

POLITECNICO DI TORINO

Department of Electronics and Telecommunications

Master's Degree in Mechatronic Engineering



Master's Degree Thesis

**End-effector tools wear prediction:
Analysis and modelling of a CNC
machine and new formulation of the
friction coefficient**

Candidate:

Emanuele PANSICA

Mentor:

Prof. Alessandro RIZZO

Supervisor:

Ing. Giovanni GUIDA - Brain Technologies

April 2022

Abstract

In recent years, we have been witnessing a technological transformation of industry, the so-called Industry 4.0. In this context, the State of Health assessment (SOH) of industrial machines is a very important topic because by having information about the SOH of the machines, it is possible to operate preventive maintenance to avoid damages that may occur and, consequently, causing waste of money and time. Implementing SOH, companies could have more efficient production systems and reduce costs related to maintenance and machine downtime.

The MOREPRO Project, owned by Brain Technologies, aims to develop a system capable of estimating the SOH of the end effector tools of a CNC machine in real-time. In particular, their main purpose is to develop a technique that is first able to estimate the friction coefficient to which the tool is subjected during metal cutting operations, and then extract from this parameter the information necessary to estimate the SOH of the tool at the end of the stroke. This thesis work aims to develop a simil FEM (Finite Element Method) analysis using the Mathworks development environment to obtain a model that can describe the interaction of the tool with the workpiece. Furthermore, from this model, artificial data can be obtained and used to develop estimation methodologies.

To achieve this, firstly, I created the model of the CNC machine and then developed the trajectories that the end effector follows in the operational space. Thanks to the identification of the points of interaction between the end effector and the workpiece, mainly developed by my colleague, who provided me with necessary data for doing final simulations, I developed a dynamic formulation of the metal cutting operation to obtain an estimate of the value of the friction coefficient. The main activities carried out in this thesis project can be divided as follows:

1. **Creation of a CNC machine model:** the CNC machine analysed can be modelled as a robot consisting of two parts. The first part aims to move and position the cutting tool while the second part aims to position and orientate the workpiece. The models of the two parts were created on MATLAB and SIMULINK using the knowledge acquired in the Robotics course. These models are necessary to obtain the 3D coordinates of the tool and the workpiece.
2. **Creation and planning of trajectories in the operational space:** to simulate the movement made by a CNC machine, it was necessary to give as input a trajectory to be followed by our model. However, for our objective, we were interested in the movements of the end-effector tool; thus, trajectories were developed in operational space. A segment in the space trajectory and a circumference in the space trajectory were developed.

3. **Dynamic model of a CNC machine and estimation of the coefficient of friction:** The cutting process, nowadays, remains a very difficult process to dynamically model. However, starting from shear theory and empirical formulation, it has been possible to estimate the forces that occur during the shear process, especially normal forces and shear force. After that, having obtained these forces, it was possible to advance a formulation relating to the coefficient of friction which depends on shear parameters, the type of material used in machining and geometry.

Acknowledgements

Prima di tutto, ringrazio il Prof. Alessandro Rizzo e l'Ing. Giovanni Guida per i suggerimenti, le preziose indicazioni e il sostegno che mi hanno dimostrato in questi mesi.

Un ringraziamento speciale va alla mia famiglia che pian piano sta diventando sempre più grande. Grazie a loro supporto e incoraggiamento, ho potuto terminare i miei studi. Inoltre sono stati sempre presenti come una luce inesauribile nel buio.

Ringrazio i miei amici di sempre di giù per il loro supporto e per essere stati sempre presente nei miei momenti sia belli che brutti.

Ringrazio i miei amici di Torino, con i quali ho condiviso gioie, ansie e preoccupazioni durante questi ultimi anni.

Ringrazio il mio collega Dario Caniglia, con il quale ho lavorato a questo progetto di tesi. Grazie al suo supporto e al suo incoraggiamento, ha reso il lavoro di questo progetto più piacevole.

Infine, ringrazio la mia collega Alessia che è stata sempre presente in questi due ultimi anni della mia laurea magistrale

Table of Contents

| | |
|--|------|
| List of Tables | VII |
| List of Figures | VIII |
| Acronyms | XII |
| 1 Introduction | 1 |
| 1.1 Wear estimation | 1 |
| 1.2 State of Health of a CNC machine | 2 |
| 1.3 MOREPRO project | 3 |
| 1.4 Partnership | 5 |
| 1.5 Thesis outline | 6 |
| 2 State of the Art | 9 |
| 2.1 State of Health estimation methods | 10 |
| 2.1.1 Model-based fault detection | 11 |
| 2.1.2 State observer | 12 |
| 2.1.3 Data-driven fault prognosis | 13 |
| 2.1.4 Vibration monitoring | 16 |
| 2.1.5 Moving horizon estimation | 17 |
| 2.1.6 Mixed/Hybrid algorithms | 18 |
| 2.2 Results from past MOREPRO team | 18 |
| 2.2.1 Cutting force analysis | 21 |
| 2.2.2 Thermal analysis | 22 |
| 2.3 Finite Element Method | 23 |
| 3 CNC machine modelling | 25 |
| 3.1 CNC machine | 25 |
| 3.2 Direct Kinematics | 28 |
| 3.2.1 Denavit-Hartenberg convention | 28 |
| 3.2.2 Direct Kinematics of the two part of the CNC machine . . . | 31 |

| | | |
|----------|---|------------|
| 3.3 | Differential Kinematics | 38 |
| 3.4 | Inverse Kinematic Algorithm | 42 |
| 4 | Trajectory Planning | 45 |
| 4.1 | Trajectory | 45 |
| 4.1.1 | Operational space trajectory | 47 |
| 4.1.2 | Timing law with trapezoidal velocity profile | 49 |
| 4.2 | Rectilinear trajectory | 51 |
| 4.2.1 | Rectilinear trajectory planning in 2-D | 52 |
| 4.2.2 | Rectilinear trajectory planning in 3-D | 54 |
| 4.3 | Circular trajectory | 57 |
| 4.3.1 | Circular trajectory planning in 2-D | 60 |
| 4.3.2 | Circular trajectory planning in 3-D | 63 |
| 4.4 | Test of the CNC machine model | 66 |
| 5 | Dynamic model of shear process and new formulation of friction coefficient | 70 |
| 5.1 | Shear theory | 70 |
| 5.1.1 | Generality | 71 |
| 5.1.2 | The orthogonal cut | 72 |
| 5.1.3 | Pijspanen Model | 74 |
| 5.1.4 | Kinematic of the shear | 75 |
| 5.1.5 | Cutting forces (Ernst-Merchant) | 76 |
| 5.1.6 | Valuation of the cut force | 80 |
| 5.2 | Dynamic analysis of the shear process | 81 |
| 5.3 | New mathematical model of the friction coefficient β | 84 |
| 5.4 | Design of the systems control | 87 |
| 5.5 | Implentation of the dynamic part on Matlab and Simulink and final test | 89 |
| 6 | Conclusions | 97 |
| 6.1 | Future works | 98 |
| | Bibliography | 100 |

List of Tables

| | | |
|-----|--|----|
| 3.1 | Denavit-Hartenberg parameters of CNC machine first part | 31 |
| 3.2 | Denavit-Hartenberg parameters of CNC machine second part | 34 |
| 3.3 | Vertices coordinates with respect to reference frame $x_3 - y_3 - z_3$. . . | 36 |
| 4.1 | Chosen parameters for planning a 2-D rectilinear trajectory | 52 |
| 4.2 | Parameters employed in a 3-D rectilinear trajectory planning | 55 |
| 4.3 | Parameters employed in a 2-D circular trajectory planning | 60 |
| 4.4 | Parameters employed in a 3-D circular trajectory planning | 63 |
| 5.1 | Typical values of the hardness in Brinell for different materials . . . | 81 |

List of Figures

| | | |
|------|---|----|
| 1.1 | General structure of MOREPRO system | 4 |
| 1.2 | Implementation of the system | 5 |
| 2.1 | Forms of maintenance according to the standard EN 13306 (2001) . | 10 |
| 2.2 | Generic structure of Kalman Filter | 12 |
| 2.3 | Artificial neural network's structure | 14 |
| 2.4 | Components of a fuzzy-logic system | 15 |
| 2.5 | General structure of discrete-time hidden Markov model | 16 |
| 2.6 | P-F curve | 17 |
| 2.7 | Simplified milling machine model | 19 |
| 2.8 | Simple model of DC motor | 20 |
| 2.9 | Cutting force decomposition in the milling process | 22 |
| 2.10 | Temperature distribution in the contact area | 23 |
| 3.1 | Example of CNC machine structure | 26 |
| 3.2 | The four common machining operations: (a) turning, (b) drilling, (c) peripheral milling and (d) surface grinding | 27 |
| 3.3 | Structure of the analysed CNC machine | 28 |
| 3.4 | The two part of the CNC machine | 29 |
| 3.5 | Example of a Denavit-Hartenberg application | 30 |
| 3.6 | Model of the CNC machine part 1 drawn in Solidwork | 32 |
| 3.7 | Model of the CNC machine part 2 | 34 |
| 3.8 | rotation around axis y_3 (blue) of the work surface in the original pose (red) | 36 |
| 3.9 | Work surface translation along x_3 (blue) | 37 |
| 3.10 | Work surface rotation around axis z_3 (blue) | 37 |
| 3.11 | Work surface after a translation and rotation around axis z_3 | 38 |
| 3.12 | Block scheme of the inverse kinematics algorithm | 44 |
| 3.13 | Block scheme of the inverse kinematics algorithm on MATLAB . . . | 44 |
| 4.1 | Schematic representation of trajectory planning | 46 |

| | | |
|------|---|----|
| 4.2 | Parametric representation of a path in the space | 48 |
| 4.3 | Position, velocity and acceleration of the timing law with trapezoidal velocity profile | 50 |
| 4.4 | 2D rectilinear trajectory planning: time evolution of the x coordinate and velocity along the x axis. | 53 |
| 4.5 | 2D rectilinear trajectory planning: time evolution of the z coordinate and velocity along the z axis. | 53 |
| 4.6 | 2D Rectilinear planned path and time evolution of the velocity along the direction of motion. | 54 |
| 4.7 | 3D rectilinear trajectory planning: time evolution of the x coordinate and velocity along the x axis. | 55 |
| 4.8 | 3D rectilinear trajectory planning: time evolution of the y coordinate and velocity along the y axis. | 56 |
| 4.9 | 3D rectilinear trajectory planning: time evolution of the z coordinate and velocity along the z axis. | 56 |
| 4.10 | 3D Rectilinear planned path and time evolution of the velocity along the direction of motion. | 57 |
| 4.11 | Parametric representation of a circumference in the space | 58 |
| 4.12 | 2D circular trajectory planning: time evolution of the x coordinate and velocity along the x axis. | 61 |
| 4.13 | 2D circular trajectory planning: time evolution of the x coordinate and velocity along the x axis | 61 |
| 4.14 | 2D circular planned path and time evolution of the velocity along the direction of motion | 62 |
| 4.15 | 3D circular trajectory planning: time evolution of the x coordinate and velocity along the x axis. | 64 |
| 4.16 | 3D circular trajectory planning: time evolution of the y coordinate and velocity along the y axis. | 64 |
| 4.17 | 3D circular trajectory planning: time evolution of the z coordinate and velocity along the z axis. | 65 |
| 4.18 | 3D circular planned path and time evolution of the velocity along the direction of motion. | 65 |
| 4.19 | Time evolution of the desired and the effective x coordinate | 66 |
| 4.20 | Time evolution of the desired and the effective z coordinate | 67 |
| 4.21 | Time evolution of the desired and the effective velocity along x-axis | 67 |
| 4.22 | Time evolution of the desired and the effective velocity along z-axis | 68 |
| 4.23 | Time evolution of the desired and the effective velocity along the direction of motion | 68 |
| 4.24 | Time evolution of the position and velocity errors | 69 |
| 5.1 | Representation and aspect of the shear process | 71 |

| | | |
|------|--|----|
| 5.2 | Case of orthogonal cut | 73 |
| 5.3 | Kinematic analysis | 75 |
| 5.4 | Dynamic equilibrium of chip section | 77 |
| 5.5 | Cerchio di Merchant , scomposizione delle forze di taglio | 79 |
| 5.6 | Dynamic equilibrium | 82 |
| 5.7 | Chip section exported by the tool | 83 |
| 5.8 | Scheme of the feedback control system | 87 |
| 5.9 | Simulink block function of the dynamic part to compute the friction coefficient | 89 |
| 5.10 | Plot of the results of the 1st test | 90 |
| 5.11 | Plot of the results of the 2nd test | 91 |
| 5.12 | Plot of the results of the 3rd test | 91 |
| 5.13 | Plot of the results of the 4th test | 92 |
| 5.14 | Plot of the results of the 5th test | 92 |
| 5.15 | 1st simulation of the model with $w=200$ rad/s | 93 |
| 5.16 | 2nd simulation of the model with $w=300$ rad/s | 94 |
| 5.17 | 3rd simulation of the model with $w=400$ rad/s | 95 |

Acronyms

FEM

Finite Element Method

SOH

State of Health

CNC

Computerized Numerical Control

KF

Kalman filter

EKF

Extended Kalman filter

DH

Denavit-Hartenberg

Chapter 1

Introduction

Recently, there has been exponential development in the field of technology. The companies have integrated some of the new technologies into the industrial automation field to increase the efficiency of the plant's production system in terms of productivity and product quality, improve working conditions and create new business models. Hence, the concept of Industry 4.0 indicates precisely this attitude of today's industrial automation to incorporate new technologies. The most relevant features of Industry 4.0 are:

- *Interconnection*: Devices capable of communicating with each other via the Internet of Things
- *Transparency of information*: Detailed information, which operators can obtain, on various aspects of the manufacturing process. Then, this information can be relevant to improve the process;
- *Technical assistance*: Systems can assist the operator with unsafe tasks or in solving problems;
- *Decentralized decision*: The systems that make up the entire apparatus can make decisions and perform their tasks autonomously.

The MOREPRO project meets precisely the framework of "Industry 4.0".

1.1 Wear estimation

In the context of Industry 4.0, the problem related to the wear estimation in real-time is very relevant. Indeed, knowing the state of health (SOH) of the production machine, or its estimation, it is possible to intervene before a failure occurs through predictive maintenance avoiding a loss of time and money. In this way, it is possible

to increase the efficiency of production and safety and decrease the loss of funds related to failures that may occur in the machines of the production system. For these reasons, in recent years, scientific research has focused on the real-time estimation of the state of health (SOH) of production machinery and predictive maintenance techniques.

Predictive maintenance is a topic that has been widely discussed. Its first form was a visual inspection. Afterwards, thanks to scientific and technological progress, it has evolved into automated methods that can employ various techniques such as neural networks, fuzzy logic and empirical and physical modelling based on data. Another factor that has made this evolution possible is the use of sensors that can be integrated into machine systems providing some relevant parameters for predicting the state of Health. [1]. The main tasks of these predictive maintenance techniques are:

- Assessment of the health status of the different machine components;
- Calculation of the solution to prevent a failure.

The importance assumed by predictive maintenance techniques, which use predictive algorithms, derives from the fact that these methods allow the identification of a potentially critical situation before it occurs and, therefore, to intervene promptly with repairs or replacements of machine components. In high-precision machining, for example, when an anomaly occurs during the machining process, in this case, non-conforming parts would be produced, causing economic damage to the company, due to a preventable failure that can be avoided with a maintenance operation done at the right time. Unfortunately, SOH estimation is very complex and often very expensive. For these reasons, these topics have become relevant in the research environment.

1.2 State of Health of a CNC machine

Nowadays, CNC (Computer Numerical Control) machines are used widely in machine tool processes, i.e. chip removal, cutting and metal finishing processes. According to the programmed instructions, the CNC machine process the raw material to obtain the final component. However, often for achieving the desired shape of the last piece, is very complex and can not be obtained through manual control of the machine. Therefore, it is used a controlled machine by a computerized system that also ensures greater precision and efficiency than a manual control of the machine. The machining process consists in removing the chip from the workpiece, that is, removing part of the workpiece material, using a cutting tool that considers the *CNC* machine as a robot is the end effector. This component is essential because it is in contact with the workpiece and its excessive wear or

damage could cause non-compliant parts or a machine failure causing both time and money loss. Therefore, the SOH estimation of the end-effectors is very significant. Unfortunately, the modelling of the interaction between tool and workpiece is very complex since there are many variables to consider. The most important ones are:

- Configuration of the robot;
- Environmental parameters;
- State of Health of the tool.

However, it is not possible to estimate the wear condition of the tool. Sensors are helpful to measure some important parameters that occur during the process: temperature, forces, voltages, pressure. After these values are known, it is possible to estimate the value of the SOH. This estimation can often be inaccurate because the sensors are usually installed in incorrect locations that can interfere with the work process. For example, a sensor that measures the temperature, can not be placed in the end effector and, for this reason, the value provided by the sensor on the temperature that occurs during the process is not accurate and is subject to errors. In addition, the parameters returned by the sensors may be insufficient to estimate the actual state of the tool, that is, the end effector. Nowadays, there are more widely used methods for estimating the state of health of the (SOH) of the tool and they are machine learning techniques combined with the simulation of a digital twin. These methods are efficient but with the disadvantage of having high calculation requirements due to the high number of parameters presented in the cutting process. The most relevant are the mechanical ones because they can give information on the state of tension to which the terminal tool is affected. The main mechanical parameters are:

- Coefficient of friction;
- Temperature;
- Chip load.

These parameters are greatly affected by the contact forces that develop during the machining process.

1.3 MOREPRO project

The main objective of MOREPRO project is to estimate the SOH of the tool of a CNC machine. The project is structured around a logical architecture distributed over three levels:

- To detect the *SOH* of critical machine components. First data is acquired through embedded sensor and then processed using machine learning and data mining techniques to obtain information on the *SOH* of the machine.
- To monitor the wear of the spindle-mounted tool, i.e. the end effector, using twin digital instruments that provide real-time environmental signals. Then, these parameters are compared with certain quantities estimated in a specific simulation environment to extrapolate some helpful information about the wear of the end effector.
- To predict over time how the *SOH* of the machine evolves to carry out preventive maintenance. Then, through preventive maintenance, it is possible to avoid the production process can be interrupted and have losses in terms of time and money.

The architecture can be divided into two levels. The first level consists of edge computing that deals first with information acquisition from the system, then the processing of data collected to provide real-time supervision of the system *SOH*. This level is very significant because it allows timely intervention to avoid the occurrence of a fault if potential danger is detected. Then, the same collected data are used in the second server level, mainly located on the cloud. In the second level, there is the creation of a database, the implementation of a digital twin and the derivation of the exact parameters to re-set the logical processing of each edge device. In this way, it is possible to ensure that the performance of the devices is optimal. In addition, according to the current system situation, it is possible to configure their behaviour. Figure 1.1 shows a general scheme of the MOREPRO project.

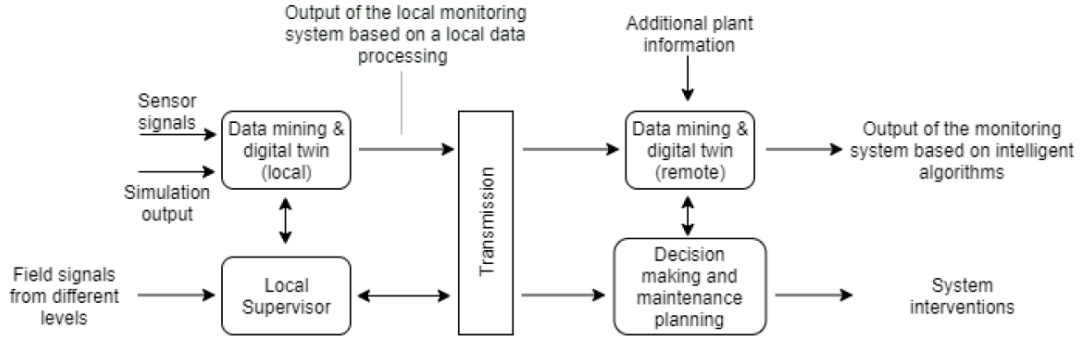


Figure 1.1: General structure of MOREPRO system

Referring to image 1.1, inside the edge device will be implemented *the data mining, digital twin and monitoring algorithms*. In this way, it is possible to have a two-way exchange of information with both the plant and the supervisor. Thanks to the use of sensors that acquire signals, it will process the real signal coming from the field together with the output of a simulation to calculate the SOH of the element under control. Thanks to the information provided by the server on the cloud, the supervisor will be able to update and refine the algorithm of the edge device itself to reconfigure and support the planning decisions. The following figure 1.2 shows a possible physical implementation of the system.

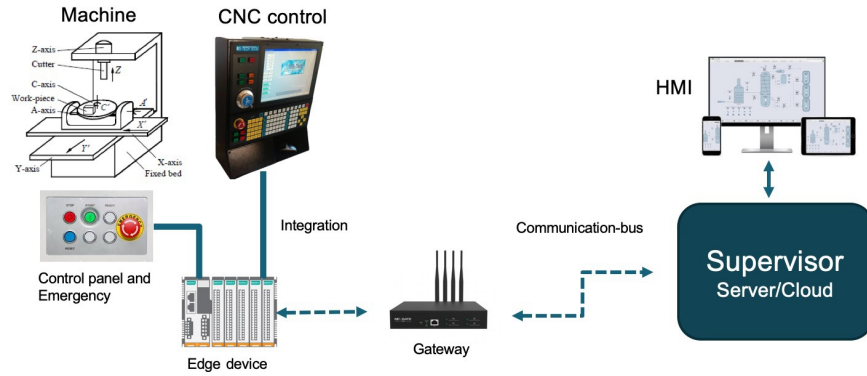


Figure 1.2: Implementation of the system

It is possible to say that often the wear estimation techniques are relied on deep learning calculations only and this operation, given the complexity of processing and a large amount of data, is carried out in the cloud. However, the MOREPRO project's objective, by incorporating edge devices into its architecture, is to process this data locally, i.e. as close as possible to the source to increase performance for cost-effective real-time monitoring. In addition, this solution has the advantage of avoiding problems that may arise from the Internet connection. Finally, it is possible to modify the operating code inside the edge devices or the architecture. For all these reasons, MOREPRO's architecture is very flexible and innovative.

1.4 Partnership

To achieve the aim of this project, several companies, each specialising in a specific field, are involved. It is relevant to analyse how each company is involved in the project to understand how a development process is faced when designing a new and innovative device. Below is a list of the companies that are involved in the MOREPRO project:

- **Brain Technologies:** plays a very relevant role in the project and can be considered the manager of MOREPRO project manager. It mainly contributes both to the definition and design of the digital architecture (in collaboration with the other partners) and the software development of the distributed intelligence system proposed by the project. Finally, Brain technologies is involved in the different phases, such as implementation, testing and validation of the final prototype.
- **MCM S.p.A:** deals with the design of high-precision production machining centres and flexible machining systems (FMS). It has defined the requirements and constraints of the algorithm through the definition of use cases. In addition, MCM plays an active role in other activities such as interfacing the machine for data collection, including the equipment of new sensors and supporting the integration of the new MOREPRO solution with the plant supervision software. Finally, it is responsible for the analysis and testing of the prototype system with its validation on the machines of CAMS, a partner in the project.
- **AL.MEC:** is a mechatronic company engaged both in the design and manufacture of boards for controlling electronic systems. In addition, it is involved in the design and manufacture of specific electronic boards and equipment in MOREPRO project to acquire data from sensors and process them.
- **CAMS:** is an expert company in its field and taken part in the project as an expert user of highly flexible production lines for producing complex value-added parts. In particular, CAMS within the project carries out the following activities:
 - definition and analysis of the requirements to develop new solutions for monitoring and predictive management of the plants.
 - identification and definition of its industrial cases of interest.
 - implementation, testing and validation phases of the final prototype within its production line equipped with the flexible systems supplied by MCM.

1.5 Thesis outline

The aim of this thesis was to develop an accurate model of the CNC machine, using the knowledge of robotics acquired during my studies. This model should be able to provide relevant artificial data. Then, the artificial data can be employed to develop appropriate algorithms to be able to have an estimate of the state of health of the tool (SOH). In this phase, I worked together with my colleague Dario Caniglia. In particular, first, we created a robotic model describing the

CNC machine considered, then we worked on the kinematic model of this machine to simulate the movements of the CNC machine and the planning part of the trajectory. Afterwards, I have studied how to dynamically model the process using the formulas and concepts found in the literature, while my colleague worked on the interaction between the cut tool and the working for identifying the contact points that occur during the simulation of the cutting process. In particular, I have done a dynamic formulation of the cut process to find a formulation that allows us to estimate the value of the beta friction coefficient during the cutting operations. The beta value is very relevant in the estimation of the state of tool wear. Then, I carried out tests in Matlab and Simulink to verify the precision of the mathematical formulation of beta for different machining conditions. In the end, I simulated the system, which my colleague and I modelled, plotting the different quantities of interest. Below is how this thesis project is structured.

In the first chapter, it is possible to find a short introduction concerning the problem of wear estimation and its complexity. Then, the MOREPRO project is presented, explaining its objective, the general architecture conceived and developed by Brain Technologies and its partner companies, and the advantages of this method. At the end of this chapter, all the companies involved in this MOREPRO project, were presented.

In Chapter 2, it is possible to find an overview of the different methods for estimating CNC in the literature, highlighting their advantages and disadvantages. The work carried out by the previous MOREPRO team was then generically reviewed. Finally, the finite element method (FEM) was briefly presented, as in this thesis work, a similar approach to the fem.

In the third chapter, there is a general description of the CNC machine. In particular, the one analysed in this thesis project is a 5-axis CNC milling machine. The CNC machine can be modelled as a robot composed of two parts. For modelling these two parts, an introduction to *Denavit-Hatenberg convection* is given, followed by an illustration of the kinematic model of the first part of the *CNC* machine that moves the cut tool (end effector). Then, there is a description of the algorithm used for kinematic inversion. As done for the first part, the same analysis was conducted on the second part of the CNC machine, i.e. *the part of the CNC machine in which the workpiece is positioned and oriented in space*.

In Chapter 4, there is an explanation of the trajectory planning part. The theory behind this task was explained, highlighting the reasons behind our choices. Then it is explained how we calculated a linear and a circular trajectory. These trajectories can be used as input for the CNC model described in the previous part.

In chapter 5, there is an introduction to the theory of cutting by referring to the literature. The dynamic formulation of the system is based mainly on the cutting theory, paying attention to the shear and normal forces at play during the cutting process, making it possible to arrive at a formulation of the beta friction coefficient

which occurs during the interaction between the tool and the workpiece. Next, it describes how the digital controllers were derived so that in the two cases of contact and non-contact, to achieve a constant spindle speed on which is placed the tool. Finally, in the last part of this chapter, they are shown the results of the various simulations carried out.

Chapter 2

State of the Art

As was expressed in the first chapter, modern companies aim to improve the efficiency of the work process to reduce costs by avoiding the waste of economic resources. For this reason, the development of new advanced maintenance techniques is very relevant. In this area, we find the *EN 13306 (2001) standard* (figure 2.1) that mainly divides maintenance techniques into two categories:

- **Corrective maintenance;**
- **Preventive maintenance.**

Corrective maintenance, also known as *run-to-failure*, is defined as "the set of maintenance actions that do not lead to increasing the value or productivity and performance of a system, but simply tend to restore the operational state before a fault or failure occurs". In summary, it is a matter of identifying the component on which a failure has occurred and replacing it or fixing it to re-establish the correct operation of the machinery or plant. So this type of approach is used when, following a failure for breakdown, the consequences and repairs do not require high costs and time availability. Furthermore, it is possible to find a further subdivision of Curative maintenance into two sub-categories, namely *Palliative maintenance* and *Curative maintenance*. *Palliative maintenance* refers to the case of a temporary repair, while *Curative Maintenance* refers to when the repair is permanent. On the other hand, the second category of maintenance, i.e. preventive maintenance, aims to carry out a maintenance intervention before the component failure occurs thanks to the real-time analysis of the data acquired by the system. Therefore, this type of maintenance can be considered time-based, i.e. the components are replaced according to a re-established schedule considering the working hours of the part and, in this case, we speak of predetermined maintenance. However, this technique does not provide optimal results as parts may be replaced before their critical condition, increasing costs. In addition, this technique does not provide us with

very reliable results. Therefore, another type of maintenance is introduced, which is predictive maintenance. The latter tries to estimate the SOH of the machine, referring to the working conditions in which the system is and to more dynamic algorithms. [2].

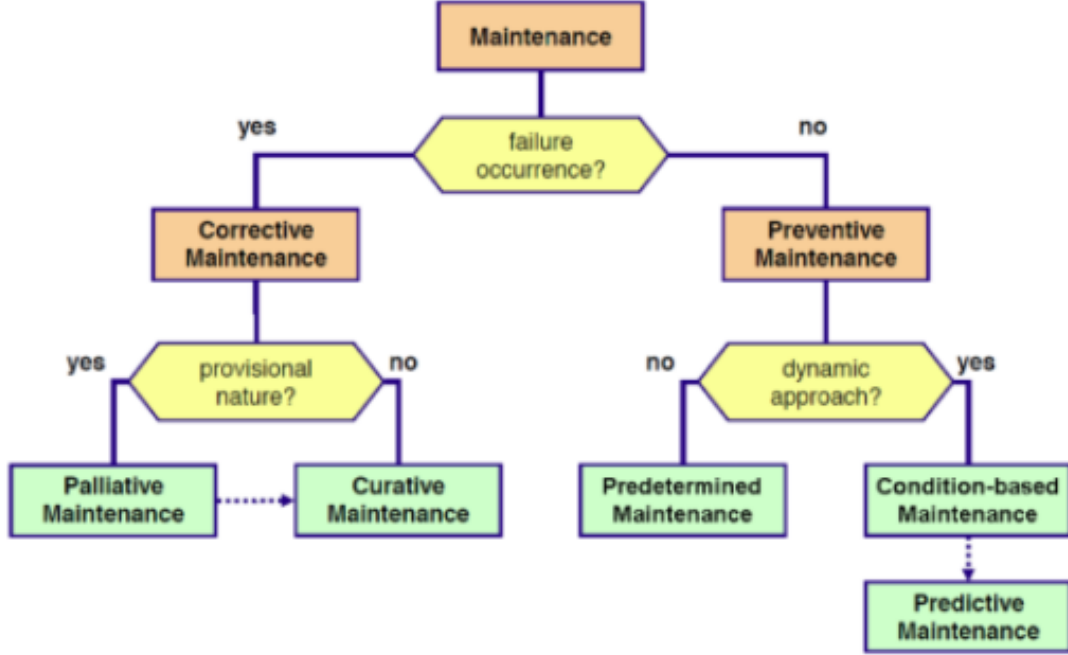


Figure 2.1: Forms of maintenance according to the standard EN 13306 (2001)

2.1 State of Health estimation methods

As stated in the previous section, the concept of predictive maintenance is introduced for estimating SOH and for coping with the disadvantages of other types of maintenance. However, several techniques are employed due to the complexity of predictive maintenance. The model, considered in my thesis work, produces artificial data that will later be useful in developing a suitable method for estimating the SOH of the component.

In the following section, there is an introduction to the different estimation techniques.

2.1.1 Model-based fault detection

The basis of this approach is a physical model of failure that allows predicting the rate of degradation of a component or its lifespan. For this reason, it is necessary to find a mathematical model to describe the failure mechanism. Furthermore, depending on how much the model is accurate, the estimation of the health will be more or less precise. Unfortunately, it is not always possible to obtain a model that adheres perfectly to reality, so there must be a compromise between a very accurate model and an estimate to compensate for the lack of knowledge about certain aspects of the installation. Mainly, this approach follows some predefined steps, which are: [3]

- **Identification of critical parts:** in this phase, we first identify the components that contribute significantly to the machine's life and then study and analyse them. This phase results be helpful, especially when dealing with very complex plants.
- **Model definition:** this is the most delicate phase. In fact, after determining the failure mechanism, a model is drawn up to describe the most important aspects of the system.
- **Evaluation of loads:** During the process, loads affect the possibility of faults in different ways. For this reason, it is helpful to understand which of these loads are most responsible for damages and how they are related to system activity.
- **Data collection and failure prediction:** After defining the model, data from the system can be obtained through sensors and then combined with the data provided by the model to get the most accurate possible estimate of the *SOH* of the system.
- **Evaluation of the model:** In the end, it is possible to determine how reliable the defined model can predict the failure mechanism by comparing the estimated failures that we have with the model with the actual failure data.[3]

Referring to the models found in the literature, they essentially can be divided into two categories:

- **Physical models:** in this case, the models describe the system behaviour through physical laws that link the various macroscopic parameters of the system, such as torque, forces, currents, etc. Moreover, using this type of model, it is possible to obtain precise results, especially when considering the microscopic characteristics of the materials. However, it has the disadvantage of being very complex and therefore very time-consuming to calculate.

- **Mathematical models:** This model type is convenient when the system is very complex by adopting a physical model. This method is, essentially, based on the use of identification procedures. Firstly, taking into account the available information about the system, the assumptions made about the structure of the system and the complexity. Then, the system is described using a mathematical function composed of computed coefficients. These types of models have the disadvantage of not having a precise relationship between the physical parameters of the system [4].

2.1.2 State observer

Another widely used approach for estimating the state of Health is the state observer. The last can be of different types depending on the case considered. Indeed, for linear systems with additive Gaussian noise terms, KF is used. When the systems present a non-linear behaviour with additive Gaussian noise, an EKF is employed. Finally, for non-linear systems with non-Gaussian noise terms, PF and Bayesian theory are used. [3].

- **KF:** The *Kalman filter* is an efficient recursive filter that allows estimating the dynamic state of the system from a series of measurements subjected to noise. The term recursive emphasises an intrinsic advantage of this method, i.e. the algorithm underlying the *Kalman filter* requires only the measures of the current instant to calculate the state estimate and not the storage of all past data. In the following figure, we have a macroscopic diagram of how the *Kalman filter* operates. [5][6]

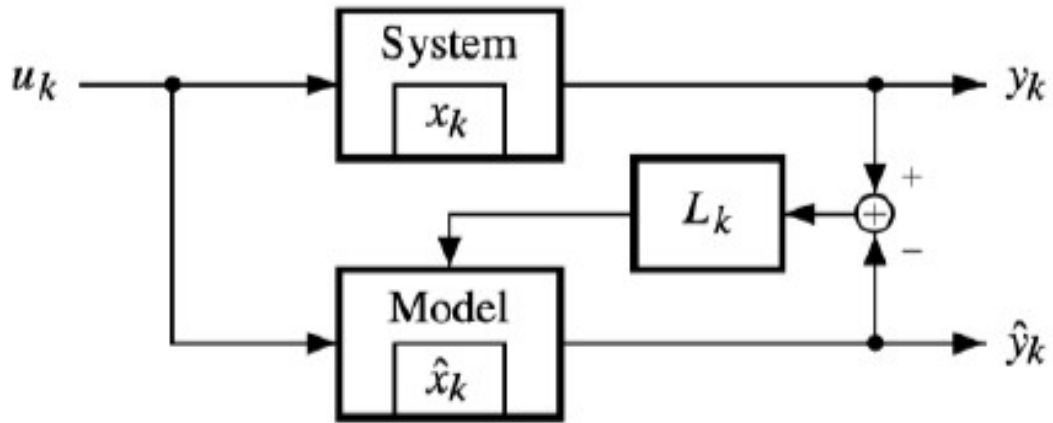


Figure 2.2: Generic structure of Kalman Filter

The *Kalman filter* operates by a parallel simulation of a real system and a system model, simulating the performance of the real one, having the same measured input u_k . The output of the actual system y_k , which can be measured using sensors, is compared with the output \hat{y}_k of the system model. The difference between the two quantities is called output error or innovation. Finally, the output error is first converted into a vector value by multiplication with L_k and then used to approximate as closely as possible the state of the model \hat{x}_k , known from the simulation to the actual system's state x_k . Subsequently, the estimated state and uncertainty estimates are updated using recursive relations that are very efficient from a computational point of view.

- **EKF:** as was written previously, this type of filter is used with non-linear systems characterised by additive Gaussian noise. The algorithm underlying this type of filter is similar to the **KF**. The only difference is that the system is linearised for considering the system as a linear time variable (*LVT*).
- **Particle filters:** *Particle filters*, or *Sequential Monte Carlo Methods*, are used when dealing with nonlinear systems in the presence of non-Gaussian noise. The objective of these filters is to calculate the posterior distribution of the state using partial observations, subject to noise, taking into account the estimate of the internal state and its associated uncertainty to obtain weighted samples, called particles. Obviously, the greater the number of particles considered, the greater the efficiency of these filters. Focusing on the estimation of the state of health (*SOH*), the algorithm underlying these filters types consists of two phases: the first phase consists of state estimation, i.e. are estimated both the current size of the fault and how the parameters vary in the environment. On the other hand, the second phase consists of the state's prediction from $(\tau + 1)$ to $(\tau + p)$ by combining the data obtained in the first phase, i.e. the estimate of the size of the fault at the considered time instant, and a damage growth model. Then, knowing the lower and upper limits of the failure zone, it is possible to estimate the prognosis interval.

2.1.3 Data-driven fault prognosis

This category includes so-called learning algorithms. These techniques can create a non-linear structure from the measurement signals and their statistics, providing good results from the input data. They are very interesting methods, as they are less complex and less expensive than methods based on models. In the following, we have a description of the most relevant techniques in this category, such as principal component analysis (PCA), partial least squares (PLS), artificial neural networks, fuzzy logic systems and graphical models like hidden Markov models (HMM).

- **Artificial neural networks:** The basic idea of this technique relies on the concept of how the biological nervous system works. It consists of a network of structures called neurons. The input of the network is a set of monitored data, while output is composed of predetermined values. After this, it is possible to find the relationship between the input and output of the system thanks to so-called training algorithms. The network of neurons are self-adaptive structures, i.e. each neuron is assigned a weight, which is then adjusted to have a more precise approximation of the system. Then, this phase of training, that is, after the correct definition of neurons' weights, the ANN can generate the desired output how a prediction of the evolution of the failure. The structure of the artificial neural network is shown in figure 2.3. [3]

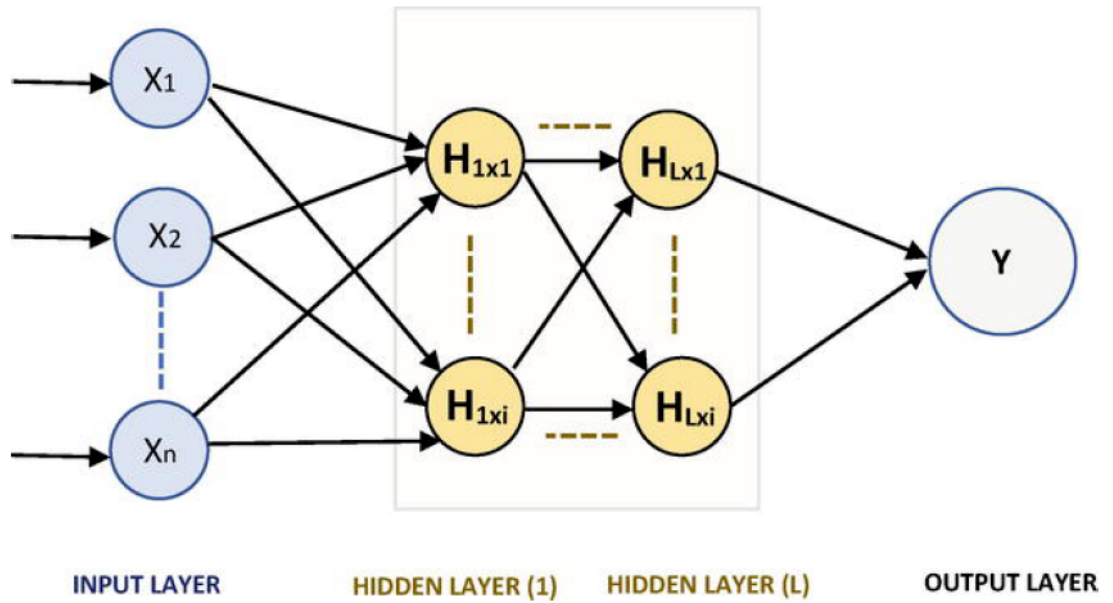


Figure 2.3: Artificial neural network's structure

- **Fuzzy-logic:** It is a widely used technique in various fields because of its many advantages. It is very similar to the one described above, but, unlike the previous one, the base of this method relies on human linguistic and reasoning skills. The Fuzzy logic technique provides a mapping between input and output signals. In the end, the mapping occurs through a list of if-then statements, also called rules. Then, considering the rules and regulating the belonging functions, relevant data are provided by this technique. After that, it is possible to achieve a forecast of the system conditions. In the end, as can be seen from the figure 2.4, the architecture of fuzzy logic is composed of

three phases[7]:

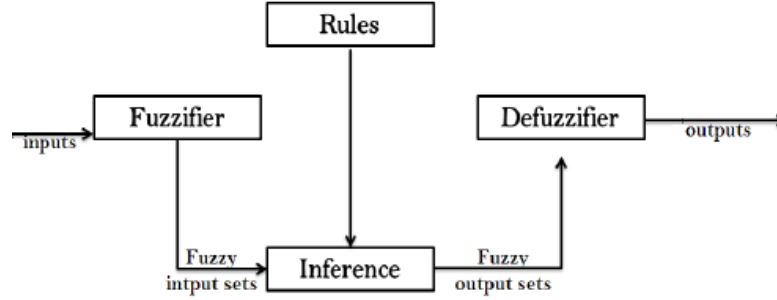


Figure 2.4: Components of a fuzzy-logic system

- *Fuzzification*: Essentially, in this phase, there is the conversion of the inputs, so-called crisp inputs, because they are measured and, subsequently, processed by control systems into a fuzzy set. The conversion takes place using fuzzy linguistic variables.
 - *Inference*: in this second phase, an inference occurs. In more detail, first, there is a determination of the level of correspondence between the current fuzzy input and each rule. After that, there is the decision on which rules are to be activated based on the type of input. In the end, the rules, which were activated before, are combined to obtain control actions.
 - *Defuzzification*: in this last phase, the output is mapped through the use of the so-called membership functions. A membership function is a graph that associates a membership value between 0 and 1 to each point in the input set. If we refer to predictive maintenance, the result is a curve, in which it is possible to find helpful information on wear conditions.
- **Hidden Markov Models**: Around the 1970s, this technique was enhanced by the Russian mathematician Andrey Andreyevich Markov, who made a major contribution to the relevant statistical theory. The HMM is a statistical model that describes the transitions between system states. Essentially, It is an extension of the Markov chain, but in this case, the states are partially observable or unobservable. Figure 2.5 shows the general structure of a discrete-time HMM considering N states, $S=(s_1, s_2, \dots, s_N)$ and M observation symbols, $V=(v_1, v_2, \dots, v_M)$.

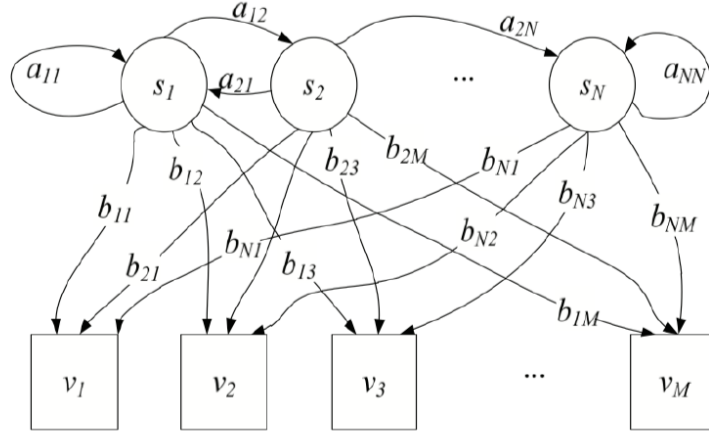


Figure 2.5: General structure of discrete-time hidden Markov model

As shown in Figure 2.1, the circles represent the states of the system. Then, the various states are interconnected each to the other by a transition, an arrow going from one state to another possible state. Moreover, it is relevant to emphasise that the hidden state of the system at time t , denoted by q_t , follows the transition rules of the states in the Markov chain, i.e. the state at time t depends only on the which at time $t-1$. It is possible to define a Markov model as $\lambda=(A,B,\pi)$ where:

- A is the transition matrix and stores the probability that state j follows state i ;
- B the observation matrix, i.e., a matrix indicating the probability that state j produces on observation k ;
- π represents the initial distribution over the hidden states.

HMMs, using the Baum-Welch algorithms, are very useful for both estimating the state of the system and estimating when a failure may occur. In addition, thanks to its characteristics, it can indicate the probability to find the system in a specific state after n iterations [3].

2.1.4 Vibration monitoring

Another technique used to estimate the state of health is vibration monitoring. As the name suggests, the basis of this technique is vibration, which, from a mechanical point of view, contains a great deal of information helpful to catch any damage

quickly. However, this method is more effective for diagnostic aspects of the system than for prediction. Indeed, in some cases, this technique provides a prognosis of the system. By referring to the PF curve shown in figure (2.6), it is possible to identify two zones in the curve, the right and the left. The first part of the diagram, the one on the left, is used for monitoring purposes where the condition varies slowly up to point P, called the deterioration observability point. On the other hand, the second part, which goes from P to point F, the point where the failure occurs, is used to predict the behaviour of the curve. [8].

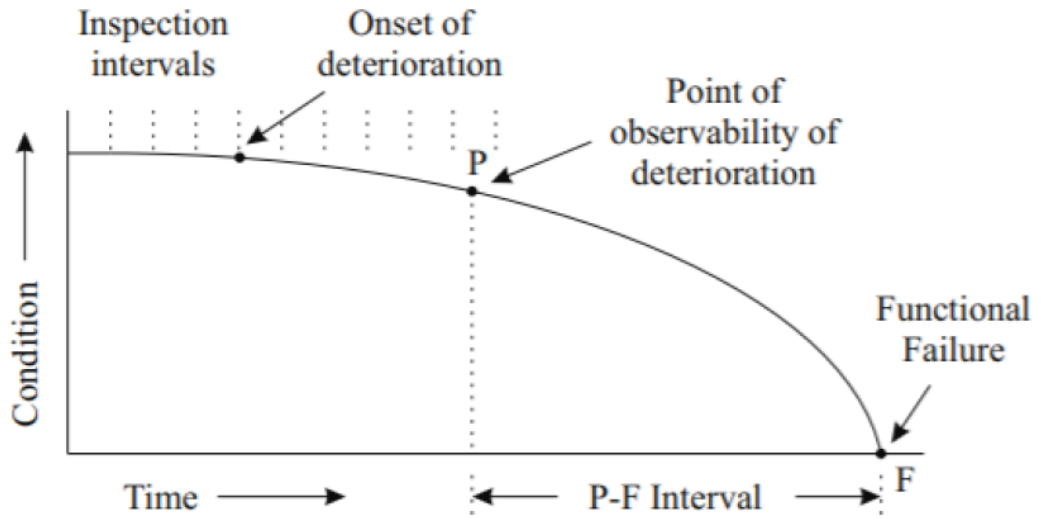


Figure 2.6: P-F curve

2.1.5 Moving horizon estimation

Another technique for estimating the state of a dynamical system is Moving horizon estimation, with the acronym MHE. It is useful when the system studied is non-linear, subject to constraints and disturbances. It is an iterative method performed by linear and non-linear problem-solving programs to minimise the objective function defined as a moving window consisting of a limited number of time steps. The object function has two terms, the first of which indicates the error in the output, based on recently made measurements, and the second term suggests the discrepancy between the estimated and predicted current state (calculated at the beginning of the moving window). [9]

2.1.6 Mixed/Hybrid algorithms

All the described methods have both many advantages and many disadvantages, unfortunately. Therefore, to get better results, several methods have been developed through the mix of some of them. In fact, combining these techniques, the disadvantages are compensated by obtaining a higher quality on the estimate of the state of health than the estimation obtained by a single method.

2.2 Results from past MOREPRO team

The MOREPRO project, owned by Brain Technologies, was already underway years ago, i.e. before I started my thesis work. For this reason, it was essential to analyse the work carried out on the project by previous colleagues in the group who worked on the project. Within the project, it is possible to find a subdivision of three sub-teams, which are: [10] [9] [5] [6]

- Requirements team: The team's members first analysed the requirements and specifications for the different parts of the project and then developed an experimental design. The experiment design is relevant for testing the functionalities, first individually and then together. The work done by this team is very suitable, as the analysis of the requirements and specifications is a preliminary step in the so-called V-shape development flow.
- Modelling team: Mainly, this team developed a preliminary kinematic and dynamic model of the *CNC* machine.
- Prediction team: The main aim of this team is to obtain an estimate of the *SOH* of the machine. To achieve this, the team used a simple model of the system to compute estimates of the *SOH*. They used the approach based on extended Kalman filters and error analysis techniques in the Matlab and Simulink environment.

As stated in the previous chapter, my thesis work regarded the modelling of the model. This model describes accurately the *CNC* machine, the kinematic modelling of the model, the planning of the trajectory and the dynamic analysis of the cutting process to obtain a formulation of the beta friction coefficient, which helps predict the *SOH* of the machine. Therefore, my work is part of the modelling team. So, to have a starting point, it was helpful to analyse the work done previously. They tried to design a simple model that could describe the interaction between tool and workpiece during the cutting process. The model was, essentially, based on a disc that makes a rotation motion in the direction of the piece to perform the cut, as can be seen in the (figure 2.7).

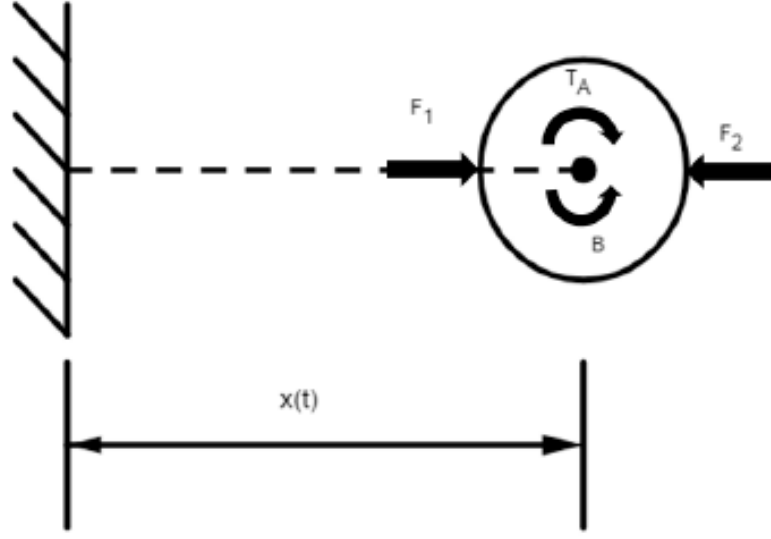


Figure 2.7: Simplified milling machine model

First, they focused on the mechanical analysis of the process and analyzing the balance of forces they obtained the following formulas:

$$\begin{cases} \ddot{\theta} = \frac{T_a - \beta \dot{\theta} F_c}{I_n} \\ \ddot{x} = \frac{F_1 - f_c(F_2 \Delta_x + cost)}{m} \end{cases} \quad (2.1)$$

The *CNC* movements are possible thanks to an electric motor. Furthermore, it was relevant to analyze the motor. Typically, the engines that drive *CNC* machines are brushless motors or servomotors. For simplicity, a DC motor was taken an account, represented simply by the circuit shown in the figure 2.8.

Referring to the figure 2.8, the following formulas were derived:

$$\begin{cases} V_s - V_b = Ri_a(t) + L \frac{di_a(t)}{dt} \\ V_b = kb\dot{\theta} \\ T_a = k_t i_a(t) - Att_{mot} \dot{\theta} \\ T_a = I_L \dot{\theta} \end{cases} \quad (2.2)$$

Finally, the dynamic equations of the electro-mechanical model were:

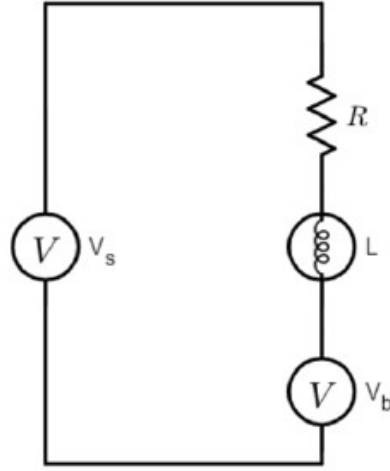


Figure 2.8: Simple model of DC motor

$$\begin{cases} \ddot{\theta} = \frac{k_t i_a}{I_n} - \frac{Att_{mot} \dot{\theta}}{I_n} - \frac{\beta F_c \dot{\theta}}{I_n} \\ \ddot{x} = \frac{F_1}{m} - \frac{F_c (F_2 \alpha + c)}{m} \\ \dot{i}_a = \frac{V_a}{L} - \frac{R i_a}{L} - \frac{k_v \dot{\theta}}{L} \end{cases} \quad (2.3)$$

where the states are:

- θ : Rotational velocity;
- \dot{x} : Linear velocity;
- \dot{i}_a : Linear velocity;

The model input are:

- V_a : Armature voltage;
- F_1 : horizontal force that moves the cutter;
- F_c : function that defines the contact with the workpiece. Its value is 1 when there is the contact and 0 in the other case.

The model parameters are:

- β : coefficient of friction;

- L : Motor inductance;
- m : Tool mass.

Starting from this simple model, the previous team tried to obtain a more accurate and complex model by adding information for getting a model, which describes the interaction between the tool (end-effector) and the workpiece more accurately. So, the previous team has searched scientific articles in the literature that provide more detailed information on the friction coefficient, being a very relevant parameter in cutting operations, to estimate the *SOH* of the tool. Within the scientific research field, there are numerous analyses concerning the friction coefficient due to the complexity of the cutting process. The friction between the cut tool and the workpiece depends on various parameters such as temperature, materials, cutting speed, applied forces, and contact surface. Among the myriad all the of analyses examined, two of them have been taken into account to obtain a comprehensive formulation of the friction coefficient, which . The two analyses are the cut force analysis and the thermal analysis. Therefore, below these two analyses have been briefly described. So, subsequently, there is an introduction of two analyses. However, for my thesis work, other concepts found in the literature, which will be introduced in the last chapter of this thesis work, concerning the shear process were taken into account to obtain a new formulation for of the friction coefficient.

2.2.1 Cutting force analysis

Referring to article [11], the cutting operation was analysed from a mechanical point of view to find a relationship between the friction coefficient and the forces applied during the cutting process. In this analysis, there was an estimation of cutting forces, considering the cutting tool divided into small elements. Then, the cutting tool was examined by mechanical analysis. In particular, as shown in figure 2.9, the force occurring during the process can be broken down into three components (dF_t , dF_r and dF_a) along the tangential, radial and axial directions.

The three force components are described mathematically as:

$$\begin{cases} dF_t(\phi, z) = K_t h_j(\phi, z) dz \\ dF_r(\phi, z) = K_r dF_t(\phi, z) = K_t K_r h_j(\phi, z) dz \\ dF_a(\phi, z) = K_a dF_t(\phi, z) = K_a K_t h_j(\phi, z) dz \end{cases} \quad (2.4)$$

Where

h_j : chip thickness in point j ;

K_r , K_t and K_a : The coefficients of the cutting force;

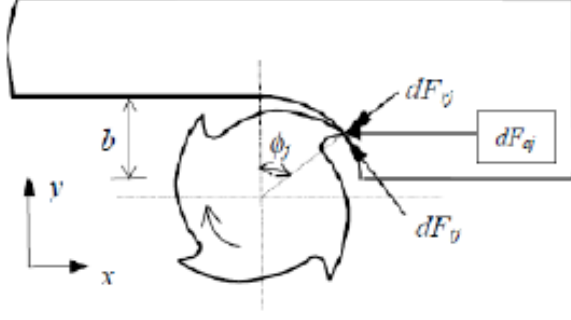


Figure 2.9: Cutting force decomposition in the milling process

ϕ : Shear angle.

The shear forces expressed according to the Cartesian reference system XYZ are:

$$\begin{cases} dF_{xj}(\phi) = dF_{tj} \cos(\phi_j) + dF_{rj} \sin(\phi_j) \\ dF_{yj}(\phi) = dF_{tj} \sin(\phi_j) - dF_{rj} \cos(\phi_j) \\ dF_{zj}(\phi) = dF_{aj} \end{cases} \quad (2.5)$$

In the end, the final result of this analysis linking the coefficient of friction to the forces occurring during the process is as follows:

$$\beta = \tan \left[\arctan \left(\frac{dF_{yj}(\phi)}{dF_{xj}(\phi)} \right) + \gamma \right] \quad (2.6)$$

2.2.2 Thermal analysis

The thermal aspect is another factor that influences the friction coefficient. Therefore, a thermal analysis was carried out for finding a relationship correlating the friction coefficient with the temperature developed in the cutting process. However, finding this relationship between friction and temperature is very complex because there is no constant temperature in the cutting zone during the cutting process.

As shown in figure 2.10, the temperature has its maximum value in the stripping zone of the tool. Then, there is a decreasing gradient moving away from this zone. Another aspect to consider is that the temperature also depends on the cutting speed during the cutting process. In particular, the higher the cutting speed, the greater the amount of heat produced. [12].

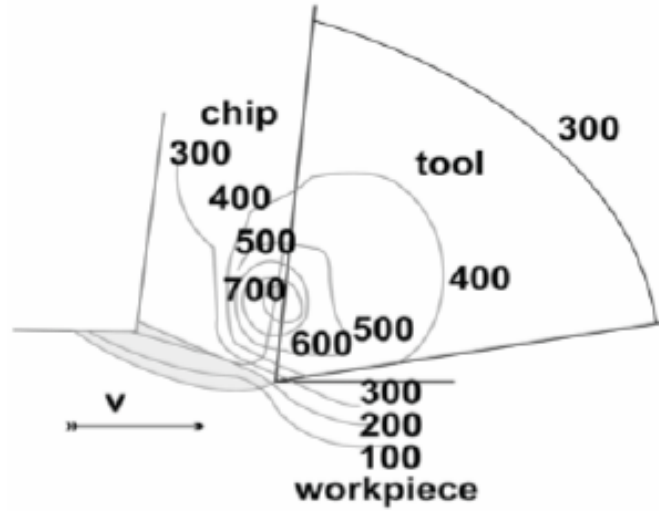


Figure 2.10: Temperature distribution in the contact area

2.3 Finite Element Method

After the analyses carried out previously, we decided to make a macroscopic analysis to obtain more information, so we considered the tool as made by many infinitesimal small elements. Firstly, we generated a model for describing the cutting process and used those elements to get a mechanical analysis of the interaction between the tool and the material. Considering other aspects, such as the heat distribution on the working surface, it's possible to have a better model. To achieve this purpose, my colleague and I, , used an approach similar to the Finite element method (FEM). The finite element method, abbreviated to FEM, is a numerical technique for deriving approximate solutions to problems described by partial differential equations. For this reason, this method subdivides the considered system into smaller parts with homogeneous characteristics, called finite elements. It is possible to obtain finite elements through a discrete space, got by a mesh development. For each finite element, a series of equations can be defined and afterwards assembled into a system of equations that models the entire system. Then, it is possible to obtain the final solution. The advantage of using this method is the detection of local effects. From a literature search, several articles have used this method to be able to analyse the different phenomena that occur during the cutting process of *CNC* machines obtaining good results [13] [14]

As result, we have chosen to refer to this method to develop our model to consider the mechanical behaviour of each finished element during the cutting process. In addition, the model can be improved, taking into account other aspects which

occur during the cutting process. Therefore, the first step to achieve our goal was to model the *CNC* machine considered in a robot that allows us to identify the position of the tool during the process. Next, the cutting tool was designed as a cloud of points to identify the elements that interact with the workpiece moment by moment. Then, the knowledge of the penetration of the cutting tool inside the workpiece at each instant time was very relevant for mechanical analysis to obtain an estimated value of the friction coefficient.

Chapter 3

CNC machine modelling

In order to obtain a simulator of the *CNC* machine to acquire artificial data on the machining process, the first step was to create a model describing the *CNC* machine. In addition, this model needs to reproduce the movements made by the machine.

3.1 CNC machine

A CNC machine is a machine tool based on computerised numerical control (hence the acronym CNC). it is very similar to a corresponding manual machine with the only difference that it has additional components, such as: [15]

- *Electric motor*: this controls the movement of the machine's axes;
- *Encoders*: this component gives information to the on-board computer on the movement and position of the axes
- *Drives*: these are special units that are very important to power and control the movements of the electric motors. Their complexity depends on the type of motor and the feedback type adopted.
- *Computer*: Can be considered as the brain of the machine. It acquires the data from the encoder and the instructions provided by the operator and the program. It also provides that information on the position and speed of the tool and guides the tool's movements during the process.

Another aspect of CNC machines is that they typically have a SCARA or Cartesian configuration (Figure 3.1). The advantages of CNC machines over manual machines are many as high accuracy and precision, making it possible to carry out work that is very complex or impossible to perform through a manual system. Furthermore,

the produced parts have the same precision, i.e. identical parts with a dimensional tolerance in the order of a micron, resulting in superior and uniform quality. Others relevant advantages are the cut in working time, which lowers unit costs and a reduction in labour costs as fewer operators are needed.

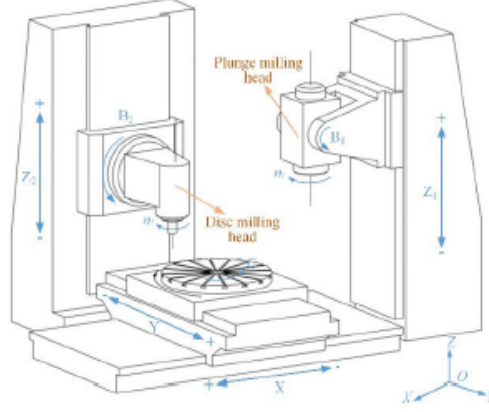


Figure 3.1: Example of CNC machine structure

Initially, CNC machines were used for high-precision machining, then, thanks to their versatility and precision, CNC technologies were introduced in other manufacturing sectors such as carpentry, upholstery, etc. Nowadays, mainly they are employed in machine tool processes, in which there are removing material from raw material to obtain the desired shape. There are many different operations in chip removal processes, such as Turning, Milling, Drilling and Grinding (Figure 33). (Figure3.2).[16]

For each machining operation, a different tool is being employed. Furthermore, it is necessary to provide parameters that are important in the machining operations, i.e. the cutting parameters, which are:

- *Cutting speed*: is the relative tangential speed between the tool and the work surface. Mainly, the unit of measurement is m/mm (ft/min). This speed can be converted into spindle rotation speed using the equation $N = \frac{v}{(\pi d)}$. The cutting speed is supplied to the machine's computer, and thus implicitly with the spindle rotation speed.
- *Feed rate*: This indicates the size of the chip formed by each tooth during the process with the unit of measurement mm/tooth. While the feed

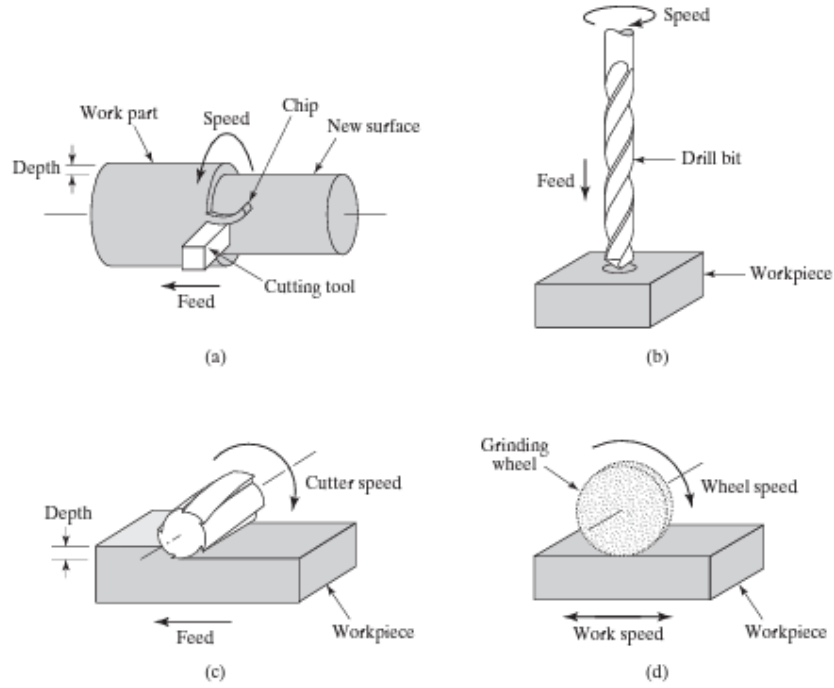


Figure 3.2: The four common machining operations: (a) turning, (b) drilling, (c) peripheral milling and (d) surface grinding

rate (mm/min), usually a parameter supplied to the computer, indicates how much the tool has moved in one minute.

- *Depth of cut*: indicates the distance the tool penetrated below the original surface of the work. Usually, its unit measurement is mm.

These cutting parameters also called cutting conditions, are very important and must be controlled during the operation of the CNC machine. In this thesis work, a CNC milling machine with five degrees of freedom, or five axes, was analyzed (figure 3.5). The allowed movements are along the three axes X, Y and Z and two rotations, which correspond to the tilting of the plane on which the material to be machined is placed. As mentioned above, The CNC machine has the configuration of a *SCARA* or *Cartesian* robot composed of two parts, as shown in figure 3.4. The first part is composed of two prismatic joints that, essentially, makes the tool (end effector of the robot) move along the x and y directions. The second part, which allows the workpiece to be moved and oriented, can be represented by a prismatic joint (translation along z) and two revolute joints (rotation A and B). To simplify the kinematic analysis of the system, we have analysed them independently. In fact,

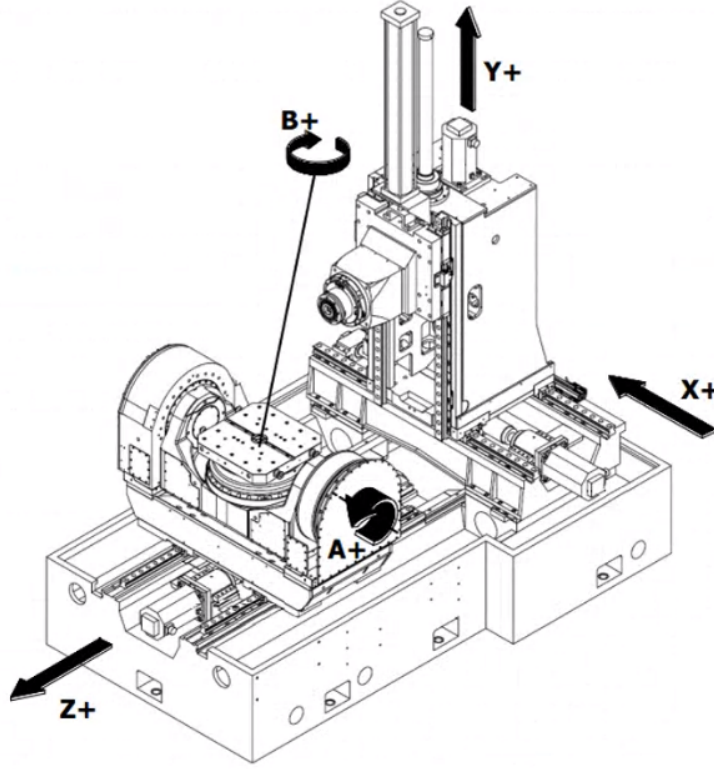


Figure 3.3: Structure of the analysed CNC machine

in this thesis project was assumed that the two parts work separately, i.e. first, the second part positions the workpiece in the correct required position, then the first part moves the tool to start the cutting process.

3.2 Direct Kinematics

3.2.1 Denavit-Hartenberg convention

Denavit-Hartenberg convention is a general and systematic method present in the literature for calculating the direct kinematic equation of an open-chain manipulator that express the relative position and orientation of two consecutive links through a matrix, the so-called homogeneous transformation matrix. Afterwards, thanks to the calculated matrices before, it is possible to find the pose of the end effector expressed to the base reference system. The first important step is to define the reference system attached to each link. A reference system is attached to the link when each point of the i -th link considered has constant coordinates to the frame

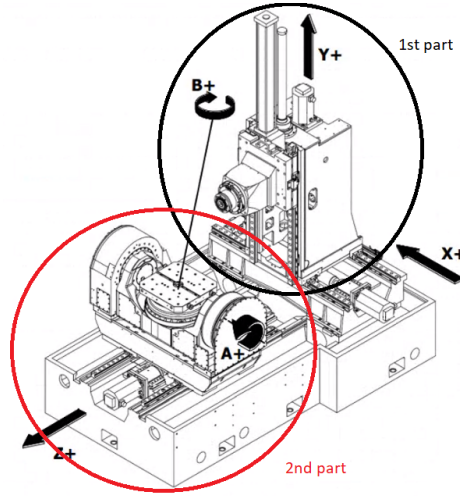


Figure 3.4: The two part of the CNC machine

i. Afterwards, when the i -th link is working, the reference system (i) will move in solidarity with the link considered. Among the various moving frames, we find the fixed frame 0, also called the base reference system. Often, frame 0 is associated with the base of the manipulator considered.

Therefore, considering a manipulator composed of n joints and $n+1$ links, it is necessary to define $n+1$ frames attached to the link using the Denavit-Hartenberg convention. So considering the generic i -th joint shown in figure 3.5, which connects the $i-1$ and i links respectively, we work according to the Denavit-Hartenberg convention to find the i -frame [17]

- The axis z_i of the i -th reference system is positioned along the axis of the i -th joint;
- The origin O_i is given by the intersection between the z_i axis and the common normal to axes z_{i-1} and z_i . Moreover, O'_i is at the intersection between common normal and axis z_{i-1} .
- The x_i axis is chosen along the common normal with a positive directed from joint i to joint $i+1$.
- Finally, the y_i axis is chosen to have a right-handed reference system..

However, there are many cases where DH convention does not provide a unique frame definition, such as:

- For frame 0, only the direction z_0 is specified while O_0 and X_0 are arbitrary.

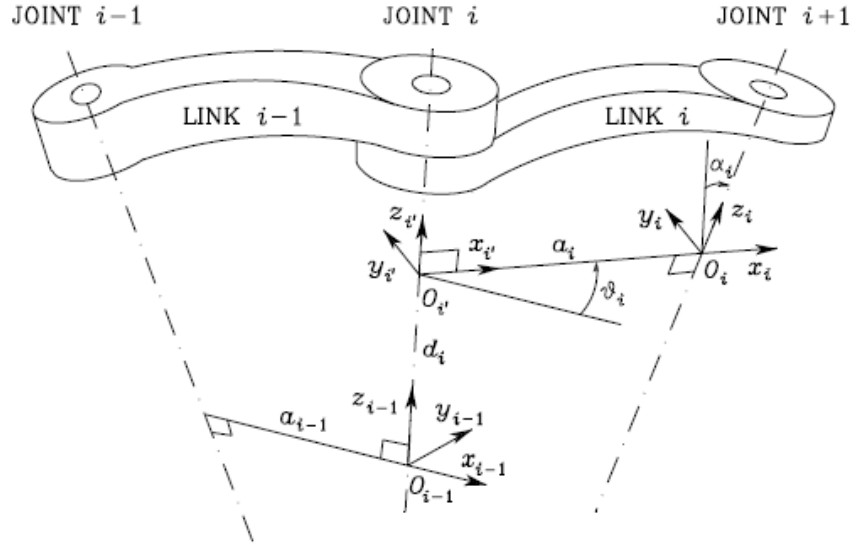


Figure 3.5: Example of a Denavit-Hartenberg application

- z_n is not unambiguously defined in frame n since there is no joint $n+1$. Typically, joint n is revolute, thus z_n is usually chosen to be aligned with the z_{n-1} axis. Instead, x_n is chosen orthogonal to z_{n-1} .
- In the case of consecutive parallel axes, the common normal is not determined. Afterwards, it is possible to choose an arbitrary one. The normal that passes through the origin of the joint reference system, it is usually chosen as the common normal.
- When two consecutive axes intersect, the common normal collapses to a point. Therefore, the direction of the x_i -axis can be chosen arbitrarily.
- When the i -th joint is prismatic, the direction of z_{i-1} is undetermined.

In the DH convention, four parameters define the pose of frame i to frame $i-1$. These four parameters are [17]:

- a_i : the distance between O_i and $O_{i'}$;
- d_i : the coordinate of $O_{i'}$ on the axis z_{i-1} ;
- α_i : the angle between z_{i-1} and z_i axes on the x_i with positive counterclockwise direction;
- θ_i : the angle between axes x_{i-1} and x_i on axis z_{i-1} .

Two of them, a_i and α_i , are constant and depend on the geometry of the considered part. Instead, considering the other two parameters, one is variable, indicated as q_i , and depends on the joint used to connect the $i-1$ link to link i , while the other is a constant. For a prismatic joint, the variable is d_i while α_i is constant. Vice-versa for a revolute joint. According to DH convention, the homogenous transformation matrix, which expresses the frame i to the frame $i-1$, is

$$A_i^{i-1}(q_i) = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

The variable parameter q_i is called joint variable. In the end, combining all the computed homogenous transformation matrix, the direct kinematic function is expressed as:

$$T_n^0(q) = A_1^0(q_1)A_2^1(q_2)...A_n^{n-1}(q_n) \quad (3.2)$$

3.2.2 Direct Kinematics of the two part of the CNC machine

Thanks to DH convention, it was possible to derive the direct kinematic function of the first part of the *CNC* machine. Figure 3.6 shows the kinematic chain with the reference triads obtained by applying the DH convention. As said previously, the first part is composed of two prismatic joints, falling within the particularities described in DH convention. However, as the direction of the z-axis was placed along their movement direction. In the end, referring to the reference systems found, it was possible to find the four parameters. Since the kinematic chain consists of two prismatic joints, d_1 and d_2 are the variables of the joints. The extrapolated parameters are in Table 3.1:

| DH parameters | a_i | α_i | d_i | θ_i |
|---------------|-------|------------|-------|------------|
| Joint 1 | 0 | 90° | d_1 | 90° |
| Joint 2 | 0 | -90° | d_2 | 90° |

Table 3.1: Denavit-Hartenberg parameters of CNC machine first part

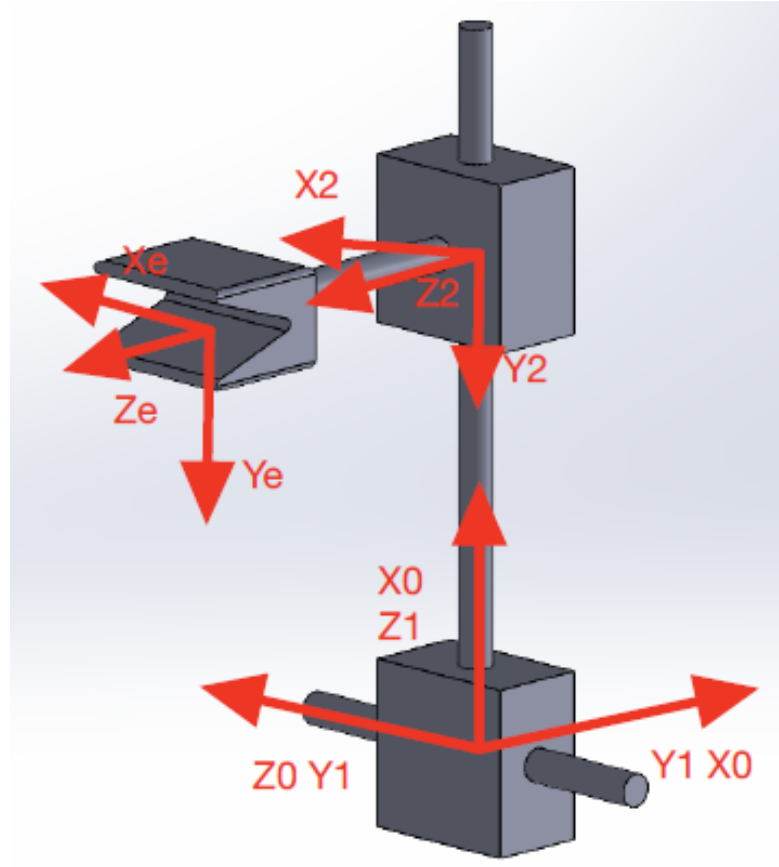


Figure 3.6: Model of the CNC machine part 1 drawn in Solidwork

Considering the parameter of the table 3.1, the following matrices were obtained :

$$A_1^0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$A_2^1 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$T_2^0 = A_1^0 A_2^1 = \begin{bmatrix} 0 & -1 & 0 & d_2 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Finally, a reference frame was placed in the base on the end effector, i.e. the cutting tool. The distance between RF_e and RF_2 is \bar{d} along the z_2 axes. Therefore, the matrix obtained at the end is:

$$T_e^0 = \begin{bmatrix} 0 & -1 & 0 & d_2 \\ 0 & 0 & -1 & -\bar{d} \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

This final matrix represents the direct kinematic function of the first part, so, knowing the values of the joint variables, it is possible to calculate the position of the end effector to the base reference frame.

Finally, the second part of the *CNC* model was analysed for obtaining the homogeneous transformation matrix associated with the workpiece (the direct kinematics equation of the second part). Initially, the reference systems of the kinematic chain were derived using the DH convention (Figure 3.7). The second part of the model that schematises the *CNC* machine for me consists of:

- *one prismatic joint*: allows the movement of the workpiece along the z_0 axis, which brings the workpiece closer to the cutting tool.
- *Two revolute joints*: orientate the workpiece.

The positioned reference systems have been considered as coincident, assuming that the two revolute joints are in a coincident position. On the contrary, in figure 3.7 they are showed as not coincident in order to obtain a clear kinematic chain diagram.

After the definition of the reference systems in the kinematic chain, it was possible to extrapolate the DH parameters. As the kinematic chain consists of a prismatic joint and two revolute joints, the joint variables are d_1 , θ_2 and θ_3 . The four extrapolated parameters are shown in Table 3.2.

As done for the first part, the following matrices were obtained:

$$A_1^0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

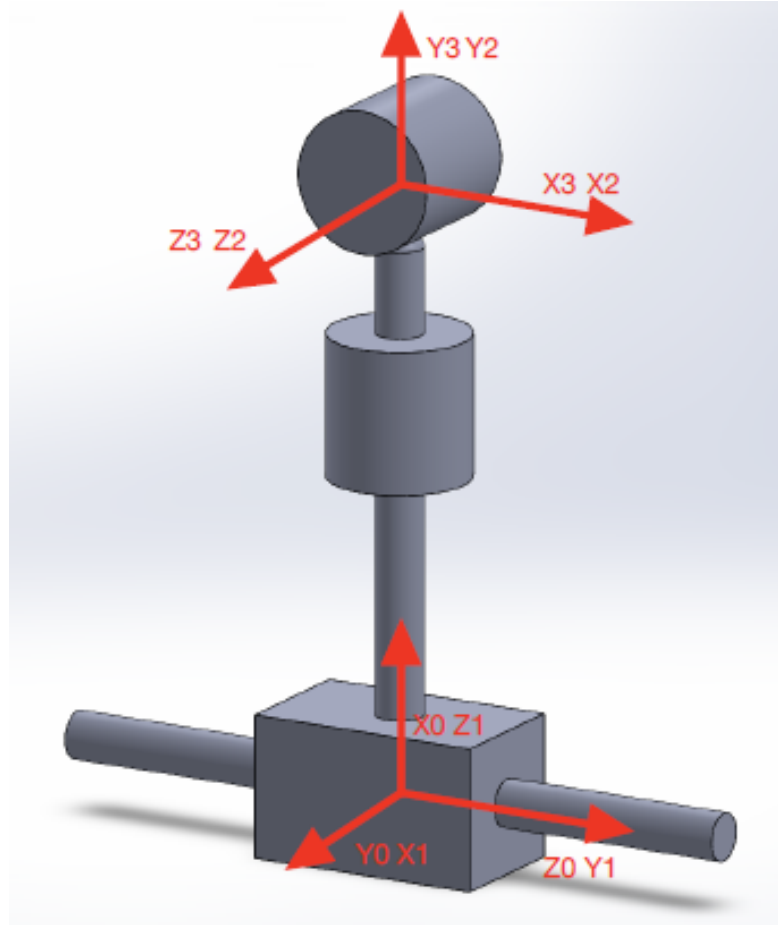


Figure 3.7: Model of the CNC machine part 2

| DH parameters | a_i | α_i | d_i | θ_i |
|---------------|-------|------------|-----------|------------|
| Joint 1 | 0 | 90° | d_1 | 90° |
| Joint 2 | 0 | 90° | \bar{a} | θ_2 |
| Joint 3 | 0 | 0 | 0 | θ_3 |

Table 3.2: Denavit-Hartenberg parameters of CNC machine second part

$$A_2^1 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & \bar{a} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$$A_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

$$T_3^0 = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} s_3 & c_3 & 0 & \bar{a} \\ c_2 c_3 & -c_2 s_3 & s_2 & 0 \\ s_2 c_3 & -s_2 s_3 & -c_2 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

For our analysis of the *CNC* machine, the cutting tool and the workpiece must be both expressed to the same reference system. For simplicity, the $x_0 - y_0 - z_0$ reference system of the first part of the *CNC* machine was chosen as the base frame of the second part, in order to express both the cutting tool and workpiece's coordinates to the same reference system. Finally, the transformation matrix between the base frame and the current $x_0 - y_0 - z_0$ reference system is:

$$T_0^b = \begin{bmatrix} 1 & 0 & 0 & -\bar{a}_1 \\ 0 & 0 & 1 & -\bar{a}_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

After calculating the transformation matrix T_3^0 , the homogeneous transformation matrix, which expresses the pose of the workpiece to the base reference system, is:

$$T_3^b = T_0^b T_3^0 = \begin{bmatrix} s_3 & c_3 & 0 & \bar{a} - \bar{a}_1 \\ s_2 c_3 & -s_2 s_3 & -c_2 & d_1 - \bar{a}_2 \\ -c_2 c_3 & c_2 s_3 & -s_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

The equation of the direct kinematics of the second part was derived to identify later the points of interaction between the tool and the surface representing the workpiece.

To simplify the analysis, instead of a 3D shape, the workpiece was modelled as a surface. The surface pose was obtained by the transformation matrix previously calculated as a function of the assumed values of the joint variables. After a geometric transformation, the surface's points have moved, changing their coordinates. To compute the new coordinate of these rotating points, firstly it was helpful to define the vertices of the surface. After that, it was important to remember that the centre of the table, where the workpiece rests, coincides with the origin of the frame attached to joint 3, whose pose is expressed to the base frame by T_b^3 . However, the work surface is not in the centre of the table but translated along x_3 . So, for example, considering that the workpiece is a cube with sides of length 10, the

coordinates of the four vertices to the frame attached to joint 3 are shown in Table 3.5.

| x_3 | y_3 | z_3 |
|-------|-------|-------|
| 5 | 0 | 5 |
| 5 | 0 | -5 |
| 5 | 10 | 5 |
| 5 | 10 | -5 |

Table 3.3: Vertices coordinates with respect to reference frame $x_3 - y_3 - z_3$

Afterwards, thanks to the homogeneous transformation matrices T_b^0 and T_0^{b3} , it was possible to obtain the coordinates of the vertices after a geometric transformation. The same procedure was followed for the points that compose the surfaces. In the end, this procedure was implemented on Matlab and Simulink where several tests were carried out. The results are shown in the following figures:

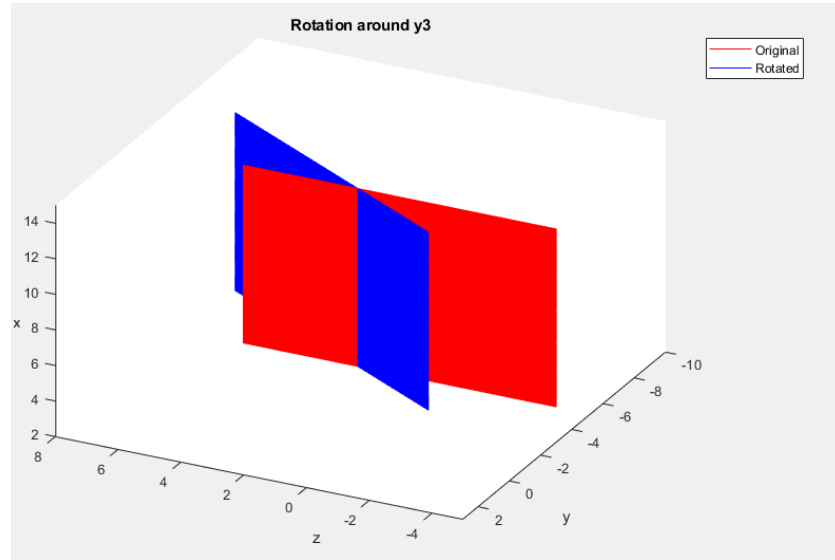


Figure 3.8: rotation around axis y_3 (blue) of the work surface in the original pose (red)

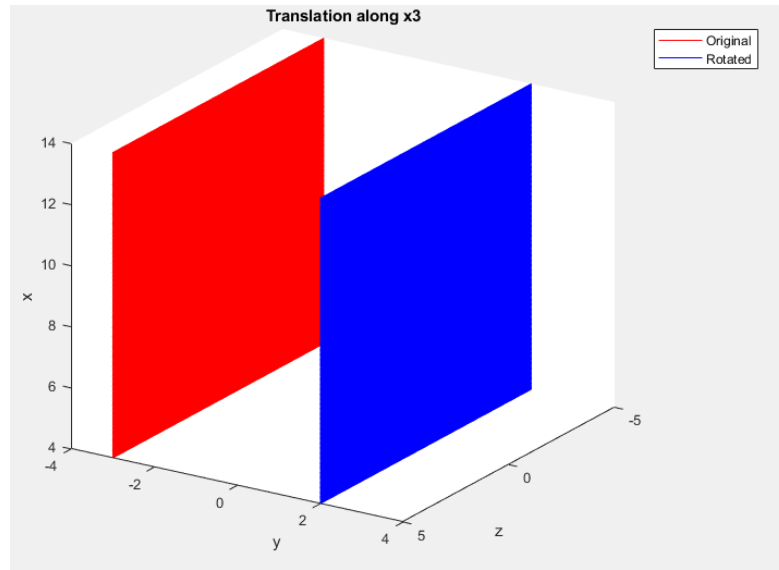


Figure 3.9: Work surface translation along x_3 (blue)

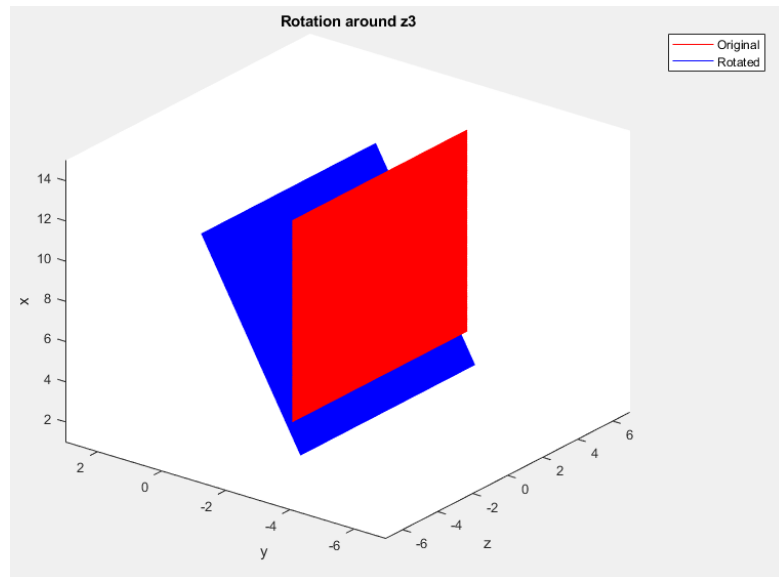


Figure 3.10: Work surface rotation around axis z_3 (blue)

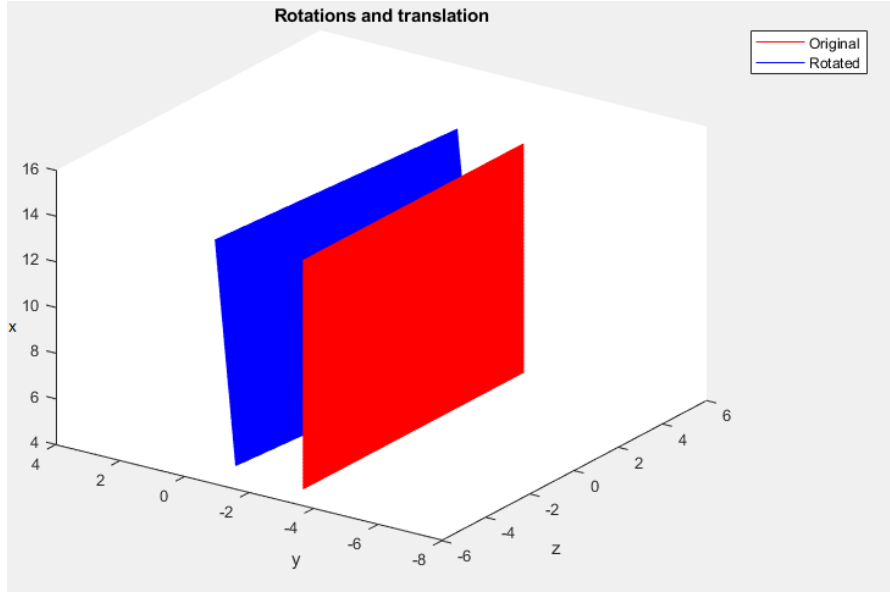


Figure 3.11: Work surface after a translation and rotation around axis z_3

3.3 Differential Kinematics

The second step in modelling the CNC machine was the analysis of differential kinematics. The last allows finding the relationship between the speeds of the various joints and the corresponding linear and angular velocity of the end effector. In particular, this relationship is well-defined by a matrix called Jacobian, which depends on the manipulator's configuration. Moreover, it is possible to have two types of Jacobian depending on the mode used to express the pose of the end-effector. In particular, we have a geometric Jacobian with the poses defined by the homogeneous transformation matrix. Instead, we have an analytical Jacobian with poses defined by minimal representation. In our case, the pose of the end effector was expressed through the homogeneous transformation matrix. Therefore, for our purpose, we have considered the geometric Jacobian. [17]

Indicating with \dot{p}_e and ω_e respectively the linear and angular velocity of the end effector and with \dot{q} the joint velocity, the relations expressing the linear velocity of the end effector \dot{p}_e , and its angular velocity ω_e , as a function of the joint velocities, are [14]:

$$\dot{p}_e = J_P(q)\dot{q} \quad (3.13)$$

$$\omega_e = J_O(q)\dot{q} \quad (3.14)$$

Where J_P is a (3xn) matrix that takes into account the contribution of joint velocities to the linear velocity of the end effector, while, J_O is a (3xn) that considers the contribution given by joint velocities to the angular velocity of the end effector. The previous formulas can be expressed in a compact form as:

$$J = \begin{bmatrix} J_P \\ J_O \end{bmatrix} \quad (3.15)$$

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J(q)\dot{q} \quad (3.16)$$

where the (6xn) matrix J is the Geometric Jacobian of the manipulator. Afterwards, skipping the theoretical details on how to obtain the Geometric Jacobian, The relation to derive the linear velocity of the end effector \dot{p}_e is:

$$\dot{p}_e = \sum_{i=1}^n \frac{\partial p_e}{\partial q_i} \dot{q}_i = \sum_{i=1}^n j_{Pi} \dot{q}_i \quad (3.17)$$

the term $j_{Pi} \dot{q}_i$ represent the contribution of the velocity of joint i to the end-effector linear velocity when the other joints do not move. This contribution depends on the type of joint:

- *Prismatic joint* ($q_i = d_i$):

$$\dot{q}_i j_{Pi} = \dot{d}_i z_{i-1} \quad (3.18)$$

and then

$$j_{Pi} = z_{i-1} \quad (3.19)$$

- *Revolute joint* ($q_i = \theta_i$):

$$\dot{q}_i j_{Pi} = \omega_{i-1,i} \times r_{i-1,e} = \dot{\theta}_i z_{i-1} \times (p_e - p_{i-1}) \quad (3.20)$$

and then

$$j_{Pi} = z_{i-1} \times (p_e - p_{i-1}) \quad (3.21)$$

Instead, the relation that expresses the angular velocity of the end effector ω_e is:

$$\omega_e = \omega_n = \sum_{i=1}^n \omega_{i-1,i} = \sum_{i=1}^n j_{Oi} \dot{q}_i \quad (3.22)$$

As for the linear velocity, the provided contribution of each joint depends on the joint type. In particular:

- *Prismatic joint*

$$\dot{q}_i j_{Oi} = 0 \quad (3.23)$$

and then

$$j_{Oi} = 0 \quad (3.24)$$

- *Revolute joint:*

$$\dot{q}_i j_{Oi} = \dot{\theta}_i z_{i-1} \quad (3.25)$$

and then

$$j_{Oi} = z_{i-1} \quad (3.26)$$

Afterwards, the Geometric Jacobian can be expressed as:

$$J = \begin{bmatrix} j_{P1} & \dots & j_{Pn} \\ j_{O1} & & j_{On} \end{bmatrix} \quad (3.27)$$

where:

$$\begin{bmatrix} j_{Pi} \\ j_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{for a prismatic joint} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a revolute joint} \end{cases}$$

The last relationship allows us to calculate the Jacobian based on the direct kinematics relationship. In this relation, z_{i-1} , p_e and p_{i-1} are a function of all the joint variables. In particular:

- z_{i-1} is the third column of the rotation matrix considered. R_{i-1}^0 :

$$z_{i-1} = R_1^0(q_1) \dots R_{i-1}^{i-2}(q_{i-1}) z_0 \quad (3.28)$$

where $z_0 = [0 \ 0 \ 1]^T$.

- p_e is the position of the end effector and it is composed of the first three elements of the fourth column of the homogeneous transformation matrix. T_0^e . The coordinate of p_e can be extracted from:

$$\tilde{p}_e = A_1^0(q_1) \dots A_n^{n-1}(q_n) \tilde{p}_0 \quad (3.29)$$

where $\tilde{p}_0 = [0 \ 0 \ 0 \ 1]^T$.

- p_{i-1} similarly to p_e , is derived from the fourth column of the transformation matrix T_0^{i-1} . The three elements of p_{i-1} can be extracted from:

$$\tilde{p}_{i-1} = A_1^0(q_1) \dots A_{i-1}^{i-2}(q_{i-1}) \tilde{p}_0 \quad (3.30)$$

Finally, considering the CNC machine in this thesis, according to the theory exposed previously, the following matrices were obtained:

$$z_0 = [001]^T$$

$$z_1 = [100]^T$$

$$J = \begin{bmatrix} z_0 & z_1 \\ 0 & 0 \end{bmatrix} \quad (3.31)$$

$$J = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.32)$$

Unfortunately, the matrix obtained is not full rank. However, from this matrix it was possible to extract a full rank matrix

$$J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.33)$$

Lastly, after deriving the full rank Jacobian matrix, the final equation that links the end-effector velocities to those of the joints is:

$$V_e = \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = J(q) \dot{q} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (3.34)$$

3.4 Inverse Kinematic Algorithm

Thanks to direct kinematics and differential kinematics introduced in the previous paragraphs, it was possible to find the solution to the inverse kinematics problem. The latter allows the calculation of the value of the joint variable $\mathbf{q}(t)$ considering the desired position of the end effector. To find the solution to the problem of inverse kinematic, between the various method, inverse differential kinematics was employed for this thesis. This method allows the computation of a feasible joint trajectory $(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ to reproduce the assigned motion trajectory to the end effector, given v_e and the initial pose. Given the differential kinematic equation [17]:

$$v_e = J(q)\dot{q} \quad (3.35)$$

The joint velocities $\dot{\mathbf{q}}(t)$ can be derived simply by considering the inverse of the Jacobian, i.e. from the following formula:

$$\dot{q} = J^{-1}(q)v_e \quad (3.36)$$

It is possible to apply this formula when the Jacobian matrix is both square and full rank, i.e. when the matrix is invertible. On the other hand, if the Jacobian matrix is not invertible, a full rank matrix is extracted from it. If the extracted matrix is not square, we use the so-called pseudo-inverse of the latter matrix. In our case, this problem does not arise since the full rank matrix extracted from the Jacobian is square, therefore invertible. Finally, if the initial posture of the manipulator is known, the positions of the joint can be calculated using the following formula:

$$q(t) = \int_0^t \dot{q}(\zeta) d\zeta + q(0) \quad (3.37)$$

This integral can be solved using numerical techniques in discrete as Euler's integration method. Given an integration interval δt , if the positions and velocities of the joints at time t_k are known, the positions of the joints at time $t_{k+1} = t_k + \delta t$ are:

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k)\delta t \quad (3.38)$$

Then, considering $\dot{q}(t_k)$ in discrete form:

$$q(t_{k+1}) = q(t_k) + J^{-1}(q(t_k))v_e(t_k)\delta t \quad (3.39)$$

However, some inaccuracies are introduced due to numerical integration, which affect the calculated value of the joint. As a result, we will have an error because

the derived pose does not correspond precisely with the desired one. A solution to this problem is to use an algorithm that calculates \dot{q} as a function of e so that the error asymptotically converges to zero. First of all, the operating space error is defined as the difference between the effective pose x_e and the desired end-effector pose x_d .

$$e = x_d - x_e \quad (3.40)$$

Deriving the above equation:

$$\dot{e} = \dot{x}_d - \dot{x}_e \quad (3.41)$$

Finally, taking into account the differential kinematic equation and combining it with the previous formula:

$$\dot{e} = \dot{x}_d - J(q)\dot{q} \quad (3.42)$$

In this algorithm, operational spatial quantities are used in the process. Therefore, the algorithm requires the analytical Jacobian, not the geometric Jacobian. Nevertheless, in our case, it was possible to use the previously calculated Geometric Jacobian because it and the Analytical Jacobian are identical concerning the linear velocity part, while they only differ for rotation. The first part taken into consideration, which partially models the *CNC* machine, is composed, as already seen, of two prismatic joints. Therefore, since they are prismatic joints, there are only linear movements and speeds, so, the calculated geometric Jacobian concerns only the linear velocity.

Considering that the Jacobian is invertible:

$$\dot{q} = J^{-1}(q)(\dot{x}_d + Ke) \quad (3.43)$$

It possible to obtain the following linear system:

$$\dot{e} + Ke = 0 \quad (3.44)$$

If K is a positive definite diagonal matrix, the system defined above is asymptotically stable. The speed of convergence of the error depends on the values of the eigenvalues of the K -matrix, with larger values of the eigenvalues, the velocity of convergence will be faster. The scheme of the algorithm is implemented in discrete time, so, there is an upper bound for the eigenvalues. In particular, the asymptotic stability of the system is guaranteed for values of eigenvalues under the upper bound. Moreover, this upper bound depends on the sampling time. Finally, Figure 3.12 shows the block diagram of this algorithm.

In this thesis, to develop and obtain a simulation of the first part of *CNC* machine, the block scheme of the system was implemented on Matlab and Simulink, as shown in figure 3.13. Analyzing the scheme, there are different blocks: 3.13.

- *Inverse Jacobian block*: computes at each instant of time the inverse of the Jacobian matrix because it depends on the pose of manipulator;

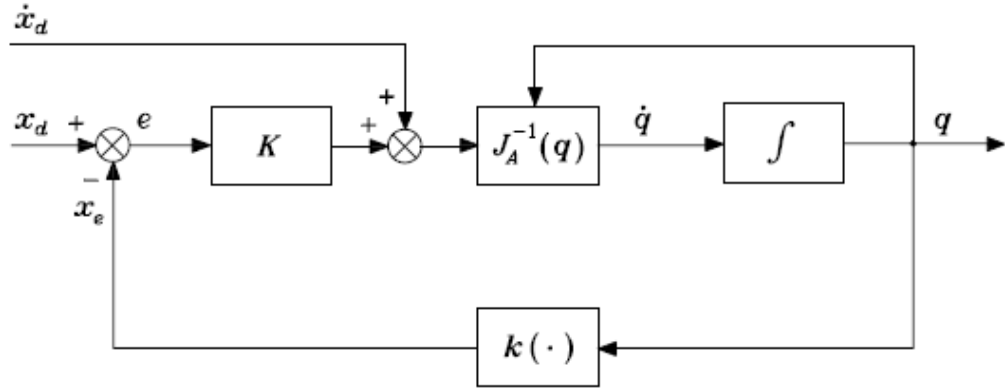


Figure 3.12: Block scheme of the inverse kinematics algorithm

- *Direct Kinematics block:* using the homogenous transformation matrix, the effective position of the end effector is computed on the basis of joint variables in time $q(t)$.
- *Integrator block:* contains the information on the initial position and joint variable

Finally, the desired position x_d and velocity \dot{x}_d are the input of the system, while the output is the effective position of the end effector.

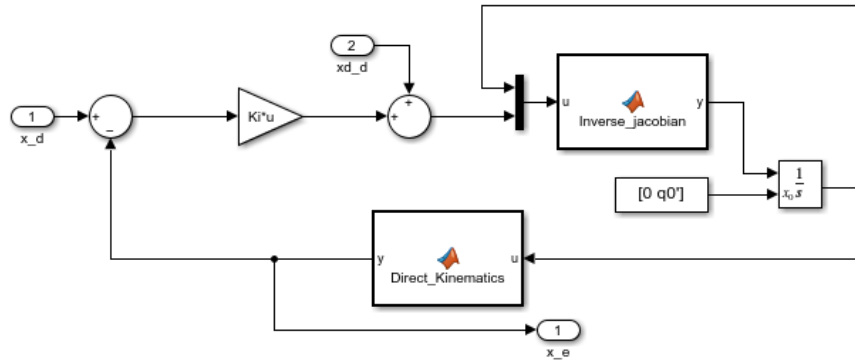


Figure 3.13: Block scheme of the inverse kinematics algorithm on MATLAB

Chapter 4

Trajectory Planning

In the previous chapter, it was analyzed the two parts of the model of the *CNC* machine considered. In particular, as was explained in the paragraph related to the inverse kinematic, it is necessary to provide the desired position and velocity of the end effector as inputs to The inverse kinematics algorithm implemented on Simulink. Therefore, this chapter will illustrate how to generate these inputs through trajectory planning. Referring to the literature, first, there is an introduction to the concept of trajectory, then an introduction to generate a trajectory, especially how to generate rectilinear or circular trajectories. Finally, the results obtained by applying these inputs to the model of the first part of the *CNC* machine are shown below.

4.1 Trajectory

The minimum requirement for a manipulator is to move from an initial pose to an assigned final pose through a smooth transition. The latter should be characterised by laws of motion, which allow the actuators to generate generalised joint forces that both do not exceed the physical limits of the actuators and do not excite the resonant modes of the structure. For this reason, it is necessary to have regular trajectories. Often, the two terms path and trajectory are used synonymously. In reality, there is a difference between the two terms. The term path means the set of points that the manipulator follows to arrive at the desired final pose. Instead, the term trajectory means a path combined with a temporal law, specifying position and velocity acceleration for each point. The trajectory planning consists of generating a temporal sequence of values taken by a given function, often a polynomial function, to interpolate the desired trajectory. To obtain this trajectory, the user provides important input parameters, such as the start and endpoints, the type of function used and the kinematic constraints, i.e. speed and maximum acceleration. Therefore, the algorithm underlying the planning of the trajectory is

shown in figure 4.1. [17]



Figure 4.1: Schematic representation of trajectory planning

Finally, there are two types of trajectories:

- **Joint space trajectory:** the values of the joint variables are calculated during the motion of the end effector by the initial pose and the end pose, which are defined by the user in the operational space. Then, it is necessary to use the Kinematic algorithm to generate the function $q(t)$, which interpolates the assigned values for the joint variables respecting the constraints. Moreover, it's necessary to say that the position and velocity must be linear function. The advantages from planning trajectory in joint space are
 1. Possibility to solve in directly way the problems linked to kinematic singularities;
 2. it is often requested low computational effort
- **Operational space trajectory:** the trajectory is planned directly in the operational space when the end effector must follow a specific path. Trajectory planning can be done in two ways:
 1. Interpolation of a sequence of prescribed points that defines the path for me.
 2. By generating the analytical motion primitive and its relative trajectory in a pointwise manner.

However, planning a trajectory in operational space is more complex than planning one in joint space. Moreover, in this case, the end effector follows the prescribed value of position and velocity.

Referring to our system, the trajectory in the operational space was chosen among the two types because, during the cutting process, the CNC machine follows

an exact path defined by the user. Moreover, it was possible to develop a simulator generically for future analysis considering other types of CNC machines. Finally, it was also possible to set the end-effector velocity during the process. However, since the first part of the CNC machine results easy, it is also possible to obtain good results using the trajectory in joint space.

4.1.1 Operational space trajectory

As a first step to planning a trajectory in the operational space, it is relevant to introduce the concept of a parametric description of the paths in the space. Consider a vector \mathbf{p} (3x1) and a parametric function $\mathbf{f}(\sigma)$ defined in the interval $[\sigma_i, \sigma_f]$. Consider the following equation: [17]:

$$p = f(\sigma) \quad (4.1)$$

The sequence of values, assumed by p in the function of σ , is defined path Γ in space. Thus, the previous equation defines the parametric representation of the path Γ for the scalar σ , called "parameter". In particular, when σ increases, the point p moves along the path according to the direction induced by the parametric representation. The path is said to be closed when the initial and final points coincide, otherwise it is said to be open.

Considering a path Γ with a fixed direction, the point p_i of the open path and the generic point p , it is possible to define the arc length of the point p as the arc length having as extremes p_i and p . In particular, its value is positive if p follows the fixed direction of the path. Otherwise, its value is negative (figure 4.2). Moreover, the point p_i is called the origin of the arc length, namely when s is equal to zero. So, using the concept of arc length s , it is possible to obtain a different parametric representation of the path Γ :

$$p = f(s) \quad (4.2)$$

In light of the previous parametric representation, the point \mathbf{p} with arch length s allows to define of three unitary vector, which are:

- **Tangent unit vector \mathbf{t} :** is a tangent vector to the path passing through p with direction oriented along the path direction
- **Normal unit vector \mathbf{n} :** is a vector, normal to vector \mathbf{t} , oriented along the line that intersects p and it stays in the osculator plane (figure 4.2);
- **Binormal unit vector \mathbf{b} :** is a vector defined to obtain a right-hand reference frame.

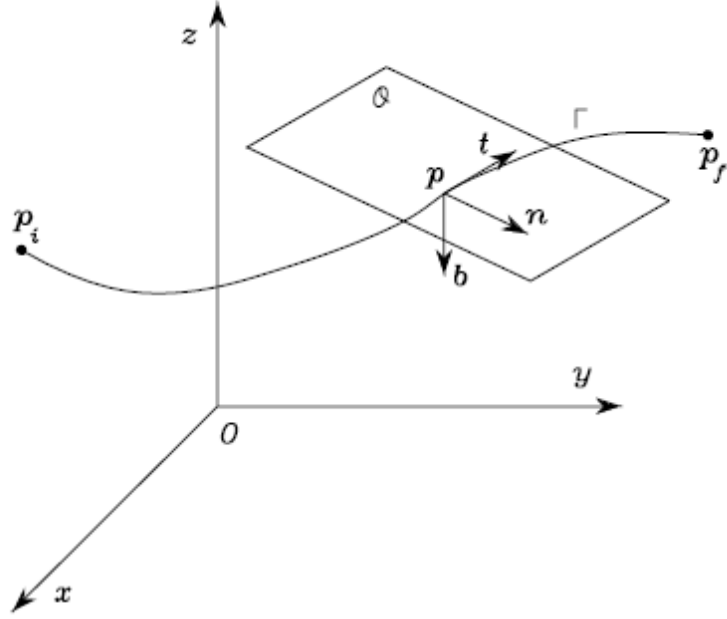


Figure 4.2: Parametric representation of a path in the space

Finally, the three unitary vectors are related to the path Γ through the following simple relations in the function of the arc length s :

$$t = \frac{dp}{ds} \quad (4.3)$$

$$n = \frac{1}{\left\| \frac{d^2p}{ds^2} \right\|} \frac{d^2p}{ds^2} \quad (4.4)$$

$$b = t \times n \quad (4.5)$$

So, as described before, with the trajectory in operational space, a function $p(t)$ is generated through the end effector can achieve the final desired pose in time t_f , through a determined path following a motion law. Finally, considering the formula (4.2) with extremes the initial point and final point, and the arch length s we obtain a path described by the temporal law $s(t)$.

The path, followed by the end effector, is defined by the user based on some specific constraints or work requirements.[17]

In this thesis, we used the timing law with a trapezoidal velocity profile.

4.1.2 Timing law with trapezoidal velocity profile

Among the various temporal laws, the timing with trapezoidal velocity profile is the one more used in the industry field for its advantages. Moreover, it allows controlling the cruise velocity of the manipulator. For this reason, we choose to adopt this timing law in the thesis because the tool has a constant cut velocity during the cutting process.

Referring to figure 4.4, assuming that the initial velocity is zero, we have:

- From the instant 0 to t_c there is a constant acceleration, so the velocity increase linearly until to achieve the set cruise velocity at the time ($t = t_c$);
- In the time interval between $t_c < t < t_f - t_c$ the velocity is constant and equal to the set cruise control while the acceleration is null.
- In the time interval between $t_f - t_c$ and t_f , there is a constant deceleration, so the velocity decrease linearly from the cruise velocity to zero ($t = t_f$).

Moreover, the two phases of acceleration and deceleration have the same time duration, producing a function with a symmetric profile. Finally, the timing law with trapezoidal law is described by the following relationships: [17].

$$s(t) = \begin{cases} \frac{1}{2}\ddot{p}_c t^2 & 0 \leq t \leq t_c \\ \ddot{p}_c t_c (t - \frac{t_c}{2}) & t_c < t \leq t_f - t_c \\ s_f - \frac{1}{2}\ddot{p}_c (t_f - t)^2 & t_f - t_c < t \leq t_f \end{cases} \quad (4.6)$$

Where \ddot{p}_c is the modulus of the two phases of acceleration and deceleration and s_f is the arch length in instant time $t = t_f$. After the following parameters are defined:

- Initial and final position;
- Cruise velocity;
- Time interval $[0, t_f]$

It is possible to compute the two parameters p_c and t_c . Finally, to verify that the chosen values are acceptable, it is necessary to check the following constraint:

$$\frac{\|s_f\|}{t_f} < |\dot{p}_c| \leq 2 \frac{\|s_f\|}{t_f} \quad (4.7)$$

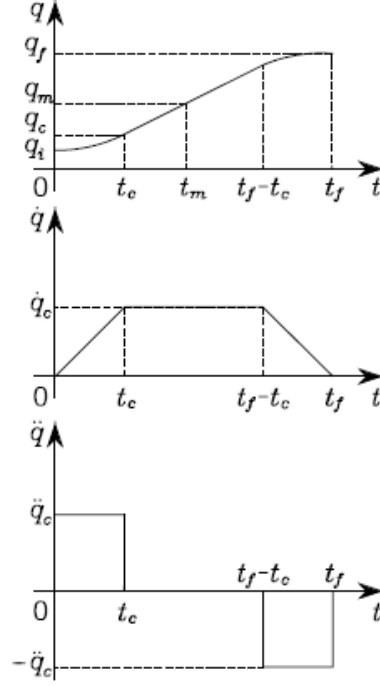


Figure 4.3: Position, velocity and acceleration of the timing law with trapezoidal velocity profile

If the constraint 4.7 is satisfied, it is possible to compute \ddot{p}_c and t_c as:

$$t_c = \frac{\dot{p}_c t_f - s_f}{\dot{p}_c} \quad (4.8)$$

$$\ddot{p}_c = \frac{\dot{p}_c^2}{\dot{p}_c t_f - s_f} \quad (4.9)$$

After that, using these computed values, it is possible to obtain the polynomials that describe the timing law (4.6). Instead, if the constraint 4.7 is not satisfied, we don't obtain a function that has the desired behaviour. In this case, to achieve the desired behaviour it is necessary to change the values for satisfying the constraints. In this thesis, referring to the concept of the literature presented briefly, it was possible to plan a trajectory considering two types of different paths, respectively the linear and circular.

4.2 Rectilinear trajectory

In this paragraph, it is used a linear trajectory to link the initial point p_i to the final point p_f . The its parametric representation is: [17]

$$p(s) = p_i + \frac{s}{\|p_f - p_i\|} (p_f - p_i) \quad (4.10)$$

Assuming that $p(0) = p_i$ and $\|p_f - p_i\|, p(\|p_f - p_i\|) = p_f$. After that, deriving the relation 4.10, we obtain the velocity, that is:

$$\dot{p} = \frac{\dot{s}(p_f - p_i)}{\|p_f - p_i\|} = \dot{s}t \quad (4.11)$$

Where t is the tangent unit vector.

Referring to the timing law with trapezoidal velocity profile and considering the time instant $t=t_f$, the arch length is $s_f = \|p_f - p_i\|$. So, the constraint to be checked for computing p_c and t_c is:

$$\frac{\|p_f - p_i\|}{t_f} < |\dot{p}_c| \leq 2 \frac{\|p_f - p_i\|}{t_f} \quad (4.12)$$

If the constraint is satisfied, it possible to compute t_c and \ddot{p}_c as:

$$t_c = \frac{\dot{p}_c t_f - \|p_f - p_i\|}{\dot{p}_c} \quad (4.13)$$

$$\ddot{p}_c = \frac{\dot{p}_c^2}{\dot{p}_c t_f - \|p_f - p_i\|} \quad (4.14)$$

After the computation of t_c and \ddot{p}_c , it is possible to obtain the following polynomials:

$$s(t) = \begin{cases} \frac{1}{2} \ddot{p}_c t^2 & 0 \leq t \leq t_c \\ \ddot{p}_c t_c (t - \frac{t_c}{2}) & t_c < t \leq t_f - t_c \\ \|p_f - p_i\| - \frac{1}{2} \ddot{p}_c (t_f - t)^2 & t_f - t_c < t \leq t_f \end{cases} \quad (4.15)$$

Finally, these relationships were implemented on Matlab to see if the obtained results from the simulation coincided with the ones present in the literature. In particular, two examples were done: planning of linear trajectory in 2D and 3D.

4.2.1 Rectilinear trajectory planning in 2-D

In this paragraph, there is an example of the planning of linear trajectory in 2D. In particular, a 2D trajectory was considered because, as mentioned before, the first part of the *CNC* machine is composed of two prismatic joints. Therefore, the allowed movements of the end effector are along the two directions x and z. Referring to the literature, the following parameters, shown in table 4.1, are used to plan the trajectory.

| | |
|-------------|----------|
| p_i | (6, 8) |
| p_f | (10, 13) |
| t_i | 0 s |
| t_f | 2 s |
| \dot{p}_c | 4 cm/s |

Table 4.1: Chosen parameters for planning a 2-D rectilinear trajectory

Through these parameters, it was possible to compute t_c and \ddot{p}_c , which are:

$$t_c = 0.3992s \quad (4.16)$$

$$\ddot{p}_c = 10.0196[cm/s^2] \quad (4.17)$$

After that, the following polynomials were obtained, with which it was subsequently possible to derive $p(t)$ and $\dot{p}(t)$:

$$s(t) = \begin{cases} 5.0098t^2 & 0 \leq t \leq 0.3992 \\ 4(t - 0.1996) & 0.3992 < t \leq 1.6008 \\ 6.4031 - 5.0098(2 - t)^2 & 1.6008 < t \leq 2 \end{cases} \quad (4.18)$$

Finally, all these part was implemented on Matlab, obtaining the following results shown in the following figure:

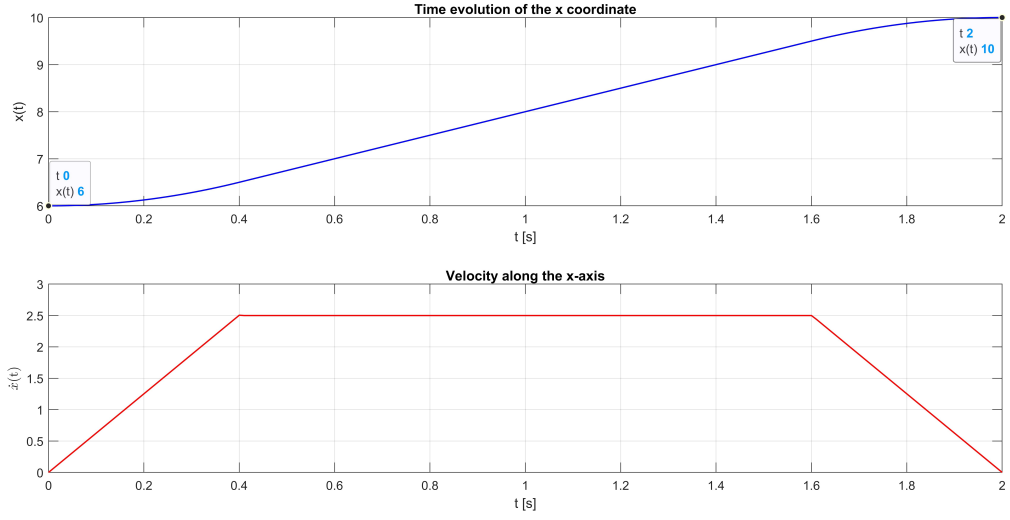


Figure 4.4: 2D rectilinear trajectory planning: time evolution of the x coordinate and velocity along the x axis.

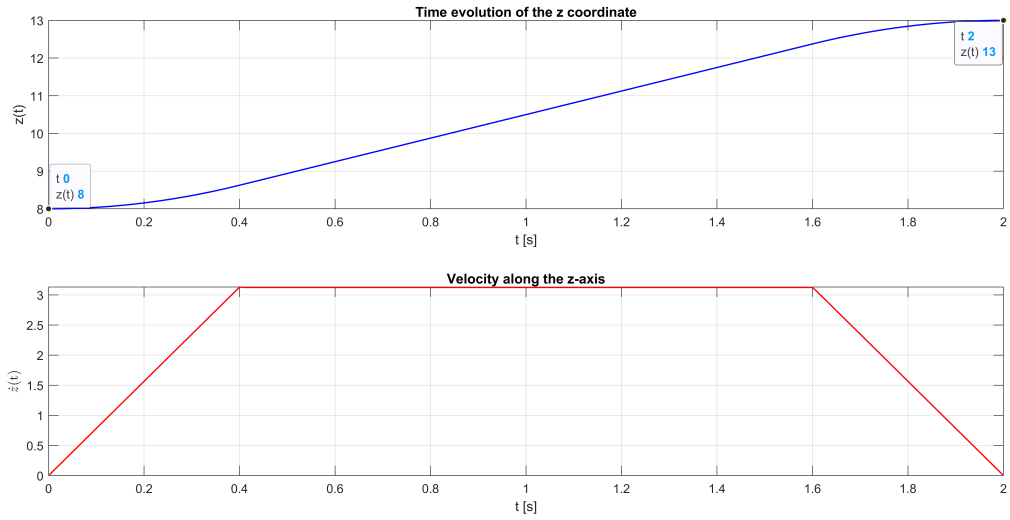


Figure 4.5: 2D rectilinear trajectory planning: time evolution of the z coordinate and velocity along the z axis.

As shown in figure 4.4 and 4.5, each coordinate reaches the given final value with $p_f = (x_f, z_f)$. Instead, the profile of the two velocities is trapezoidal. Finally, the figure 4.6 shows time evolution of z coordinate in comparison

to the x coordinate, that describes a rectilinear trajectory, and the time evolution of velocity with trapezoidal profile.

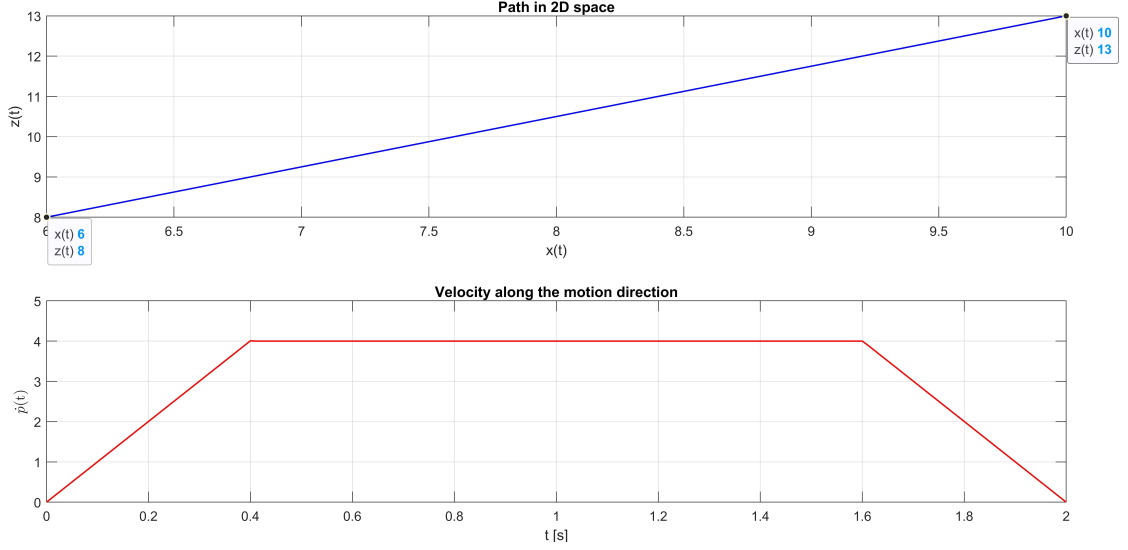


Figure 4.6: 2D Rectilinear planned path and time evolution of the velocity along the direction of motion.

4.2.2 Rectilinear trajectory planning in 3-D

Finally, the algorithm, implemented on Matlab for planning trajectories, was also tested in 3D. In this way, the simulator can be used for cases 2D and 3D in the future. Similarly to the case 2D, it was necessary to define the parameters shown in tab 4.2:

After that, the parameters p_c e t_c were obtained:

$$t_c = 0.3267[s] \quad (4.19)$$

$$\ddot{p}_c = 15.3055 \frac{cm}{s^2} \quad (4.20)$$

| | |
|-------------|-------------|
| p_i | (5, 8, 5) |
| p_f | (10, 14, 8) |
| t_i | 0 s |
| t_f | 2 s |
| \dot{p}_c | 5 cm/s |

Table 4.2: Parameters employed in a 3-D rectilinear trajectory planning

Finally, the arch length is:

$$s(t) = \begin{cases} 7.6528t^2 & 0 \leq t \leq 0.3267 \\ 5(t - 0.1633) & 0.3267 < t \leq 1.6733 \\ 8.3666 - 7.6528(2 - t)^2 & 1.6733 < t \leq 2 \end{cases} \quad (4.21)$$

Using the obtained polynomials, it was possible to compute $p(t)$ and $\dot{p}(t)$. Figure 4.7, 4.8 and 4.9 show the results obtained, such as the time evolution of the three coordinates (x, y, z) and the time evolution of the velocity along the respective axes.

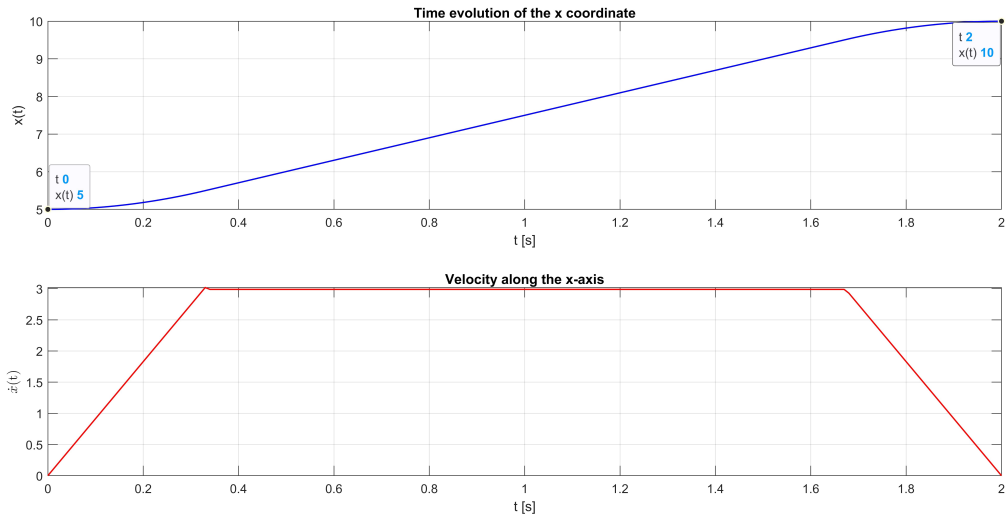


Figure 4.7: 3D rectilinear trajectory planning: time evolution of the x coordinate and velocity along the x axis.

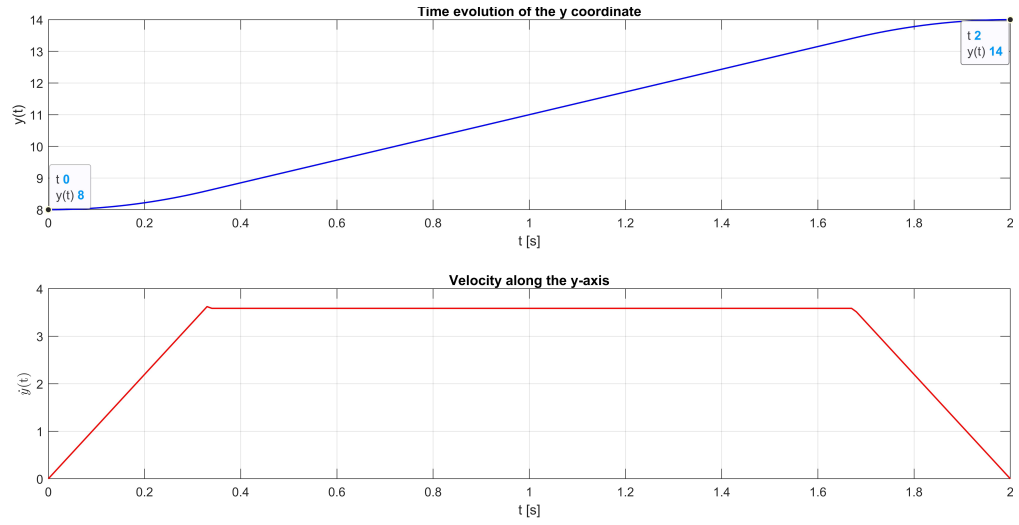


Figure 4.8: 3D rectilinear trajectory planning: time evolution of the y coordinate and velocity along the y axis.

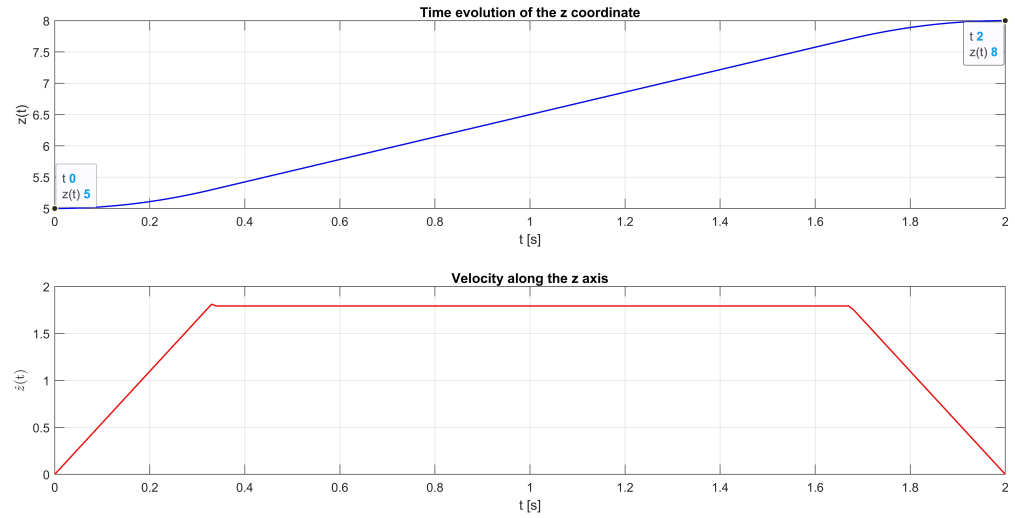


Figure 4.9: 3D rectilinear trajectory planning: time evolution of the z coordinate and velocity along the z axis.

Finally, figure 4.10 displays the 3D rectilinear path between the initial point and the final desired point of the end effector, As expected, the evolution of the velocity is shown along the motion's direction with a trapezoidal profile.

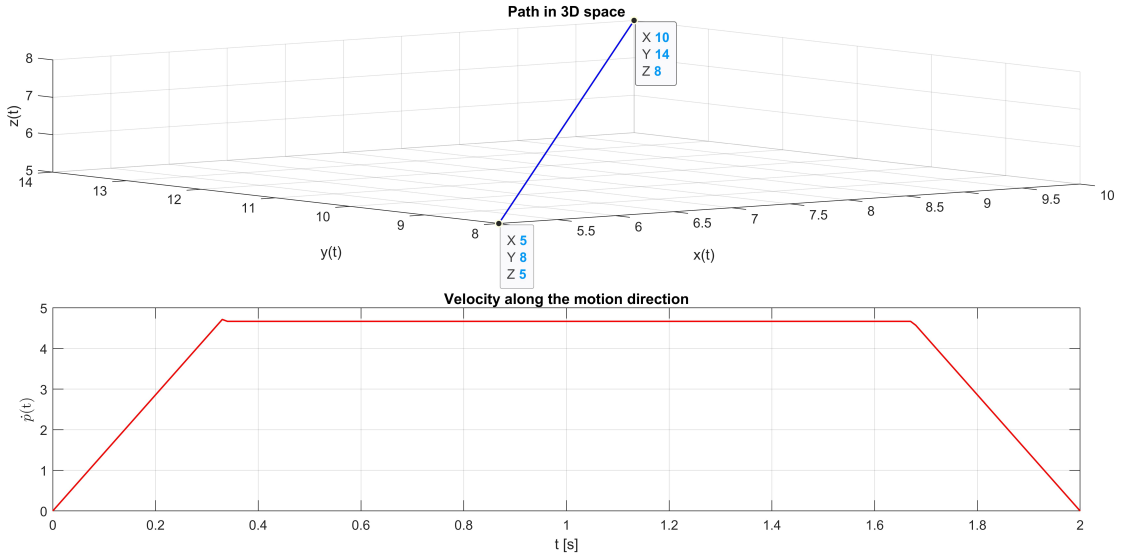


Figure 4.10: 3D Rectilinear planned path and time evolution of the velocity along the direction of motion.

Considering the *CNC* machine considered in this thesis, the algorithm implemented on Matlab worked correctly both for 2D and 3D cases. For this application, it only handled the end-effector position and not the orientation. Thus, if in the future other types of *CNC* machines are considered, the algorithm will be updated to also manage the aspect linked to the orientation of the end effector for the planning a trajectory.

4.3 Circular trajectory

Another type of trajectory is circular, which will be exposed briefly to the theory present in literature [17]. As shown in figure 4.11, to define a circle Γ in the space it's necessary to define the following parameters:

- The unitary vector r along the circumference axis
- The position vector d of a point along the circumference axis
- The position vector p of a point on the circumference

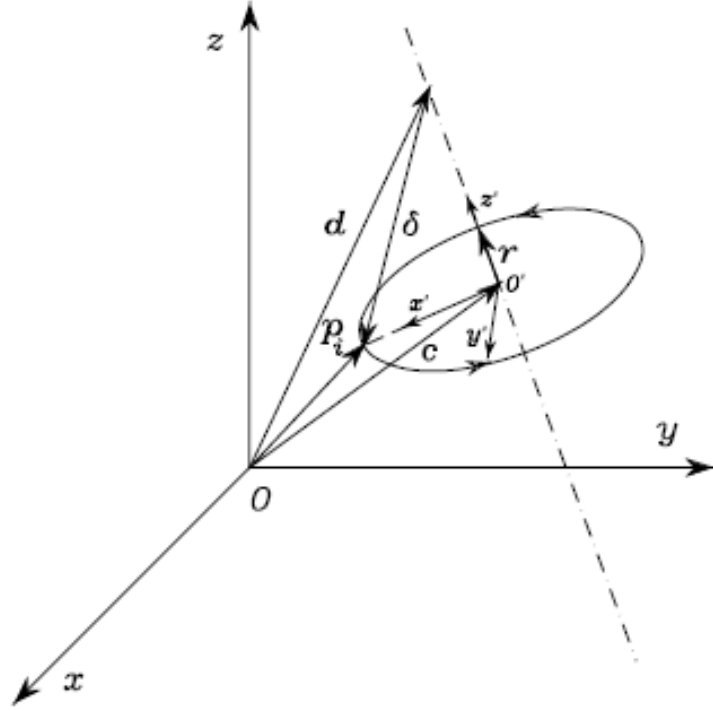


Figure 4.11: Parametric representation of a circumference in the space

The point p_i must not be on the axis so that the circumference does not degenerate into a point. Hence, we have the following constraint, which must be satisfied for planning a circular trajectory:

$$|\delta^T r| < \|\delta\| \quad (4.22)$$

where $\delta = p_i - d$.

If the constraint is satisfied, it is possible to compute the centre of the circumference c as:

$$c = d + (\delta^T r)r \quad (4.23)$$

After obtaining the centre of the the circumference c , the radius ρ can be computed as: Knowing the position vector c , the radius ρ is computed:

$$\rho = \|p_i - c\| \quad (4.24)$$

As the last step, for obtaining its parametric representation, as shown in figure 4.11, the reference system $O'-x'y'z'$ is defined in the following way:

- X' -axis oriented along the direction of the vector $(\pi-c)$;
- Z' -axis on the direction of vector r ;
- Y -axis was chosen in a way to have a right-handed frame compared to the base frame.

Then, to express the unitary vectors compared to the base frame, the following rotation matrix is introduced, that refers the frame $O'-x'y'z'$ with respect to $O-X Y Z$ frame:

$$R = [x' y' z'] \quad (4.25)$$

It is relevant to say that x', y' and z' are the unit vector of the frame $O'-x'y'z'$ expressed in the frame $O-X Y Z$. Finally, referring to the all elements just described, the path representation $p(s)$ is:

$$p(s) = c + R p'(s) = c + R \begin{bmatrix} \rho \cos\left(\frac{s}{p}\right) \\ \rho \sin\left(\frac{s}{p}\right) \\ 0 \end{bmatrix} \quad (4.26)$$

Finally, deriving the formula 4.26 with respect to s , the velocity is described by the following formula:

$$\dot{p} = \frac{dp}{ds} = R \begin{bmatrix} -\dot{s} \sin\left(\frac{s}{p}\right) \\ \dot{s} \cos\left(\frac{s}{p}\right) \\ 0 \end{bmatrix} \quad (4.27)$$

Lastly, similarly to what has been done for the rectilinear trajectory, these relationships were implemented in Matlab in time-discrete. Afterwards, two examples have been carried out to verify the correctness of the implementation, which are:

- Planning of circular trajectory in 2D;

- Planning of circular trajectory in 3D.

4.3.1 Circular trajectory planning in 2-D

In this example, there is the planning of a circular trajectory in the plane xz because the cutting tool of the *CNC* machine, considered in this thesis, can move only in this plane. For this reason, first, the chosen parameters are shown in tab 4.3 below:

| | |
|----------------------|------------------|
| \mathbf{r} | $[0 \ 1 \ 0]^T$ |
| \mathbf{d} | $[4 \ -5 \ 4]^T$ |
| \mathbf{p}_i | $[6 \ -6 \ 4]^T$ |
| t_i | 0 s |
| t_f | 2 s |
| $\dot{\mathbf{p}}_c$ | 7 cm/s |

Table 4.3: Parameters employed in a 2-D circular trajectory planning

After the choice of the parameters, it was possible to compute the position vector of centre of the circumference, the radius ρ , the time t_c the acceleration $\ddot{\mathbf{p}}_c$ as:

$$\mathbf{c} = [4 \ -6 \ 4]^T$$

$$\rho = 2$$

$$t_c = 0.2048 \text{ s}$$

$$\ddot{\mathbf{p}}_c = 34.1790 \text{ cm/s}^2$$

Then, it was relevant to point out that being a circular trajectory, the final arch length is $s_f = 2 \pi \rho$. Finally, the polynomials that describe the arch length $s(t)$ are:

$$s(t) = \begin{cases} 17.0895t^2 & 0 \leq t \leq 0.2048 \\ 7(t - 0.2048) & 0.2048 < t \leq 1.7952 \\ 12.5664 - 17.0895(2 - t)^2 & 1.7952 < t \leq 2 \end{cases} \quad (4.28)$$

The formula 4.28, rotation matrix R and the radius ρ were used to obtain $p(t)$ and $\dot{p}(t)$. In figure 4.12 and 4.13 the results, obtained from the simulation, are shown :

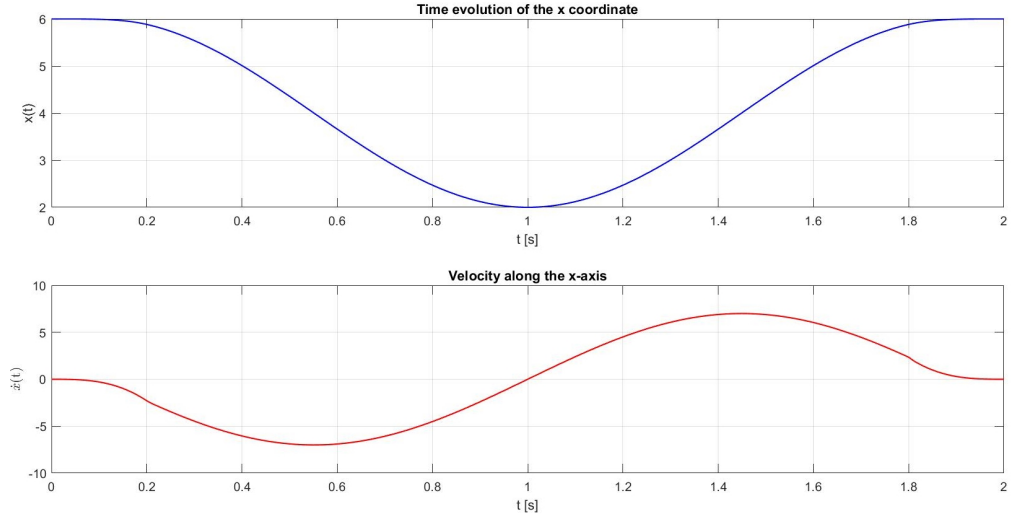


Figure 4.12: 2D circular trajectory planning: time evolution of the x coordinate and velocity along the x axis.

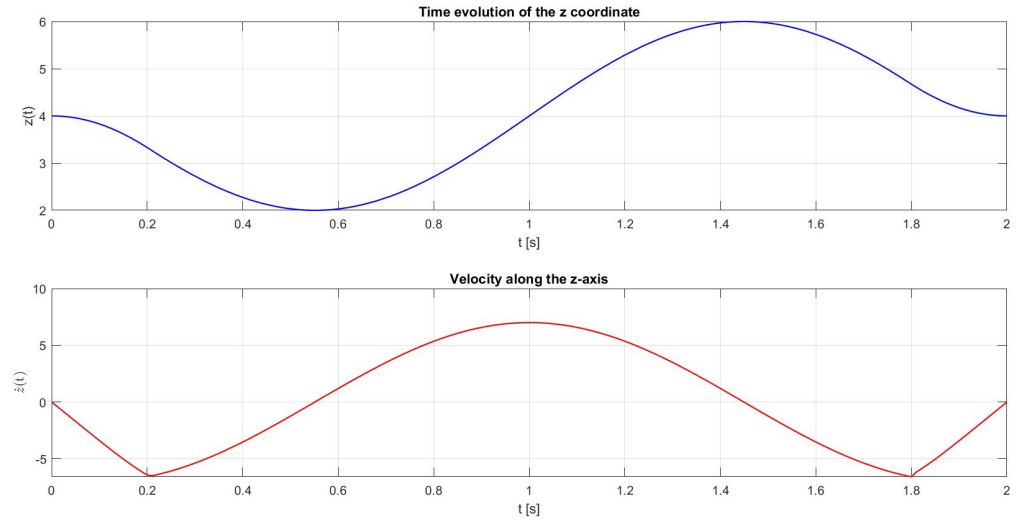


Figure 4.13: 2D circular trajectory planning: time evolution of the x coordinate and velocity along the x axis

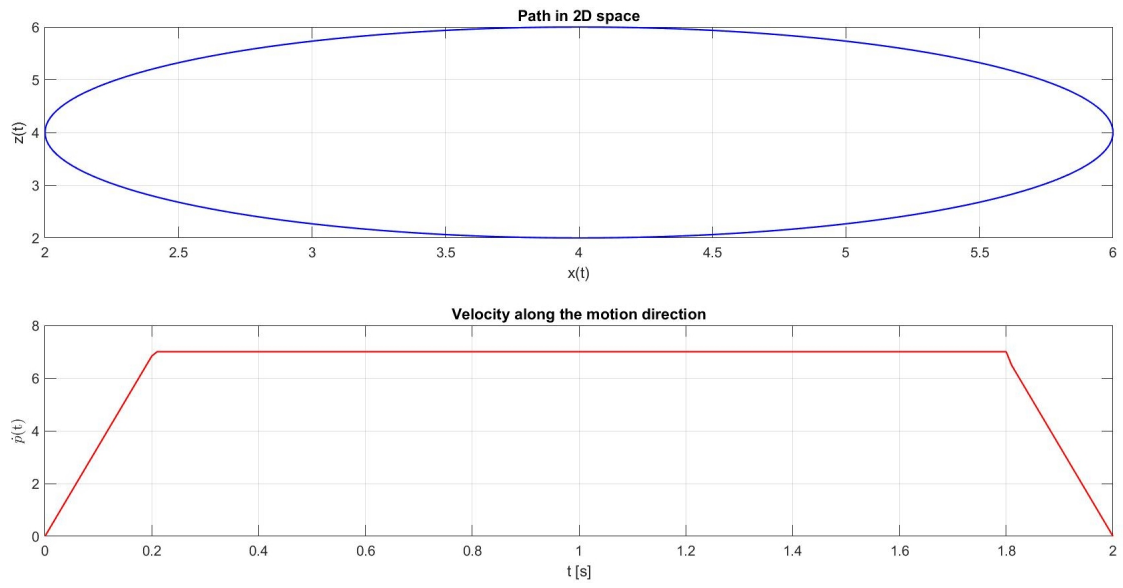


Figure 4.14: 2D circular planned path and time evolution of the velocity along the direction of motion

4.3.2 Circular trajectory planning in 3-D

In this paragraph, there is an example of the planning of circular trajectory in 3D. Following the same procedure used for the case 2D, the chosen parameters are shown in table 4.4 below:

| | |
|-------------|--------------------------------|
| r | $[0.5774 \ 0.5774 \ 0.5774]^T$ |
| d | $[3 \ 3 \ 2]^T$ |
| p_i | $[3 \ 5 \ 6]^T$ |
| t_i | 0 s |
| t_f | 2 s |
| \dot{p}_c | 10 cm/s |

Table 4.4: Parameters employed in a 3-D circular trajectory planning

After the choice of the parameters, the other obtained parameters are:

$$c=[5 \ 5 \ 4]^T$$

$$\rho = 2.8284$$

$$\ddot{p}_c = 44.8739 \text{ cm/s}^2$$

$$t_c = 0.2228 \text{ s}$$

Finally, the polynomials that describe the arch length $s(t)$ are:

$$s(t) = \begin{cases} 22.4369t^2 & 0 \leq t \leq 0.2228 \\ 10(t - 0.1114) & 0.2228 < t \leq 1.7772 \\ 17.7715 - 22.4369(2 - t)^2 & 1.7772 < t \leq 2 \end{cases} \quad (4.29)$$

As in the previous case, then $p(t)$ and $p(t)'$ are computed. The results obtained from the simulation on Matlab are shown in figure 4.15,4.16, 4.17 and 4.18 below:

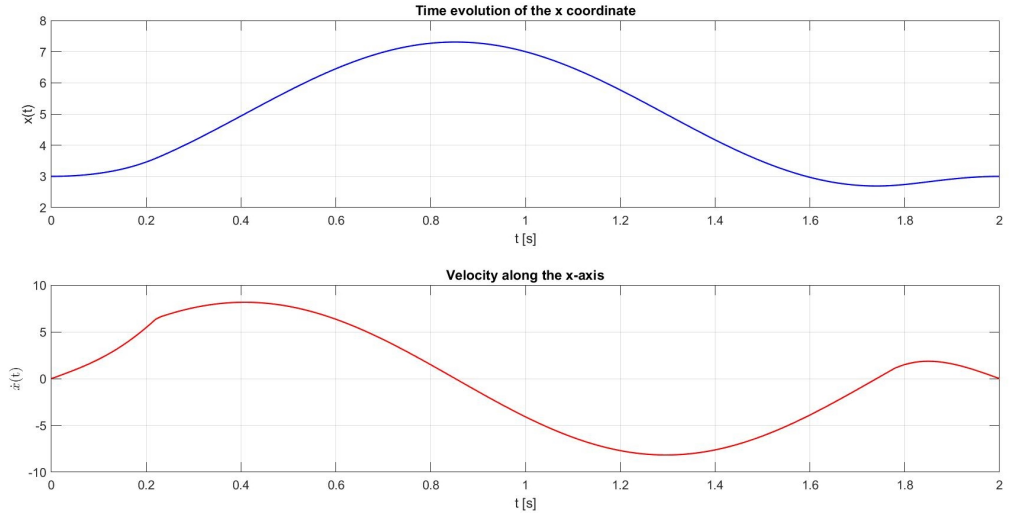


Figure 4.15: 3D circular trajectory planning: time evolution of the x coordinate and velocity along the x axis.

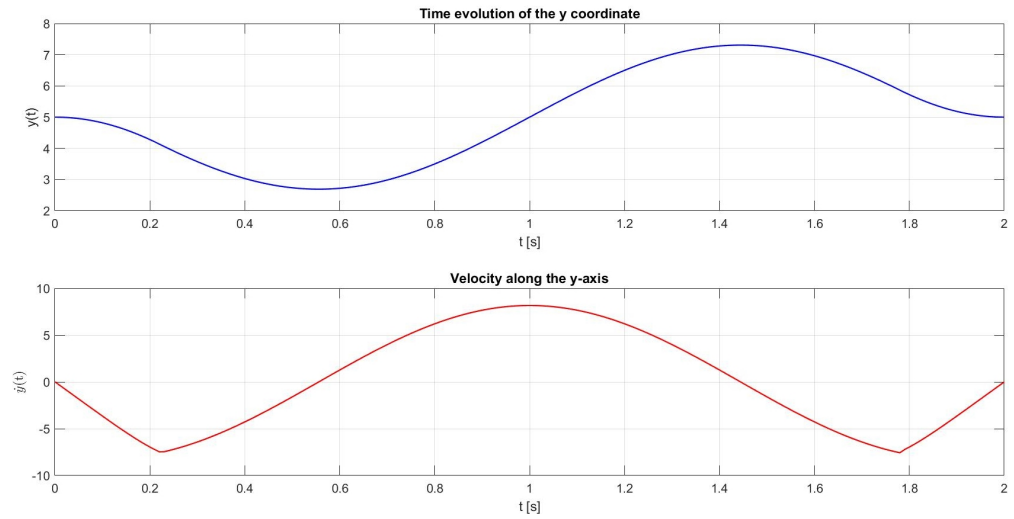


Figure 4.16: 3D circular trajectory planning: time evolution of the y coordinate and velocity along the y axis.

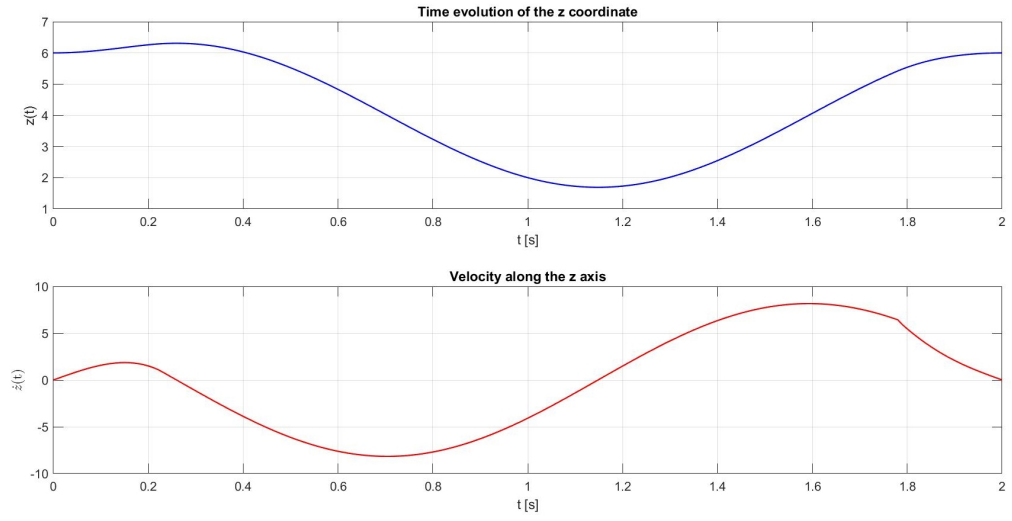


Figure 4.17: 3D circular trajectory planning: time evolution of the z coordinate and velocity along the z axis.

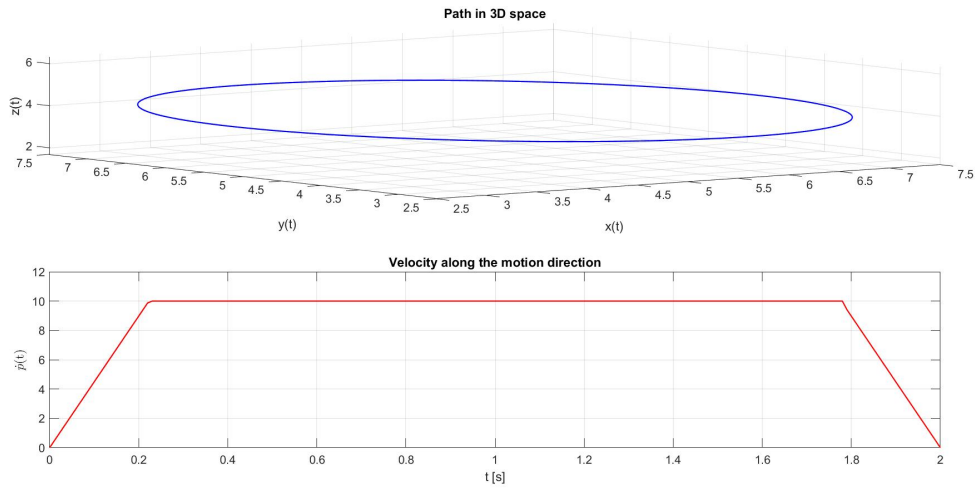


Figure 4.18: 3D circular planned path and time evolution of the velocity along the direction of motion.

Also in this case, looking at the results obtained, the algorithm implemented in Matlab works in both cases.

4.4 Test of the CNC machine model

Finally, after implementing and testing the simulators of planning of rectilinear or circular trajectories in 2D and 3D, the 2D trajectories analyzed were used as input for the first part of the model of *CNC* machine obtaining good results for both rectilinear and circular trajectory. In particular, using as input the rectilinear trajectory, the test of the model was successful and the results are shown in the following figure:

In figure 4.19 and 4.20, it is possible to observe the time evolution of the desired coordinates (x_d, z_d) and effective coordinate (x_e, z_e) of the end effector. In the figure 4.21 and 4.22, there is the representation of the time evolution of the velocity along the two axes, respectively x and z axis. Finally, similarly the previous result, in the figure 4.23 there is the time evolution of the velocities along the motion direction, respectively the desired velocity and the effective one. To verify how much the model is precise, there is a plot of the error present in the various cases.

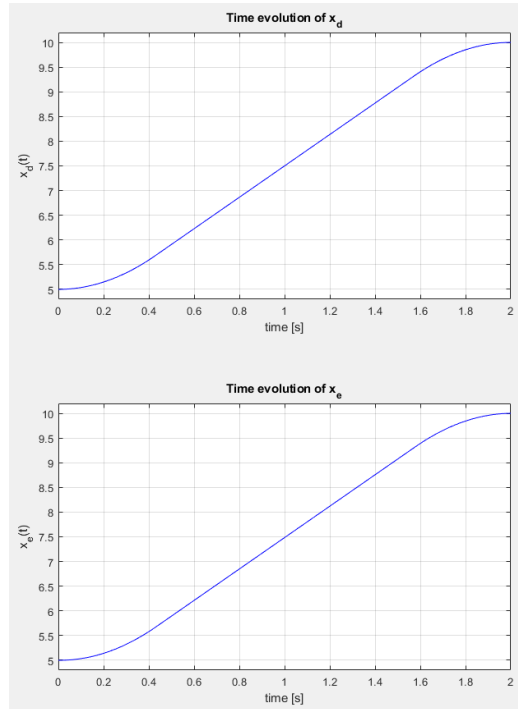


Figure 4.19: Time evolution of the desired and the effective x coordinate

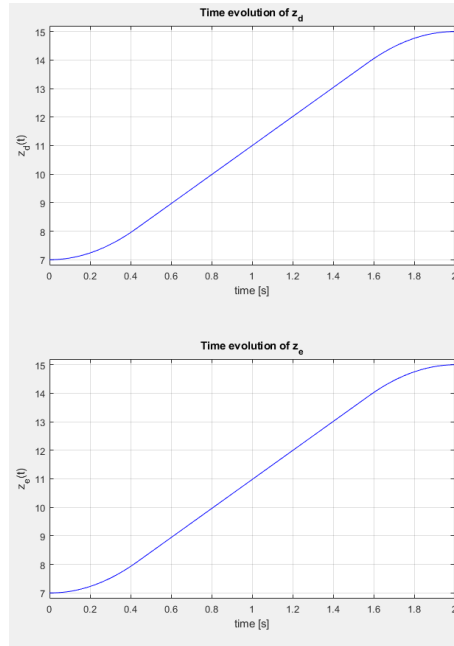


Figure 4.20: Time evolution of the desired and the effective z coordinate

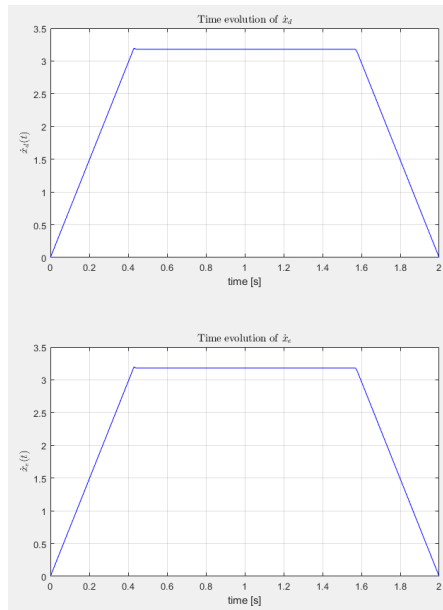


Figure 4.21: Time evolution of the desired and the effective velocity along x-axis

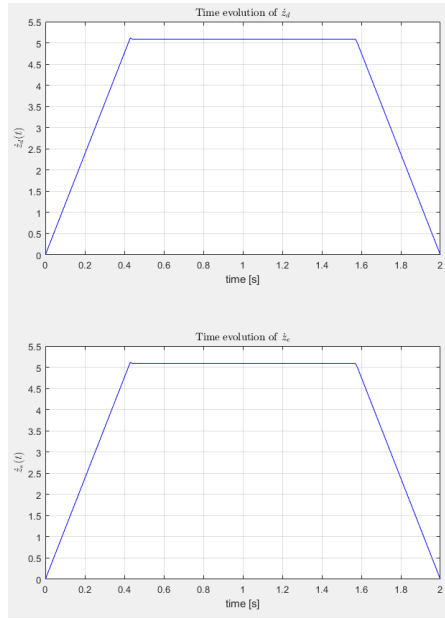


Figure 4.22: Time evolution of the desired and the effective velocity along z-axis

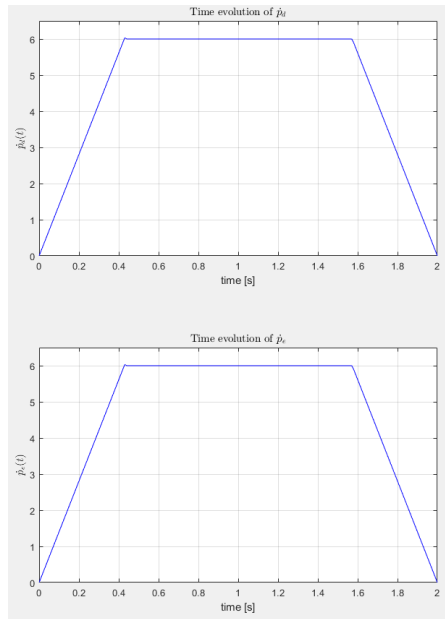


Figure 4.23: Time evolution of the desired and the effective velocity along the direction of motion

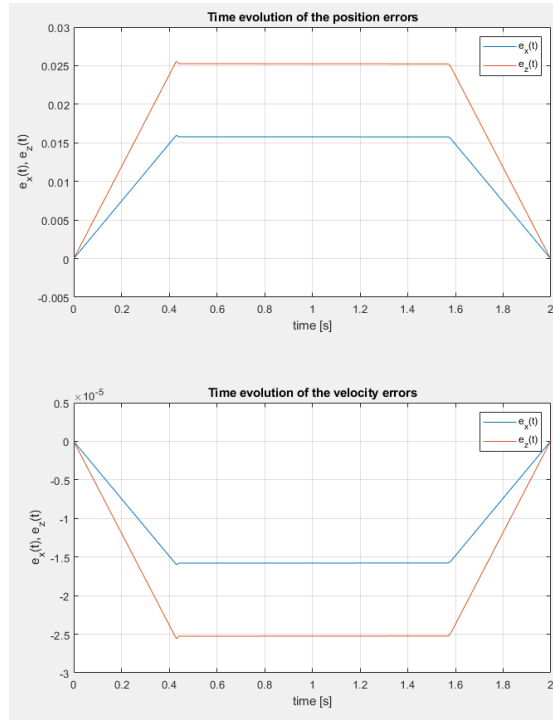


Figure 4.24: Time evolution of the position and velocity errors

As it is possible to observe in figure 4.24, the error is small enough. Thus, the model is able to follow the desired behaviour with a small margin of error.

With this part, I achieved one of the goals of this thesis, which is the realization of the model able to reproduce the movements of the *CNC* machine considered. In the future, a new control strategy can be used to simulate the movements of the *CNC* machine reducing the error.

Chapter 5

Dynamic model of shear process and new formulation of friction coefficient

Nowadays, the cutting process is very difficult to model dynamically due to the large number of variables that occur during the process. However, in this thesis work, I was able to advance a dynamic formulation of the process, referring to the theory of shear and empirical formulas found in the literature. It was possible to estimate the forces present in the shear process, which are the normal force and the cut force were derived. After that, considering the geometric factors, type of material used in the machining process and cutting parameters, it was possible to develop a formulation relating to the friction coefficient. Therefore, both to simulate and to verify that the formulation gives sensible results, two digital controllers were designed to have a constant spindle rotation speed during the simulations, as happens when using *CNC* machines for cutting operations. Finally, several tests were completed in different working conditions and a simulation was carried out on the model developed in this work following the trajectory defined in the last chapter.

5.1 Shear theory

In this paragraph, there is an introduction to the shear theory to understand the concepts used to obtain the forces that occur during the cutting process.

5.1.1 Generality

The cutting process (fig. 5.1) is a plastic deformation process where the tool, moving relative to the workpiece, removes a layer of surface material, called machining allowance. Then, it is transformed into chips to generate a surface with the specified properties. [18]

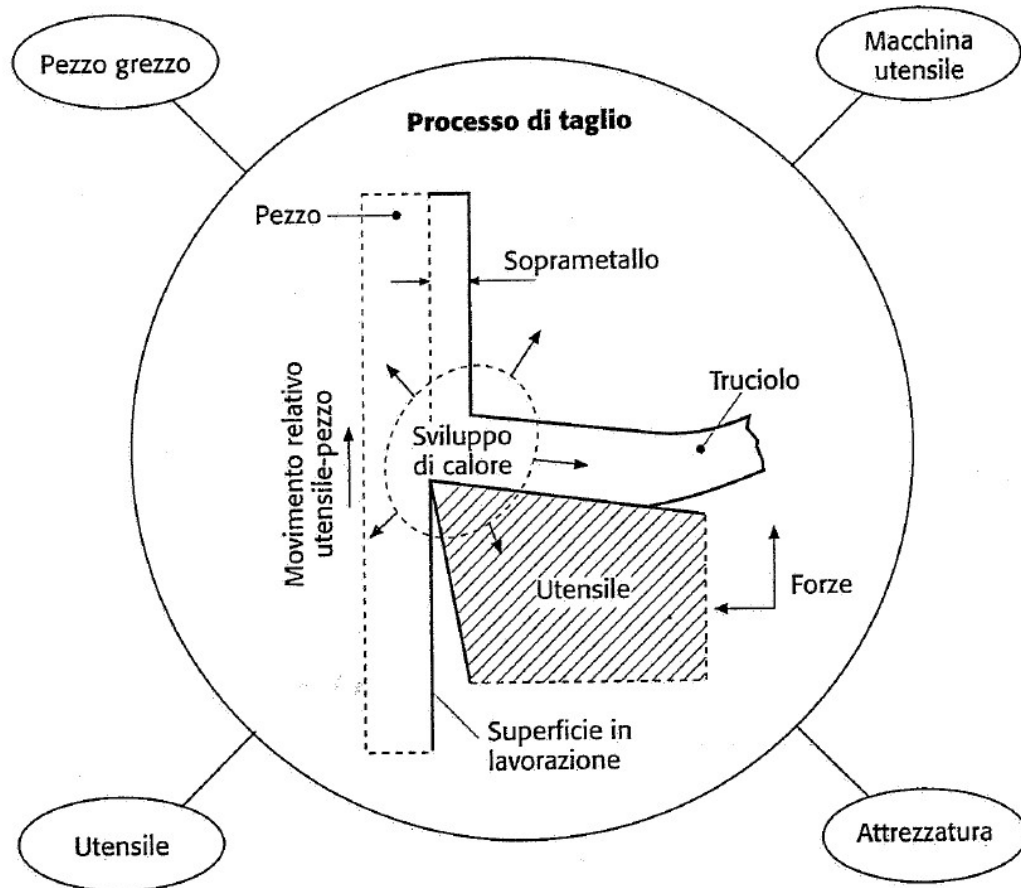


Figure 5.1: Representation and aspect of the shear process

The process occurs through the motions provided by the machine tool in condition of the ambient temperature, but during the process, a high heat is generated due to the friction and the plastic deformation. Therefore, the tool, having a certain geometry and hardness, must maintain its properties

for a certain time and, for this reason, it is composed of the material suitable. Moreover, the workpiece tends to move or deform due to solicitations that occur in the process. As result, the cutting process requires:

- A blank;
- A machine tool;
- A tool;
- An equipment.

Moreover, during the metalworking, it is possible to observe that the exported chip has both the thickness and the hardness largest than the base material. [18]

5.1.2 The orthogonal cut

It is introduced the orthogonal cut to analyse the mechanism of chip formation. In this case, it is possible to consider a vertical plane to obtain a perpendicular section to the tool's edge to have a bidimensional model of the chip formation (figure 5.2). To refer to this situation, the following assumptions are:

- Cutting edge width greater than workpiece width;
- Constant cutting speed along the cutting edge;
- Cutting edge perpendicular to the cutting speed ;
- Perfectly sharp tool (no contact in the backplane).

Referring to figure 5.2, a relevant parameter to analyse the cutting process is the cutting ratio (its inverse is called "shrinking factor"), defined as:

$$c = \frac{s}{s_1} \quad (5.1)$$

$$r = \frac{s_1}{s} \quad (5.2)$$

where:

- s_1 is the thickness of undeformed chip;

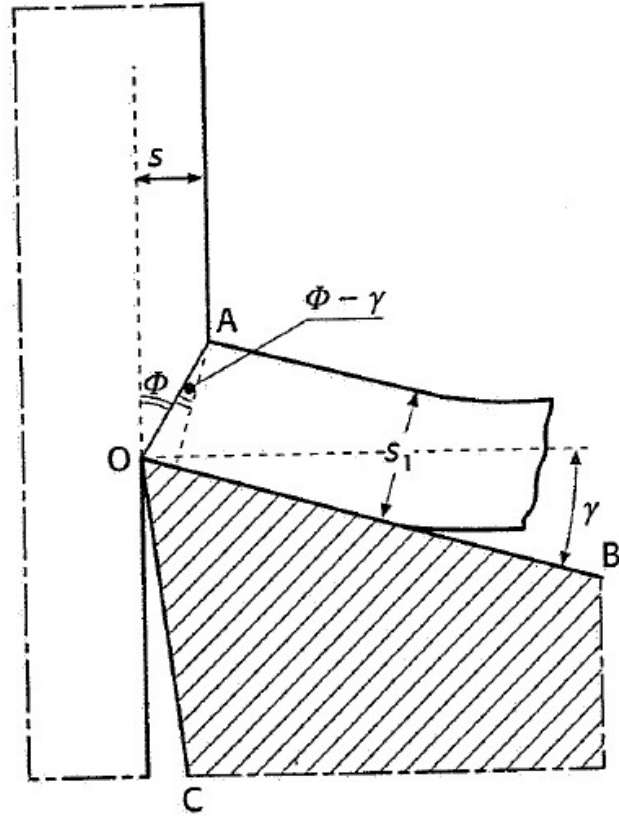


Figure 5.2: Case of orthogonal cut

- s is the thickness of the chip.

Moreover, it is possible to define the following geometric characteristics:

- The upper surface of the tool is called the breast and the surface above the machined surface is called the back;
- The principal rake angle γ is the angle between the breast and the normal to the direction of cut;
- The back rake angle α is the angle between the back and the direction of cut;
- The rake angle ϕ defines the inclination of the sliding.
- The shear angle β

Considering the scheme of the orthogonal cut, shown in figure 5.2, the following formulas are defined:

$$c = \frac{\sin(\phi)}{\cos(\phi - \gamma)} \quad (5.3)$$

$$r_c = \frac{\cos(\phi - \gamma)}{\sin(\phi)} \quad (5.4)$$

So, the sliding angle α can be expressed in function of the principal rake angle ϕ as:

$$\tan \phi = \frac{c \cos(\gamma)}{1 - c \sin(\gamma)} \quad (5.5)$$

In the case where $\gamma=0$, the cutting ratio c is equal to zero and, consequently, the sliding angle is equal to 45° degrees. [18]

5.1.3 Pijspanen Model

The Pijspanen model is used to schematize the process of chip development. The deformation occurs by the sliding of rigid blocks with parallelogram shape along the sliding plane direction. It is possible to quantify the shear deformation acting on the material in terms of sliding deformation γ_s . Considering both a rigid block with the characteristic dimension dx and the sliding defined by stretch x , we have: [18]

$$\gamma_s = \frac{\Delta s}{\Delta x} = \frac{AB}{CO} = \frac{(AO + OB)}{CO} = \frac{AO}{CO} + \frac{OB}{CO} \quad (5.6)$$

Geometrically, AO and AB are given by:

$$AO = CO \tan(\phi) \quad (5.7)$$

$$OB = CO \tan(\phi - \gamma) \quad (5.8)$$

the value of the sliding deformation is:

$$\gamma_s = \cot \phi + \tan(\phi - \gamma) \quad (5.9)$$

In particular, considering the formula 5.5, it is possible to state that:

- If γ_s increases, γ decrease. Consequently, the necessary forces for the deformation increase;

- For each γ , there is a value of ϕ for that γ_s is minimum.

Minimizing γ_s , we have:

$$\frac{\delta\gamma_s}{\delta\phi} = -\frac{1}{\sin(\phi)^2} + \frac{1}{\cos(\phi - \gamma)^2} = 0$$

$$\sin(\phi) - \cos(\phi - \gamma) = \sin(\phi) - \sin(90^\circ - \phi + \gamma) = 0$$

$$2\phi - \gamma = \frac{\pi}{2} \quad (5.10)$$

Referring to the formula 5.10, if $\gamma=0$ then we have γ_s for $\phi= 45^\circ \rightarrow c=1$. As noted, the shrinking factor is equal to one because an ideal model is used in the analysis. However, the shrinking factor cannot be one in reality.

5.1.4 Kinematic of the shear

The deformation velocity, the derivative of the shear deformation with respect to the time, is:

$$\dot{\gamma}_s = \frac{d\gamma_s}{dt} = \frac{ds}{dt} \frac{1}{\Delta x} = \frac{v_s}{\Delta x} \quad (5.11)$$

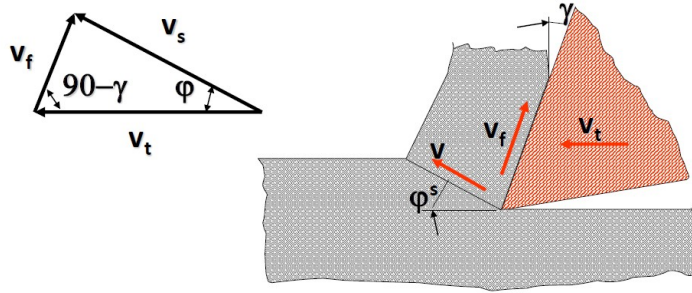


Figure 5.3: Kinematic analysis

Referring to the velocity triangular, shown in figure 5.3, it is possible to determine the shear velocity V_t and the sliding one V_s thanks to the sine's theorem:

$$\begin{aligned}\frac{V_t}{\sin \frac{(\pi}{2} + \phi - \gamma)} &= \frac{V_s}{\sin \frac{(\pi}{2} - \gamma)} = \frac{V_f}{\sin (\phi)} \\ \frac{V_t}{\cos (\phi - \gamma)} &= \frac{V_s}{\cos (\gamma)} = \frac{V_f}{\sin (\phi)} \\ V_s &= V_t \frac{\cos (\gamma)}{\cos (\phi - \gamma)}\end{aligned}\tag{5.12}$$

where:

- V_s is the sliding velocity
- V_t is the cut velocity
- V_f is the flux velocity

Considering that the tool advances gradually removing the chips, for continuity, the material flux must be constant through the section considered. Thus, from the condition of a constant volume of the chip, we have: [18]

$$\begin{aligned}V_t h_o b &= V_f h_1 b \\ \frac{V_t}{V_f} &= \frac{h_0}{h_1} = c \\ V_f &= V_t c = V_t \frac{\sin (\phi)}{\cos (\phi - \gamma)}\end{aligned}\tag{5.13}$$

5.1.5 Cutting forces (Ernst-Merchant)

The cutting forces can be analysed referring to the *Ernst-Merchant theory*. In the latter, there is an analysis of the forces that are present during the cutting process. To simplify the analysis, the *Ernst-Merchant* model is based on the following hypothesis:

- Orthogonal cut process;
- Continuous chip formation for sliding along the sliding plane;
- Absence of friction in the contact between side of tool and workpiece;

- Constant coefficient of friction between the chip and breast of the tool.

To study the cutting force, it is necessary to consider the chip section in dynamic equilibrium (figure 5.4).

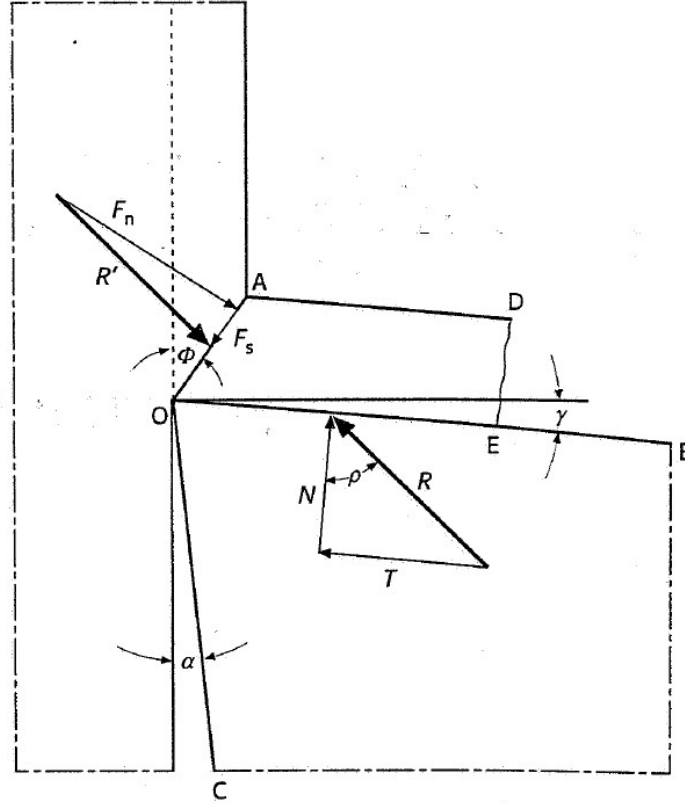


Figure 5.4: Dynamic equilibrium of chip section

In the chip section two forces act: the first one is the resulting force R applied by the tool on the workpiece, while the other one is the reaction force R' , equal and contrary, applied by the workpiece in the sliding plane. The two resulting forces R and R' lie on two parallel lines, whose distance can be neglected (in this way the chip curvature is neglected). The force R is the resulting force of the force N , normal force to the tool's breast, and the force F_t , the friction force between the chip and the tool. Instead, the force R' is the resulting force of the sliding force F_s , which is on the direction of the sliding plane, and the normal force F_n , which is orthogonal to the sliding

plane. Moreover, it is possible to decompose the resulting force R' along the cut direction obtaining F_z and F_x , which are respectively parallel and normal to cut direction. Finally, considering the resulting force R applied to the point that represents the edge of the tool, after the drawing of a circumference with diameter R , it is possible to obtain the mathematical formulas of the various force component (figure 5.5): [18]

$$F_z = R \cos (\rho - \gamma) \quad (5.14)$$

$$F_x = R \sin (\rho - \gamma) \quad (5.15)$$

$$F_s = R \cos (\phi + \rho - \gamma) \quad (5.16)$$

$$F_n = R \sin (\phi + \rho - \gamma) \quad (5.17)$$

$$N = R \cos (\rho) \quad (5.18)$$

$$T = R \sin (\rho) \quad (5.19)$$

Finally, the sliding surface results to be stressed by a normal and a tangential tension, respectively σ_s and τ_s , which are:

$$\sigma_s = \frac{F_n}{sl} \sin (\phi) = \frac{R \sin (\phi + \rho - \gamma) \sin (\phi)}{sl} \quad (5.20)$$

$$\tau_s = \frac{F_s}{sl} \sin (\phi) = \frac{R \cos (\phi + \rho - \gamma) \sin (\phi)}{sl} \quad (5.21)$$

In the *Ernst-Merchant theory*, the principle of minimum energy is used. Among the various forces, the cut force F_z is the most important force in the cut direction because it is responsible for the work done during the cutting. The cutting force necessary to cause the sliding along the sliding plane identified by γ is the smallest force that causes the chip formation.

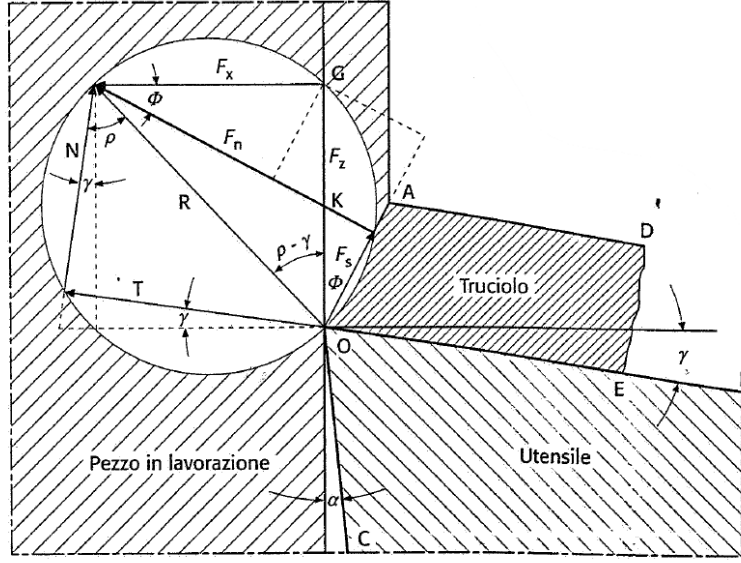


Figure 5.5: Cerchio di Merchant , scomposizione delle forze di taglio

Then, knowing the friction angle and the front rake angle, there is a value of the sliding angle that minimizes the cutting force F_z . In particular, the formula is:

$$\begin{cases} F_z = R \cos(\rho - \gamma) \\ F_s = R \cos(\phi + \rho - \gamma) \end{cases}$$

$$F_z = \frac{F_s \cos(\rho - \gamma)}{\cos(\phi + \rho - \gamma)} \quad (5.22)$$

Deriving the formula 5.22 with respect to γ and setting it equal to zero, it becomes:

$$\frac{dF_z}{d\phi} = \frac{\cos(\phi) \cos(\phi + \rho - \gamma) - \sin(\phi) \sin(\phi + \rho - \gamma)}{\sin(\phi)^2 \cos(\phi + \rho - \gamma)} = 0$$

$$\cos(\phi) \cos(\phi + \rho - \gamma) - \sin(\phi) \sin(\phi + \rho - \gamma) = \cos(\phi + \phi + \rho - \gamma) = 0$$

$$2\phi + \rho - \gamma = 90^\circ \quad (5.23)$$

Finally, deriving again the formula 5.23, the obtained relation is:

$$\phi = 45^\circ - \frac{1}{2}(\rho - \phi) \quad (5.24)$$

Referring to the formula 5.23, if the sliding angle decreases, the damping angle β , while, If the sliding angle γ increases when the rank angle ϕ increases [18].

5.1.6 Valuation of the cut force

The relations within the *Ernst-Merchant theory* are obtained considering the case of the orthogonal cut. However, these formulas can be also used when the cut is oblique if the second cutting is smaller than the principal cutting and the curvature of the machined surface is not big enough. Without entering in detail, the cut force can be computed through the K_s method as:

$$F_t = K_s A \quad (5.25)$$

Where:

- K_s is the cut pressure $[\frac{N}{mm^2}]$
- A is the chip section

The K_s method is technological because the parameter K_s is determined based on measurements of the cutting force during the real working process. The parameter K_s can be also computed by an empirical formation, that is the Kronenberg relation:

$$K_s = K_{so} A^{-1/n} \quad (5.26)$$

where K_{so} is the specific cut pressure. The latter can be obtained using the relation proposed by Kronenberg. For the steel is:

$$K_{so} = 2.4 R_m^{0.454} \beta^{0.666} \quad (5.27)$$

where:

- $R_m [\frac{N}{mm^2}]$ is the tensile strength

- β is the angle that can be obtained from the Ernest-Merchant

For the cast iron:

$$Ks_o = 0.9(HB)^{0.4}\beta^{0.666} \quad (5.28)$$

Where HB is the hardness of the material (typical values are shown in table 5.1)

| Material | Brinell Hardness HB |
|---------------------|---------------------|
| Brass | 80 ÷ 120 |
| Bronze | 60÷70 |
| Aluminum and alloys | 65÷70 |
| Ultra-light alloys | 50 ÷ 60 |

Table 5.1: Typical values of the hardness in Brinell for different materials

In literature, there are other formulas to compute the specific cut pressure, but they are not presented because I referred to the formula proposed by Kronenberg. [18] Then, the final equation to obtain the cut force is:

$$F_t = K_{so}A^{1-1/n} \quad (5.29)$$

Rewritten as:

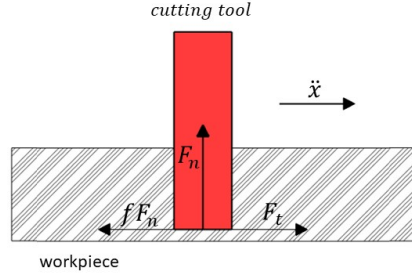
$$F_t = 2.4R_m^{0.454}\beta^{0.666}A^{1-1/n} \quad (5.30)$$

5.2 Dynamic analysis of the shear process

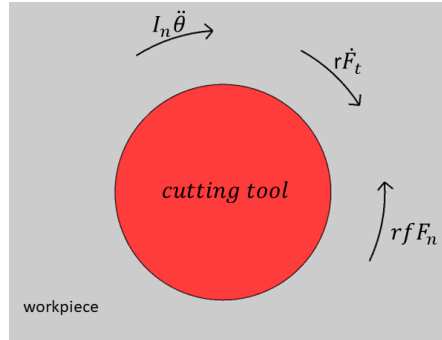
Following the introduction of the shear theory present in literature, it was analysed the dynamic interaction between the tool and the workpiece to find the forces that act during the process. After obtaining the forces, a new formulation of the friction coefficient β can be estimated. To analyse the system, the tool was approximated to a rotating cylinder, which translates along the direction of the workpiece (direction of feed) to perform the cutting process. Moreover, the tool and the workpiece were assumed to be both steel. Considering the system as shown in figure 5.6, the following relationships were obtained:

$$m\ddot{x} = F_t - fF_n \quad (5.31)$$

$$I_n \ddot{\theta} = r F_t - r f F_n \quad (5.32)$$



(a) Translation equilibrium



(b) Rotational equilibrium

Figure 5.6: Dynamic equilibrium

where:

- F_t is the cut force;
- $F_n * f$ is the sliding friction force. The parameter f depends on the type of material of two parts in contact. In our case, the two surfaces are steel, so the value suggested by the literature for f is 0.57;

- I_n is the inertia of the cylinder that represents the tool;
- F_t^*r is the torque responsible for the spindle rotation;
- $F_n^*f^*r$ is the resistive torque generated by the sliding friction force.

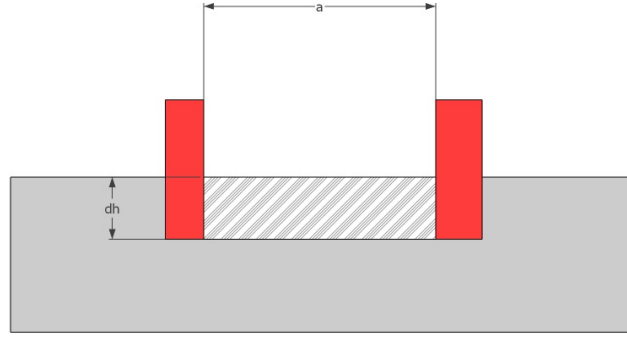


Figure 5.7: Chip section exported by the tool

In this case, referring to the formula 5.25, as shown in figure 5.7, the product between the feed (a) and the depth of cut (dh) provides the chip area. The parameter dh can be obtained thanks to the algorithm that identifies the interaction points. Instead, the parameter K_s is the cut pressure that can be computed according to the Kronenberg formulation as:

$$K_s = K_{so}A^{-1/n} \quad (5.33)$$

where K_{so} is the specific pressure that can be obtained through the following formulation given by Kronenberg:

$$K_{so} = 2.4R_m^{0.454}\beta^{0.666} \quad (5.34)$$

Then, the final equation, that allow to compute the cut force, is:

$$F_t = 2.4R_m^{0.454}\beta^{0.666}A^{1-1/n} \quad (5.35)$$

According to the Merchant circle, the normal force F_n is:

$$\begin{cases} F_t = R \cos (\beta - \gamma) \\ F_n = R \sin (\beta - \gamma) \end{cases}$$

$$\begin{cases} R = \frac{F_t}{\cos (\beta - \gamma)} \\ F_n = R \sin (\beta - \gamma) \end{cases}$$

$$F_n = F_t \tan (\beta - \gamma) \quad (5.36)$$

Referring to the formula 5.23, β (damping angle) can be extracted through the Ernst-Merchant formulas as below:

$$\beta = \frac{\pi}{2} - 2\phi + \gamma \quad (5.37)$$

Finally, the resistive torque generated M_{res} by the sliding friction force is:

$$M_{res} = frF_n = frF_t \tan (\beta - \gamma) = fr2.4R_m^{0.454}\beta^{0.666}A^{1-1/n} \tan (\beta - \gamma) \quad (5.38)$$

5.3 New mathematical model of the friction coefficient β

The previous paragraph describes how to obtain formulas to compute the forces that are present during the process. In particular, the attention was focused on the formula that allows computing the normal force F_n , which is responsible for the resistive torque. This paragraph explains how the formula for calculating beta is derived.

The considered system can be described dynamically by the following differential equation:

$$I_n \ddot{\theta} = C_m - \beta \dot{\theta} \quad (5.39)$$

Where:

- I_n is the inertia of the tool

- C_m is the torque provided by the electric motor for having the rotation of the spindle, on which is mounted the tool;
- $\beta \cdot \dot{\theta}$ is the resistive term that opposes the spindle rotation.

The goal was to obtain a mathematical formulation that allows computing the value of the friction coefficient beta according to:

- Cut condition;
- Type of material used in the processing;
- Geometry of the cut and the workpiece.

Therefore, I assumed that the friction coefficient beta varies in time with the following formula:

$$\beta(t) = \beta_0 + \Delta\beta(t) \quad (5.40)$$

Where:

- β_0 is supposed to be an initial value of β because, as soon as the tool interacts with the workpiece, there is immediately a resistance to the spindle rotation, so a resistive torque is exercised by the workpiece. Considering the previous work done inside the MOREPRO project, I assumed that β_0 is equal to 0.30, but this value will be confirmed later through some experimental tests.
- $\Delta\beta(t)$ represents the variation of the friction coefficient beta in time and it varies based on the resistive torque defined by the following formula:

$$\Delta\beta(t)\dot{\theta} = M_{res}(t) = F_n f r$$

$$\Delta\beta(t) = \frac{F_n f r}{\dot{\theta}} \quad (5.41)$$

The final formula that describes the friction coefficient β in time, considering different aspects, is:

$$\beta(t) = \beta_0 + \frac{F_n f r}{\dot{\theta}} = \beta_0 + \frac{2.4 R_m^{0.454} \beta^{0.666} A^{1-1/n} \tan(\beta - \gamma)}{\dot{\theta}} \quad (5.42)$$

Finally, tool wear occurs during the process resulting in an increase of the cutting force and normal force, so the friction coefficient also increases. Considering this phenomenon, Taylor's law is introduced which links the cutting velocity to the tool life through the formula below:

$$V_t t^n = C1 \quad (5.43)$$

Similarly to the Taylor law, the below formula is used to consider the tool wear in the process:

$$tool_{wear} = 1.2 V_t t^n \quad (5.44)$$

where, according to the range of values present in the literature for different materials, n is equal to 0.18. After the definition of the formula above, the cut force is assumed to increase its value by ten per cent. Therefore, the formulas to compute the forces are:

$$\begin{cases} F_{t,wear} = F_t(1 + 1.1 tool_{wear}) = 2.4 R_m^{0.454} \beta^{0.666} A^{1-1/n} [1 + 1.1(1.2 V_t t^n)] \\ F_{n,wear} = F_{t,wear} \tan(\beta - \gamma) \end{cases} \quad (5.45)$$

Finally, considering the formula 5.42 and 5.45, the final formula to obtain the friction coefficient β is:

$$\beta(t) = \beta_0 + \frac{F_{n,wear} f r}{\dot{\theta}}$$

$$\beta(t) = \beta_0 + \frac{2.4 R_m^{0.454} \beta^{0.666} A^{1-1/n} [1 + 1.1(1.2 V_t t^n)] \tan(\beta - \gamma) f r}{\dot{\theta}} \quad (5.46)$$

5.4 Design of the systems control

During the cutting process, the spindle speed is constant and the torque provided by the electric motor varies. Consequently, two digital controllers were implemented to do a final simulation of the system in Matlab and Simulink and to verify the new obtained formulation to compute the friction coefficient β . The scheme of the two controllers is shown in figure 5.8.

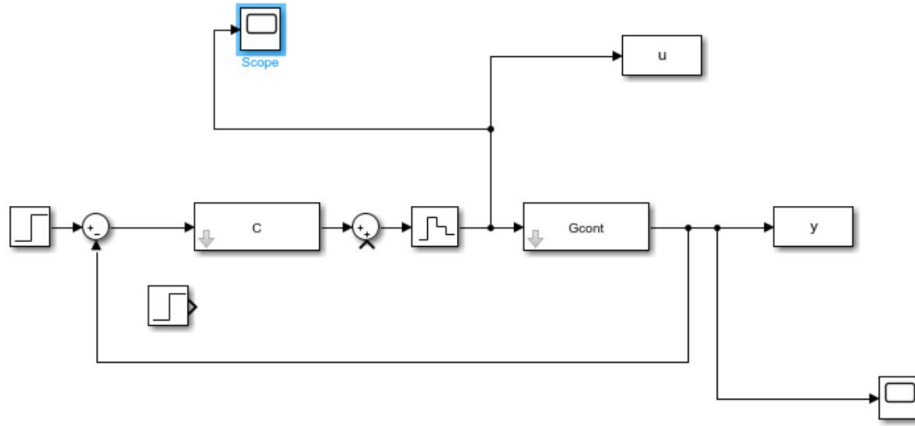


Figure 5.8: Scheme of the feedback control system

The two controllers, one for the contact between the tool and the workpiece and one for the case of no contact, were designed in according to the notion acquired in the Digital control technologies and architecture course. The first step to design the controllers is to consider the dynamic equation that describes the system: The state-space representation of the system, considering as input the torque ($u(t) = C(t)$) and as output the spindle speed ($y(t) = \omega(t)$), is:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (5.47)$$

where the matrices A, B, C and D are:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-\beta}{I_n} \end{bmatrix} B = \begin{bmatrix} 0 \\ \frac{1}{I_n} \end{bmatrix} C = \begin{bmatrix} 0 & 1 \end{bmatrix} D = 0$$

In case of no contact ($\beta=0$), the matrix A is: $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

The transfer function in continuous time domain that describes the system in both case are:

$$G_p(s) = \frac{\frac{1}{I_n}}{s + \frac{\beta}{I_n}} \quad (5.48)$$

Finally, to design the two controllers, the following constraints were chosen:

- $T_r < 0.003$ where T_r is the rise time;
- $T_s < 0.01$ where T_s is the settling time;
- $\tilde{s} < 12\%$, where \tilde{s} is the overshoot.

The transfer functions obtained for the two digital controllers after the design are:

$$Gc(z)_{nocontact} = \frac{0.037973}{z - 0.3179} \quad (5.49)$$

$$Gc(z)_{contact} = \frac{0.037973(z - 0.2793)}{z - 1} \quad (5.50)$$

5.5 Implementation of the dynamic part on Matlab and Simulink and final test

Finally, this dynamic formulation was implemented as a function block in Simulink, as shown in figure 5.9.

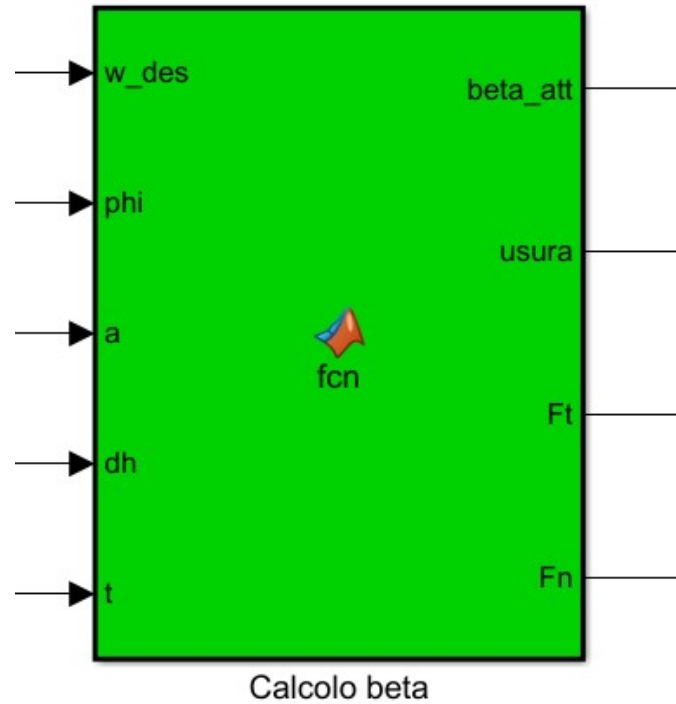


Figure 5.9: Simulink block function of the dynamic part to compute the friction coefficient

This block permits to compute of the friction coefficient β at each instant t based on the cutting condition. In particular, the inputs of this block are:

- $Wdes$: is the set spindle rotation
- a : is the feed
- phi : is the angle that describes the orientation of the workpiece with respect to the tool

- dh : is the depth of cut at each instant of time
- t : is a linear function to consider the time

The output of this block is the friction coefficient beta that was computed. Finally, considering the tool as a single cutter, several simulations were carried out to analyse the output in different cutting conditions. To simulate a real process, the following parameters were taken from the catalogue of Fraisa (a company that produces the cutting tool for *CNC* machines):

- $a=20$ mm;
- $d = 20$ mm, where d is the diameter of the cutter

The results of the simulation are:

TEST 1 ($w = 200$ rad/s, $a = 20$ mm, $r = 10$ mm, $dh = 1$ cm)

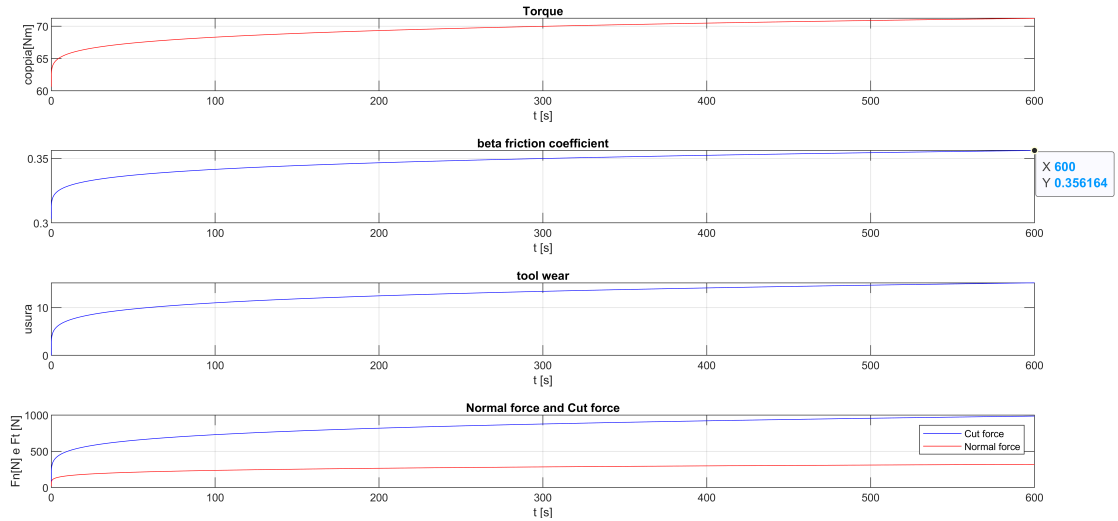


Figure 5.10: Plot of the results of the 1st test

TEST 2 ($w = 200$ rad/s, $a = 20$ mm, $r = 10$ mm, $dh = 2$ cm)

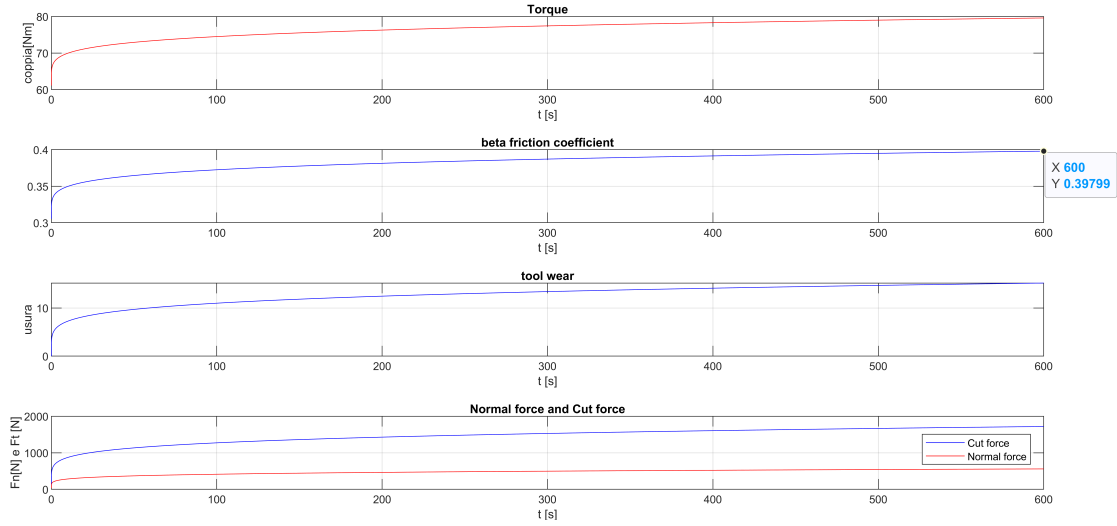


Figure 5.11: Plot of the results of the 2nd test

TEST 3 ($w = 200$ rad/s, $a = 20$ mm, $r = 10$ mm, $dh = 3$ cm)

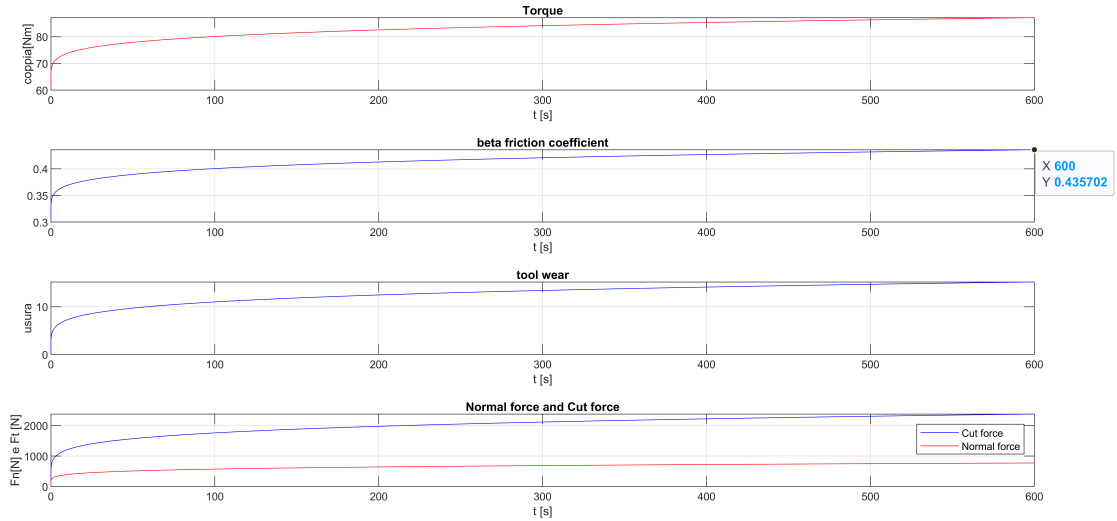


Figure 5.12: Plot of the results of the 3rd test

TEST 4 ($w = 200$ rad/s, $a = 20$ mm, $r = 10$ mm, $dh = 4$ cm)

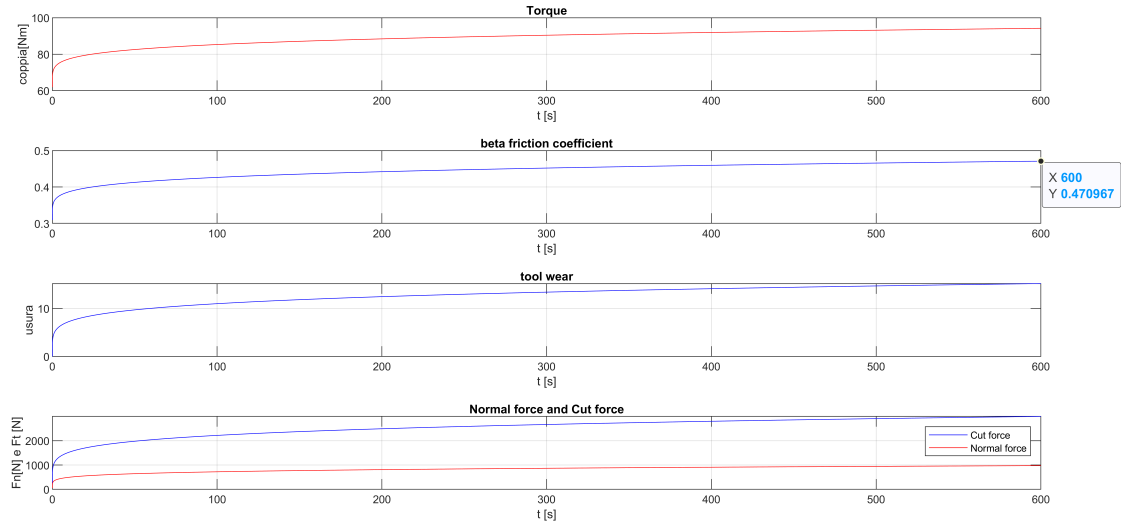


Figure 5.13: Plot of the results of the 4th test

TEST 5 ($w = 200$ rad/s, $a = 20$ mm, $r = 10$ mm, $dh = 5$ cm)

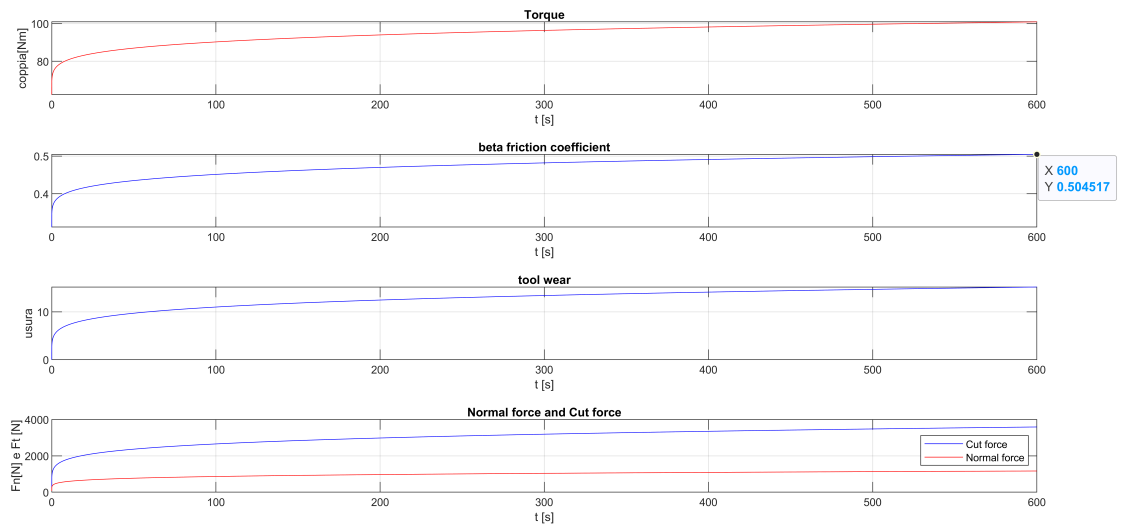


Figure 5.14: Plot of the results of the 5th test

In figures 5.10, 5.11, 5.12, 5.13 and 5.14, there is the plot of different parameters relevant for us , there are:

- Time evolution of the torque;
- Time evolution of the friction coefficient β ;
- Time evolution of the terms that consider the wear tool during the process;
- Time evolution of the forces F_n and F_t .

Referring to the plot of the simulations of the previous figures, when the depth of cut dh increases, the tool starts to wear out more quickly. For this reason, the span of the friction coefficient beta is higher in time when the depth of cut dh increases. As a result, according to the formula 5.46 (the found formula to obtain beta), the forces, consequentially also the torque, increase, as shown in the plots of the various simulations. Finally, several simulations of the model of the *CNC machine* designed in this thesis were performed using the tool with a diameter of 20 mm, which follows the *2D* planned trajectory. The simulations were performed by changing only the spindle speed with the same cutting conditions to see how the torque varies. The results are shown in the following figures.

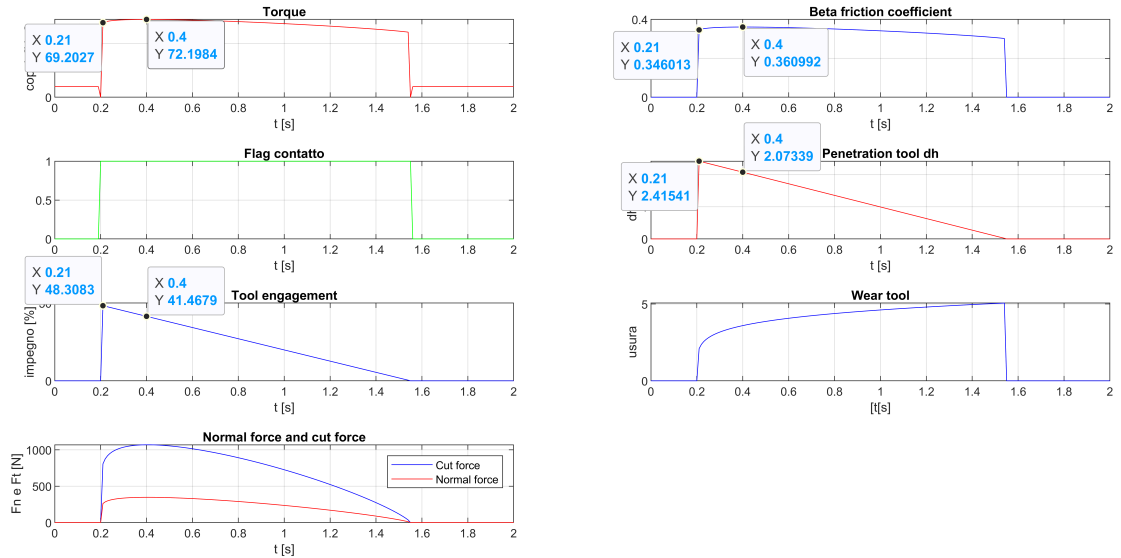


Figure 5.15: 1st simulation of the model with $w=200$ rad/s

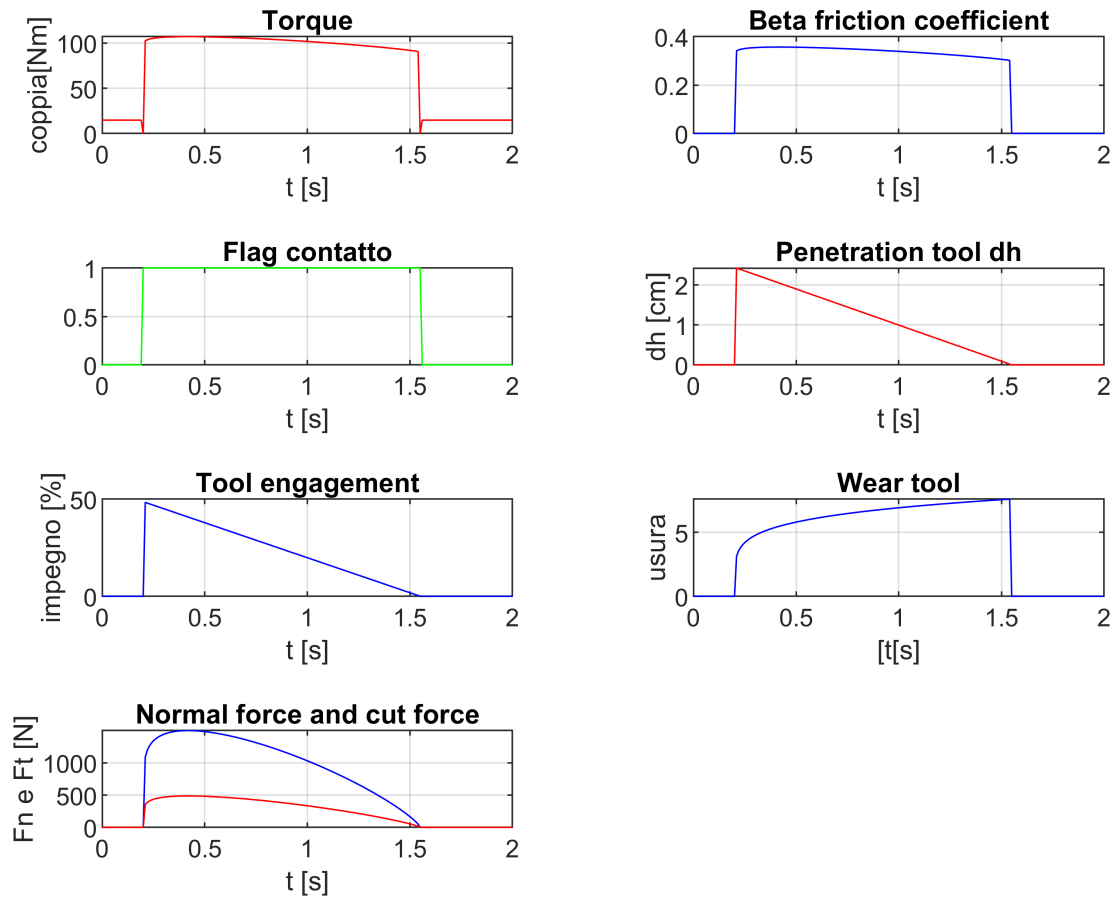


Figure 5.16: 2nd simulation of the model with $w=300$ rad/s

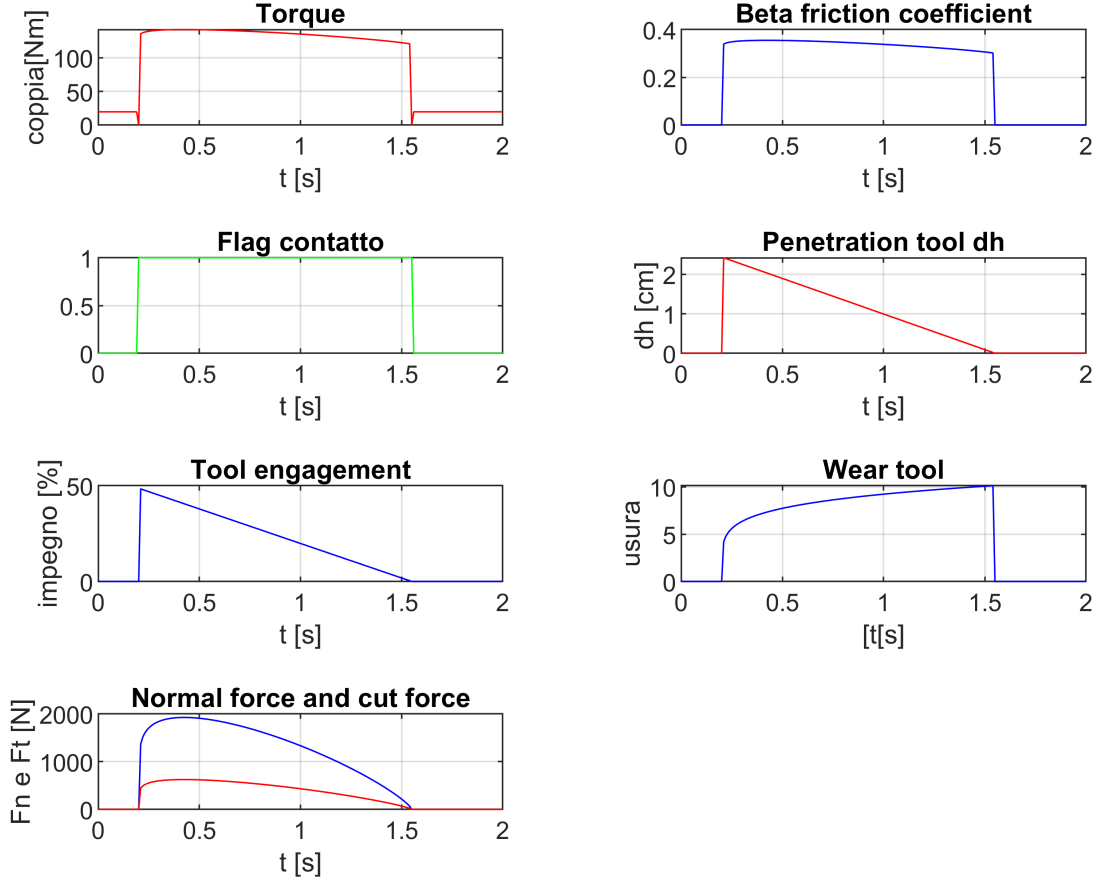


Figure 5.17: 3rd simulation of the model with $w=400$ rad/s

For each simulation was plotted the quantities relevant for us that are:

- The torque;
- Time evolution of the friction coefficient β ;
- time evolution of the two forces F_t and F_n ;
- Time evolution of the wear of the tool;
- A flag diagram that indicates if there its contact or not (1 or 0);
- The percentage of the used tool;
- time evolution of the penetration of the tool (*or depth of cut dh*).

Referring to the figures 5.15, 5.16 and 5.17, it is possible to see that in the time interval between 0.2 and 0.4 the friction coefficient β increases (also the forces and the torque supplied by the electric motor) despite the penetration of the tool decreases. Finally, increasing both the friction coefficient and the cutting force, the electric motor shall provide a higher torque to perform the cutting process with a spindle speed constant. Therefore, the results obtained from the simulations are sufficiently coherent with respect to those expected achieving the main goal of this thesis. The latter was to develop a complete simulation of the system considering the dynamic aspect of the process to obtain a new formulation to obtain the friction coefficient β .

Chapter 6

Conclusions

The aim of this thesis was to create a model of a CNC machine to carry out simulations to produce artificial data which could be used to find an estimation technique capable of estimating the SOH of the cut tool(end effector). To achieve this goal, the first step was to analyse the work completed by previous groups in the MOREPRO project. Then, a CNC machine was analysed and modelled, specifically a 5-axis CNC milling machine. During the modelling of the CNC machine, to better study the interaction between the tool and the workpiece, an approach similar to the FEM approach was followed. The tool was modelled as a cloud of points and the workpiece as a surface using geometric solutions. Finally, the interaction between the workpiece and the tool was dynamically analysed to derive a formulation that would allow the calculation of the friction coefficient, a very important parameter to be able to estimate the SOH of the tool.

In particular, the main activities carried out in this thesis are:

- The analysis and creation of a model of the CNC machine to simulate the movements made by the machine through the analysis of direct, inverse and differential kinematics;
- The development of a trajectory planning algorithm. In particular, to plan a rectilinear trajectory or a circular trajectory. In this thesis work, 2D trajectories were planned according to the CNC machine considered. However, the algorithm also works in the 3D case. Finally, the planned 2D rectilinear trajectory was provided as input to the inverse kinematics

algorithm to create a simulation in which the created model moves following the desired trajectory;

- A dynamic analysis of the interaction between the tool and the workpiece was carried out, where formulas were derived to calculate the forces that occur during the cutting process based on the material used in the machining process and the cutting conditions. After this, having obtained the formulas for calculating the forces, a mathematical formulation was found which allows the friction coefficient to be calculated;
- The design of the control systems to have a constant speed in the two cases of contact or non-contact between the tool and the workpiece.

Finally, simulations were first carried out to check that the mathematical formulation found to calculate the coefficient of friction returned sensible results. Then, after my colleague worked on the algorithm that identifies the contact points between the tool and the workpiece during machining, a complete simulation was carried out of the model created in this work in which the tool follows the desired trajectory provided. The results obtained were as expected, so the objectives set for this thesis work have been achieved.

6.1 Future works

The dynamic interaction between the tool and the workpiece is a very complex phenomenon to model dynamically because of the many variables to be taken into account in the cutting process. However, by referring to concepts in the literature, it was possible to find a mathematical formulation to be able to calculate the friction coefficient. Future work will be to confirm the correctness of this formulation using experimental tests. Furthermore, this formulation can be improved by taking into account other aspects that occur during the cutting process, for example, the thermal aspect. Finally, as already mentioned in the previous chapters, the simulation made of the system can be improved in the future in the following ways:

- Development of a better strategy to be able to simulate the movements of the CNC machine instead of the 'inverse kinematics algorithm, to reduce the error between the desired and the actual trajectory;

- Development of a planning algorithm for more complex trajectories;
- More accurate tool and workpiece modelling.

Finally, the data obtained from the simulations can be used in future MORE-PRO project work to develop a method capable of estimating SOH of the cutting tool.

Bibliography

- [1] Hashem M Hashemian. «State-of-the-art predictive maintenance techniques». In: *IEEE Transactions on Instrumentation and measurement* 60.1 (2010), pp. 226–236 (cit. on p. 2).
- [2] Rafael Gouriveau, Kamal Medjaher, and Nouredine Zerhouni. *From prognostics and health systems management to predictive maintenance 1: Monitoring and prognostics*. John Wiley & Sons, 2016 (cit. on p. 10).
- [3] Aleksandra Marjanović, Goran Kvašček, Predrag Tadić, and Željko Đurović. «Applications of predictive maintenance techniques in industrial systems». In: *Serbian Journal of Electrical Engineering* 8.3 (2011), pp. 263–279 (cit. on pp. 11, 12, 14, 16).
- [4] J. Kallrath. «ed. Modeling Languages in Mathematical Optimization, Applied Optimization». In: *Information* 88 (2004) (cit. on p. 12).
- [5] Antonia Verde. «End-effector tools wear prediction: machine and interaction modeling, system identification based on the EKF approach.» PhD thesis. Politecnico di Torino, 2021 (cit. on pp. 12, 18).
- [6] Dario Caniglia. «End-effector tools wear prediction: interaction with the workpiece modelling in a quasi-FEM approach». PhD thesis. Politecnico di Torino, 2021 (cit. on pp. 12, 18).
- [7] WorldQuant Perspectives. *Reshaping the world with Fuzzy Logic*. April 2018 (cit. on p. 15).
- [8] Alessandro Paolo Daga and Luigi Garibaldi. «Machine vibration monitoring for diagnostics through hypothesis testing». In: *Information* 10.6 (2019), p. 204 (cit. on p. 17).
- [9] Michele Pinto. *End-effector tools wear prediction: a multimodel approach*. 2021 (cit. on pp. 17, 18).
- [10] Luca Cecere. «End-effector tools SoH prediction: parameter identification and reliability estimation.» PhD thesis. Politecnico di Torino, 2021 (cit. on p. 18).

- [11] Sunday J Ojolo, Olumuwiya Agunsoye, Oluwole Adesina, and Gbeminiyi M Sobamowo. «Cutting Force and Friction Modelling in High Speed End-Milling». In: *International Manufacturing Science and Engineering Conference*. Vol. 56826. American Society of Mechanical Engineers. 2015, V001T02A029 (cit. on p. 21).
- [12] Abdil Kus, Yahya Isik, M Cemal Cakir, Salih Coşkun, and Kadir Özdemir. «Thermocouple and infrared sensor-based measurement of temperature distribution in metal cutting». In: *Sensors* 15.1 (2015), pp. 1274–1291 (cit. on p. 22).
- [13] Jaroslav Mackerle. «Finite element analysis and simulation of machining: an addendum: A bibliography (1996–2002)». In: *International Journal of Machine Tools and Manufacture* 43.1 (2003), pp. 103–114 (cit. on p. 23).
- [14] Zhao Haitao, Yang Jianguo, and Shen Jinhua. «Simulation of thermal behavior of a CNC machine tool spindle». In: *International Journal of Machine Tools and Manufacture* 47.6 (2007), pp. 1003–1010 (cit. on p. 23).
- [15] Wikipedia. *Macchina a controllo numerico* — *Wikipedia, L'enciclopedia libera*. [Online; in data 21-marzo-2022]. 2021. URL: http://it.wikipedia.org/w/index.php?title=Macchina_a_controllo_numerico&oldid=120779625 (cit. on p. 25).
- [16] Mikell P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson, 2015 (cit. on p. 26).
- [17] Luigi Villani Bruno Siciliano Lorenzo Sciavicco and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009 (cit. on pp. 29, 30, 38, 42, 46–49, 51, 57).
- [18] Francesco Giusti Marco Santochi. *Tecnologia meccanica e studi di fabbricazione*. CEA, 2000 (cit. on pp. 71, 72, 74, 76, 78, 80, 81).