frac{1}{R_2'})^{-}1 \tag{5.23}$$

$$C_E = (\frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2}) \tag{5.24}$$

$$\lambda = 0.75(1 - \mid \cos\theta \mid^{2.17657})^{0.24586} \tag{5.25}$$

$$\cos\theta = \frac{K_D}{1.5}\sqrt{(\frac{1}{R_1}^{2} - \frac{1}{R_1'})^2 + (\frac{1}{R_2} - \frac{1}{R_2'}) + 2(\frac{1}{R_1} - \frac{1}{R_1'})(\frac{1}{R_1} - \frac{1}{R_1'})\cos 2\phi} \tag{5.26}$$

Variable	Description
δ	Current penetration at the point of contact. When contact is not occurring, $\delta = 0$.
R_1	Principal radius of curvature of first element surface at the point of contact.
R_2	Principal radius of curvature of the second element surface at the point of contact.
R_1'	Minor principal radius of curvature of first element surface at the point of contact.
R_2'	Minor principal radius of curvature of second element surface at the point of contact.
E_1	Young's Modulus of first element surface
E_2	Young's Modulus of second element surface
ν_1	Poisson's ratio of first element surface.
ν_2	Poisson's ratio of second element surface.
e	Coefficient of restitution. A value of 1 generates an elastic collision. A value of 0 generates a plastic collision. A value of 0.7 is suggested as a starting point.
v_{pen}	Current penetration velocity at point of contact.
v_s	Transition Velocity. This parameter, which is a positive number, controls the harshness of the contact force. As this value is decreased and approaches zero, the contact damping force will become more harsh and the solution performance can degrade. When this value is increased to infinity, the amount of damping applied will vanish.
ϕ	Angle between planes which contain R_1 and R_2.

Figure 5.20: Table with the meaning of parameters (from [131]).

As regards the stiffness parameter of the force [132], models are defined by a look-up table, in the Analytical Contact formulation the look-up table has been replaced by λ. In formula 5.21, the hyperbolic tangent is used for smoothing the force function as the velocity is transitioning through zero. The restitution coefficient e and the transition velocity V_e are two parameters that can be tuned by the user. Fig. 5.21 shows the variation of Hertzian force with respect to the penetration depth with three different restitution coefficients and a constant transition velocity of $0.001m/s$. The figure shows that for $\delta > 0$ the contact force is zero because there

is no contact, while when $\delta < 0$ the force is generated. The red curve represents the case in which $e = 1$, in this case the damping contribution to the contact can be considered 0, while in the other two cases with lower e it is possible to notice the hysteresis area that represents the energy dissipated in the impact.

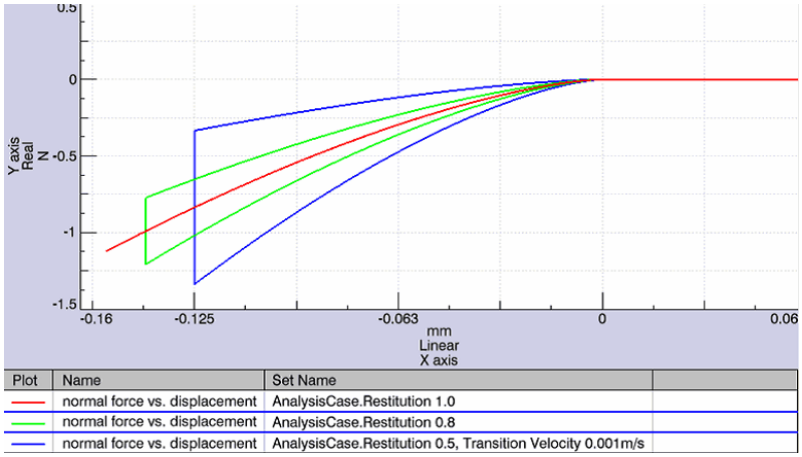


Figure 5.21: Analysis of the restitution factor from([131])

Fig. 5.22 represents Hertzian force with respect to the transition velocity with a constant value of the restitution factor. In this figure, it is possible to notice that, with an increasing value of transient velocity, the hysteresis curve tends to have smoother angles, but with high values the force tends to a line.

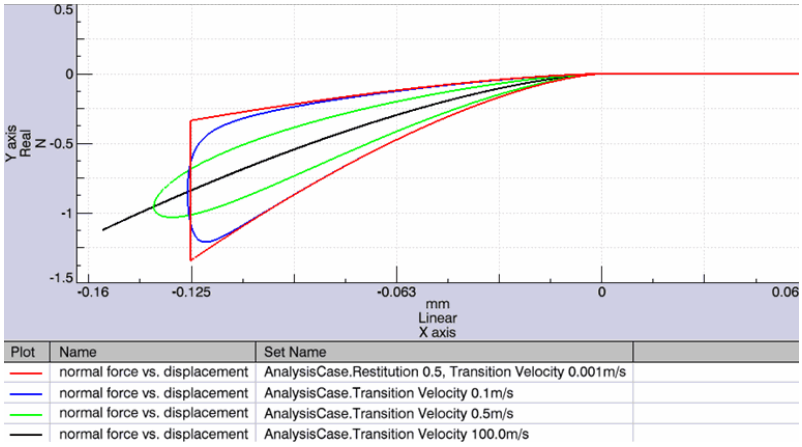


Figure 5.22: Analysis of the transition velocity from ([131]).

After the evaluation of the contact formulation, the model constrains have been imposed.

5.3.2.2 Bearing Model Constrains

The outer ring is fixed by means of a bracket joint, while the inner ring is connected to the outer one by means of a bushing. The bushing connector allows to

define stiffness and damping values along the translation (x, y, z) and the rotation (Ax, Ay, Az). The total number of degree of freedom of the model is 54. In this case all the stiffness values have been defined as $0N/mm$, because the stiffness of the bearing is defined by the contact stiffness of each rolling element. For the damping factor the following values $(x = 100Ns/m, y = 100Ns/m, z = 100Ns/m)$ and $(Ax = 0.01Nms/rad, Ay = 0.01Nms/rad, Az = 0.01Nms/rad)$ have been used because the use of damping coefficient in the bushing is helpful for the reduction of the simulation time since it makes the problem smoother for the solver.

5.3.3 Model Validation

In order to validate the multibody model, it has been compared with a state-of-the-use solution called Bearing Element already implemented in Simcenter 3D Motion.

5.3.3.1 Bearing Element in Simcenter 3D Motion

The Bearing Element can be used for modelling angular-contact ball bearings, and deep groove ball bearings. The model requires as input the displacement in 6-dof of the inner ring with respect to the outer ring. Starting from the global displacement of the inner ring, the local displacement in the position of each rolling element is computed. Therefore, the local displacement is used to compute the equilibrium of each rolling element accounting for the centrifugal load. The equilibrium of the rolling element returns the contact forces and the contact angles, which are subsequently used to compute the total reaction forces on the bearing center due to the stiffness contribution. To model the contact, two different contact models can be selected: a Hertz-based contact model referred to as Dry and a lubricated contact model accounting for the EHL regime referred to as EHL Steady. As far as the damping contribution is concerned, a damping matrix describing the global energy dissipation of the bearing is used. This model is based on the following assumptions:

- Steady-state conditions: the solution is computed in a time independent way. This assumption remains true if the bearing is excited with frequencies below the natural frequencies;
- The lubricated contact is considered as fully flooded (e.g. no starvation);
- The rolling elements have a pure rolling contact with the raceways, without any slipping;
- The bearing tilts of small angles;
- The cage is not modelled. The rolling elements can only displace on the radial plane;
- The contacts are considered to be frictionless.

The main advantage of this element lies on the fact that the use of few parameters characterizing the internal geometry of the bearing makes it possible to quickly create the model in an automatic manner involving a minimal effort. The element behaves as a force element ensuring an accurate and yet fast solution thanks to the dedicated solution method. The disadvantage lies on the fact that it can consider only the theoretical geometry of the bearing and consequently it is not possible to model damages.

5.3.3.2 Comparative study

A DGB6202 deep groove ball bearing has been modelled for the comparison (Fig. 5.23 shows the geometry of the bearing used for the comparison.)

Parameter	Value
Inner bore diameter	15 mm
Outer diameter	35 mm
Bearing width	11 mm
Inner groove radius of curvature	3.07 mm
Outer groove radius of curvature	3.24 mm
Pitch diameter	25.26 mm
Radial clearance	~ 0 um
Ball diameter	6 mm
Number of rolling elements	8

Figure 5.23: DGB6202 ball groove bearing geometry parameters.

Subsequently the two models have been tested by introducing either a force or a moment or both together along all the directions while the bearing was rotating at relatively high speed. The results have been compared in terms of the displacement and/or tilting associated to the load and in terms of computing time.
If a bearing is loading while it is rotating, the displacement response associated to the bearing can be split into two main contributions: a constant displacement due to the average bearing stiffness for that given load; and a variable displacement due to the rolling elements rotating about the bearing axis. The frequency of the variable contribution corresponds to the ball passing frequency either of the inner or outer ring depending on the constraints. In this case, being the outer ring fixed, the variable part of the displacement will correspond to the outer ring ball passing frequency. It is of great importance to accurately estimate both contributions since the constant contribution – usually greater than the variable one – will influence the position of the components connected to the bearing and the modes of the structure. Instead the variable contribution introduces excitation frequencies that can excite

the surrounding structure with consequent increase in noise levels. In this study the two contributions are split and compared separately to assess their accuracy.

5.3.3.3 Test scenarios

To compare the two bearing models, the following test scenarios are used:

- 4 radial loads along x and y: $0.5kN$, $1.5kN$, $2kN$, $5kN$;
- 3 axial loads along z: $0.5kN$, $1.5kN$, $2kN$;
- 3 tilting moments along x and y: $1Nm$, $3Nm$, $5Nm$;
- 2 mix loads:
  - Radial force along x of $2kN$ and a tilting moment along y of $1Nm$;
  - Radial force along y of $0.5kN$, an axial load along x of $2kN$ and a tilting moment along x of $1Nm$.

When the bearing is rotating on its own axis, the response to a load – force or moment – in x and y directions is equivalent and it presents only a different phase. This symmetry has been tested and confirmed. Hence only the results concerning the loading along x are shown. The cases under study including their own numbers are reported in Fig. 5.24

Case	Fx (N)	Fy (N)	Fz (N)	Tx (Nm)	Ty (Nm)
1	500	-	-	-	-
2	1500	-	-	-	-
3	2000	-	-	-	-
4	5000	-	-	-	-
5	-	-	500	-	-
6	-	-	1500	-	-
7	-	-	2000	-	-
8	-	-	-	1	-
9	-	-	-	3	-
10	-	-	-	5	-
11	-	2000	-	1	-
12	500	-	2000	-	1

Figure 5.24: Cases studied for the comparison.

The cases 1-10 are meant to evaluate the displacements directly associated with the applied load, while the cases 11-12 are meant to evaluate the coupling terms

among different loading directions. The models are running at $3000RPM$ in order to have the centrifugal loads playing a role in the behaviour of the bearings.

As regards the damping values used in both models, the following damping constants are used:

- Radial and axial damping coefficient: $100Ns/m$;
- Rotational damping along the radial directions: $0.01Nms/rad$.

5.3.3.4 Comparison of the mean displacements

The mean values of the displacement are calculated by averaging the displacement over the time of simulation making sure to truncate the simulation, in precise spots in order to remove the initial transient and the initial run-up. Moreover, it is truncated in a way that preserves the natural periodicity of the signal. The results are shown in Fig. 5.25 that collects the cases with radial and axial load, Fig 5.26 that collects the cases with tilting moments and Fig. 5.27 that collects the results from the cases with combined loads, where dq indicates the displacement in q direction ($q = x, z$) and daq the tilting in q direction ($q = x, z$).

	Multibody Bearing		Bearing Element		
Case	dx (mm)	Fx/dx (N/mm)	dx (mm)	Fx/dx (N/mm)	Err
1	0.010767	46438	0.0108	46296	-0.31 %
2	0.022347	67123	0.02252	66608	-0.77 %
3	0.027063	73902	0.02727	73340	-0.76 %
4	0.049829	100343	0.05018	99641	-0.70 %
Case	dz (mm)	Fz/dz (N/mm)	dz (mm)	Fz/dz (N/mm)	Err
5	0.079951	6254	0.08031	6226	-0.45 %
6	0.106250	14118	0.1067	14058	-0.42 %
7	0.114592	17453	0.1151	17376	-0.44 %

Figure 5.25: Comparison of cases 1-7, radial and axial loads.

Fig. 5.25 shows how the constant contribution of the displacement due to radial or axial load is very similar in the two models. This leads to a very small error ($< 1\%$) between the multibody model – considered as reference – and the Analytical Model. Furthermore, with the increase of loads, the error seems to become stable at about 0.7% for radial loads and 0.45% for axial loads. Moreover, the stiffening effect due to the nonlinear contact formulation is underlined in the figures. The dependency

of the stiffness on the load is one of the main advantages of using a physic-based model for the bearing rather than a stiffness matrix. The cases, where the force or the moment is applied to Y direction, are equivalent to the cases where the force or the moment is applied to X direction. This is due to the symmetry of the problem with respect to Z axis. In fact, these cases only differ in the phase of the variable term, while the constant part of the displacement is equivalent.

	Multibody Bearing		Bearing Element		
Case	dax (rad)	Tx/dax (Nm/rad)	dax (rad)	Tx/dax (Nm/rad)	Err
8	0.005248	190.55	0.005213	191.83	0.66 %
9	0.006942	432.15	0.006896	435.04	0.67 %
10	0.007918	631.47	0.007866	635.65	0.65 %

Figure 5.26: Comparison of cases 8-10, tilting along radial direction.

Fig. 5.26 reports the results regarding the tilting moments. As in the previous cases, the error remains lower than 1%, converging to a value close to 0.7%. Here the increment of secant stiffness is even more pronounced since the increment of the contact angle between the rolling elements and the raceways adds stiffening to the components.

	Multibody Bearing			Bearing Element		
Case	dy (mm)	dz (mm)	dax (rad)	dy (mm)	dz (mm)	dax (rad)
11	-0.02694	0.05650	-0.006072	-0.02737	0.05709	-0.006065
			Err	-1.60 %	-1.04 %	0.12 %
Case	dx (mm)	dz (mm)	day (rad)	dx (mm)	dz (mm)	day (rad)
12	-0.01126	-0.11387	-0.001966	-0.01142	-0.1144	-0.001963
			Err	-1.47 %	-0.47 %	0.15 %

Figure 5.27: Comparison of cases 11-12, combined loads.

Fig. 5.27 shows the results regarding the last two cases, where a combination of radial, axial and tilting loads is applied. This case is of great importance since the Analytical Model is designed to accurately capture the coupling terms between different directions. Even in these cases, the error remains small also considering

the complex loading condition of the bearing. In this case the stiffening cannot be taken into consideration since the two cases involve different loaded directions. A summary of the error in the different cases is reported in Fig. 5.28 where it is clear how the error always remains small in all the studied cases.

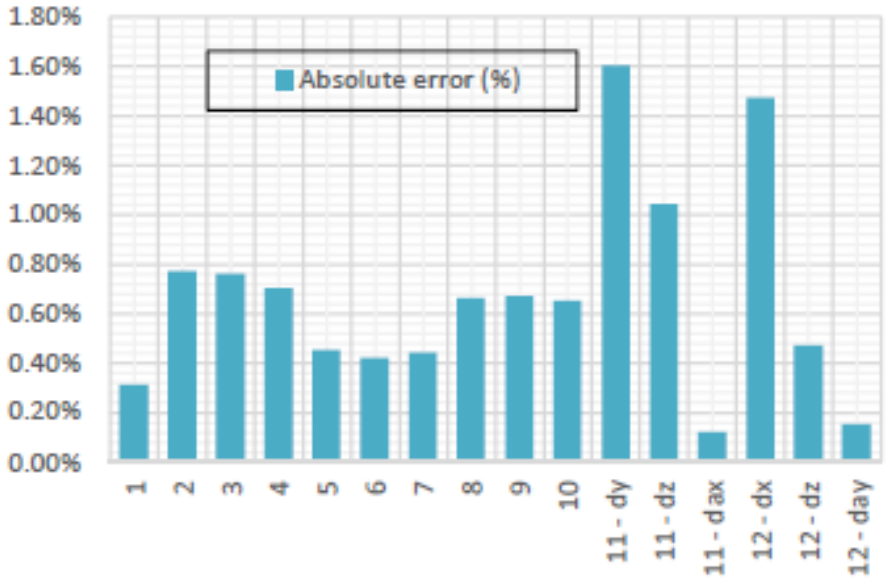


Figure 5.28: Absolute value of the error in % between the Bearing Element and the multibody model.

5.3.3.5 Comparison of the variable displacements

This section compares the two models to assess the accuracy of the Analytical Model in predicting the stiffness fluctuations due to the ball passing phenomena. In order to pursue the object, the mean value of the displacement is removed from the responses of the bearings and the curves of the two models are overlaid on the same plot. In order to reduce the number of plots reported, the comparison is shown only for the case with the greatest error in the constant part. Instead, for cases 11-12 the variable part is not reported since it follows the accuracy of the cases with single load.

Fig. 5.29 shows the comparison regarding Case 2 where a radial load of $1500N$ is applied. The curves of the Bearing Element and the Multibody Model are almost overlapping, which confirms that the contact models are equivalent. In addition to it, the curves show the same frequency, which confirms that the assumptions introduced in the Bearing Element do not present significant errors and in particular the no-slip hypothesis and the steady state solution of the equilibrium for the Bearing Element. In Fig. 5.29 it is also clear how the fluctuation of the displacement is smaller than the mean displacement showed in the previous section. For NVH studies, instead,

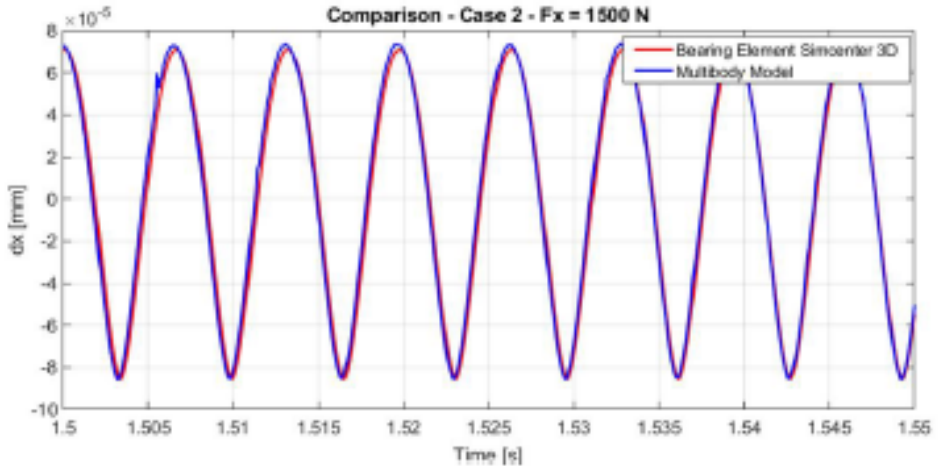


Figure 5.29: Displacement comparison: Case 2 $Fx = 1500N$.

the displacement fluctuations are of great importance since they are transmitted and potentially amplified by the surrounding structure, generating vibrations and noises.

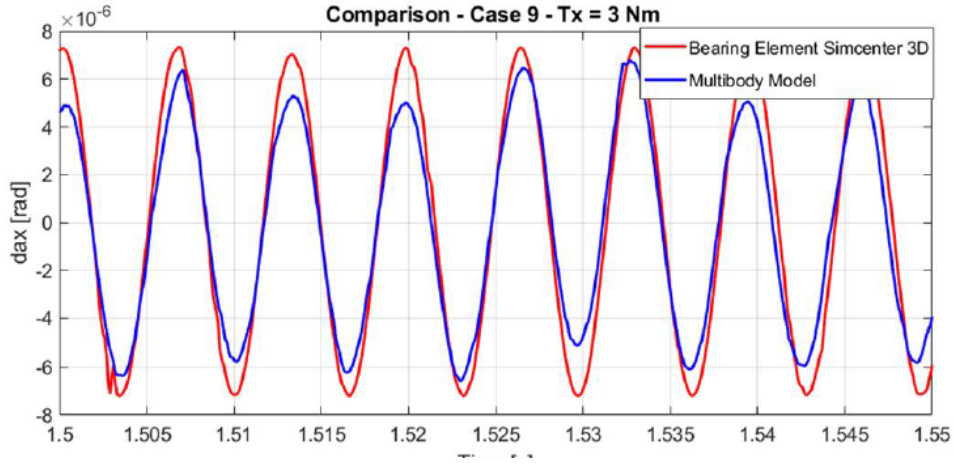


Figure 5.30: Displacement comparison: Case 9 $Tx = 3Nm$.

The two curves show a greater deviation with respect to the case with radial load. This can be due to the different contact detection algorithm used in the two models. However, even if a small deviation is observed, the two curves are still very close to each other. Moreover, the multibody model shows a transient at low frequency that obviously cannot be captured by the Bearing Model. As regards the cases where the bearing is loaded with only axial load, the displacement does not show any fluctuation since, due to the axial load, all the rolling elements are equally loaded, hence no stiffness variation is observed.

To conclude the multibody model shows a good accuracy with respect to the dedicated Bearing Element of Simcenter 3D Motion.

5.3.3.6 Comparison of computational time

This section compares the computing time between the two solutions. The main advantage of the Bearing Element is the minimal amount of bodies that have to be defined in the multibody environment. This guarantees a minimal number of degrees of freedom of the system. Moreover, the contact detection between the rolling elements and the raceways is based on the nominal geometry of the rolling elements and the raceways and allow a fast definition of the contact point. The multibody model, instead, intrinsically involves a larger amount of degrees of freedom, for this reason it requires a more expensive algorithm for the contact detection. The advantage, instead, of using the full multibody model rather than the Bearing Element, is that the geometry can differ from the nominal one – e.g. ovalization of the raceways or defects of the raceways can be modeled. This makes the full multibody model more suitable for the purpose of the simulation of damaged bearings. Fig. 5.31 reports the improvement of computing time when using the Bearing Element rather than the full multibody model. The improvement is good even on such a simple model. The Bearing Element is up to 16 times faster for complex cases, while for the easier cases – with radial load – it shows to be from 3 to 6 times faster.

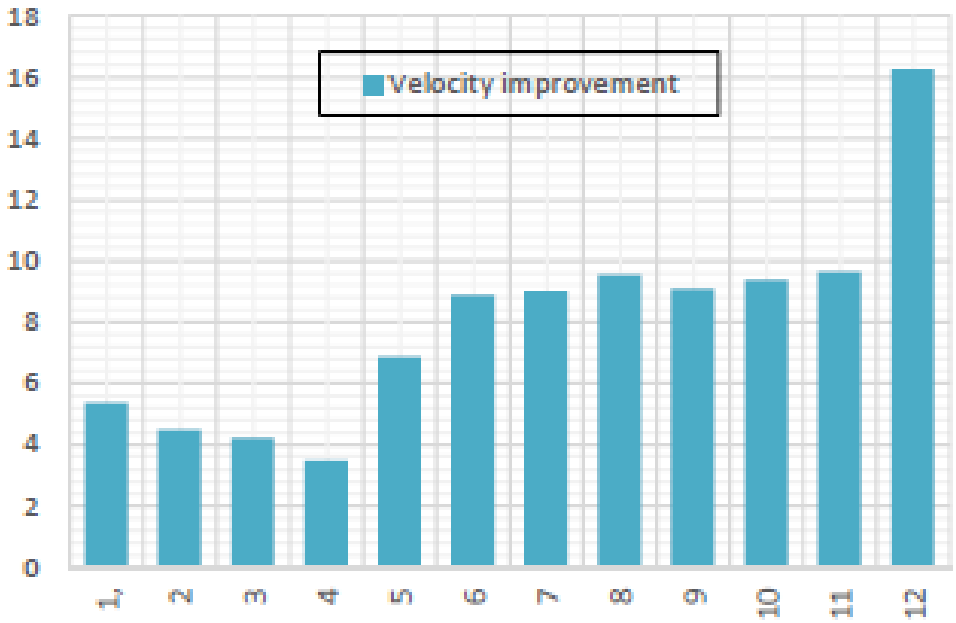


Figure 5.31: Computational velocity difference between the Bearing Element and the full Multibody model.

Fig. 5.32 reports the details regarding the computing time for the two models when a radial or axial load is applied. From the reported times, it is clear how the Bearing Element has a more stable computing time with respect to the multibody Bearing, which is faster for higher loads.

Fig. 5.33 reports the computing time for the cases where a tilting moment is applied. Here, due to the complex trajectory of the rolling elements, the Bearing

	Multibody Bearing	Bearing Element	
Case	CPU Time (s)	CPU Time (s)	Improvement
1	61.158	11.4	5.37
2	51.212	11.5	4.45
3	49.090	11.7	4.20
4	42.552	12.3	3.46
Case	CPU Time (s)	CPU Time (s)	Improvement
5	67.131	9.91	6.77
6	86.987	9.8	8.88
7	88.971	9.9	8.99

Figure 5.32: CPU time comparison in cases 1-7, radial and axial loads.

Element shows to be around one order of magnitude faster than the multibody model of the full bearing.

	Multibody Bearing	Bearing Element	
Case	CPU Time (s)	CPU Time (s)	Improvement
8	93.377	9.82	9.53
9	88.495	9.76	9.12
10	93.259	9.97	9.42

Figure 5.33: CPU time comparison in cases 8-10, tilting along radial direction.

The reduction in computational time becomes greater when loads are combined, as reported in Fig. 5.34 and in particular, in Case 12 where the bearing is loaded radially, axially and with a tilting moment.

5.3.3.7 Damaged model

After the validation of the healthy bearing, the damage case has been taken into consideration. In order to model the damage along one of the two rings with Simcenter 3D motion, the healthy surface of the ring has been created without an angular piece. In the place of the missing angular piece, a new surface has been built, it represents the damage. The damage surface has been created by increasing the radius of the ring. With this type of damage modelling, it is possible to simulate a brinelling/pitting damage on the inner and/or the outer ring. It can also have

	Multibody Bearing	Bearing Element	
Case	CPU Time (s)	CPU Time (s)	Improvement
11	77.142	9.61	8.03
Case	CPU Time (s)	CPU Time (s)	Improvement
12	189.751	11.69	16.23

Figure 5.34: CPU time comparison in cases 11-12, tilting along radial direction.

different dimensions, for example different depth or width. An "analytical contact" has been defined between the damage surface and each ball.

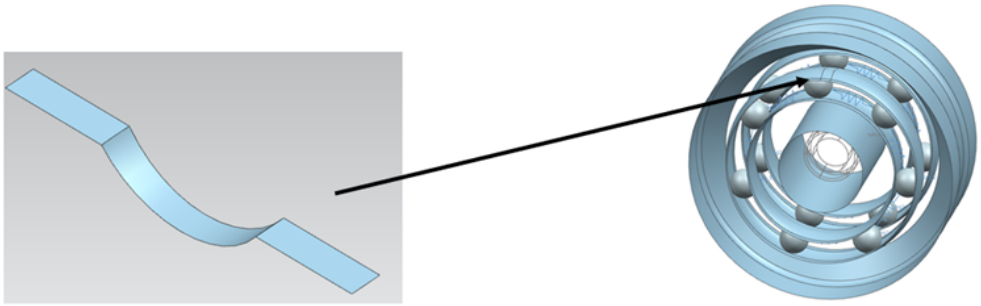


Figure 5.35: Damage surface.

With this type of modelling, it is possible to simulate both the decompression of the sphere at the entrance to the damage area, because the damage surface has a larger radius than the healthy one, and the compression on the way out of the ball from the damaged part to the healthy one.

5.3.3.8 Cart bearing model

The bearings of the cart are double row deep-groove ball bearings. For each cart there are two different types of bearings that differ only in dimensions. The dimensions of the bearings are in Tab. 5.2. The bearings have a V shape (Fig. 5.36) in order to be in contact with the guide. The contact between the external rolling surface and the rail has been modelled as an analytical contact between the revolution surface of the bearing and the extrusion surface of the roller with the following parameters:

- Restitution coefficient: 0.7
- Friction coefficient: 0.2

- Transition velocity: $10/ms$

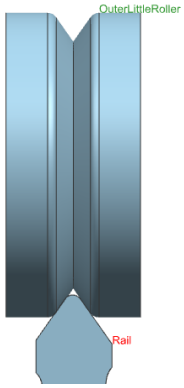


Figure 5.36: Bearing rail contact.

5.3.3.9 Cart Model

In the simulation, the geometry of the cart has been simplified with respect to the real one, but it maintains all the inertia and mass properties of the real one. The cart consists of three double row ball bearings, a frame body and a slider body (Fig. 5.37).

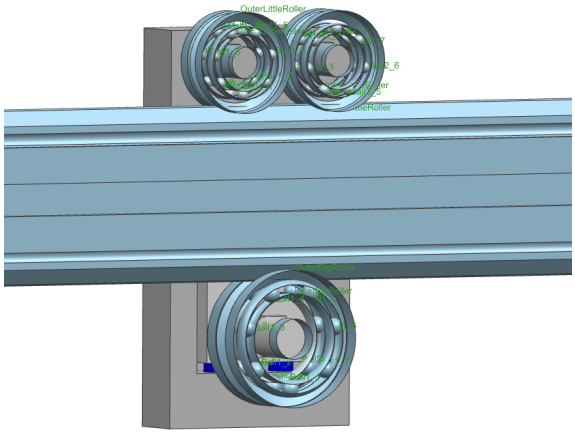


Figure 5.37: Complete cart rail model.

The top bearings are of the same type and they are smaller than the one in the bottom. Each top bearing is connected to the frame body by means of a bushing connector. The bottom bearing is connected to the slider by means of a bushing connector with the same damping values of the other two bearings. The slider body is used in order to maintain a constant preload on the three bearings when the cart is connected to the rail. As showed in Fig. 5.38, the slider is connected to the frame

body by two slider joints. Each slider joint has its direction in the center of the corresponding guide rail. For each guide rail a preload spring with a length of 2.525 and a stiffness of $20N/mm$ has been defined.

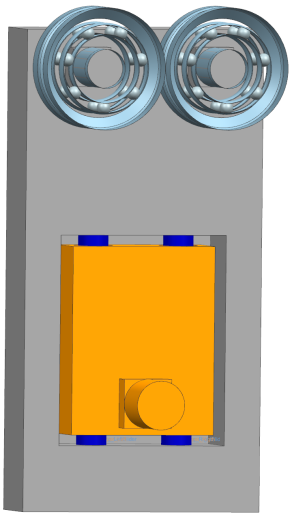


Figure 5.38: Bottom bearing housing with preload system (orange).

5.3.4 Model tuning and validation

5.3.4.1 Experimental Setup

The experimental activity has been carried out using a specific test rig made up of Beckhoff XTS system (Fig. 5.6) with GXF Hepco guidance and the cart shown in Fig. 4.8 in order to verify and validate the proposed model. The system taken into account consists of a straight rail 1.750 m long. For the length of the machine, 7 straight motors have been used. A single cart has been sensorized with an accelerometer PCB 356A02 SN 40918 placed on the top surface. The sensor has been connected to the EL3632 analog input IEPE terminal with a sampling frequency of 20 kHz.
During the test the mover goes from the left to the right and from the right to the left following a trapezoidal motion profile (Fig. 5.40). The profile consists in an acceleration phase, a constant velocity zone and a deceleration zone. The four tests have been conducted with four different constant velocity peaks of 500 mm/s, 1000 mm/s, 1500 mm/s, 2000 mm/s . Each test consists in a forty-second record of the system variables while the mover is executing the motion profile. The tests have been performed at the aforementioned velocities with two different configurations In the first configuration, all the bearings of the mover are healthy (Fig. 5.39), in the second configuration the bearing placed on the top left is damaged on the outer ring surface. The damaged bearing is the same bearing used for the test in Chapter 5.2, Fig. 5.9 shows the damage of the bearing.

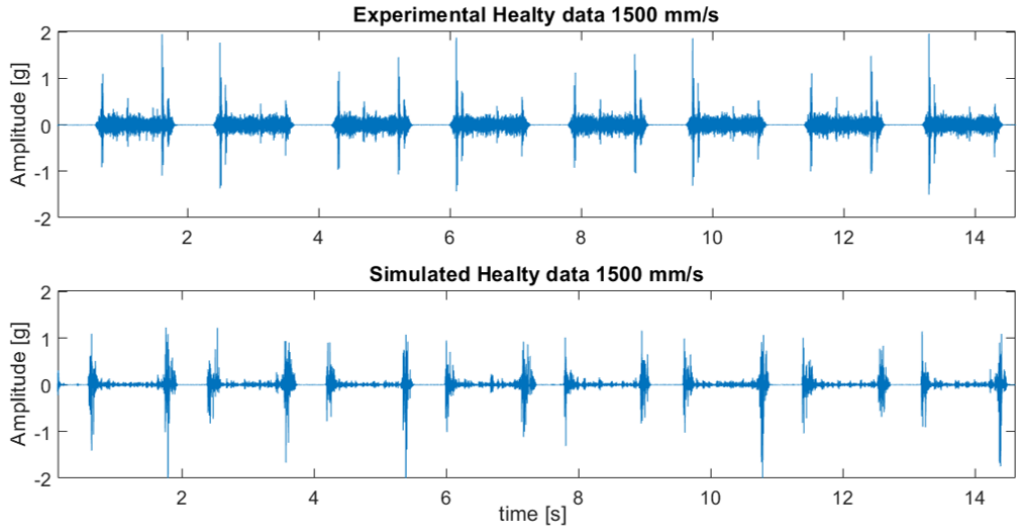


Figure 5.39: Comparison of experimental and simulated vibration response of the cart at 1500 mm/s.

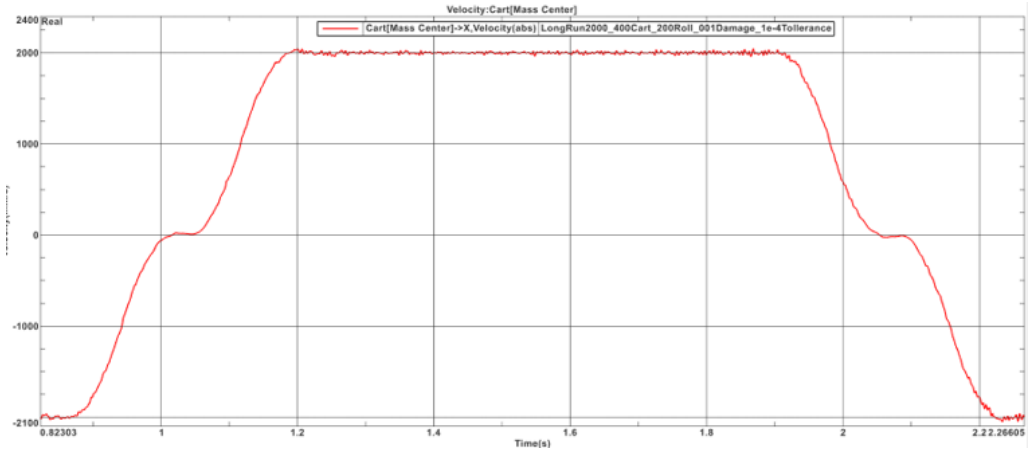


Figure 5.40: Cart Motion profile at 2000 mm/s.

5.3.4.2 Data Pre-Processing

The pre-processing of the vibration data has been executed as follows:

- The vibration data has been cut with respect to the velocity in order to have only the vibration signal in the constant velocity zones of the motion profiles.
- The cut vibration signals have been windowed with a Hanning window and connected one after the other in order to have a constant velocity vibration signal.
- In order to identify the frequency of the bearings an envelope spectrum has been computed for the constant velocity vibration signal.

The pre-processing of the data has been implemented both for the real data and for the simulated ones.

Several tests have been carried out with healthy bearings in order to tune the bushing connector damping values between the bearings and the cart. A good correlation between the simulation and the real values has been found with these parameters (x= 200 Ns/m, y=200 Ns/m, z=200 Ns/m) and (Ax=0.01 Nms/rad, Ay=0.01 Nms/rad, Az=0 Nms/rad).

5.3.5 Model validation

After the tuning of the damping parameters, even the damage surface of the model has been tuned. In this case, different radii of the damaged surface have been considered in order to have a good correlation between the simulation data and the real data. For the damage tuning, only the case of 500 mm/s has been used and a good data correlation has been defined with a radius of the damaged surface equal to 0.001 mm, higher than the theoretical one. All the other cases have been used for validation.
Fig. 5.41 and Fig. 5.42 show the envelope spectrum of the real and simulated vibration data at 1500 mm/s and 2000 mm/s.

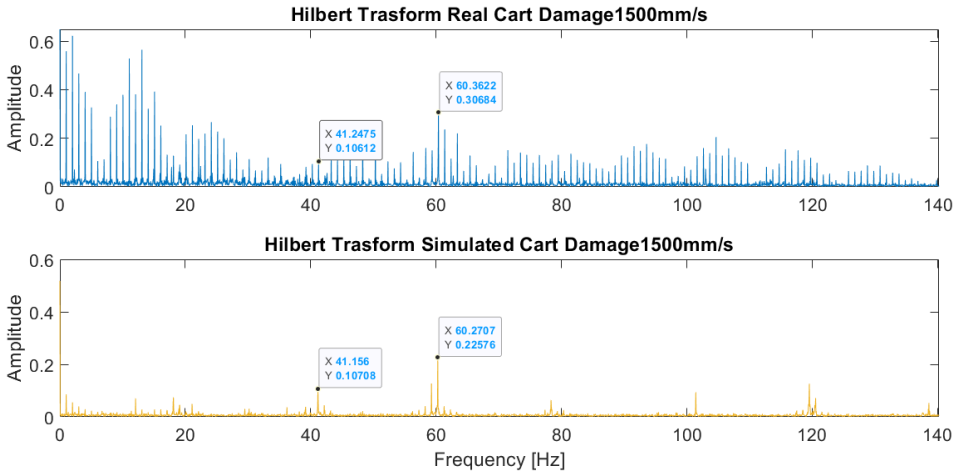


Figure 5.41: Frequency comparison 1500 mm/s.

In the experiment at 1500 mm/s, the BPFO of the damaged top bearing is visible in the experimental and real data. Even the BPFO of the bottom roller bearing is visible in both cases and they are comparable both in frequency and in amplitude.

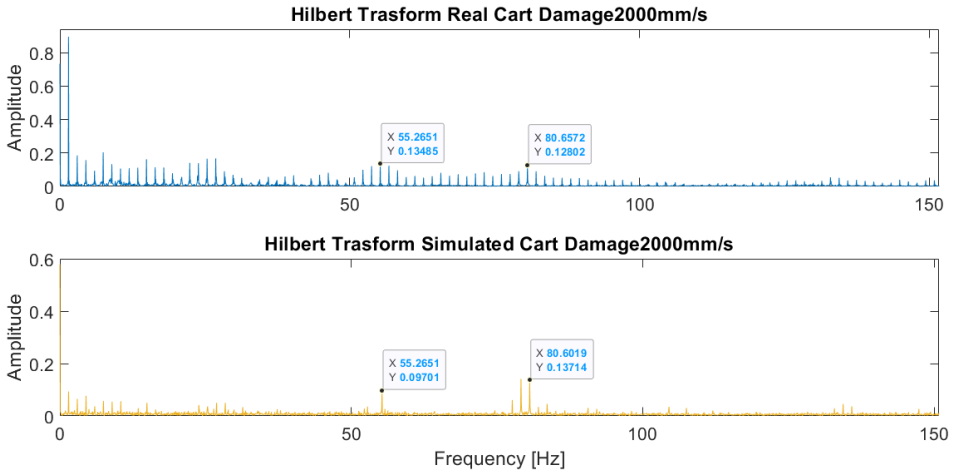


Figure 5.42: Frequency comparison 2000 mm/s.

5.3.6 Conclusions referring to multibody model

This chapter has shown the feasibility of the use of a commercial multibody software for the simulation of an Independent Cart System for condition monitoring application. The model has a good accuracy in the simulation of brinelling/pitting damages with a very high computational efficiency. With this model it is possible:

- To simulate the behaviour of healthy bearings.
- To simulate brinelling/pitting damages.

- To consider variable motion profiles.
- To choose any shape of the rail track.

The results of the simulation can be used in order to evaluate the response of the system under incipient damages of bearings. The model can be generalized by assembling several carts with different conditions of bearings and using the simulated vibrations such as training data of machine learning algorithms or testing model-based techniques for fault detection and identification.

Chapter 6

Conclusions and future work

The thesis shows the implementation of condition monitoring methods for Independent Carts System. It describes the advantages of using this technology for high speed and high flexible manufacturing applications, but, at the same time, it highlights the complexity of the use of this system in the condition monitoring field. The first part deals with the problem of the integration of this system with different vendors' controllers. Two methods (Linear Approximation and Discrete Approximation) of integration and synchronization of the motion task are explained. They are useful not only for the control of the Independent Carts System, but they also allow the record and the analysis of the work parameters of the system for the implementation of Condition Monitoring System. The Linear Approximation method allows to reduce the complexity of computation maintaining good synchronization performances, while the Discrete Approximation shows better performances but with a higher computational effort. The first algorithm is more suitable for PLCs with low computational power, while the second one for more expensive and high performance PLCs. The second part of the thesis considers a Data-driven approach for the implementation of a monitoring system for Independent Carts Technology. It consists in a preliminary Data-driven implementation since there is no previous case of bearing faults concerning this machine. For this reason, a methodology based on the use of artificial damages is proposed such as to reproduce the most similar fault behaviour of the machine. This procedure can be extended to the cases in which it is necessary to implement a condition monitoring system before the release of a machine on the market. The aim of the data-driven model developed with this procedure is to carry out a condition monitoring system that can identify the most dangerous or probable failures. At the same time, the models will be updated with all the new datasets coming from the field in order to increase the type of detectable failures and the accuracy of prediction. The procedure consists in the use of pre-processed healthy and faulty data for training random forest and support vector machine algorithms with the aim of fault detection and fault identification. A logical disjunction (logical OR) between the output predictions of the two algorithms is used in order to improve the accuracy in detecting inner/outer ring bearing damages and blocked bearings. The third part of the thesis considers a model-based approach for the development of condition monitoring system. In this case, two different methods are

used for the simulation of the system: a signal-based formulation and a multi-body formulation. Both formulations take into consideration the geometry of the bearings and of the cart, the variable motion profiles and load profiles. As regards the signal-based formulation, the vibration signal produced by the cart is represented by a series of impulses modulated by the load variation, the motion profile and a one DOF system. In this case, it is possible to simulate different types of fault bearings by including a pulse train of the same frequency as that of the fault, the impulse train is defined with respect to the position of the cart and not with respect to the time. This is due to the variable velocity of the system. The multi-body model consists in rigid bodies representing the cart, bearings and rail. In this case, the study has been focused on the modelling of the bearings in the healthy and damaged state. An analytical formulation is used for the simulation of the contact between the rolling elements and the rails of the bearings. This formulation allows to use not only the theoretical shape of the bearings but also different shapes for the simulation of damaged surfaces. The healthy bearing model has been validated by the comparison with a Bearing Element of Simcenter 3D Motion. The comparison shows the responses of the two models, which are comparable with each other with an error in the bearing displacement lower than 2%. Both the signal-based model and the multi-body model have been validated in healthy and faulty cases through an experimental campaign carried out with a cart sensorized by an accelerometer. With the simulated data coming from the two models, it is possible to reduce the complexity of the system by simulating only the most important phenomena of the system; it is also possible to use the simulated data to develop condition monitoring thresholds and to train machine learning and deep learning algorithms.

6.1 Future Work

This thesis is a preliminary study for the development of a global condition monitoring framework for Independent Carts System. Different aspects, such as control systems, data transfer, data-driven and model-based approaches have been taken into account in order to establish a fault detection system. Concerning the data-driven approach and the model- based approach, attention has been concentrated only on the most probable faults of the system that are bearing faults. In future, it will be also necessary to consider other types of faults, for example electrical failures, mechanical failures of the cart or functional failures of the machine. This study has been carried out only with data coming from experimental campaigns because the product is not yet in production. When it starts to be in the field, more data will be available and even neural network algorithms could be trained with good performances. With a larger number of healthy data, it will be also possible to train anomaly detection algorithms in order to verify every variation in the global performance of the system and to detect possible failures. All the data-driven models developed for this application will need to be retrain with the new data coming from the field. For this reason, it will be also necessary to define a policy for the automatic retrain of all the machine learning algorithms and for their release in the

field. As regards the model-based approach, the next step will be the use of the simulated healthy and faulty data as inputs for the training of the fault detection algorithms. The models can be generalized by simulating several carts moving at the same time. Furthermore, the models can be improved by the transfer path analysis of the system, which is useful for the identification of the vibration energy flow from the bearings to the cart and for the implementation of the flow itself in the models. For the multi-body model, it will be possible to define the cart as a flexible body in order to consider the frequency response of the cart during the simulation. The last step of this study will regard the carrying out of a complete Prognostic and Health Management framework. In order to achieve the aim, prognostic algorithms have to be developed, they will be useful for identifying the RUL of the different components of the system.

Appendix A

Appendix

Let S be a signal composed of K points of amplitude x_i:

1. **Mean**:Average of all values of the signal/sample

$$x_m = \frac{1}{K} * \sum_{i=1}^{K} x(i) \tag{A.1}$$

2. **Standard Deviation**: Deviation from the mean of the signal/sample.

$$x_{std} = \sqrt{\frac{\sum_{i=1}^{K}(x(i) - x_m)^2}{K-1}} \tag{A.2}$$

3. **Variance**: Square of Standard Deviation.

$$x_{var} = \frac{\sum_{i=1}^{K}(x(i) - x_m)^2}{K-1} \tag{A.3}$$

4. **Root Mean Square**: Square root of the mean of squares of a signal/sample.

$$x_{rms} = \sqrt{\sum_{i=1}^{K} \frac{x(i)^2}{K-1}} \tag{A.4}$$

5. **Maximum Amplitude**: Value of the maximum amplitude of the signal/sample.

$$x_{max} = max(x(i)) \tag{A.5}$$

6. **Minimum Amplitude**: Value of the minimum amplitude of the signal/sample.

$$x_{min} = min(x(i)) \tag{A.6}$$

7. **Peak to Peak Value**:Difference between maximum and minimun peak values.

$$x_{ppv} = x_{max} - x_{min} \tag{A.7}$$

8. **Square Root of Amplitude**: Value of the root of Amplitude.

$$x_{sra} = (\frac{1}{K}\sum_{i=1}^{K}\sqrt{|x(i)|})^2 \tag{A.8}$$

9. **Skewness**: Measure of lack of symmetry.

$$x_{skew} = \frac{\sum_{i=1}^{K}(x(i) - x_m)^3}{(K-1)x_{std}^3} \tag{A.9}$$

10. **Kurtosis**: Measure of the spikiness of the signal/sample relative to a normal distribution.

$$x_{kurt} = \frac{\sum_{i=1}^{K}(x(i) - x_m)^4}{(K-1)x_{std}^4} \tag{A.10}$$

11. **Kurtosis Factor**: It is Kurtosis value divided by the square of the mean of squares of amplitudes.

$$x_{kurtFactor} = \frac{\frac{\sum_{i=1}^{K}(x(i)-x_m)^4}{(K-1)x_{std}^4}}{(1/K\sum_{i=1}Kx(i)^2)^2} \tag{A.11}$$

12. **Clearance Factor**: Ratio of maximum amplitude value to square of mean of root of absolute values.

$$x_{clf} = \frac{x_{max}}{(1/K\sum_{i=1}^{K}\sqrt{|x(i)|})^2} \tag{A.12}$$

13. **Shape Factor**: Value of how much the shape of a signal is affected, other than shifting or scaling.

$$x_{sf} = \frac{x_{rms}}{(1/K\sum_{i=1}^{K}\sqrt{|x(i)|})} \tag{A.13}$$

14. **Impulse Factor**: Ratio of maximum amplitude value to mean of absolute values.

$$x_{if} = \frac{x_{max}}{1/K\sum_{i=1}^{K}|x(i)|)} \tag{A.14}$$

15. **Crest Factor**: Ratio of the maximum amplitude to the RMS value of the signal/sample.

$$x_{cf} = \frac{x_{max}}{xmin} \tag{A.15}$$

16. **Sum** Sum of all signal point values in a sample/signal.

$$x_{sum} = \sum_{i=1}^{K} x(i) \tag{A.16}$$

17. **Entropy**:Is a calculation of the uncertainty and randomness of a sampled signal.Given a set of probabilities, $(p1, p2, ..., pn)$, the entropy can be calculated as:

$$e(p) = -\sum_{i=1}^{K} p(z_i) log2p(z_i) \tag{A.17}$$

18. **Activity** is the variance of the signal.

$$Activity = \sigma_x^2 \tag{A.18}$$

19. **Mobility**: is the square root of the ratio of the activity of the first derivative and the activity of the vibration signal.

$$Mobility = \frac{\sigma_x^{'}}{\sigma_x} \tag{A.19}$$

where $\sigma_x^{'}$ is the standard deviation of the first derivative of the vibration signal.

20. **Complexity**: is calculated as the ratio of mobility of the first derivative and the mobility of the vibration signal.

$$Complexity = \frac{\frac{\sigma_x^{''}}{\sigma_x^{'}}}{\frac{\sigma_x^{'}}{\sigma_x}} \tag{A.20}$$

21. **Max Power Spectrum**: Value of the maximum power of the frequency spectrum.

$$x_{fmax} = max(Power(n)) \tag{A.21}$$

22. **Max Envelope**: Maximum value of the envelope of the signal/sample.

$$x_{env} = max(Env) \tag{A.22}$$

23. **Frequency Center**: Average of all values of spectrum of the signal/sample.

$$f_c = \frac{\sum_{i=1}^{K} f * S(n)}{\sum_{i=1}^{K} S(n)} \tag{A.23}$$

24. **Root Mean Square Frequency**: Square root of the mean of squares of spectrum of a signal/sample.

$$f_{rms} = \frac{\sqrt{\sum_{i=1}^{K} f^2 * S(n)}}{\sum_{i=1} KS(n)} \tag{A.24}$$

25. **Root Variance Frequency**: Deviation from the center of the frequency of the signal/sample.

$$f_{std} = \frac{\sqrt{\sum_{i=1}^{K} (f - f_c)^2 * S(n)}}{\sum_{i=1}^{K} S(n)} \tag{A.25}$$

Bibliography

[1] K. Andrew. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, pages 1483–1510, 2006.

[2] European Standard CEN (European Committee for Standardization). Maintenance terminology. page EN 13306:2001, Brussels 2001.

[3] M. Eti, S. Ogaji, and S. Probert. Maintenance schemes and their implementation for the afam thermal-power station. *Applied Energy*, 82:1255–265, 2005.

[4] C Sheut and LJ Krajewski. A decision model for corrective maintenance management. *The International Journal of Production Research*, 32(6):1365–1382, 1994.

[5] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[6] M. Cooconcelli, L. Capelli, J. Cavalaglio Camargo Molano, and D. Borghi. Development of a methodology for condition-based maintenance in a large-scale application field. *Machines*, 6(2), 2018.

[7] H. Fleischmann, J. Kohl, and J. A Modular Franke. Architecture for the design of condition monitoring processes. pages 410–415, 2016.

[8] S. Park and J. Kim, Y.and Won. Application of iot for the maintaining rolling stocks. *Qual. Innov. Prosper*, 21:71–83, 2017.

[9] Zio E. Review reliability engineering: old problems and new challenges. *Reliability Engineering and System Safety*, 94:125–41, 2009.

[10] K. Worden and Dulieu-Barton J.M. An overview of intelligent fault detection in systems and structures. *Structural Health Monitoring*, 3(1):85–98, 2004.

[11] A. Birolini. *Reliability Engineering 8th ed.* Springer Berlin/Heidelberg, 2017.

[12] O'Connor and P. Kleyner. *A Practical Reliability Engineering*. Wiley and Sons:Hoboken, 2012.

[13] A. Heng, S. Zhang, and J. Tan, A.C.C.and Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical System and Signal Processing*, 23, 2009.

[14] Z. Hameed, Y.S. Hong, Y. Cho, S. Ahn, and Song C. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Sustain. Energy Rev*, 13, 2009.

[15] E.P. Carden and P. Fanning. Vibration based condition monitoring: A review. *Struct. Health Monitoring*, 3, 2004.

[16] R. Ruiz-Gonzalez, J. Gomez-Gil, F.J. Gomez-Gil, and V. Martínez-Martínez. An svm-based classifier for estimating the state of various rotating components in agro-industrial machinery with a vibration signal acquired from a single point on the machine chassis. *Sensor*, 14, 2014.

[17] R. Potter. A new order tracking method for rotating machinery. *Sound Vibration*, 24, 1990.

[18] K. Fyfe and E. Munck. Analysis of computed order tracking. *Mechanical System and Signal Processing*, 11, 1997.

[19] K. Bossley, R. McKendrick, C. Harris, and C Mercer. Hybrid computed order tracking. *Mechanical System and Signal Processing*, 13, 1999.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer: Berlin, Germany, 2009.

[21] A. Albalate and W. Minker. Semi-supervised and unsupervised machine learning: Novel strategies. *Iste/Hermes Science Pub*, 2010.

[22] Randal R. *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications.* Wiley and Sons:Hoboken, 2010.

[23] Sankar Mahadevan and Sirish L Shah. Fault detection and diagnosis in process data using one-class support vector machines. *Journal of process control*, 19(10):1627–1639, 2009.

[24] D. Dasgupta and S. Forrest. Artificial immune systems in industrial applications. *In Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials„ Honolulu, HI, USA*, 1, 1999.

[25] S. Yin, S. Ding, X. Xie, and H Luo. A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans. Ind. Electron*, 61, 2014.

[26] Z. Ge. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemom. Intell. Lab. Syst*, 171, 2014.

[27] M. Cerrada, R.V. Sánchez, C. Li, F. Pacheco, D. Cabrera, J. Valente de Oliveira, and R. Vásquez. A review on data-driven fault severity assessment in rolling bearings. *Mech. Syst. Signal Process.*, 99, 2014.

[28] R. Liu, B. Yang, and X. Zio, E.and Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Process.*, 108, 2018.

[29] B. Samanta and K. Al-Balushi. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mech. Syst. Signal Process.*, 17, 2003.

[30] Dongdong Wei, KeSheng Wang, Stephan Heyns, and Ming J Zuo. Convolutional neural networks for fault diagnosis using rotating speed normalized vibration. pages 67–76, 2018.

[31] B. Paya, I. Esat, and M. Badi. Artificial neural network based fault diagnostics of rotating machinery using wavelet transforms as a preprocessor. *Mech. Syst. Signal Process.*, 11, 1997.

[32] A. Moosavian, H. Ahmadi, A. Tabatabaeefar, and M. Khazaee. Comparison of two classifiers; k-nearest neighbor and artificial neural network, for fault diagnosis on a main engine journal-bearing. *Shock Vib.*, 20, 2013.

[33] N. Li and C. Mechefske. Induction motor fault detection and diagnosis using artificial neural networks. *Int. J. COMADEM*, 9, 2013.

[34] J. Patel and S. Upadhyay. Comparison between artificial neural network and support vector method for a fault diagnostics in rolling element bearings. *Procedia Eng.*, 144, 2016.

[35] M. Cocconcelli, R. Rubini, R. Zimroz, and W. Bartelmus. Diagnostics of ball bearings in varying-speed motors by means of artificial neural networks. *In Proceedings of the Eight International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, Cardiff, UK*, 2, June2011.

[36] M. Cocconcelli and R. Rubini. Support vector machines for condition monitoring of bearings in a varying-speed machinery. *In Proceedings of the 8th International Conference on Condition Monitoring, Cardiff, UK*, 2011.

[37] V. Sugumaran, V. Muralidharan, and K. Ramachandran. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mech. Syst. Signal Process*, 21, 2007.

[38] V. Sugumaran, G. Sabareesh, and K. Ramachandran. Fault diagnostics of roller bearing using kernel based neighborhood score multi-class support vector machine. *Expert Syst. Appl*, 34, 2008.

[39] L.and Chen Guo and X J.and Li. Rolling bearing fault classification based on envelope spectrum and support vector machine. *J. Vib. Control*, 15, 2009.

[40] F. Di Maio, P. Baraldi, E. Zio, and R. Seraoui. Fault detection in nuclear power plants components by a combination of statistical methods. *IEEE Transactions on Reliability*, 62(4):833–845, 2013.

[41] G. Niu, Y. Zhao, M. Defoort, and M. Pecht. Fault diagnosis of locomotive electro-pneumatic brake through uncertain bond graph modeling and robust online monitoring. *Mechanical Systems and Signal Processing*, 50-51:676–691, 2015.

[42] P. Baraldi, F. Di Maio, D. Genini, and E. Zio. Comparison of data-driven reconstruction methods for fault detection. *IEEE Transactions on Reliability*, 64(3):852–860, 2015.

[43] P. Baraldi, F. Di Maio, P. Turati, and E. Zio. Robust signal reconstruction for condition monitoring of industrial components via a modified auto associative kernel regression method. *Mechanical Systems and Signal Processing*, 60:29–44, 2015.

[44] P. Baraldi, G. Bonfanti, and E. Zio. Differential evolution-based multi-objective optimization for the definition of a health indicator for fault diagnostics and prognostics. *Mechanical Systems and Signal Processing*, 102:382–400, 2018.

[45] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Comput. Chem. Eng*, 7, 2003.

[46] S. Kerst, B. Shyrokau, and E. Holweg. A semi-analytical bearing model considering outer race flexibility for model based bearing load monitoring. *Mech. Syst. Signal Process*, 104, 2018.

[47] J. Heikkinen, B. Ghalamchi, R. Viitala, J. Sopanen, J. Juhanko, A. Mikkola, and P. Kuosmanen. Vibration analysis of paper machine's asymmetric tube roll supported by spherical roller bearings. *Mech. Syst. Signal Process*, 104, 2018.

[48] H. Cao, L. Niu, S. Xi, and X. Chen. Mechanical model development of rolling bearing-rotor systems: A review. *Mech. Syst. Signal Process*, 102, 2018.

[49] V.E. Silva Souza, A. Lapouchnian, and J. Mylopoulos. System identification for adaptive software systems: A requirements engineering perspective. *In International Conference on Conceptual Modeling*, 2011.

[50] M. Shahriar, P. Borghesani, and A. Tan. Electrical signature analysis-based detection of external bearing fault in electromechanical drivetrains. *IEEE Trans. Ind. Electron*, 2017.

[51] D. Abboud, M. Elbadaoui, S. Becquerelle, and M. Lalmi. The application of the cyclic coherence for distributed planet fault detection in planetary gears. *In Proceedings of the 1st World Congress on Condition Monitoring (WCCM 2017), London, UK, 13–16 June*, 2017.

[52] Z. Feng, H. Ma, and M. Zuo. Vibration signal models for fault diagnosis of planet bearings. *J. Sound Vib*, 370, 2016.

[53] S. Delvecchio, G. D'Elia, and G. Dalpiaz. On the use of cyclostationary indicators in ic engine quality control by cold tests. *Mech. Syst. Signal Process*, 60, 2015.

[54] F. Cong, J. Chen, G. Dong, and M Pecht. Vibration model of rolling element bearings in a rotor-bearing system for fault diagnosis. *J. Sound Vib.*, 332, 2013.

[55] T. Tóth, L.; Tóth. Construction of a realistic signal model of transients for a ball bearing with inner race fault. *Acta Polytech. Hung.*, 10, 2013.

[56] J. Stack, T. Habetler, and R. Harley. Fault-signature modeling and detection of inner-race bearing faults. *IEEE Trans. Ind. Appl.*, 42, 2006.

[57] S. Ericsson, N. Grip, E. Johansson, L.E. Persson, R. Sjöberg, and J.O. Strömberg. Towards automatic detection of local bearing defects in rotating machines. *Mech. Syst. Signal Process.*, 19, 2005.

[58] BR. https://www.br-automation.com/en/products/versatile-transport-systems/acopostrak/.

[59] Rockwell Automation. , itrak the intelligent track system increase machine flexibility and throughput to enhance overall productivity, https://tinyurl.com/ydh9nade.

[60] Beckhoff Automation. , xts. the extended transport system,https://tinyurl.com/y7vzhujy.

[61] Siemens. Multi carrier system, https://new.siemens.com/global/en/markets/machinebuildin carrier-system.html.

[62] R. Anderl. Industrie 4.0 - advanced engineering of smart products and smart production. 2014.

[63] H Kopetz. *Real-time Systems, Design Principles for Distributed Embedded Application*. Kluwer Accademic Pubblication, 2002.

[64] P. Antsaklis and J. Baillieul. *Special issue on technology of networked control systems*. IEEE, 2007.

[65] E. R. Alphonsus and M. O. Abdullah. A review on the applications of programmable logic controllers (PLCs). *Renewable and Sustainable Energy Reviews*, 60:1185–1205, 2016.

[66] J. P. Thomesse. Fieldbus technology in industrial automation. *Proceedings of the IEEE*, 93(6):1073–1101, 2005.

[67] S. Vitturi, L. Peretti, L. Seno, M. Zigliotto, and C. Zunino. Real-time ethernet networks for motion control. *Computer Standards and Interfaces*, 33:465–476, 2011.

[68] J. D. Decotignie. Analysis and design of integrated control for multi-axis motion systems. *Proceedings of the IEEE*, 93(6):1102–1117, 2005.

[69] IEEE-8023.3. *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.* standard by IEEE, 2000.

[70] ODVA. *Technology Overview Series: ControlNet.* ODVA.org, 2015.

[71] K. Bender. *PROFIBUS, the Fieldbus for Industrial Automation.* Prentice-Hall, 1993.

[72] M. Knezic, B. Dokic, and Z. Ivanovic. Topology aspects in etherCAT networks. pages T1–1–T1–6, 2010.

[73] C. Dripke and A Verl. Challenges in distributed interpolation with multi-components systems in future-oriented manufacturing units. In *Proceedings of the 47th International Conference on Computers and Industrial Engineering*, pages 215–222, 2017.

[74] UA specification, 2008.

[75] OPC UA part 1 - concepts 1.00 specification, 2006.

[76] IEEE std 802.1qbu, 2016.

[77] IEEE std 802.1qbv, 2015.

[78] IEEE std 802.1qca, 2015.

[79] M. Schleipen. OPC UA supporting the automated engineering of production monitoring and control systems. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 640–647, 2008.

[80] M. Wang, H. Luo, M. Li, J. Dong, R. Mao, and T. Zhao. The application of OPC UA technology in motion control system. pages 93–95, 2014.

[81] M. Gutierrez and R. Dobrin. Synchronization quality of IEEE 802.1AS in large-scale industrial automation networks. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 273–282, 2017.

[82] National Instruments, Designing distributed TSN ethernet-based measurement systems, 2018.

[83] F. F. Perez-Pinal, C. Nunez, R. Alvarez, and I. Cervantes. Comparison of multi-motor synchronization techniques. pages 1670–1675, 2004.

[84] Y. Xue and J. Wang. Design of an ethernet/IP-based two-axis servo system. 37(15):403–406, 2004.

[85] D. Sun, X. Shao, and G. Feng. A model-free cross-coupled control for position synchronization of multi-axis motions: Theory and experiments. 38(1):1–6, 2005.

[86] S. K. Jeong and S. S. You. Precise position synchronous control of multi-axis servo system. *Mechatronics*, 18:129–140, 2008.

[87] S. S. Yeh and P. L. Hsu. Analysis and design of integrated control for multi-axis motion systems. *IEEE Transactions on Control Systems Technology*, 11(3):375–382, May 2003.

[88] Y. Xiao and K. Y. Zhu. Optimal synchronization control of high-precision motion systems. *IEEE Transactions on Industrial Electronics*, 53(4):1160–1169, August 2006.

[89] D. Kolberg and D. Zuhlke. Lean automation enabled by Industry 4.0 technologies. 48(3):1870–1875, 2015.

[90] H. Li, F. Zhang, J. Zhang, N. Zhang, L. Wang, and X. Yang. Research and realization of a spinning machine control and monitoring system by industrial ethernet communication between IPC and OMRON PLC. pages 1901–1906, 2016.

[91] X. Xu, G. Y. Gu, Z. Xiong, X. Sheng, and X. Zhu. Development of a decentralized multi-axis synchronous control approach for real-time networks. *ISA Transactions*, 68:116–126, 2017.

[92] I. Furstner and L. Gogolak. Synchronizing the motion of multiple electric motors – new possibilities for smart motion control. pages 105–110, 2016.

[93] C. Pang, J. Yan, and S. Jennings. Distributed IEC 61499 material handling control based on time synchronization with IEEE 1588. In *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, pages 126–131, 2011.

[94] S. He, L. Huang, J. Shen, G. Gao, G. Wang, X. Chen, and L. Zhu. Time synchronization network for EAST poloidal field power supply control system based on IEEE 1588. *IEEE Transactions on Plasma Science*, pages 1–5, 2018.

[95] IEEE-1588-2008. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.* standard by IEEE, 2008.

[96] J. C. Eidson. *Measurement, Control, and Communication Using IEEE 1588.* Springer, 2006.

[97] V. Shiffer. The CIP family of fieldbus protocols and its newest member—ethernet/IP. pages 377–384, 2001.

[98] L. Breiman. Random forest. *Machine LEarning*, 45:5–32, 2001.

[99] towardsdatasciences. Random forest. 2018. [Online; accessed 10-August-2019].

[100] Wikipedia contributors. Support vector machine — Wikipedia, the free encyclopedia. 2018. [Online; accessed 29-August-2018].

[101] N. Cristianini and J Shawe-Taylor. An introduction to support vector machines: and other kernel-based learning methods. *Cambridge University Press New York*, 2000.

[102] M. Cerrada, R. Sanchez, C. Li, F. Pacheco, D. Cabrera, J. de Oliveira, and R. Vasquez. On the algorithmic implementation of multiclass kernel-based vector machine. *Journal of Machine Learning Research*, 2:265–292, 2001.

[103] B. Scholkopf, C. Burges, and A Smola. Advances in kernel methods: Support vector learning. *MIT Press*, 1999.

[104] A. Widodo and B.S Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mech. Syst. Signal Process*, 21, 2007.

[105] W. Caesarendra and T. Tjahjowidodo. 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, 2017.

[106] deeplearning accademy. Performance comparison among neural network and machine learning. 2018. [Online; accessed 10-August-2019].

[107] Cameron Sobie, Carina Freitas, and Mike Nicolai. Simulation-driven machine learning: Bearing fault classification. *Mechanical Systems and Signal Processing*, 99:403–419, 2018.

[108] Idriss El-Thalji and Erkki Jantunen. A summary of fault modelling and predictive health monitoring of rolling element bearings. *Mechanical systems and signal processing*, 60:252–272, 2015.

[109] PD McFadden and JD Smith. Model for the vibration produced by a single point defect in a rolling element bearing. *Journal of sound and vibration*, 96(1):69–82, 1984.

[110] PD McFadden and JD Smith. The vibration produced by multiple point defects in a rolling element bearing. *Journal of sound and vibration*, 98(2):263–273, 1985.

[111] Nader Sawalhi and RB Randall. Simulating gear and bearing interactions in the presence of faults: Part i. the combined gear bearing dynamic model and the simulation of localised bearing faults. *Mechanical Systems and Signal Processing*, 22(8):1924–1951, 2008.

[112] Jose L Gomez, Adeline Bourdon, Hugo André, and Didier Rémond. Modelling deep groove ball bearing localized defects inducing instantaneous angular speed variations. *Tribology International*, 98:270–281, 2016.

[113] Arnaz S Malhi. Finite element modeling of vibrations caused by a defect in the outer ring of a ball bearing. *Project Report finite element method and applications, MIE*, 605:1–6, 2002.

[114] Zeki Kiral and Hira Karagülle. Simulation and analysis of vibration signals generated by rolling element bearing with defects. *Tribology International*, 36(9):667–678, 2003.

[115] Gianluca D'Elia, Marco Cocconcelli, and Emiliano Mucchi. An algorithm for the simulation of faulted bearings in non-stationary conditions. *Meccanica*, 53(4-5):1147–1166, 2018.

[116] Marco Cocconcelli, Jacopo Cavalaglio Camargo Molano, Riccardo Rubini, Luca Capelli, and Davide Borghi. Bearing fault model for an independent cart conveyor. In *International Conference on Condition Monitoring of Machinery in Non-Stationary Operation*, pages 211–220. Springer, 2018.

[117] Radoslav Tomović. Calculation of the boundary values of rolling bearing deflection in relation to the number of active rolling elements. *Mechanism and machine theory*, 47:74–88, 2012.

[118] Jacopo Cavalaglio Camargo Molano, Stefano Rossi, Marco Cocconcelli, and Riccardo Rubini. Dynamic model of an independent carts system. In *Advances in Italian Mechanism Science*, pages 379–387. Springer, 2017.

[119] Jérôme Antoni. Cyclic spectral analysis of rolling-element bearing signals: Facts and fictions. *Journal of Sound and vibration*, 304(3-5):497–529, 2007.

[120] Robert Bond Randall. *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.

[121] H Sjovall. The load distribution within ball and roller bearings under given external radial and axial load. *Tekniks Tidskrift Mek.*, (9), 1933.

[122] Yi Guo and Robert G Parker. Stiffness matrix calculation of rolling element bearings using a finite element/contact mechanics model. *Mechanism and machine theory*, 51:32–45, 2012.

[123] T. A. Harris and M. N. Kotzalas. Essential concepts of bearing technology. *CRC press*, 2006.

[124] T. A. Harris and M. N. Kotzalas. Advanced concepts of bearing technology: rolling bearing analysis. *CRC press*, 2006.

[125] L. Xiangyang and Wanqiang C. Rolling bearing fault diagnosis based on physical model and one-class support vector machine. *ISRN Mechanical Engineering*, 2006.

[126] A. Sopanen, J. Mikkola. Dynamic model of a deep-groove ball bearing including localized and distributed defects. part 1:theory. proceedings of the institution of mechanical engineers. *Journal of Multi-body Dynamics*, 217(3), 2003.

[127] A. Sopanen, J. Mikkola. Dynamic model of a deep-groove ball bearing including localized and distributed defects. part 2: Implementation and results theory. *Journal of Multi-body Dynamics*, 217(3), 2003.

[128] U. Leturiondo, O. Salgado, and D. Galar. Multi-body modelling of rolling element bearings and performance evaluation with localised damage. *Maintenance and reliability*, 18(4), 2016.

[129] L. Scurria, Cavalaglio Camargo Molano J., P. Jiranek, T. Tamarozzi, and D. Fauconnier. Rolling element bearings – advanced modeling for multibody simulations. *WCX SAE World Congress Experience, In Press*, 2002.

[130] S. jain and Hunt H. A dynamic model to predict the occurrence of skidding in wind-turbine bearings. *Journal of Physics: Conference Series*, 305(1), 2011.

[131] Siemens Digital Industry Software. Simcenter 3d motion analytical contact formulation. *https://docs.plm.automation.siemens.com.*

[132] W. C. Young and Budynas R. G. Roark's formula for stress and strain. 2002.