

Politecnico di Torino
Department of Electronics and Telecommunications

Institute for Engineering and Architecture



Thesis
For the Degree of
Master of Science in Mechatronics Engineering

**End-effector tools wear prediction:
a multi model approach**

By
Michele Pinto

Submission Date: April, 2021

Mentor: Prof. Alessandro Rizzo
Supervisor: Ing. Giovanni Guida - Brain Technologies

Abstract

Nowadays there is a need for many companies emerging in the context of Industry 4.0 to save costs, increase efficiency, and improve the factory management. For this reason the research on predictive maintenance techniques and state of health (SoH) estimation of a production machine is one of the most relevant areas in the scientific field.

This thesis is inside the research and development MOREPRO project owned by Brain Technologies. Morepro aims to bring to the field an innovative solution based on multi-level distributed intelligence logic to improve the management of production plants thanks to new capabilities. The objective concerns the development of a prototype which is capable of:

- Monitoring the SoH of CNC machines and their critical components through embedded sensors signals coupled with machine learning and data mining techniques.
- Monitoring the wear condition of the machine's tool using a digital twin approach, combining real-time signals with estimated quantities in a virtual simulation environment.
- Developing predictive models able to estimate the on-line SoH and the trend of degradation states of machine and system components over time.

The contribution of this thesis work can be divided in three main sections:

- 1. Development of a basic CNC machine model:** the first phase is the physical model's basic design of a CNC machine, in order to be able to develop a prediction algorithm. The approach was to combine the state equations of a DC motor with the mechanical equations of a CNC machine and simulate the plant to collect the values of the analysed state variables.
- 2. Predictive multimodel:** the core of the project where a prediction analysis on the state variable and wear's parameter estimation are developed using a bank of Extended Kalman Filters and a logic of residual error management.
- 3. Interaction model upgrade and predictive multimodel update:** an in-depth modelling of the interaction between CNC end-effector and work-piece, through the study of how various parameter can impact the end-effector's wear condition. An update of the EKF's bank is made accordingly to the upgrade of the model. Finally various tests were carried out to check the overall system correct behaviour.

Table of contents

List of Figures	3
List of Tables	5
Acronyms	7
1 Introduction	8
1.1 Wear estimation	8
1.2 CNC machine SoH	9
1.3 Edge Computing advantages	10
1.4 MOREPRO project	11
1.4.1 Partnership	13
1.5 Work Organization	13
1.5.1 Morepro team	14
1.5.2 Work flow	16
1.6 Thesis outline	18
2 State of art	20
2.1 Methods for estimating SoH	21
2.1.1 Past project references	28
2.1.2 Comparison between the methods	29
3 Physical Model	30
3.1 Mechanical part	31
3.2 Electrical part	32
3.3 Plant model	34
3.3.1 Simulink implementation	34
3.4 Most significant parameter choice	40
4 Parameter identification	41
4.1 Set-membership identification	42
4.1.1 Application	43
4.1.2 Results	44
5 EKF Bank: multi model approach	46
5.1 EKF	47
5.2 Residual error analysis	49
5.2.1 Residual error comparison	59
5.3 Evaluation tests	62
5.3.1 Reset time choice	73
5.4 Multi model: algorithm structure	77

5.4.1	Switching estimator	79
5.4.2	Best model choice	80
6	Model upgrade	83
6.1	Friction coefficient models	84
6.1.1	Friction coefficient correlated with the tool-chip contact length	84
6.1.2	Thermal effect in friction coefficient	86
6.1.3	Thermal analysis	87
6.1.4	Other correlated model	89
6.1.5	Model comparison	90
6.2	Final plant model	92
7	Multi model update	94
7.1	EKF integration	94
7.2	Functional Tests	95
7.3	Sensitivity Tests	98
	Conclusions and future works	104
	References	105

List of Figures

1.1	Example of CNC machine schematic diagram	9
1.2	Synthetic structure of MOREPRO system	12
1.3	Overall system structure	12
1.4	V-shape development flow.	14
1.5	Team organization chart.	15
1.6	Workflow schematic.	16
1.7	Workflow contribution.	16
2.1	Forms of maintenance	21
2.2	Schematic of state update	23
2.3	P-F curve	24
2.4	Schematic of a Artificial Neural Network	25
2.5	Components of a Fuzzy logic system	26
2.6	Schematic of HMM	27
2.7	Flow diagram of condition indicator construction. The gray boxes indicate the condition indicators.[14]	28
3.1	Simplified milling machine model	31
3.2	Simplified schematic of a DC motor	32
3.3	Contact logic	35
3.4	Plant Simulink.	36
3.5	Simulink implementation of the model	37
3.6	Contact force control input plot.	38
3.7	Summary plots of Model's main parameters.	39
4.1	Block scheme of the system	42
5.1	Diagram of nonlinear discrete time system in state-space form . . .	48
5.2	Simulink scheme of the EKF.	49
5.3	Mean test.	52
5.4	Covariance test.	53
5.5	PSD test.	54
5.6	Correlation test.	55
5.7	RMS test.	56
5.8	Integral test.	57
5.9	Boxplot of the angular acceleration error.	60
5.10	Boxplot of the current derivative error.	61
5.11	Boxplot of the average error.	61
5.12	Boxplot error with $200 \left[\frac{rad}{s^2} \right]$ angular velocity.	62
5.13	Boxplot error with $210 \left[\frac{rad}{s^2} \right]$ angular velocity.	63
5.14	Boxplot error with $220 \frac{rad}{s^2}$ angular velocity.	64

5.15	Boxplot error with 230 $[\frac{rad}{s^2}]$ angular velocity.	64
5.16	Boxplot error with 0.4 $[m]$ position reference.	65
5.17	Boxplot error with 0.47 $[m]$ position reference.	66
5.18	Boxplot error with 0.53 $[m]$ position reference.	66
5.19	Boxplot error with 0.6 $[m]$ position reference.	67
5.20	Boxplot error with 20 % duty cycle.	68
5.21	Boxplot error with 40 % duty cycle.	68
5.22	Boxplot error with 60 % duty cycle.	69
5.23	Boxplot error with 80 % duty cycle.	70
5.24	Boxplot error with 4 number of cycles.	71
5.25	Boxplot error with 6 number of cycles.	71
5.26	Boxplot error with 8 number of cycles.	72
5.27	Boxplot error with 10 number of cycles.	73
5.28	T reset analysis	74
5.29	T reset choice	75
5.30	Integral error in nominal condition with Reset time of 30s.	76
5.31	Simulink implementation of the algorithm.	78
5.32	Simulink implementation of error logic.	78
5.33	Switching estimator	79
5.34	Best model choice with nominal condition	81
5.35	Best model choice with β variation	82
6.1	Model upgrade path	83
6.2	Proposed friction model.	85
6.3	Tool-workpiece interaction temperature curves.	87
6.4	Influence of cutting speed on tool-chip interface temperature.	89
6.5	β obtained from the tool-chip contact length model	90
6.6	β obtained from the temperature dependent model	91
6.7	β obtained from the machining speeds model	91
7.1	Boxplot based on chip load with nominal condition	96
7.2	chip load analysis from 0.2 to 0.7	97
7.3	chip load analysis from 0.8 to 1.1	98
7.4	Test on angular velocity reference.	99
7.5	Test on number of cycles.	100
7.6	Test on number of Filters.	101
7.7	Test on position reference.	102
7.8	chip load range analysis from 0.7 to 1.7	103

List of Tables

2.1	Comparison between SoH estimation methods.	29
4.1	Parameters Uncertainty Intervals	44
4.2	Parameters identification.	45
5.1	Nominal values of the errors for each method.	50
5.2	FOM mean	52
5.3	FOM Covariance	53
5.4	FOM PSD	54
5.5	FOM Correlation	55
5.6	FOM Correlation	56
5.7	FOM Integral	57
5.8	Nominal CNC parameters.	59
5.9	Nominal working conditions.	60
5.10	Test with 210 $[\frac{rad}{s^2}]$ angular velocity.	63
5.11	Test with 220 $[\frac{rad}{s^2}]$ angular velocity.	63
5.12	Test with 230 $[\frac{rad}{s^2}]$ angular velocity.	64
5.13	Test with 0.4 $[m]$ position reference.	65
5.14	Test with 0.47 $[m]$ position reference.	65
5.15	Test with 0.53 $[m]$ position reference.	66
5.16	Test with 0.6 $[m]$ position reference.	67
5.17	Test with 20 % duty cycle.	67
5.18	Test with 40 % duty cycle.	68
5.19	Test with 60 % duty cycle.	69
5.20	Test with 80 % duty cycle.	69
5.21	Test with 4 number of cycles.	70
5.22	Test with 6 number of cycles.	71
5.23	Test with 8 number of cycles.	72
5.24	Test with 10 number of cycles.	72
5.25	Friction coefficient associated to each filter	80
5.26	Test with nominal friction coefficient	81
5.27	Test with friction coefficient variation	81
6.1	Temperature values.	88
6.2	Nominal CNC parameters updated	93
7.1	Chip Load associated to each filter	95
7.2	Test with nominal chip load	95

Acronyms

MPC	model predictive control
SoH	State of health
ERMES	Extendible Range MultiModal Estimator Sensing
DOE	Design of experiment
KF	Kalman filter
PF	Particle Filter
EKF	Extended Kalman filter
EM	Electric Machine
T	Temperature
RMS	Root Mean Square
LSM	Least Square Method
RT	Reset time
CNC	Computer numerical control
IOT	Internet Of Things
RUL	Remaining Useful Life
EOL	End of Life
DC	Direct current
AI	Artificial Intelligence

SCARA Selective Compliance Assembly Robot Arm

PM Predictive Maintenance

ANN Artificial Neural Network

HMM Hidden Markov Model

MQL Minimum Quantity Lubrification

1 Introduction

1.1 Wear estimation

The estimation in real-time of the state of a production machine is one of the most significant topic in scientific research. The estimation of the SoH together with predictive maintenance techniques are slightly becoming a relevant issue because of their direct relation that link them to the efficiency of the production and to the saves of the costs. As Industry 4.0 continues to become reality, many companies are struggling with AI algorithms implementation that can lead to to major cost savings, higher predictability, and the increased availability of the systems. Indeed, the benefits of predictive strategies are definitely very strategic. Thus, the increasing demand of monitoring systems that allows to keep track of the production as much efficiently as possible have led to the development of many predictive maintenance methods [2]. The main functions of those algorithms are:

- SoH (state of health) estimation of a machine, motor or single component.
- Calculation of patterns that can help prediction and prevention of failures.

Currently, such methods are predominantly based on machine learning algorithms that lead to very good results in terms of efficiency and precision but they often doesn't take into account of the computational effort and real-time requirements. Nevertheless, predictive maintenance doesn't require anything more than mathematical computation on when machine conditions are at a state of needed repair or even replacement so that maintenance can be performed exactly when and how is most effective. Moreover, when the processing has high precision requirements, the predictive algorithms are particularly useful. Nowadays, those requirements are very common in companies which rest their production on such fields as aerospace, oil & gas, automotive and so on. The complexity of those high precision processes depends on many aspects:

- Kind of processing.
- Modelling and simulation of robotic, mechanical and electric the systems.
- Wrought materials.
- Different tools such as milling machines, cutting machines, end-effectors and so on.
- Production timings requirements.

Dealing with those complexity level can be very hard and expensive for companies, consequently, it is always more present the need of a method that can be easily applied regardless of the wrought material, the kind of processes and the field

of application. To sum up, the key functionalities of prediction algorithm are consistent when there is abstraction with respect to processing types, real-time characteristics and efficiency both in terms of computational effort and naturally, in terms of cost savings.

1.2 CNC machine SoH

CNC (Computer numerical control) [2] machine are high precision machines which actuate manufacturing processes of material subtraction that usually require computerized control action to guarantee high precision and efficiency. A subtractive manufacturing process typically employs machine tools to remove layers of material from a stock piece known as the blank or workpiece and produces a custom-designed part. This processing type is almost independent from the material of which the workpiece is composed: plastics, metals, foam, glass etc.. This is the reason why CNC machines finds application in most of industrial processing fields.

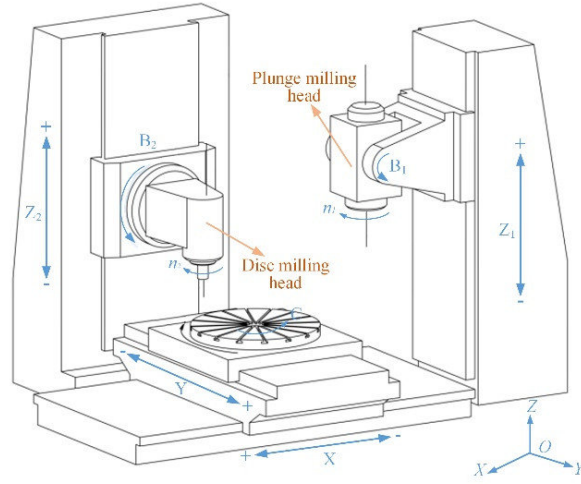


Figure 1.1 Example of CNC machine schematic diagram

As it is shown in figure 1.1 this machines have typically a SCARA or a cartesian robotic configuration with an end-effector which usually is a cutter. The modelling of the cutter contact is quite difficult because of the high number of variables that must be considered, the most relevant are:

- Robotic configuration.
- Environmental parameters.
- Wear condition of the machine: SoH.

All of those elements needs to be kept under control constantly in order to guarantee the efficiency and the precision of the machine. In particular, there is no way to check the State of health of the end-effector in a direct way. There could be possible to install sensor to check temperature, voltage, pressure and estimate a possible SoH of the tools. However, even with the knowledge of variables that can be measured by sensors, it is difficult to extract information about actual condition of the machine, firstly because is very likely that sensors cannot be set up in the right position, secondly because the knowledge of those parameters could not be enough to understand the real condition. For instance, obviously, a temperature sensor cannot be positioned near enough to the cutter to measure the correct temperature but must be positioned further, and that definitely lead to constant and inevitable measurement errors. Nowadays, the majority of the systems which estimate the SoH commonly propose digital-twin solutions. The limitation of such system is the difficulty to isolate the tool's wear from the others monitored effects. Moreover, such monitoring systems combine machine learning techniques and digital-twin simulation to estimate SoH, not taking into account computational requirements. Digital-twin models are very useful when the variables that need to be controlled are numerous, but the most influent parameters in the SoH estimation are the ones related to the cut process and the end-effector: the most stressed mechanical elements. Therefore, the parameters which are directly linked to the SoH are a lot and some of the most important are:

- Friction coefficients.
- Temperature.
- Chip load.

Those elements are strictly related to the contact forces, that's why understanding and modeling those element is fundamental for the estimation of the state of health of CNC machine end-effector.

1.3 Edge Computing advantages

As it was mentioned in the introduction, most of the existing architectures regarding the wear estimation and the predictive maintenance entrust the majority of their computational power in the cloud. Since to execute deep machine learning calculations there is the necessity of hardware resources, they have no choice but to rely on cloud computing solution. However, a centralized server, even if geographically far, can be definitely useful because of the potentially infinite number of resources that can be accessed and also, because of the huge data storage capacity available in the servers. On the other side, Edge Computing is an IT distributed architecture which allows to elaborate data locally, as much close as

possible to the source. It is based on distributed calculation concept, which relies its principles in the separation of the code execution and in the storage of data only when it is strictly necessary. This solution compensate some of the cloud computing shortcomings and provide some advantages:

- Low-latency: edge computing devices are installed locally and compensate latency that prevent the execution in real-time.
- Costs: Since hardware requirements are very low, the costs of those micro-processors is almost insignificant.
- Reliability and Security: Since most of the times the edge computing does not depend on internet connection and servers it offers an uninterrupted service. Users do not need to worry about network failures or slow internet connections.
- Scalability: Updates and modification on a cloud computing architecture can be very expensive. Edge computing do not require a datacenter to store data and it is easy to add and remove devices from the network architecture.

For sure, the limitation that characterize an edge computing device are strong constraints and developing a system which is comparable in performance with the powerful machine learning tools can be rather challenging, but definitely it is a way that it is worth to study.

1.4 MOREPRO project

Considering all the thematic exposed above, the MOREPRO project wants to bring on the field a new and innovative proposal, which is not present in any production system nowadays. It is basically based on a logic architecture distributed in three different levels:

- Monitoring of the SoH of machine and plant critical components through embedded sensors and, consequently, applying machine learning and data mining techniques.
- Keeping track of the SoH of the machine using digital twins tools. The goal is to combine real-time environment signals along with some estimated quantities in a specific simulation environment.
- Developing of forecast models, able to estimate the SoH of the machine and the time evolution decay of the plant/machine.

The general development architecture can be subdivided in two main levels. A first **field level** (Edge), where signals will be acquired and processed for a local supervision of the SoH. This is extremely useful to have a rapid reaction when

any danger anomaly is detected. The same signals are then deployed to a second **server level**, mainly located on the cloud, which will be able to set up a proper bank of data, implement *digital twin* techniques and compute the right parameters to reconfigure the elaboration logic of every single edge device. The crucial part is the continuous interoperability between the two levels and the possibility to reconfigure the architecture *on the fly* depending on the case problem. The figure below represents a general scheme on which the project will be based on.

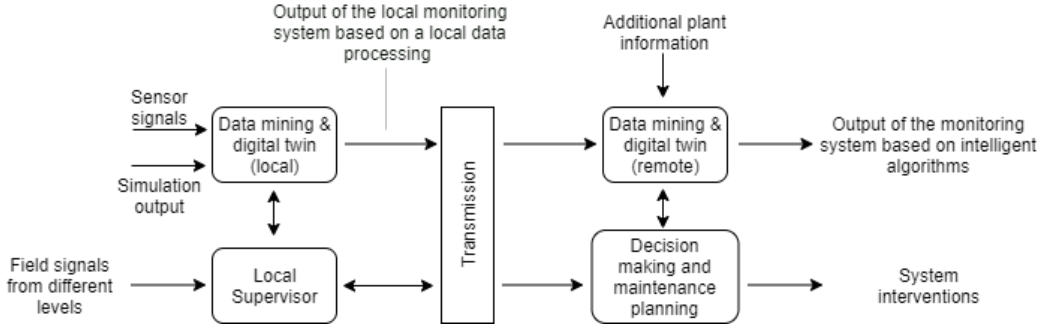


Figure 1.2 Synthetic structure of MOREPRO system

With reference to the figure, the *edge device* will implement the data-mining, digital-twin and monitoring algorithm allowing the bidirectional data exchange with both the plant and the supervisor. In practice, it will process the real signal coming from the field along with the simulation output in order to compute a SoH of the considered element under monitoring. On the other side, the supervisor will be able to adjourn and perfection the algorithm of the device itself in order to reconfigure and support the planning decisions. A possible physical implementation can be seen in the next figure.

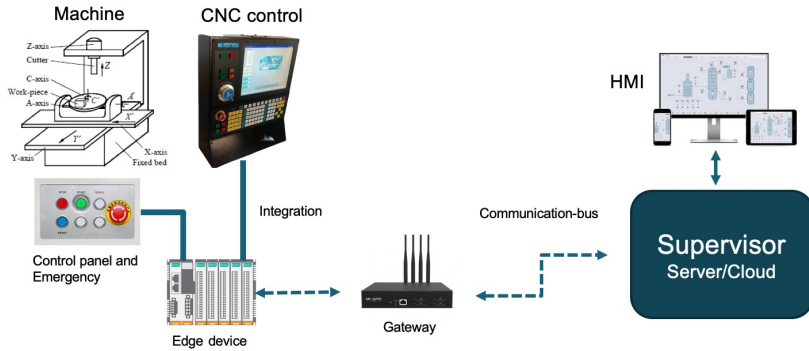


Figure 1.3 Overall system structure

1.4.1 Partnership

To the aim of this project several companies are involved. Thus, it is relevant to see how each of them is involved in the work, in order to understand how a development process is usually treated when a completely new and innovative device must be designed.

- **brain Technologies:** it will in particular contribute to the definition and design of the digital architecture (in collaboration with the other partners) and to the software development of the distributed intelligence system proposed by the project, including the architecture of the control supervisors whose action is propagated both in the devices and in the cloud. brain Technologies will also contribute to the implementation, testing and validation phases of the final prototype.
- **MCM S.p.A:** it will contribute to the analysis of user needs and to the proposal of new or improved functionalities of the processing systems as drivers for the development of the monitoring and predictive management methods of the plants covered by the project. Other activities in which MCM will play an active role include: 1) interfacing the machines for data collection, also through the installation of new sensors; 2) supporting the integration of the new MOREPRO solutions with the plant supervisor software, 3) analysis and testing of the prototype system with its validation at the production unit of CAMS, a partner in the project.
- **AL.MEC:** it will contribute to the design and manufacture of electronic boards and components necessary for data collection from machines and sensors, their mash-up and processing on board the machine and sending standardised information to predictive maintenance systems.
- **CAMS:** it is participating in the project by contributing its vision and expertise as a user of highly flexible production lines for the manufacture of complex, high value-added parts. CAMS will support in particular 1) the first phase of definition and analysis of the requirements that will guide the subsequent development of the new plant monitoring and predictive management solutions, 2) the identification and definition of its cases of industrial interest, 3) the implementation, testing and validation of the final prototype in its own production lines equipped with flexible MCM systems.

1.5 Work Organization

MOREPRO project is starting out in september 2020, and the work must be organized in order to start the development as fast as possible. In this situation model-based software design can be very suitable. Model-based approaches recom-

ment to follow precise development procedures, the so called V-shaped represents a process to be chased in order to guarantee efficiency and cost-effectiveness during such project natural advancement.

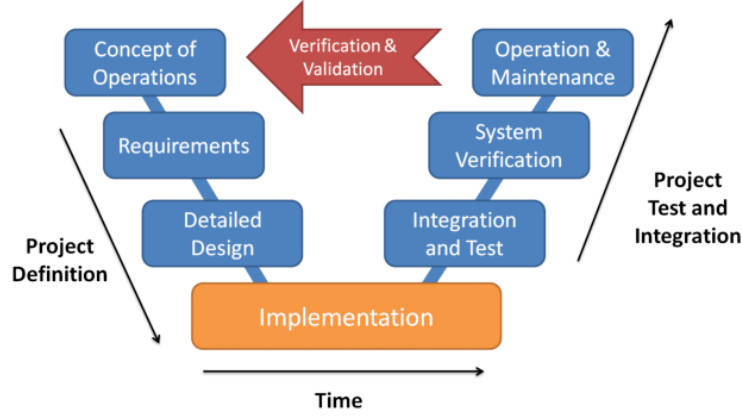


Figure 1.4 V-shape development flow.

During the first phases of the realization of scientific projects such as MOREPRO, model-based approaches as the one shown in figure 1.4 are necessary for the organization of the work. This become even more true as much as the number of people that join the project increases. Therefore, in the following paragraphs there will be a brief introduction to the team components and their aim in the V-shaped development, followed by a presentation of the work flows and the aims of the project team.

1.5.1 Morepro team

From the collaboration between Politecnico di Torino and brain Technologies srl it is arised a team of graduate students supervised by brain Technologies engineers, with the aim of developing the first phases of the project development flow. As it was previously mentioned, the project is only at the first stage, so once it is defined the concept of operations, the aim of this team is to obtain a first implementation after the first six months of work. Despite the development flow suggests to focus first on the requirements and analysis, it has been decided to employ one member of the team to do a requirement analysis, two members working on a detailed modelling of the problem, and the three remaining members working on the core implementation. This choice comes from the necessity to get a fulfilling conclusion satisfying all time-requirements.

Therefore, there are three different sub-teams:

1. Prediction team: This team will focus the attention on the prediction analysis and parameter estimation. After the development of a simple model, the aim

becomes to deeply study parameter identification through kalman-filters and residual error analysis techniques. Estimation of wear and SOH of a CNC machine is the main objective, to get to this, multi-model approach will be implemented and tested in detail, using simulative environment such as MATLAB and Simulink.

2. Modelling team: This team is created to obtain a preliminary detailed modelling of the kynematics and dynamics of a CNC Machine as first. Secondly, the main objective is to study and specify the interaction between end-effector and workpiece.
3. Requirements team: This team is in charge to carry out an overall view of the project, analyzing requirements and specifics for each part of the project. Finally, another important role of this team is to develop a design of experiment in order to opportunely test the functionalities individually and together.

After the first three months of work, the team sub-division is not valid anymore because each team-component will be focused on developing further features listed in 1.5.2.

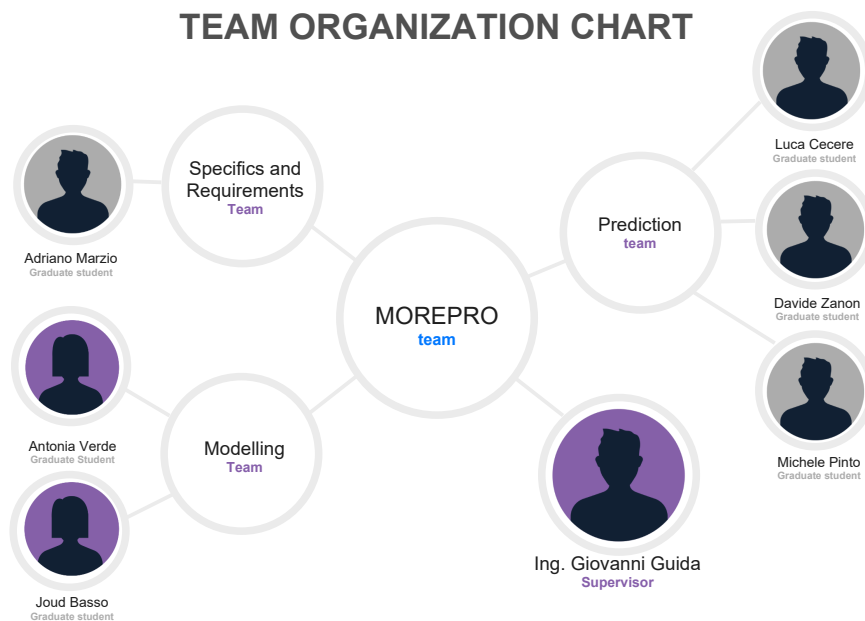


Figure 1.5 Team organization chart.

1.5.2 Work flow

An organization of the work flow is fundamental to help streamline and automate repeatable tasks, minimizing room for errors and increasing overall efficiency. The MOREPRO project work flow can be synthesized in the following schematic:

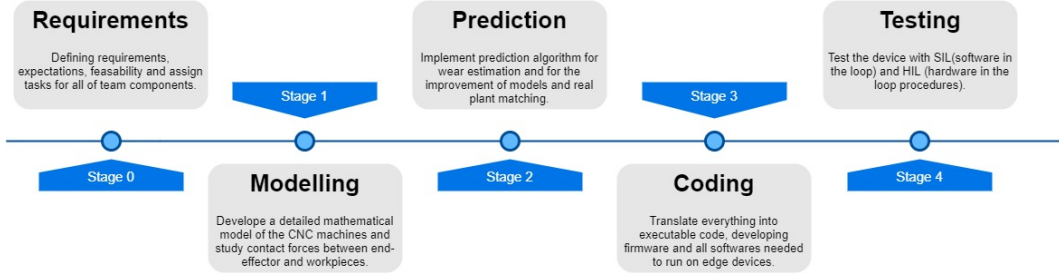


Figure 1.6 Workflow schematic.

As it was explained in the previous paragraph, the tasks have been assigned to be executed in parallel, however they are meant to be put together, however, it is important to keep in mind a clear idea of the pre-determined work-flow.

For what regards this thesis, the work-flow contribution in terms of what is carried out in team and what individually is well defined and it can be represented as follows:

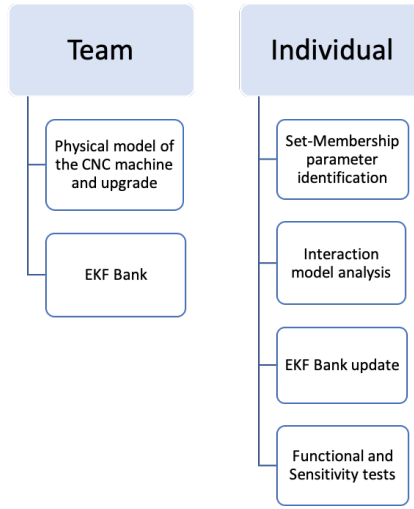


Figure 1.7 Workflow contribution.

The individual points can be summarized in:

- Set-membership identification on the unknown parameters of the model.

- In-depth modelling of the interaction between the end-effector and the work-piece.
- Update of the multi model according to the new friction coefficient model and various tests (functional and sensitivity) to verify the correct behaviour of the multi model system.

1.6 Thesis outline

The contents of this thesis are organized as presented in the following.

In Chapter **1**, an introduction related to the field of wear estimation, the application areas and their main characteristics is reported. The CNC machine is presented as a case study and in order to estimate its SoH the advantages of an edge computing approach are analysed. This is followed by the presentation of the MOREPRO project, with objectives, overall system structure and partnership. Finally the work organization between the team is described along with the work flow timeline.

In Chapter **2**, an in-depth study of the state of the art literature in topics of interest for this work is carried out. First there is a presentation of the different types of system maintenance and then specifically the various techniques for estimating SoH are examined such as Model-based approach, State observer and Learning algorithm. Finally, a summary table of all the methods with the main advantages and disadvantages is also proposed.

In Chapter **3**, the design of the physical model of a CNC machine is described, in order to be able to develop a prediction algorithm. The approach is to combine the state equations of a DC motor with the mechanical equations of a CNC machine and simulate the plant to collect the values of the state variables and to discriminate which parameter mostly affects the SoH.

In Chapter **4**, a parameters identification is carried out. In particular, the approach used is the set-membership one due to the inability to identify some parameters with a Kalman Filters Identification method tested by a colleague. The algorithm is applied to the state equation which allowed such an approach.

In Chapter **5**, the multi model approach is implemented. The algorithm involves a bank of N Extended Kalman Filter, each based on a different wear hypothesis taking into account the friction coefficient as the most significant parameter. The state variables of the plant model is estimated by the filters and a corresponding residual error is analysed through a suitable logic in order to choose the filter model that best approximate the plant condition.

In Chapter **6**, an in-depth modelling of the interaction between CNC end-effector and workpiece is developed in order to have a better information from the simulation performed. A case study is the dependence of friction coefficient on temperature and therefore it is added to the initial model. Another influencing factors emerged from literature are the machining speeds. Finally, by putting the different dependencies together, the friction coefficient is expanded as a function of other parameters or variables. It is proven that the one with the greatest impact on the change in SoH is the chip load. The final model is then modified by integrating

the state equations with the new formulation of friction coefficient.

In Chapter 7, accordingly to the model upgrade, an update of the EKF bank is needed. The chip load becomes the new wear condition hypothesis on which each filter is based. Finally, a series of tests were carried out: functional tests in order to analyze the correct behaviour of the multi model system and to verify the matching between the best model choice with the hypothesis assumed; a sensitivity analysis on the residual error produced by the bank of EKF with the aim of identifying, among the most significant parameters of the CNC machine model, how their possible variations affect the integral error produced and how a different input conditions affect the state observer behaviour.

Concluding remarks obtained from this thesis will be presented after Chapter 7. Moreover, promising new project lines that emerge from this work will be outlined as well.

2 State of art

System reliability is one of the main issues in the nowadays industry, thus the development of advanced system maintenance techniques is an emerging field based on the information collected through system or component monitoring (or system state estimation) and equipment failure prognostics (or system state forecasting). According to the standard EN 13306 (2001), such techniques can be grouped into two main categories. The first one is **corrective maintenance** and it consists of replacing the component and repairing the damage after some major breakdown. This kind of approach is used when the consequences of a failure are not so critical and the intervention on the field does not require a lot of costs and time. In particular, we refer to *palliative maintenance* when the repair is provisional, and *curative maintenance* when it is definitive. The second one is **preventive maintenance** and it refers to provide an alarm before faults reach critical levels so as to prevent system performance degradation, malfunction, or even catastrophic failures. When the maintenance intervention is time-based, meaning that the components are replaced based on a predefined schedule which relies on the working hours of the component, it is referred as *predetermined maintenance*. Obviously, this approach is not optimal, since the components are being replaced before the end of their lives, therefore increasing the costs.

A possible solution is to use *condition-based maintenance*, which refer to the analysis of real-time data in order to find in the change of their characteristic a possible failure. However, this approach do not guarantee to design a maintenance policy with certainty. On the contrary, *predictive maintenance* try to estimate the SoH of the machine, relying on more dynamic algorithms. [6]

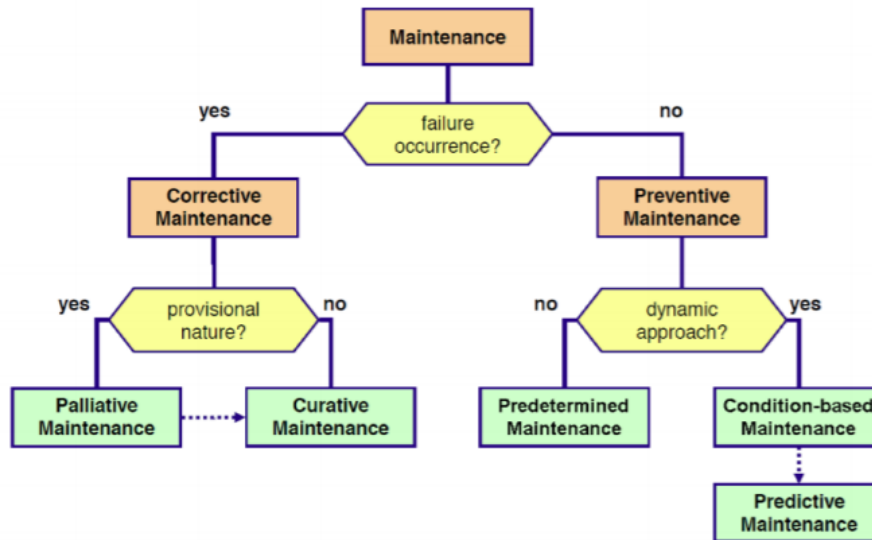


Figure 2.1 Forms of maintenance

2.1 Methods for estimating SoH

1. Model-based approach

This approach makes use of physical failure model in order to predict the degradation rate of a component or its lifetime. In practice, a mathematical model able to capture the failure mechanism must be developed. It seems obvious that the more accurate and sophisticated the model is, the more precise will be the SoH estimate of the machine under control. However, it is not always possible to obtain a model that perfectly adheres to the reality, that is why a trade-off between a very precise model and an estimate that allows to hide the lack of knowledge of the plant is needed. Usually, this approach follows some prefixed steps:

- **Critical part selection:** it is important, especially in very complex plant, to focus the study only on the part that actually contribute to the lifetime duration of the machine.
- **Failure mechanism determination and model definition:** intuitively, this is the most difficult part, where a suitable model must be designed in order to capture the most relevant aspects.
- **Governing loads evaluation:** it is important to understand which loads affects most the failure and how they are related to the operational usage of the system.

- **Data collection:** once the model is defined, it is possible to collect data from the field.
- **Failure prediction:** combining the monitored data with those one coming from the model it is possible to have an actual estimation of the health of the plant.
- **Model validation:** finally, it is possible to determine how the model is reliable by comparing the failure prediction with actual failure data. [7]

In particular, having a view at the models available in literature, we can distinguish between different kind of models:

- **Electromechanical models:** in this case we have models that describe the behavior of the plant by means of equations that link macroscopic parameters such as forces, currents, torques, etc. This approach results to be very accurate but at the same type they are very time-consuming in terms of computation.
- **Mathematical models:** these are based on the calculation of coefficients of linear and non-linear mathematical functions, needed to interpolate the data obtained experimentally through the measurement of some relevant quantities. The negative aspect is that these functions result not to connect in a natural way the physical quantities between them, often finding relationships that have no real link with the actual dynamic of the plant. [16]

2. State observer

State observer is a very popular approach to system maintenance. For linear systems with additive Gaussian noise terms, KF can be used for prediction. However, when dealing with nonlinear systems with additive Gaussian noise terms EKF are more suitable. For nonlinear systems with non-Gaussian noise terms, the PF also called sequential Monte Carlo method, which are based on the sequential importance sampling (SIS) and the Bayesian theory, lead to a suboptimal solution to state estimation problem [8]

- **KF** is an established technology for dynamic system state estimation that is mostly used in many fields including: target tracking, global positioning, dynamic systems control, navigation, and communication. The KF covers a set of recursive equations that are repeatedly evaluated as the system operates [9]. Any causal dynamic system generates its outputs as some function of the past and present inputs. It is often also convenient to think of the system having a “state” vector (which may not be directly measurable such as the SoH of a machine) where

the state takes into account the effect of all past inputs on the system. Present system output may be computed with present input and present state only, past inputs do not need to be stored. The KF can be viewed macroscopically in this way:

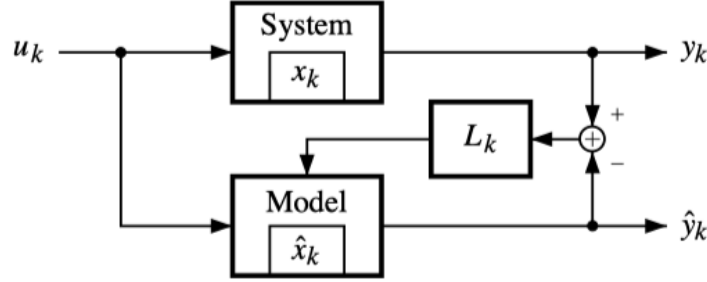


Figure 2.2 Schematic of state update

The true system has a measured input u_k and a measured output y_k . It also has an unmeasured internal state x_k . A model of the system runs in parallel with the true system, simulating its performance. This model has the same input u_k and has output \hat{y}_k . It also has internal state \hat{x}_k , which has known value as it is part of the model simulation. The true system output is compared with the model output, and the difference is an output error, or innovation. This innovation is converted to a vector value by multiplying with the Kalman gain L_k , and used to adapt the model state \hat{x}_k to more closely approximate the true system's state. The state estimate and uncertainty estimates are updated through computationally efficient recursive relationships.

- **EKF** (Extended Kalman Filter) is used in order to deal with nonlinear systems. In practice, it is based on a linearization of the system such that is possible to treat it as a linear time-variant (LTV). Since this algorithm will be widely used during the Thesis work it will be introduced and discussed more in detail in the next phases.
- **Particle filters** are nonlinear state observers that approximate the posterior state distribution using the set of weighted spots, called particles. The particles consist of samples from the states-space and a set of weights which represent discrete probability masses. A better estimate can be obtained by increasing the number of particles. Particle filtering has a wide applicability in fault prediction because of the simple implementation. The algorithm consists of two steps: the first one is state

estimation, and the second one is long-term prediction. The state estimation involves estimating the current fault dimensions and changing parameters in the environment. The next step is the state prediction, which uses the current fault dimension estimate and the fault growth model, to generate state prediction from $(\tau + 1)$ to $(\tau + p)$. Once the long-term prediction is estimated, given the lower and upper bounds of a failure zone (H_{lb} and H_{ub}), the prognosis confidence interval can be estimated.

3. Vibration monitoring

VM is a particular way of analyze the SoH of a machine by using, as obvious, vibrations as an indicator. This technique is particularly used because vibrations bring an high content of information, in the sense that a possible damage is almost instantaneously captured by them. However, vibration-based monitoring applications focus more on *diagnostic* aspects than predicting ones. Nevertheless, in some cases this method can be used and useful for making a prognosis of the system. Thus, looking at the PF-curve in the figure, it is possible to distinguish between a first part on the left, where after a certain time of inspection a point of deterioration observability (P) used for monitoring purpose, and a second part on the right, where the objective is to predict the behavior of the curve till the failure, used for prognosis.[7, 10]

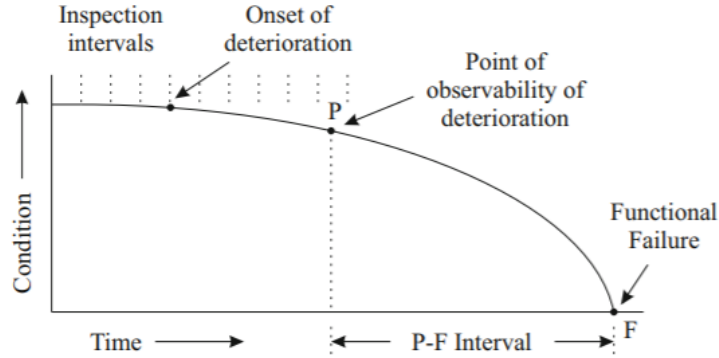


Figure 2.3 P-F curve

4. Moving Horizon Estimation

MHE is a powerful technique for facing the estimation problems of the state of dynamic systems in the presence of constraints, nonlinearities and disturbances [11]. MHE is an optimization approach that uses a series of measurements observed over time, containing noise (random variations) and other imprecisions and produces estimates of unknown variables or parameters. It

requires an iterative method that relies on linear programming or nonlinear programming solvers to find a solution. The basic concept is to minimize an estimation cost function defined on a moving window composed of a finite number of time stages. The cost function includes the usual output error computed on the basis of the most recent measurements and a term that penalizes the distance of the current estimated state from its prediction (both computed at the beginning of the moving window).

5. Learning algorithm

These techniques use measurement signals and their statistics to create non-linear structures which can provide desirable outcomes given the input data. These structures include a wide range of methods, such as principal component analysis (PCA), partial least squares (PLS), artificial neural networks, fuzzy-logic systems and graphical models like hidden Markov models (HMM).

- **ANN** propose methodologies similar to those in the biological nervous system. For a set of available monitoring data which are used as inputs and predefined, known outputs it is possible to use some of the training algorithms, such as backpropagation algorithm, to map the connection between the input and output. Neural networks are selfadaptive structures whose weights between neurons are adjusted by minimizing the criteria to match a model to desired outputs. The training procedure allows the network to learn the relationship among the data without engaging the model of the system. Once the weights are set, the ANN is ready to generate the desired output as a fault evolution prediction.

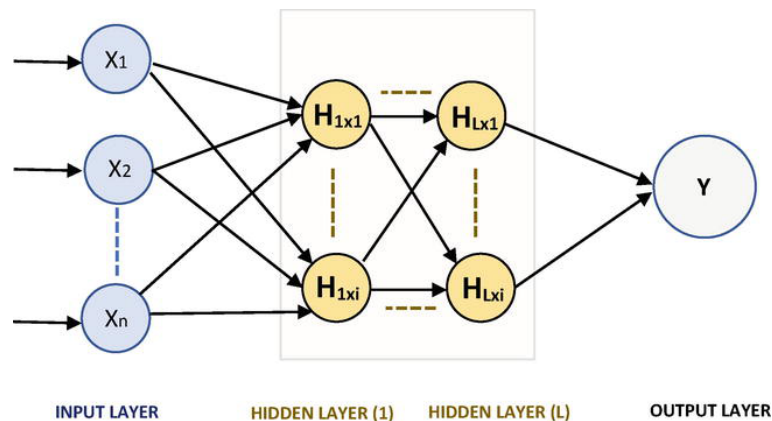


Figure 2.4 Schematic of a Artificial Neural Network

- **Fuzzy logic** also provide mapping between the input and output signals. It can be said to be an extension of the multi-value logic. In a

wider sense, is almost synonymous with the theory of Fuzzy sets, referring to classes of objects with fuzzy boundaries, in which the concept of membership takes on a matter of degree [12]. Unlike neural networks, they are based on linguistic and reasoning human capabilities. By defining the appropriate if-then rules and adjusting membership functions, fuzzy systems can give very accurate prognosis.

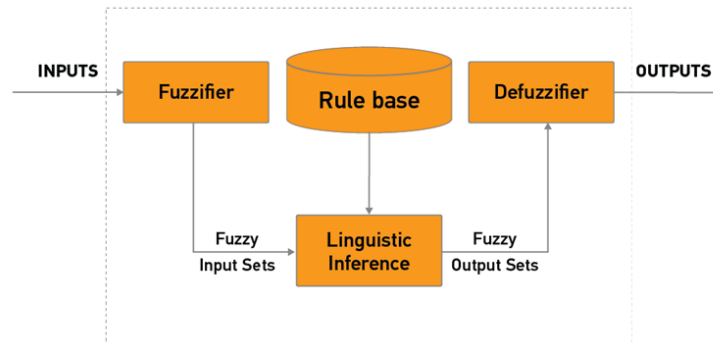


Figure 2.5 Components of a Fuzzy logic system

The common fuzzy logic system processes data in three sequential stages: fuzzification, inference and defuzzification. In the fuzzification step, a crisp, or well-defined, set of input data is gathered and converted to a fuzzy set using fuzzy linguistic variables that is, fuzzy linguistic terms. Second, an inference is made based on a set of rules. Last, in the defuzzification step the resulting output is mapped using so-called membership functions. A membership function is a curve that maps how each point in the input space is related to a membership grade. Using the wear estimation example, various levels of wear in a given set would receive a membership grade between 0 and 1; the resulting curve would not define “new” but instead would trace the transition from worn to new [13].

- **Hidden Markov Models** is a statistical model which can be used to describe system transitions between states. It represents an extension of a regular Markov chain with unobservable or partially observable states. The general structure of a discrete-time HMM with N states, $S = (s_1, s_2, \dots, s_N)$ and M observation symbols, $V = (v_1, v_2, \dots, v_M)$ is shown in the schematic below.

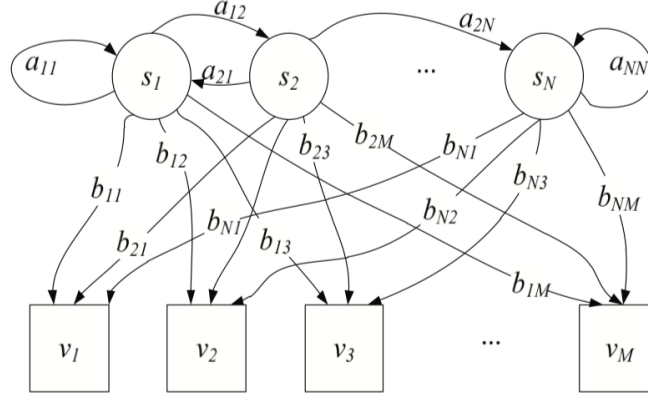


Figure 2.6 Schematic of HMM

The states are interconnected so that a transition between any two states is possible. The hidden state at time t is denoted as q_t and the state-transition rule follows the Markov property, meaning that the state q_t depends only on the state q_{t-1} . The transition matrix $A = \{a_{ij}\}$ stores the probability of state j following state i . The observation matrix $B = \{b_j(k)\}$ shows the probability of observation k being produced from the j -th state. The initial state array $\pi = \{\pi_i\}$ holds the information about initial probabilities; thus, the formulation of HMM is: $\lambda = (A, B, \pi)$.

HMMs can be used to estimate the occurrence of a breakdown, before it happens. Using the Baum-Welch algorithm, HMM can be trained in order to give desired outputs related to system health, for the monitored data inputs. HMM offer a reasonable estimation of the RUL time, meaning the time when the system will be in the specified, faulty state. Also, it is possible to estimate the probability of system being in specified state after n iterations.

6. Frequency domain condition indicators

Another possibility regards the analysis of frequency domain indicators. This kind of research was fundamental for the whole thesis streamline, because it is the base of information extrapolation from signals. A deep study can be done about all frequency domain indicators but the article "Developing a real-time data-driven battery health diagnosis method, using time and frequency domain condition indicators" [14] perfectly sum up the main features in a brief article. This article is about battery health diagnosis, but the main principles can be applied also in the study-case of this thesis. The flow diagram of the construction of condition indicators which is used in the study is depicted in figure 2.7

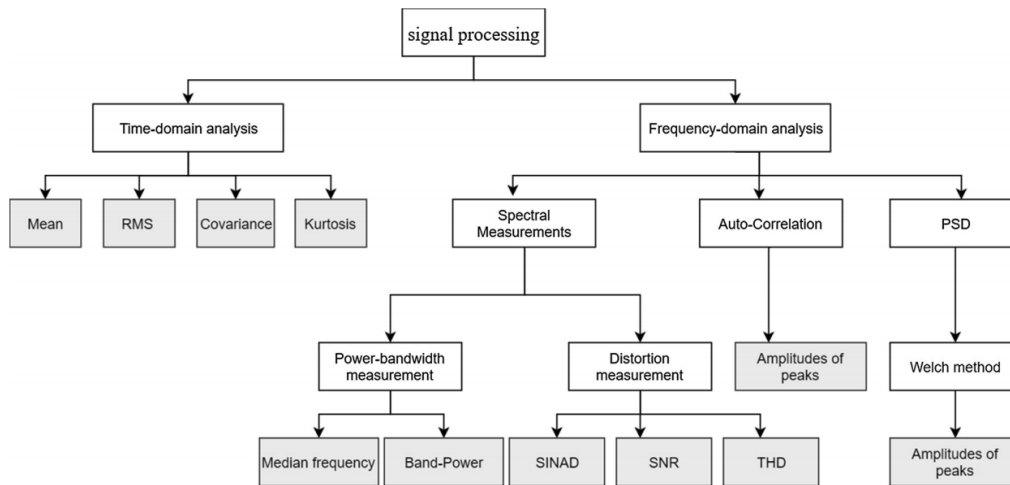


Figure 2.7 Flow diagram of condition indicator construction. The gray boxes indicate the condition indicators.[14]

7. Hybrid algorithm

In the literature it is possible to find some methods that make use of some of the theories exposed so far in order to increase the estimation quality of the SoH. This is done in order to overcome the limitations of a single approach. It will be seen that also in this Thesis work a mixed/hybrid approach will be carried out, using some of techniques exposed above, such as EKF and multimodal analysis.

2.1.1 Past project references

This project research, and the whole thesis work, is part of a continuing evolving series of projects handled by brain Technologies srl. Since the origin of MOREPRO comes from the evolution of some ideas developed in the previous projects, it is necessary to have a preparatory overview of the ideas and the principles make up the past projects.

The projects that precede this work are:

1. The **BAT-MAN** research and development, which is an industrial project owned by brain Technologies and it is the starting point of the application of the innovative approach based on EKF batteries approach and whose main goal is the realisation of an electronic device capable of detecting and forecasting, in real-time, the working conditions of a Lead-Acid battery.
2. The **ERMES** (Extendible Range MultiModal Estimator Sensing), which is an algorithm designed by Brain Technologies whose innovative value is

to identify the methodologies to apply to the problem of the diagnosis of an accumulation system, and in particular to the problem related to the estimation of the SoH of batteries. The proposed ERMES algorithm for the estimate the state of health (SoH) and the state of charge (SoC) is based on the model with the augmented state, which means to consider the uncertain parameters related to SoH and SoC as states and not simply as output. The algorithm involves the generation of a battery model based on an equivalent circuit and a bank of N EKF (Extended Kalman Filter) each based on a different SoH hypothesis. Since this approach is very similar to the one adopted in this thesis, a more detailed explanation of the multi-model and the residual error analysis approach is available in the dedicated chapter of this thesis (chapter ??, reference related to this project is *Virtual Sensing for the Estimation of the State of Health of batteries* [15]).

2.1.2 Comparison between the methods

	Advantages	Disadvantages
Model-Based	High reliable results when the model is accurate	High computational effort
Kalman Filters	High accuracy and online estimation	High calibration and strong hypothesis on the model
Particle Filters	Ease implementation, ability to cope with large scale system	Strong sample size dependence
Vibration Monitoring	Speed of fault detection	Hardly suitable for prognosis scope
Moving Horizon	High noise filtering	Very high computational effort, not able to cope with high dynamics
Learning algorithm	High accuracy and estimation	Need of an huge set of data
Hybrid algorithm	Online estimation, correction of disadvantages of other methods	Strongly depends on the model precision

Table 2.1 Comparison between SoH estimation methods.

3 Physical Model

Mathematical modeling is the art of translating problems from an application area into tractable mathematical formulations whose theoretical and numerical analysis provides insight, answers, and guidance useful for the originating application [16]. Nevertheless, the modeling of a CNC machine can be a very challenging objective, this is due to the complexity and the high number of elements that those technology tools can achieve.

Starting from a blank sheet, the general idea beside this Thesis work is to develop a model able to represent in the most effective way the real condition of the plant under assumption. However, considering the high complexity of a CNC machine, it has been decided to start from a very basic model in order to allow an embryonic prediction algorithm as soon as possible and obtain some effective results from a simple simulation environment. The main objective of the simulation is to understand and emulate the behaviour of a particular manufacturing system on a computer prior to physical production, thus reducing the amount of testing and experiments on the shop floor. By using a virtual system, less material is wasted and interruptions in the operation of an actual machine on the workplace can be avoided. The goal of the modern manufacturing technologies is to produce already the first part correctly in the shortest period of time and in the most cost effective way. Since the product complexities increase and the competitive product life cycle times are reduced, the construction and testing of physical prototypes become major bottlenecks to the successful and economically advantageous production of modern machine tools [17]. It is clear that, in this way, it is possible to discriminate better which parameter/quantity mostly affects the case under assumption. In a second moment, it will be up to the modelling team to further complicate the model in order to have a better adherence to the real case.

As for all mechatronic devices, it is possible to distinguish between a mechanical and an electrical part, parts that are not independent but they work together to exploit the necessary tasks.

3.1 Mechanical part

As regards the mechanical part, a very simple model of a milling machine is used. In particular, having a look at the figure below, a rotational disc is considered that translates in the piece direction in order to cut it.

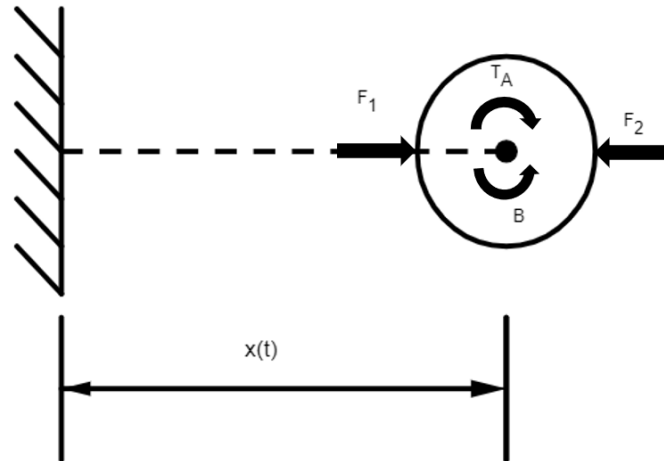


Figure 3.1 Simplified milling machine model

The approach used to get the equations governing the system is the Newton one, a balancing between forces and torques involved. Since the schematic is very simple, defining the various quantities:

- $\dot{\theta}$: Rotational velocity.
- \dot{x} : Linear velocity.
- F_1 : Horizontal force that moves the cutter.
- F_2 : Normal Force due to contact.
- f_c : Binary function that defines the presence of contact. Indeed, it assumes 1 value when the work piece is present or 0 otherwise.
- T_a : DC motor torque applied to the cutter.
- I_n : Inertia of the motor and the cutter.
- β : Contact rotational friction.
- Δ_x : Depth of cutting.
- *cost*: Minimum contact force (introduced in order to avoid model discontinuities).

It is possible to trace the two Newton equations:

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

3.2 Electrical part

For the electrical part instead, modern CNC machines are driven by brush-less or servo motors. The most important characteristics required for the servo motors that drive CNC machines are: fast response to instructions, good acceleration and deceleration properties, the capability to control velocity safely in all velocity ranges and to control very precise the position [18]. Machines with computer numerical control need controllers with high resolution that gives good precision. At this time, both classical and modern control techniques are used, such as PID controllers, feedback control, feedforward control, adaptive control or auto tuning methods.

In order to get a basic framework easy to manage, a DC motor is implemented to drive and interface with the mechanical part. The figure 3.2 shows a simplified DC motor circuit used to pull out the electrical equations.

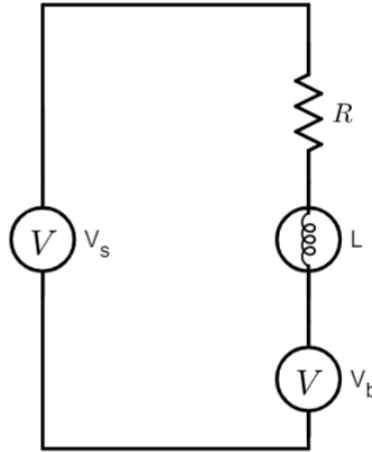


Figure 3.2 Simplified schematic of a DC motor

Defining the following quantities:

- V_s : supply Voltage.
- i_a : Armature current.

- T_a : DC motor torque applied to the cutter.
- k_t : Motor torque proportionality constant.
- L : Inductance.
- R : Resistance.
- V_b : Back E.M.F
- Att_{mot} : Engine friction.
- k : Proportionality constant.
- b : Total flux
- I_L : Motor inertia.

it is possible to derive the equations for the supply Voltage and the Torque applied to the cutter.

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

Thus, playing a little bit with the equations:

$$\begin{aligned} k_t i_a(t) - Att_{mot}\dot{\theta} &= I_L \dot{\theta} \\ V_s &= Ri_a(t) + L \frac{di_a(t)}{dt} + kb\dot{\theta} \end{aligned}$$

Since the supply Voltage is related to the angular velocity through the equation:

$$V_s = \frac{T_a}{k}R + k\omega$$

rearranged for angular velocity:

$$\omega = \frac{V_s}{k} - \frac{T_a}{k^2}R$$

Two main variables affect the speed of the motor in our final equation: the supply Voltage and the Load Torque.

3.3 Plant model

Finally, the electromechanical model used is mainly based on the following dynamic equations:

$$\begin{aligned} \dot{i}_a &= \frac{V_s}{L} - \frac{Ri_a}{L} - \frac{k_v\dot{\theta}}{L} \\ T_a &= k_t i_a(t) - Att_{mot}\dot{\theta} \\ \ddot{\theta} &= \frac{T_a - \beta\dot{\theta}F_c}{I_n} \\ \ddot{x} &= \frac{F_1 - f_c(F_2\Delta_x + cost)}{m} \end{aligned}$$

Thus, replacing the torque equation in the angular acceleration one, the final state equations of the model are obtained:

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

where the states are:

- $\dot{\theta}$: Rotational velocity.
- \dot{x} : Linear velocity.
- i_a : Dc current of the motor.

3.3.1 Simulink implementation

The equations described to date can be translated into model through suitable simulation environment program. For this thesis work, it has been decided to implement the model in Matlab and simulink, because are suitable for this sort of simulations. Implementing such mathematical model using those tools is very intuitive, particularly if the MATLAB Guidelines are followed. Applying these guidelines can improve the consistency, clarity, and readability of your models. The guidelines also help you to identify model settings, blocks, and block parameters that affect simulation behavior or code generation. MATLAB guidelines can

be found in the mathworks site [19].

In the figure 3.3 it is shown how it is implemented a simple contact logic using boolean operators, which is intended to handle the contact with the workpiece.

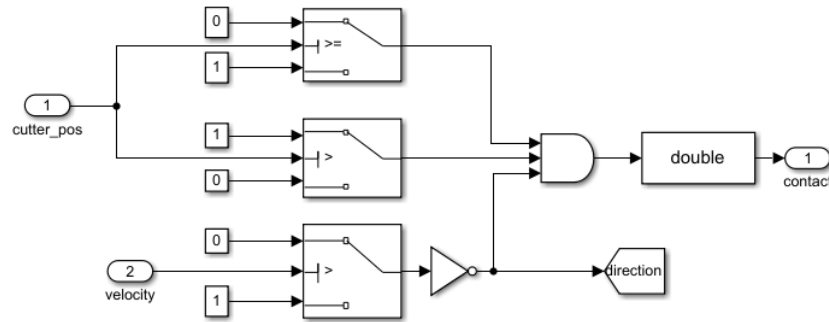


Figure 3.3 Contact logic

For the implementation of the equations described in the paragraphs 3.2 and 3.1, it was decided to create a **dynamic** MATLAB function and to use integrator blocks to integrate the output and feedback where needed:

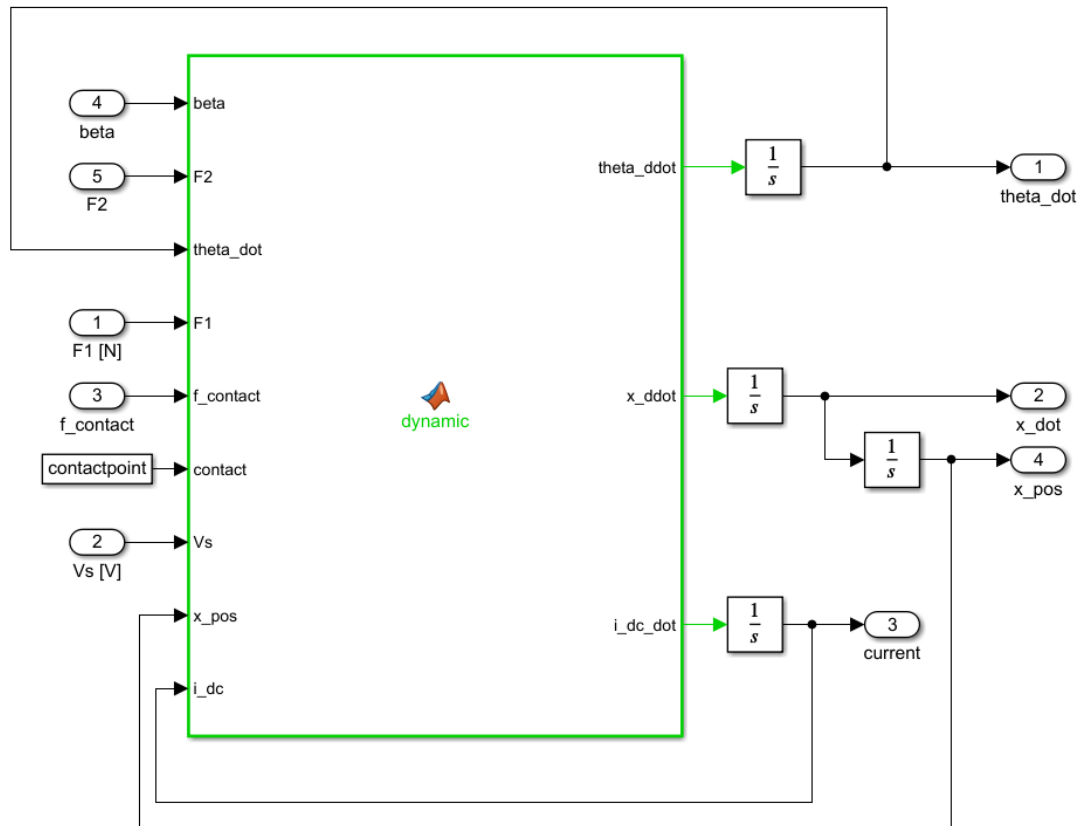


Figure 3.4 Plant Simulink.

Initial condition of the integrators are all set to 0, as well as the starting condition of the contact. The output of the plant coincide with the states of the system:

1. $\ddot{\theta}$: angular acceleration.
2. \ddot{x} : linear acceleration.
3. \dot{i}_{dc} : derivative of the current.

In the next page (figure 3.5), there is the overall simulink implementation of the model.

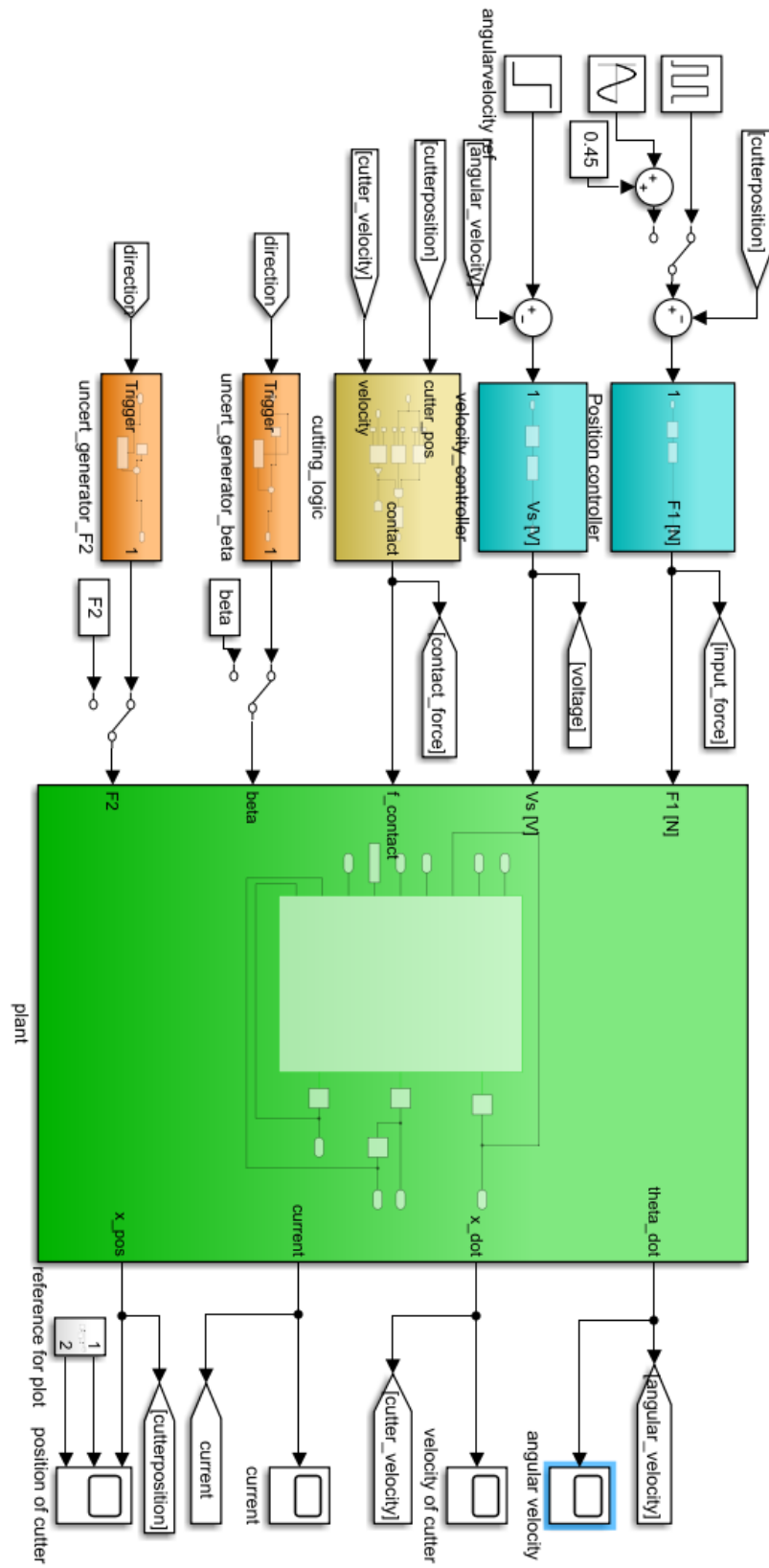


Figure 3.5 Simulink implementation of the model

To sum up, the simulink blocks are:

- Green block: plant implementation.
- Orange block: simple triggers which introduce 2% of uncertainty on the inputs.
- Cyan blocks: PID controllers on the position and angular velocity.
- Yellow block: contact logic.

For what regards the PID controllers, they have been tuned using a MATLAB predefined tool(Control System Toolbox) in such a way to find a good balance between robustness and efficiency. Instead, the next plot represents the contact logic output:

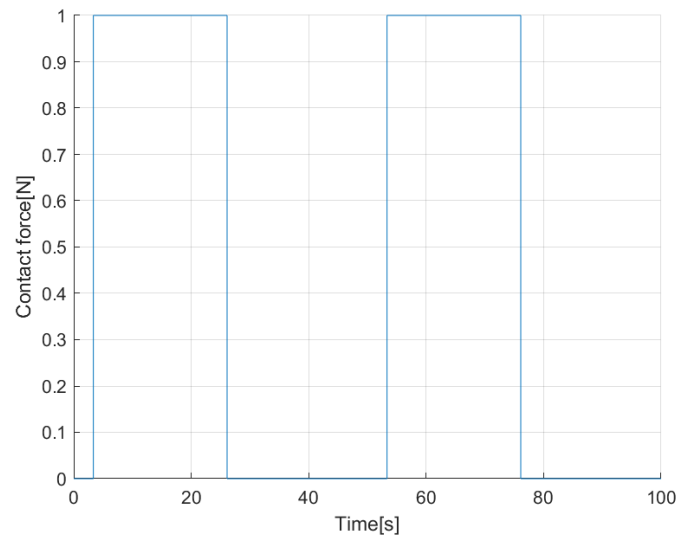
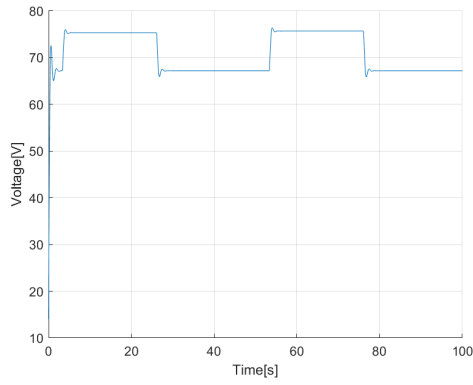
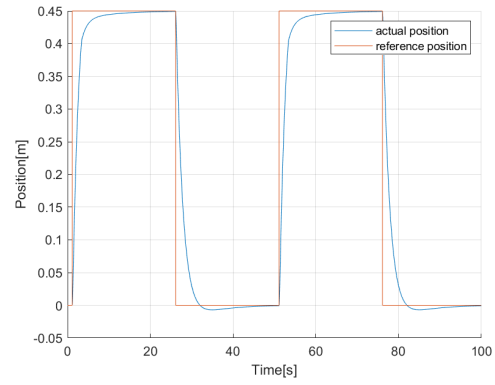


Figure 3.6 Contact force control input plot.

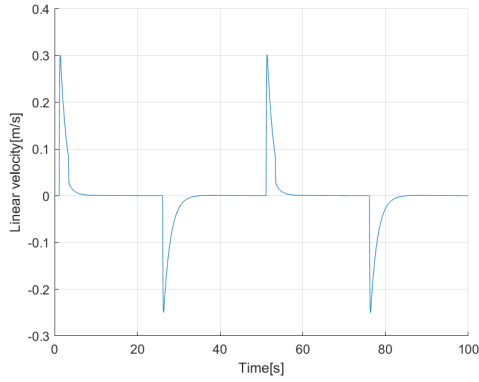
As it was expected, since the contact logic is made up so that the output is 1 when there is contact and 0 when there is no contact, the contact force input plot oscillate in a discrete way between those two values. Finally, in the figure 3.7 are depicted all outputs and main parameters of the model so that the physical behaviour is described.



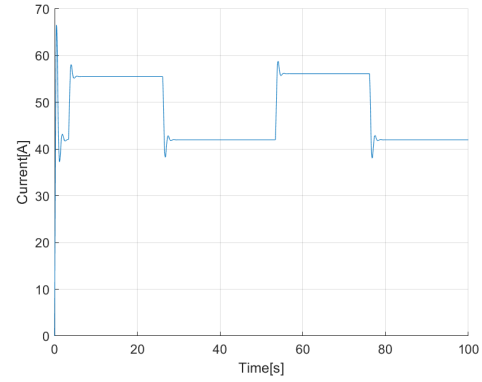
(a) Voltage plot



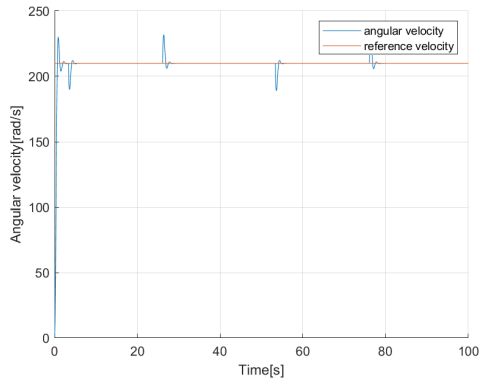
(b) Position plot



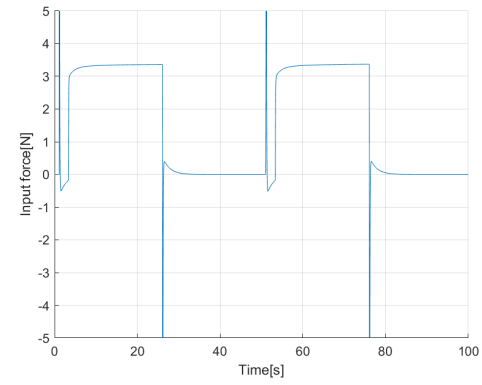
(c) Linear velocity plot



(d) Current plot



(e) Angular velocity



(f) Input force plot

Figure 3.7 Summary plots of Model's main parameters.

3.4 Most significant parameter choice

A literature research on the parameters that most influence end-effector wear has shown that the **friction coefficient** plays a key role in the interaction between the workpiece and the end-effector tool.

In this first phase of development, it is decided to adopt the friction coefficient, referred to as β , as the parameter on which to base the filter wear hypothesis.

From now on, in the multi-model approach and therefore in residual error analysis and evaluation tests, β will be used as a parameter to be estimated and from which to extrapolate the SoH of the machine.

Clearly, the interaction between the tool and the workpiece is more complex than a simple coefficient, since it depends on various factors such as temperature, the used material, relative speed, applied forces, cooling media, etc.

Thus, the key factor is that the analysis carried out during this Thesis work must work independently of the specific choice of the parameter in a way that a consequently complication of it could lead to results that are not so far from the ones obtained considering β as representative. This is the reason why, an in-depth modelling of the interaction will be the goal achieved in Chapter 6.

4 Parameter identification

Since in the first phase of the project there are not real-time signals available from embedded sensors and structural data of the CNC machine, a step further to the development of the physical model, is to identify unknown parameters.

System identification means estimate the possible values of the unknown parameters that are inside the model, in order to build the final model of our dynamical system. There are many possible way to follow but the basic idea to solve this problem is to use the inputs and outputs measured in order to build the mathematic relations between the parameters. The system identification can be done in two possible way:

- gray box model: the model of the system is available but there are different unknown parameters to be estimated.
- black box model: the model is not available, the whole estimate of the parameters is done only building connections between the inputs and outputs values.

The parameters to estimate are the following:

$$\theta = [K_t; I_n; m; F_2; L; Res; K_v]$$

with

- K_t : motor's constant
- I_n : inertia
- m : tool's mass
- F_2 : contact force
- L : inductance of the motor
- Res : resistance of the motor
- K_v : motor's constant

First an identification approach is followed by a colleague using Kalman filters but it turned out that some parameters could not be identified with such an approach. In particular, K_t and I_n are not inside an acceptable range of values, probably because they are not correlated with the collected data from the plant simulation. Therefore, the goal is to employ a set-membership identification for the state equation which allowed such an analysis on the unknown parameters of the model.

4.1 Set-membership identification

Set-membership identification theory is based on the following main ingredients:

- A-priori information about the system to be identified:



Figure 4.1 Block scheme of the system

$\omega(t) = f(\theta, r(t))$, where:

- $u(t)$ =input signal;
- $\omega(t)$ =output signal;
- $r(t)$ =regressor that depends on the past values of $u(t)$ and $\omega(t)$;
- $f \in F$ and F is a given class of function.
- A-priori information on the noise in the collected data:
 - Noise model structure (Equation error, Output error, Error in variable);
 - The noise is assumed to belong to a given bounded set.
- A set of input-output data corrupted by noise (experimentally collected).

The main objective of the Set-membership theory is:

1. to identify a good model for the system under study;
2. to quantify the accuracy of the identified model.

Since the input-output data collected experimentally are uncertain, also the obtained model will be affected by uncertainty.

In the Set-membership identification theory we call **Feasible Solution Set (FSS)** the set of all the models that satisfy the equation of the system given by the a-priori info on the system class for all the collected data corrupted by bounded noise.

When the class of systems F is parametrised by a parameter vector θ , we can replace the FSS with another set called **Feasible Parameter Set (FPS)** that is the set of all the parameter values that satisfy the equations describing the model and it is denoted with D_θ .

The computation of θ_i requests the minimization of a linear function of θ subjected

to a set of linear equalities and inequalities in the variables θ , thus in order to solve a polynomial optimization problem a Matlab relaxation method called SparsePOP is used.

In addition, the **Parameter Uncertainty Intervals (PUI)** should be defined:

$$PUI_{\theta_i} = [\underline{\theta}_i, \bar{\theta}_i]$$

where:

$$\begin{cases} \underline{\theta}_i = \min_{\theta \in D_\theta} \theta_i \\ \bar{\theta}_i = \max_{\theta \in D_\theta} \theta_i \end{cases} \quad (4.1)$$

4.1.1 Application

Considering the state equation of the system:

$$\ddot{\theta} = \frac{k_t i_a}{I_n} - \frac{\beta \dot{\theta}}{I_n} - \frac{Att_{mot} \dot{\theta}}{I_n}$$

it is necessary to convert it into a form that can be suitable to the application of the Set-membership algorithm.

First of all we should move from the time domain to the Laplace domain and write the equation in the form of a transfer function where:

- Output: angular velocity of the cutter
- Input: current of the DC motor

Remembering that β is a dynamic variable, in the context of parameter identification we can assume it to be constant as we are identifying the model parameters at a nominal level. We obtain:

$$\frac{\dot{\theta}}{I} = \frac{k_t/I_n}{s(s + \frac{0.7}{I_n})}$$

where the only parameter to be estimated are k_t and I_n . Now by means of the forward Euler discretization method, with sample time T , we obtain the transfer function in discrete time:

$$\frac{\dot{\theta}}{I} = \frac{T^2(k_t/I_n)}{z^2 + (\frac{0.7T}{I_n} - 2)z - \frac{0.7T}{I_n} + 1}$$

that corresponds to the classical SparsePOP form:

$$\frac{Y}{X} = \frac{\theta_3}{z^2 + \theta_1 z + \theta_2}$$

The parameter vector to be estimated is: $\theta = [\theta_1, \theta_2, \theta_3]$ where:

$$\begin{cases} \theta_1 = \frac{0.7T}{I_n} - 2 \\ \theta_2 = \frac{0.7T}{I_n} + 1 \\ \theta_3 = T^2(k_t/I_n) \end{cases} \quad (4.2)$$

Applying the SparsePOP relaxation algorithm the PUI's values obtained are:

PUI	min	max
θ_1	-2.65	-1.88
θ_2	1.05	1.57
θ_3	-0.16	0.01

Table 4.1 Parameters Uncertainty Intervals

By averaging the minimum and maximum of each interval we have that:

$$\begin{cases} \theta_1 = 2.265 \\ \theta_2 = 1.31 \\ \theta_3 = -0.075 \end{cases} \quad (4.3)$$

4.1.2 Results

Now substituting the results obtained from 4.3 in 4.2 the values estimated are:

- $I_n = 0.125$
- $K_t = 1.68$

These values are in an acceptable range, thus they can be used for the project analysis. To recap, the parameters value used are:

	Used value
K_t	1.5
$F_2[N]$	3
$L \text{ [mH]}$	0.1
$\text{Res [k}\Omega\text{]}$	0.6
$I_n[Kgm^2]$	0.13
K_v	0.2
$m \text{ [Kg]}$	3

Table 4.2 Parameters identification.

5 EKF Bank: multi model approach

State observers are mainly used to provide an estimate of the internal state of a given real system, from measurements of the input and output of the real system. This utilization is very suitable when there is noise and it is needed to be reduced, or when there is a state which cannot be measured directly and there is the necessity of have a more accurate estimation of it. Using a Kalman-filter in order to understand the state and the working condition looking at the residual error is not a common utilisation of such state-observers. What it is need to be done for this scope is a deep analysis of the residual errors. The latter, are defined as the module of the difference between a state estimation and the real state:

Supposing that $x(t)$ is a state of a system $\mathcal{M}(x(t))$:

$$Residual_error = |\hat{x}(t) - x(t)| \quad \forall t \quad (5.1)$$

The **residual error** can seem very similar to an **estimation error**, but there is a slight but very important difference. On one hand, the residual error is the difference between the state estimation and the state coming from the output of the real plant, so there is no need to know the internal exact formulation of the plant. On the other hand, the calculation of the estimation error suppose to perfectly know the real value of the parameter to be estimated. This difference is rather crucial, because it is not possible to suppose the real value of the internal state of the system. Moreover, it is important to mention that the residual error can be affected by measurement noise. For this thesis work it was supposed to have a really low measurement error on the states because the approach is intended to be as much simple as possible at first.

Starting from the considerations done until now, is it possible to relate the residual error to a state or to a set of parameters that can represents the SoH?

How much the other parameters changes affect the residual error calculation?

Which is the state on whom the difference of the residual errors are more highlighted?

Is it really possible to apply the multimodel approach for the estimation of SoH?

The questions that need to be solved to answer to the last one are numerous, and during this chapter there will be the proof of concepts and some possible answer to the listed questions, mainly based on empirical approach.

The starting question is: What is the effect of a parametric variation on the residual error, and which is the weight of this variation? Initially, the focus was the search for papers and/or documents that take into account the effects of parametric variations on the residual error produced by the observer: a possibility is

to consider faults as parametric variations that induce a change in system behavior. Nevertheless those kind of approach are quite time-consuming because require strong theoretical analysis. Another available option, rather more practical, is the search for a method that foresees a sensitivity analysis with the aim of identifying which are the parameters whose variations have a relevant effect on the output and consequently these parameters could be used as criteria to do the scheduling and eventually decide which will be the partition method for the state space. Sensitivity analysis is the method most frequently used during research on this topic and seems to give the best results. This method consists in getting a many data from experiment strongly varying the condition, so that there is a strong background where a a global sensitivity analysis(GSA) can be performed[20].

5.1 EKF

The Extended Kalman filter is a method to estimate both the states of the system and also his parameters; it is an iterative procedure, composed by different equations that are fast evaluated as the system changes during time. In each step there is the estimation not only of the system states but also of the covariance matrix, indicator of the uncertainty of the states estimate. A "large" value of covariance indicates a high level of uncertainty while a "small" one indicates confidence in the estimate. As seen previously, our system is represented by the following state equations:

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

where the states are:

- $\ddot{\theta}$: rotational acceleration
- \dot{x} : linear velocity
- i_a : dc current of the motor

We can notice the form of a classical nonlinear system $\dot{x} = f(x, u)$ and starting from the following state-space model in a discrete-time domain:

$$\begin{cases} x_{k+1} = f(x_k, u_k) + \omega_k \\ y_k = h(x_k) + v_k \end{cases} \quad (5.2)$$

where x_k are the states, u_k are the inputs, y_k is the output, ω_k is the disturbance and v_k is a measurement noise. $f(\cdot)$ is a nonlinear state transition function that

describes the evolution of states x from one time step to the next. The nonlinear measurement function $h(\cdot)$ relates x to the measurements y at time step k . At each time step, $f(x_k, u_k)$ and $h(x_k)$ are linearized by a first-order Taylor-series expansion. We assume that $f(\cdot)$ and $h(\cdot)$ are differentiable at all operating points (x_k, u_k) .

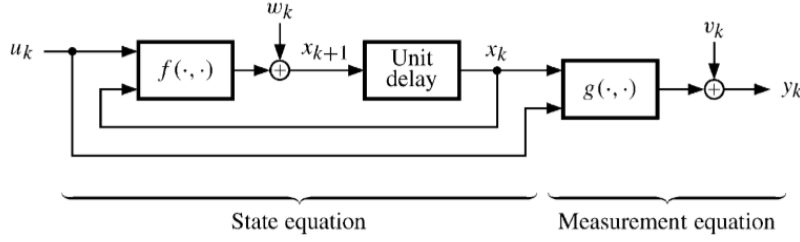


Figure 5.1 Diagram of nonlinear discrete time system in state-space form

The inputs u_k are:

- V_a : armature voltage
- F_1 : horizontal force that moves the cutter
- F_c : function that define the contact with the object.

The states x_k are the same of the plant model while the output y_k we suppose to coincide with the states. We must define the following quantities:

- $F_k = \frac{\partial f}{\partial x_k}(x_k, u_k)$ = Jacobian of f computed in (x_k, u_k)
- $H_k = \frac{\partial h}{\partial x_k}(x_k)$ = Jacobian of h computed in x_k
- \hat{x}_k = estimate of x_k computed at step k
- x_k^p = prediction of x_k computed at step $k-1$
- P_k = covariance matrix of $x_k - \hat{x}_k$
- Q^d = covariance matrix of ω_k
- R^d = covariance matrix of v_k

As regards the matrices Q^d and R^d , since we have no information on the disturbances, we chose them as diagonal matrices by a trial and error procedure. The algorithm can be summarized with the following step:

1. Prediction

way to treat a signal. One of the most complete article is the one cited in the state of the art chapter of this thesis [14]. This article plainly explain how to analyze residual errors using both frequency domain and time domain indicators. Among all, some of the most simple and efficient according to the article are:

- Mean.
- Integral.
- RMS.
- Correlation.
- PSD.
- Covariance.

To verify which is the best, it can be applied and experimental approach. In particular, it is possible to set up some test to verify which of this methods, applied on the residual error, it is most suitable. It must be keep in mind that the objective is to find a method that can highlight the difference between changing of beta. That's because the aim is to make the system very sensitive to little change of beta, but confidently less sensitive to other parameters variations. So the approach will be to test 20 little variation of beta, starting from the nominal condition and increasing of 20% every step. It will also be reported a little variation on the horizontal input force of about 2%. The nominal values (calculate with beta nominal) of the errors elaborated for each state and for each considered are reported in the following table:

Method	Nom. Rot. acc.	Nom. linear acc.	Nom. curr. der.
Mean	1.7111	0.0242	0.4959
RMS	3.6821	0.0509	1.1879
Correlation	1.3979	1.0000	3.8928
covariance	10.6403	0.0020	1.1663
integral error	58.7780	1.2503	33.7761

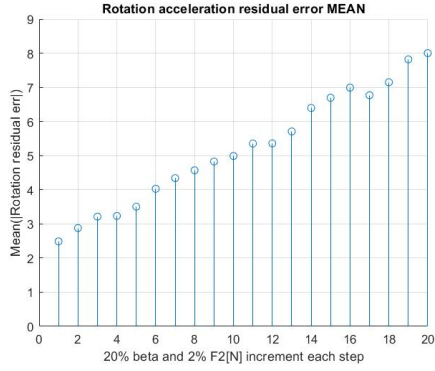
Table 5.1 Nominal values of the errors for each method.

In order to understand the results, it is also defined a FOM (Figure of merit) as a simple index that describe how far the non-nominal condition model is with respect to the nominal one. This FOM index is the ratio between the absolute value of the residual error and the absolute value of the residual in nominal condition:

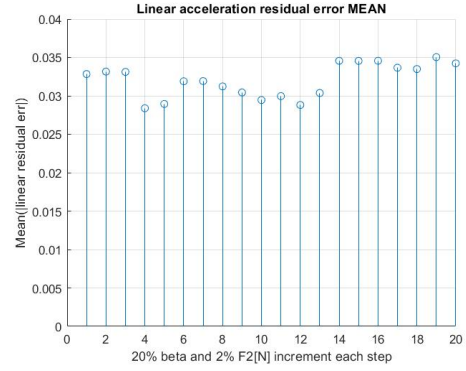
$$FOM = \frac{|\text{Residual error parameter}|}{|\text{Nominal residual error parameter}|}$$

Keeping in consideration the nominal values reported in the table 5.1, the following plot will show which signal manipulation can be considered as the most suitable. Let's start the tests from each method:

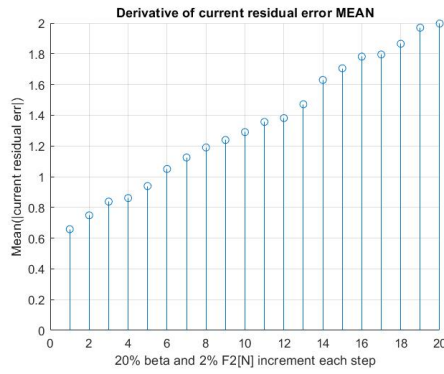
1. MEAN:



(a) Rot. acc.



(b) Lin. acc.



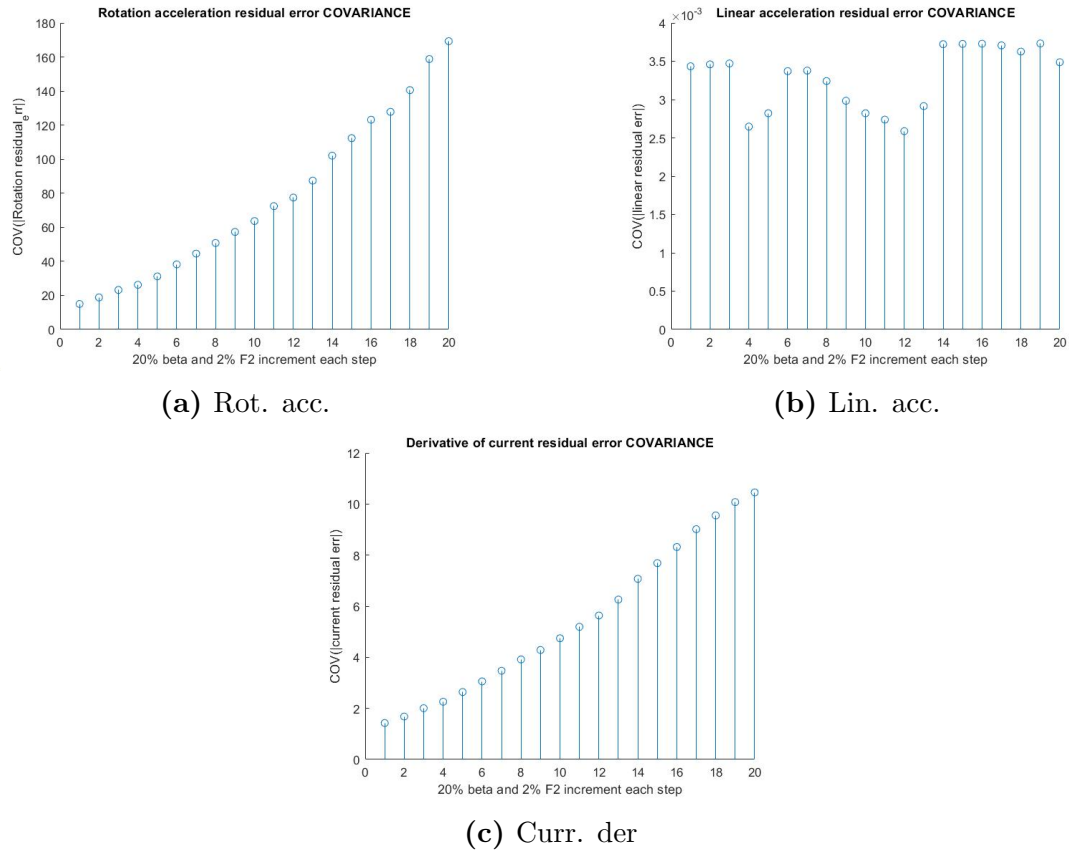
(c) Curr. der

Figure 5.3 Mean test.

FOM angular	FOM linear	FOM current
4.6759	1.4162	4.0266

Table 5.2 FOM mean

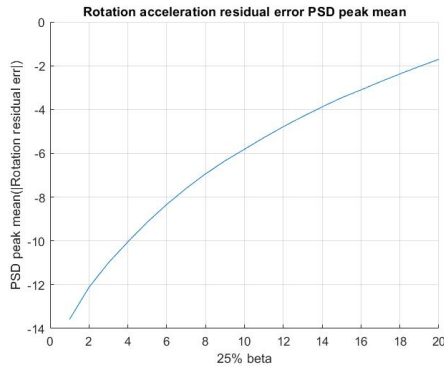
2. Covariance:

**Figure 5.4** Covariance test.

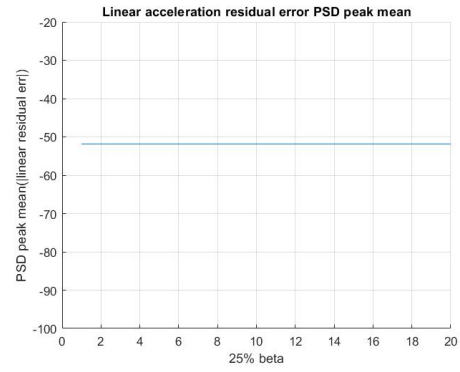
FOM angular	FOM linear	FOM current
5.9107	1.7358	7.9633

Table 5.3 FOM Covariance

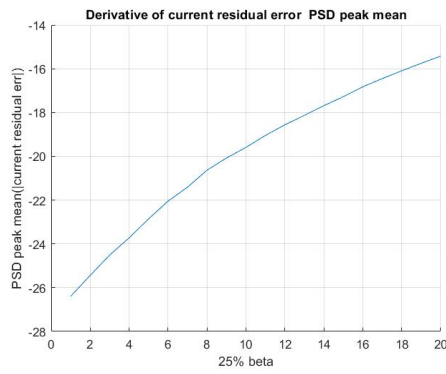
3. **PSD:** For the power spectral density, it has been considered an interpolation of the maximum peaks.



(a) Rot. acc.



(b) Lin. acc.



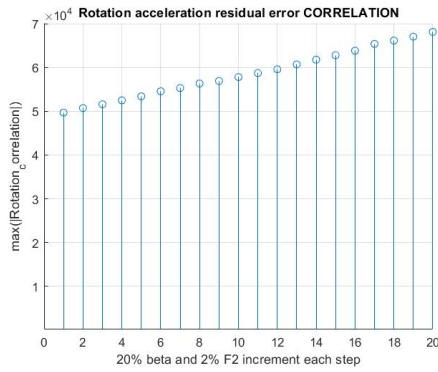
(c) Curr. der

Figure 5.5 PSD test.

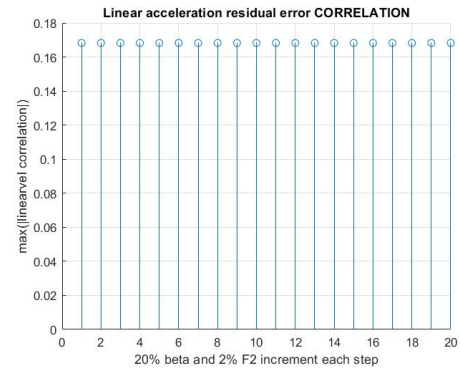
FOM angular	FOM linear	FOM current
7.527	1.001	5.743

Table 5.4 FOM PSD

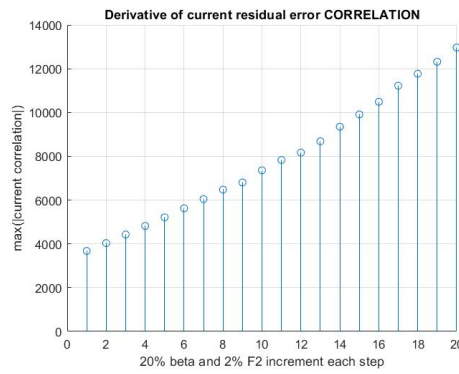
4. Correlation:



(a) Rot. acc.



(b) Lin. acc.



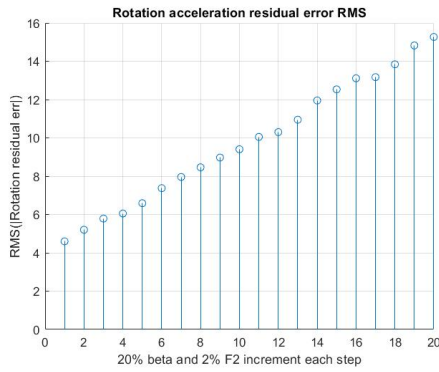
(c) Curr. der

Figure 5.6 Correlation test.

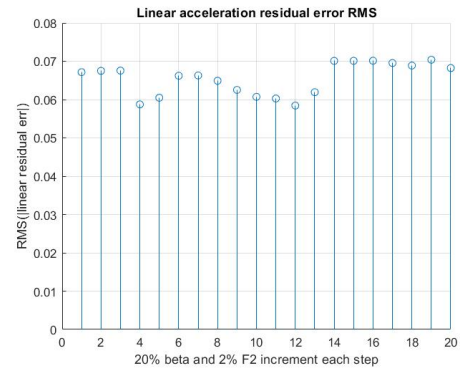
FOM angular	FOM linear	FOM current
1.3979	1.0000	3.8928

Table 5.5 FOM Correlation

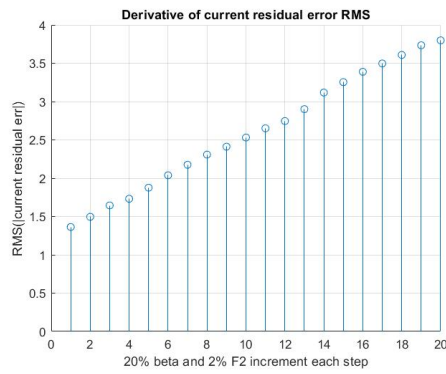
5. RMS:



(a) Rot. acc.



(b) Lin. acc.



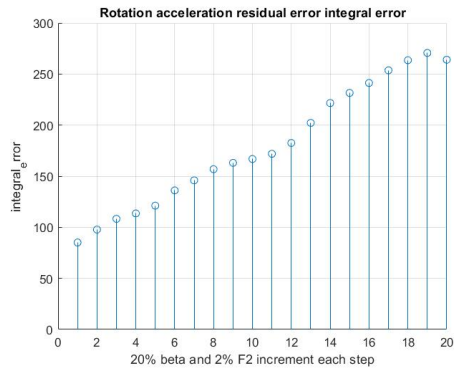
(c) Curr. der

Figure 5.7 RMS test.

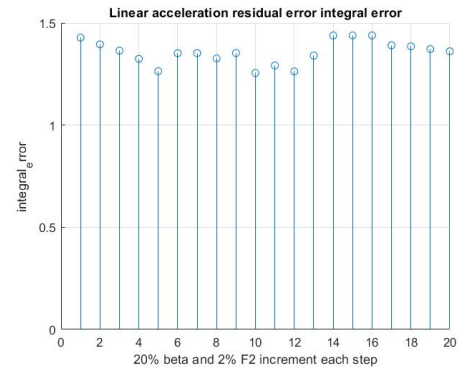
FOM angular	FOM linear	FOM current
4.1469	1.3404	3.1979

Table 5.6 FOM Correlation

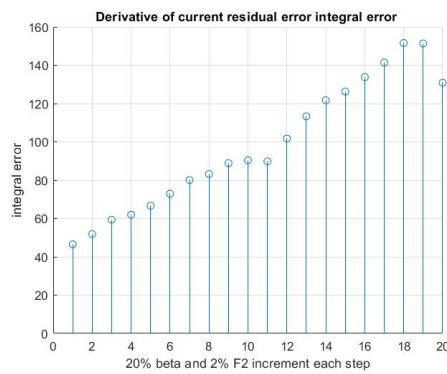
6. Integral:



(a) Rot. acc.



(b) Lin. acc.



(c) Curr. der

Figure 5.8 Integral test.

FOM angular	FOM linear	FOM current
4.4911	1.0886	3.8739

Table 5.7 FOM Integral

Results can be summed up in the following tables:

- Angular acceleration:

	<i>mean</i>	<i>integral</i>	<i>RMS</i>	<i>cov</i>	<i>corr_{max}</i>	<i>PSD</i>
$\beta \uparrow$	1.93	58	5.8	30	6.1	-19.9
	2.05	61	6	32	6.4	-18.6
	2.19	66	6.2	34	6.8	-17.4
	2.32	70	6.5	36	7.2	-16.8
\Downarrow	2.44	73	6.7	39	7.6	-15.9
	2.55	77	6.9	41	8	-15
	2.76	80	7.2	44	8.4	-14.2
	2.88	84	7.4	46	8.7	-13.5
	2.98	87	7.7	50	9.1	-12.7
	3.1	91	7.9	53	9.6	-12.2

- Linear acceleration

<i>mean</i>	<i>integral</i>	<i>RMS</i>	<i>cov</i>	<i>corr_{max}</i>	<i>PSD</i>
0.005	0.15	0.01	0.002	0.15	-63

The results on the linear acceleration, varying F_2 , differ so little from the results obtained with the nominal values that they are irrelevant.

- Current derivative

	<i>mean</i>	<i>integral</i>	<i>RMS</i>	<i>cov</i>	<i>corr_{max}</i>	<i>PSD</i>
$\beta \uparrow$	0.67	20	1.8	3.0	9.2	-25.9
	0.71	21	1.9	3.2	9.2	-25.8
	0.75	22	2.0	3.5	9.2	-25.2
	0.79	23	2.1	3.9	9.2	-24.9
	0.83	25	2.2	4.3	10	-24.8
\Downarrow	0.87	26	2.3	4.7	11	-24.3
	0.91	27	2.4	5.2	13	-23.7
	0.95	28	2.5	5.6	15	-23.2
	0.99	29	2.6	6.2	16	-22.6
	1.03	31	2.8	6.7	18	-22.6

Beyond the good results for the integral error reported above, most of the methods appear suitable for the scope. Indeed, choosing to utilize the rotation acceleration error or the current related error, there is not a method that really take advantages on the others. The RMS method and the integral are almost equivalent in terms of

FOM, that indicates that are both good for the aim. So the decision to choice for the integral error method comes from another consideration: the **integral error** is cumulative and takes into account the previous state of the system. The RMS is very good and will be used to elaborate the errors as well as the integral error. Nevertheless, considering that mechanical system states naturally needs time to evolve and change, taking into account a cumulative way to treat the residual error is definitely the best choice. The integral error behaviour will be widely treated from this point until the end of this thesis work.

5.2.1 Residual error comparison

Once decided that the integral of the residual errors is the most suitable choice to carry out a multivariate analysis, it is possible to see what of the variables available contain more information. This is done using a simulation environment composed of:

- The **Plant Model** obtained in 3.3.
- The **EKF** described in 5.1.
- A logic of management and decision of the integral of the residual errors that contains a possible integral reset as it will be seen.

The estimator allows the absolute error computation of the angular acceleration error and of the derivative of the current. Thus, exploiting a boxplot analysis on the integral of such errors, it is possible to decide which kind of error best describe our model. The simulation is carried out considering the nominal parameters of the machine, described in the table below.

	Nominal value
Mass [kg]	3
Radius [m]	0.3
Resistance [k Ω]	0.6
Inductance [mH]	0.1
Torque constant	1.5
Voltage constant	0.2
Motor Inertia [kgm ²]	0.001
Friction coefficient	0.1

Table 5.8 Nominal CNC parameters.

The same machine is considered to work in nominal condition when:

	Nominal value
Angular velocity reference $[\frac{rad}{s^2}]$	210
Position reference [m]	0.5
Duty cycle [%]	50
Number of cycles	4
Contact point [m]	0.4
Workpiece length [m]	0.09

Table 5.9 Nominal working conditions.

In the same environment, a variable e_c is defined and used to discriminate which of the residual errors available will be considered. In particular:

- $e_c = 1$: only the angular acceleration error is considered.
- $e_c = 2$: only the current derivative error is considered.
- $e_c = 3$: an average between the two errors is computed and considered.

Thus, considering a one hundred seconds simulation, the friction coefficient β of the plant is made to change between five different values while the filter one is keep fixed to the nominal one. In this way, analyzing the boxplots of the three different errors it is possible to see which of the three one allow a better distinction between the various friction coefficient cases.

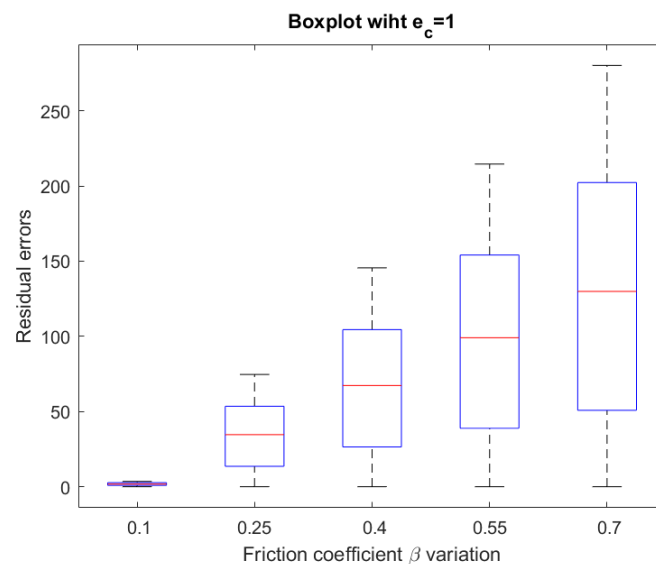


Figure 5.9 Boxplot of the angular acceleration error.

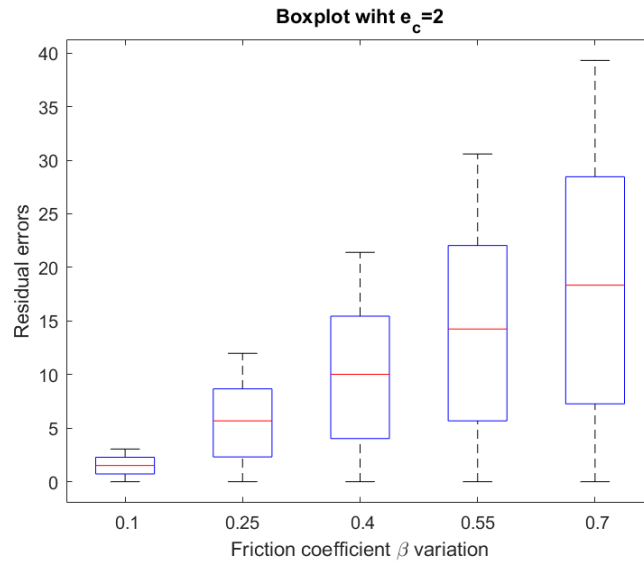


Figure 5.10 Boxplot of the current derivative error.

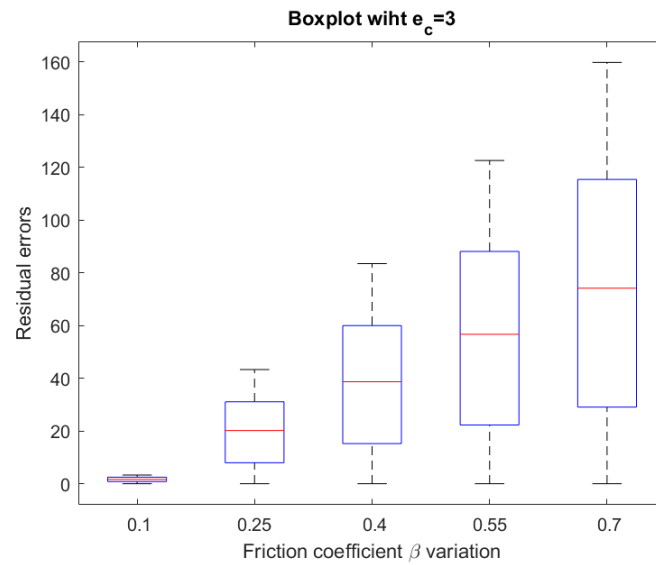


Figure 5.11 Boxplot of the average error.

Having a look at the results, it is possible to see that in the first and in the last case a better separation between the matched value and the other ones is obtained. On the contrary, considering the derivative of the current the separation is not so marked as the other ones. Thus, a first suggestion is that the e_c variable must be set to 1 or to 3 to obtain more remarkable results. Moreover, focusing just on

these values, when considering the angular acceleration, looking the median values of the boxes, an higher distance between the nominal error is obtained. Finally, it is possible to conclude that when setting e_c equal to one better results will be expected in the next.

5.3 Evaluation tests

Till now, all the simulations were carried out assuming that the machine always exploit the same kind of lavoration. Thus, it is convenient to test/stress the environment with different input conditions in order to see if the state observer works well in any case and which kind of processing affects more the algorithm. In particular, a kind of multivariate error analysis is made, changing one variable at a time:

- **Angular Velocity**

With all the other parameters fixed, only the angular velocity is made to change:

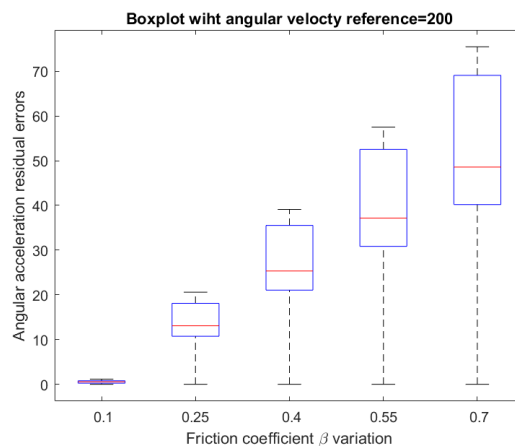


Figure 5.12 Boxplot error with 200 $[\frac{rad}{s^2}]$ angular velocity.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.10 Test with 210 [$\frac{rad}{s^2}$] angular velocity.

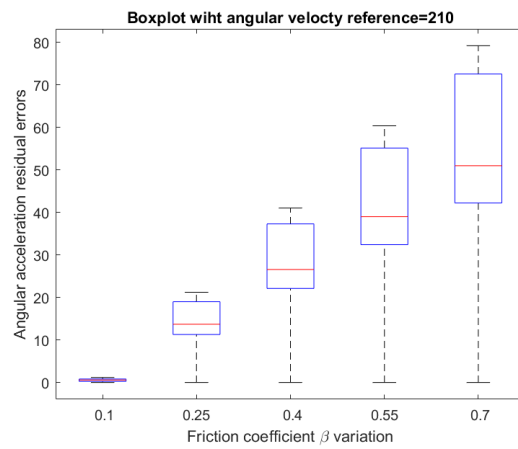


Figure 5.13 Boxplot error with 210 [$\frac{rad}{s^2}$] angular velocity.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	220
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.11 Test with 220 [$\frac{rad}{s^2}$] angular velocity.

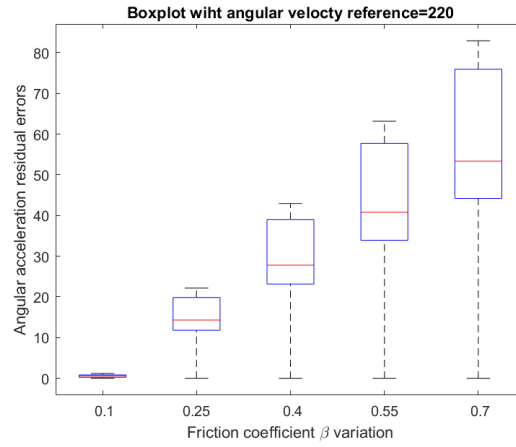


Figure 5.14 Boxplot error with $220 \frac{rad}{s^2}$ angular velocity.

	Nominal value	Testing value
Angular velocity $[\frac{rad}{s^2}]$	210	230
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.12 Test with $230 [\frac{rad}{s^2}]$ angular velocity.

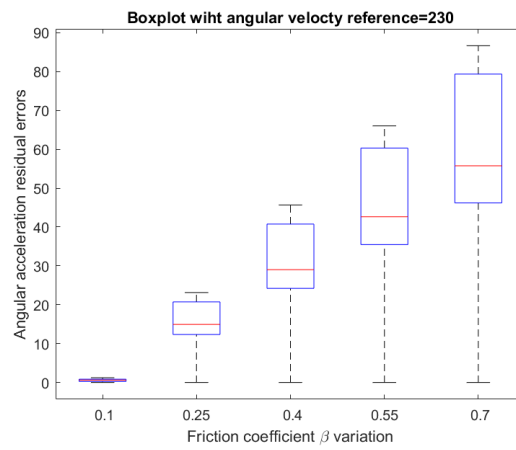


Figure 5.15 Boxplot error with $230 [\frac{rad}{s^2}]$ angular velocity.

- **Position**

With all the other parameters fixed, only the position reference is made to change:

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.4
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.13 Test with 0.4 [m] position reference.

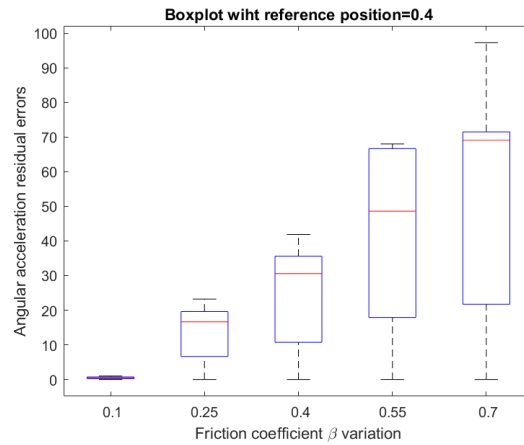


Figure 5.16 Boxplot error with 0.4 [m] position reference.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.47
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.14 Test with 0.47 [m] position reference.

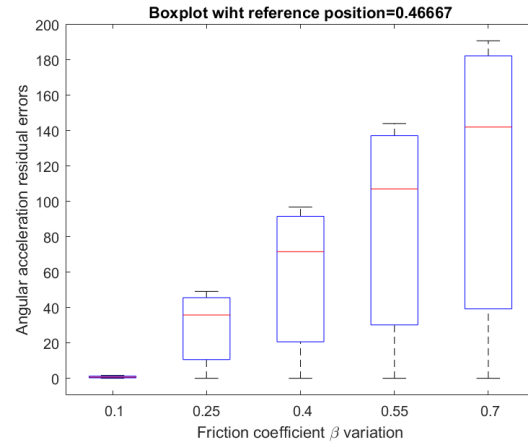


Figure 5.17 Boxplot error with 0.47 [m] position reference.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.53
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.15 Test with 0.53 [m] position reference.

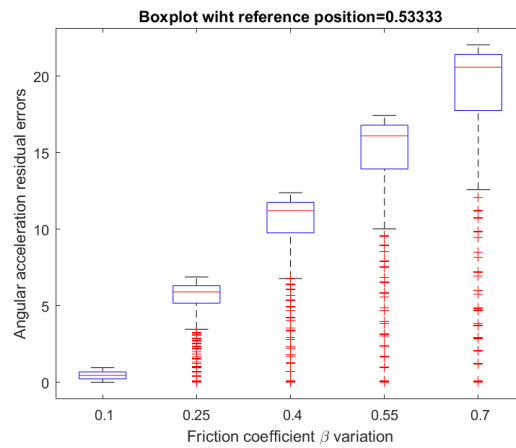


Figure 5.18 Boxplot error with 0.53 [m] position reference.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.6
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.16 Test with 0.6 [m] position reference.

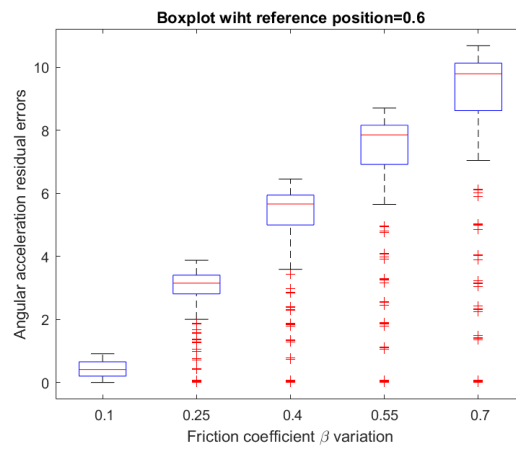


Figure 5.19 Boxplot error with 0.6 [m] position reference.

- **Duty cycle**

With all the other parameters fixed, only the duty cycle is made to change:

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	20
Number of cycles	4	4

Table 5.17 Test with 20 % duty cycle.

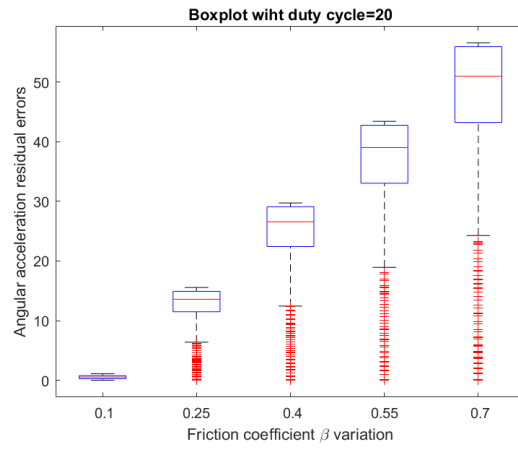


Figure 5.20 Boxplot error with 20 % duty cycle.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	40
Number of cycles	4	4

Table 5.18 Test with 40 % duty cycle.

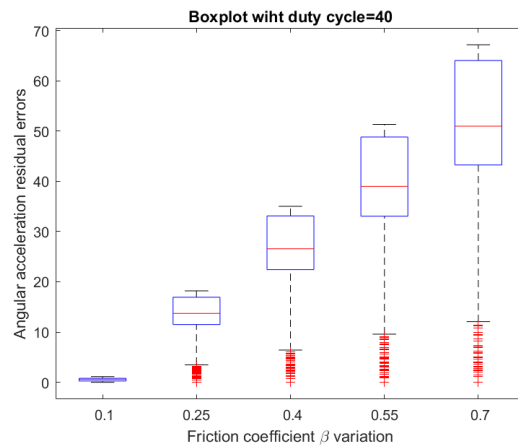


Figure 5.21 Boxplot error with 40 % duty cycle.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	60
Number of cycles	4	4

Table 5.19 Test with 60 % duty cycle.

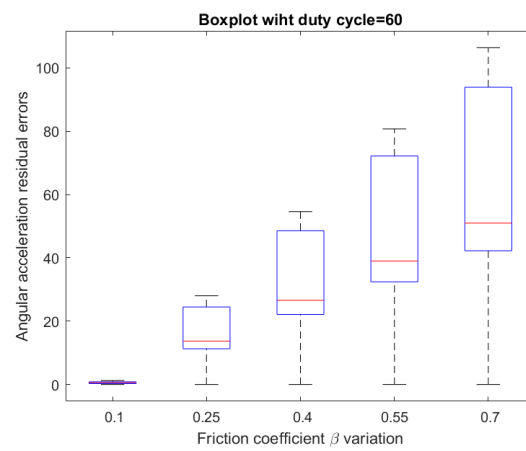


Figure 5.22 Boxplot error with 60 % duty cycle.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	80
Number of cycles	4	4

Table 5.20 Test with 80 % duty cycle.

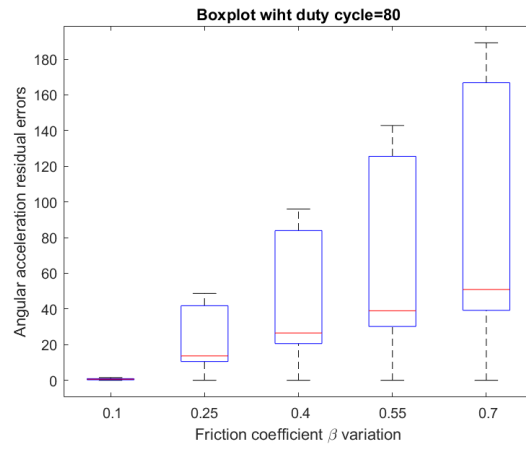


Figure 5.23 Boxplot error with 80 % duty cycle.

- **Number of cycles**

With all the other parameters fixed, only the number of cycles is made to change:

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.21 Test with 4 number of cycles.

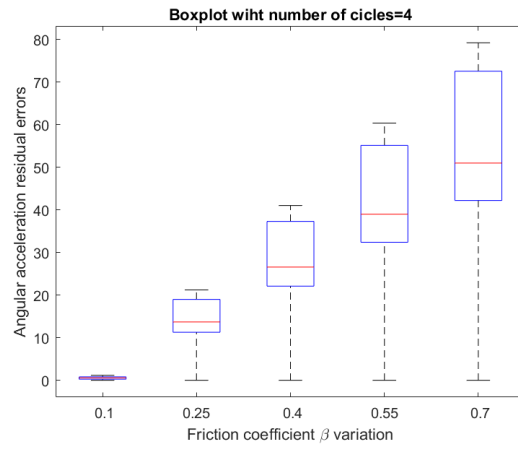


Figure 5.24 Boxplot error with 4 number of cycles.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	6

Table 5.22 Test with 6 number of cycles.

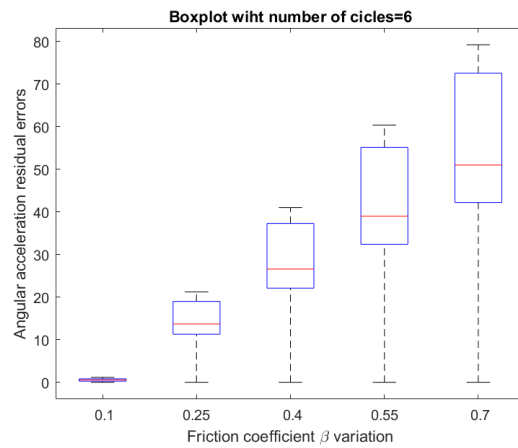
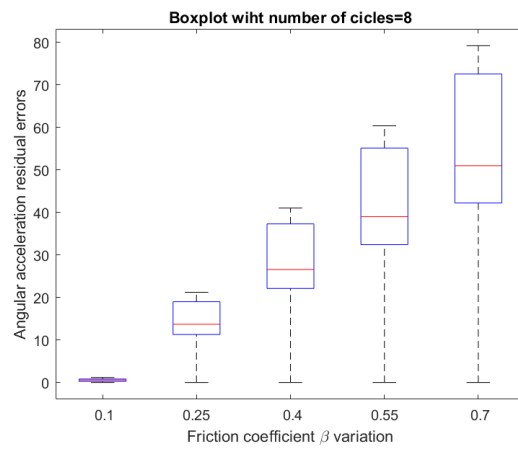


Figure 5.25 Boxplot error with 6 number of cycles.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	8

Table 5.23 Test with 8 number of cycles.

**Figure 5.26** Boxplot error with 8 number of cycles.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Position [m]	0.5	0.5
Duty cycle [%]	50	50
Number of cycles	4	10

Table 5.24 Test with 10 number of cycles.

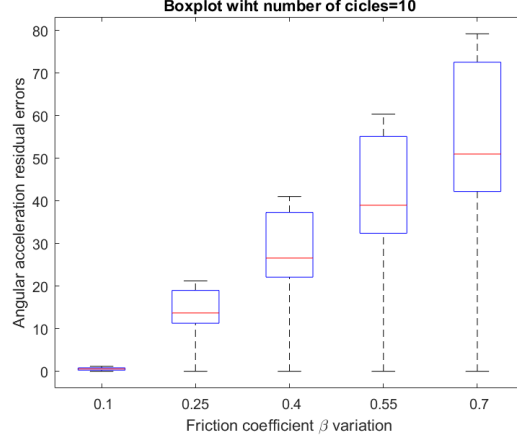


Figure 5.27 Boxplot error with 10 number of cycles.

Having a look at the various tests performed so far, it is possible to see that there are no sensible variation when considering different working conditions with respects to the nominal ones. Thus, the error associated with the correct model to estimate is always smaller then the other ones. Moreover, in some particular cases there is a better distinction between the boxplots, indicating a more accuracy on the estimation algorithm. Moreover, it is necessary to state that during these tests a reset of the integral error was considered, whose choice is justified in 5.3.1.

5.3.1 Reset time choice

A crucial aspect of residual error analysis is the choice of the integral's reset time. An integration period should be chosen mainly for two reasons:

- clearly, after a certain period of time while it is growing up, it will reach its maximum value distorting the results;
- there may be situations in which there are transient errors depending on many factors such as the work period, the type of machining process, ecc. that can influence the integral.

Thus, in order to choose an optimal reset time, a boxplot analysis was carried out by varying it through the simulation range.

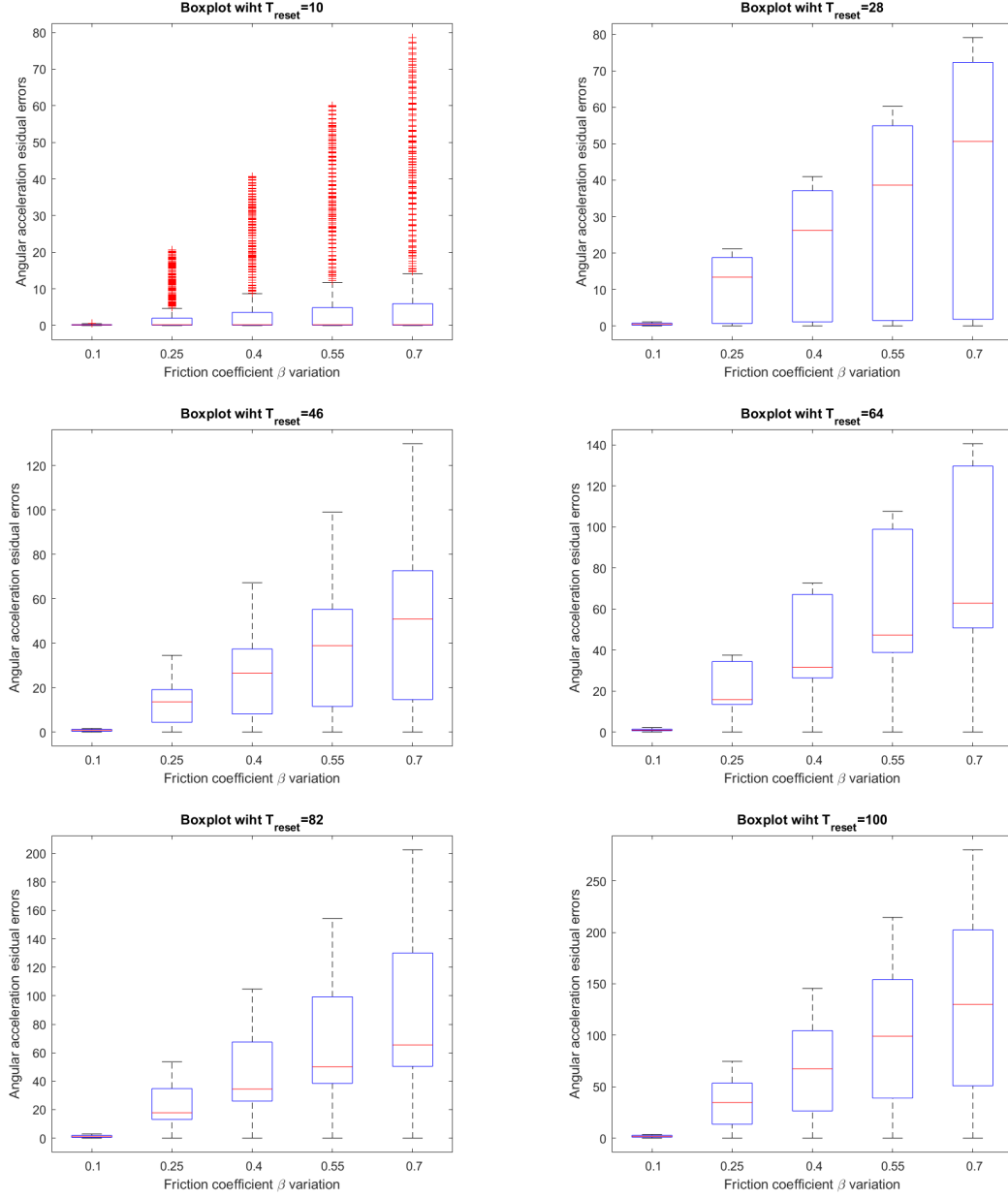


Figure 5.28 T reset analysis

From the boxplots above can be seen that with a reset time of 10s, the error variation is quite small compared to the others. This is probably due to the fact that in 10s time there are no sensible dynamic variations in the system that would capture an estimation mismatch. From 30s onwards the results are quite similar but we decide to investigate a narrower range right after 30s because by increasing more and more the reset time we can run into the problems listed above.

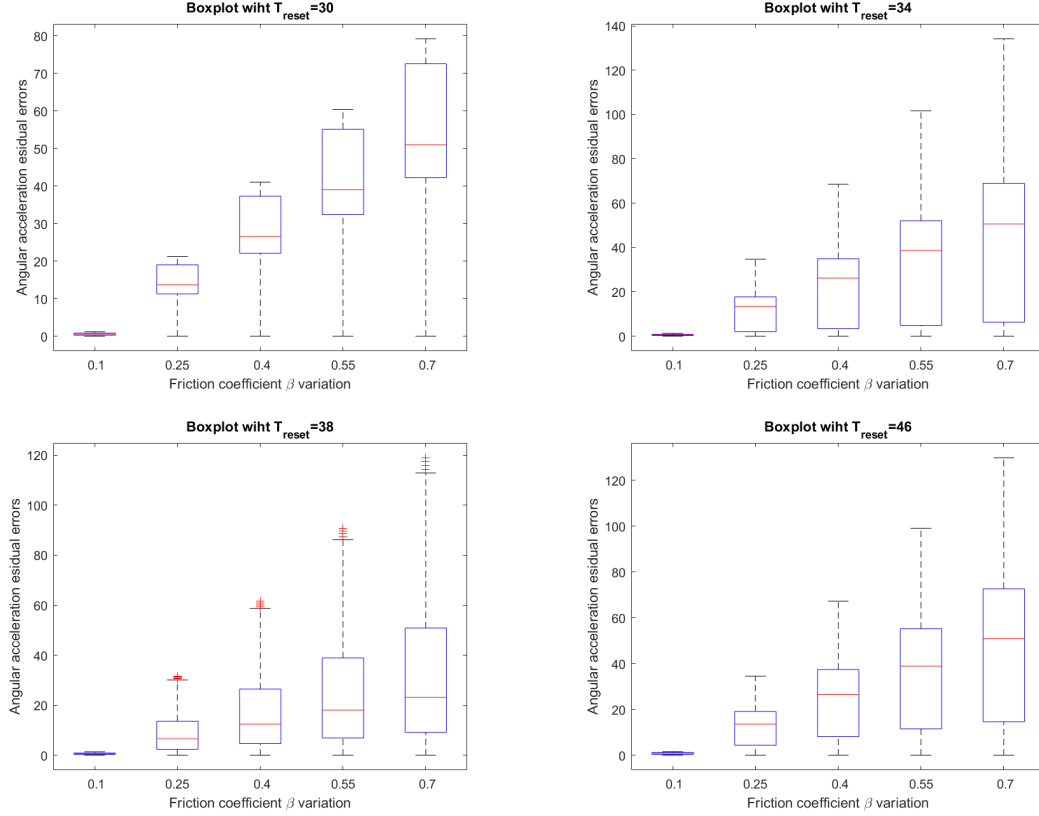


Figure 5.29 T reset choice

The results highlight that choosing a reset time of 30 seconds is the best choice as far as our framework is concerned, because it corresponds to a processing period. In practice we do not know exactly how long a processing period takes so a better choice could be to choose a reset time large enough to capture the dynamic variations of the system concerned.

In the figure 5.30 we can notice the behaviour of the angular acceleration error's integral with nominal values for all the parameters when a Reset time of 30s has been chosen.

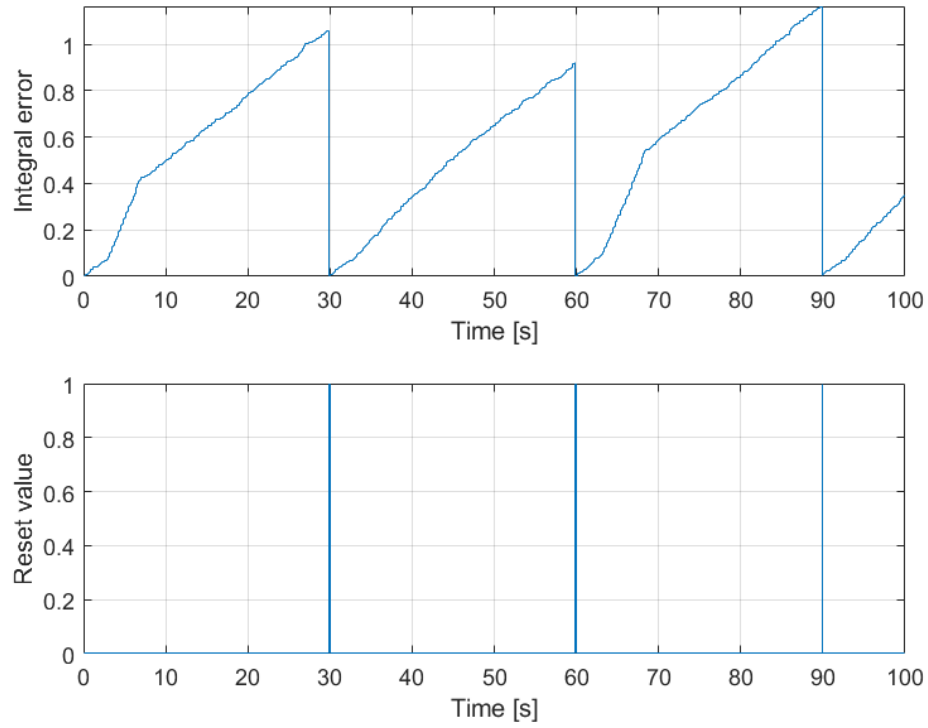


Figure 5.30 Integral error in nominal condition with Reset time of 30s.

5.4 Multi model: algorithm structure

With all the considerations made so far, it is possible to implement the final estimation algorithm, that will be then validate through an appropriate **unit testing** in ???. This part will represent the core of the MOREPRO project and of the edge device that will be implemented on the CNC machine to have an on-line SoH monitoring.

As far as the algorithm is concerned, it is mainly composed of three distinct parts:

- The **CNC model** that represents the dynamics equations governing the system;
- The **Extended Kalman Filter bank** where each filter is based on a different friction coefficient hypothesis and which get as input the same inputs applied to the model mentioned before and the outputs at the terminals produced by the latter and aims to estimate, based on the assigned β hypothesis, the acceleration at the terminals obtained by linearizing the CNC model around the specific working point.
- A **logic of decision and management** of the integral of the residual errors that include a reset, a *best model* choice and possibly the **reliability** of such choice.

In the following figure the general Simulink structure is depicted, summarizing all the components described above.

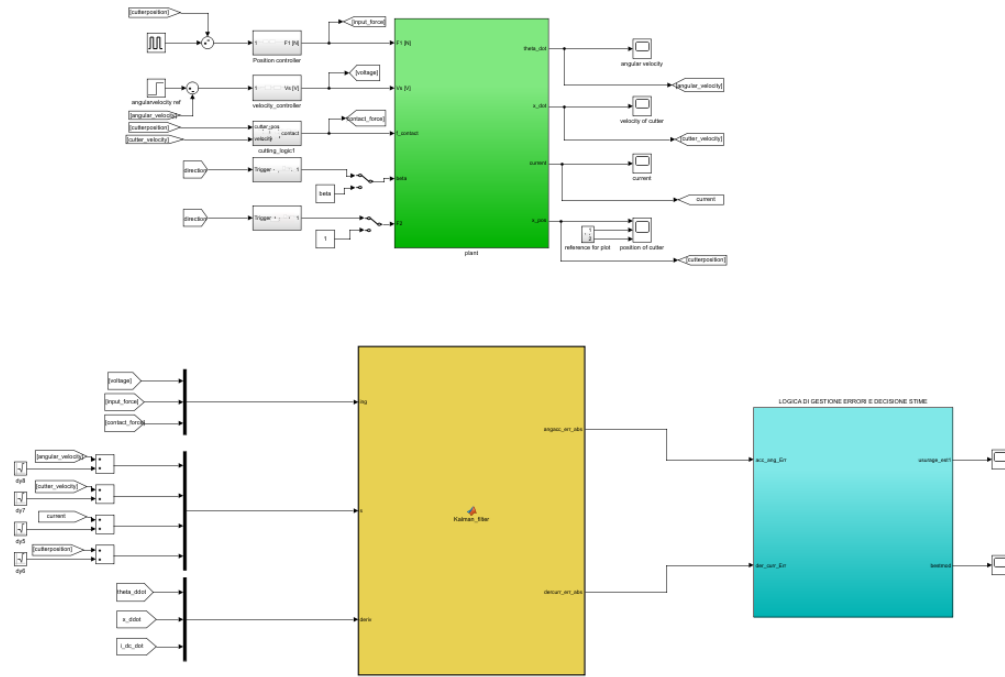


Figure 5.31 Simulink implementation of the algorithm.

In the green box are contained the dynamic behavior and the state equations of the plant while in the yellow one is contained the entire EKF bank. The residual error estimation is then forwarded to the light blue block which represents the logic of error management, whose internal structure is represented in 5.32.

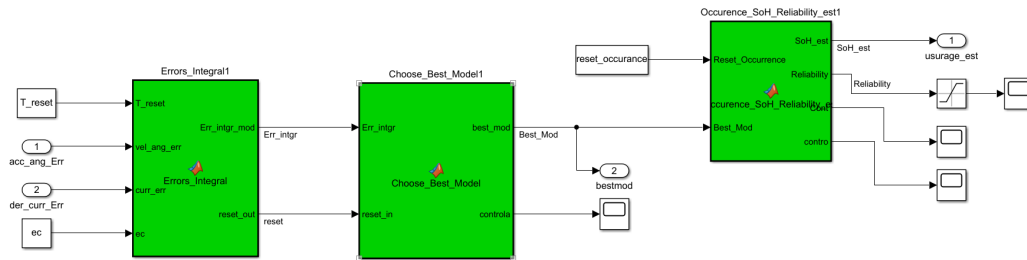


Figure 5.32 Simulink implementation of error logic.

In the next pages it is possible to find an in-depth explanation of the various mentioned structures.

5.4.1 Switching estimator

It is considered a problem of state estimation with a parameter variation in a finite range. The idea is to put N EKF in parallel, where each of them works with a different "wear condition" hypothesis. In particular, the friction coefficient β is chosen as switching parameter, obtaining N independent EKF, each with a fixed value of β .

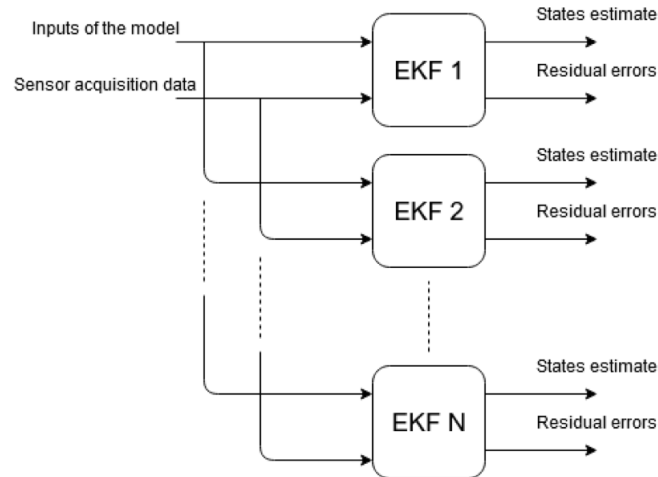


Figure 5.33 Switching estimator

The working principle of the switching estimator is:

- each EKF will purpose its estimate;
- each EKF works with a different friction coefficient, switched over a finite set of values;

It has been decided to put **N=6 EKF** with a β range values from 0.1 to 1 linearly spaced as detailed in table 5.25. The aim is to identify the filter with the minimum residual errors, which means that filter works with the friction coefficient more similar to the real one and that represents better the condition of the machine.

Filter	β value
#1	0.1
#2	0.28
#3	0.46
#4	0.64
#5	0.82
#6	1

Table 5.25 Friction coefficient associated to each filter

5.4.2 Best model choice

A key part of the algorithm is dealing with the residual errors and extrapolate useful information from it as the errors contains an intrinsic assessment of the EKF's quality. The underlying idea is to choose the model with the smallest residual error as the best model because it will have the closest friction coefficient to the real one with all other parameters unchanged.

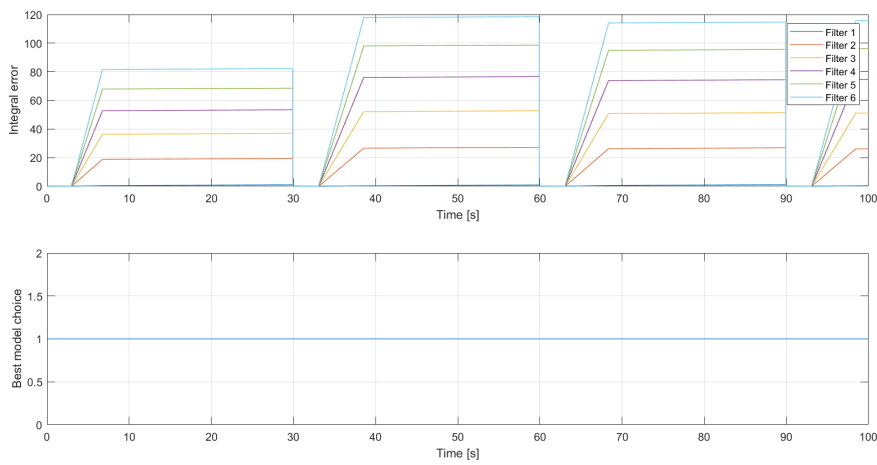
In order to implement this management, a Matlab function has been developed which is dependent on both the input errors, the reset of the integral and also a "dwell time" which will be explained shortly. In principle, the choice works through these steps:

- Initially a **dwell time** is set, that is a period in which the function can not check the errors data because it is supposed as a period for a dynamic evolution of the system so that there are relevant data in the estimates.
- When there is a **reset of the integral**, the function cannot select the model because the data is not reliable as no past information are collected.
- After resetting the integral and the dwell time at which the transient has passed, the function analyses the errors data and assigns the model with the lowest error as the **best model**.

The figure 5.34 shows that with the condition listed in 5.26 the best model is always the first because, as it should be, it is the one that has the same value as the nominal one.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Friction coefficient	0.1	0.1
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.26 Test with nominal friction coefficient

**Figure 5.34** Best model choice with nominal condition

While testing a friction coefficient variation in the range $0.1 \div 0.5$ the best model changes according to the less residual error like shown in figure 5.35.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Friction coefficient	0.1	$0.1 \div 0.5$
Duty cycle [%]	50	50
Number of cycles	4	4

Table 5.27 Test with friction coefficient variation

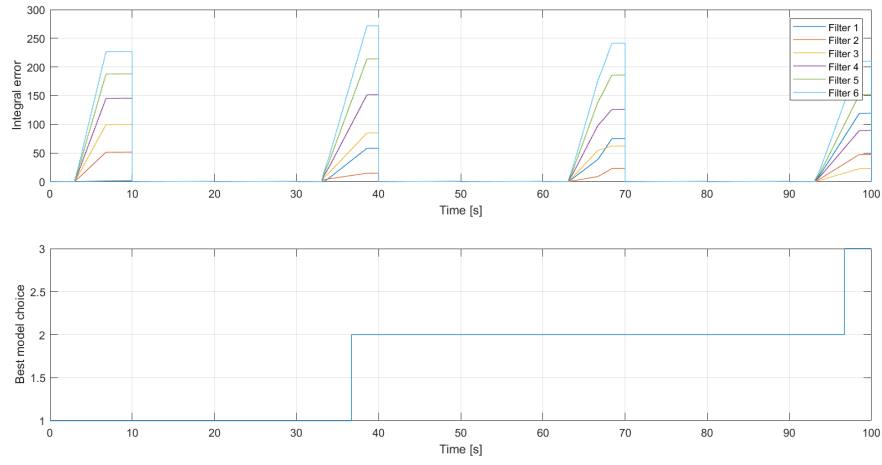


Figure 5.35 Best model choice with β variation

6 Model upgrade

Modelling is a very important and necessary step in Robotics; the obtained mathematical model is used in order to perform some simulations that can allow to better understand the robot movements and behaviours. The most important benefit in modelling and simulate a system is that using environment of simulation, like MathWorks (MATLAB, Simulink) is less expensive than perform real tests with the system and helps also to collect data. The initial models, normally, are as simple as possible in order to explain the main characteristic of the system and to develop an embrional prediction algorithm. But in order to have good information from the simulation performed, it is better to create a model closer enough to reality and for this reason increase the level of detail and accuracy of the model. One of the crucial steps of the project is to built the model of the used machine and of the interaction between the end-effector and the material to be shaped. The starting model of the CNC machine, is composed by an electrical and mechanical equations that can synthesize the work of the machine. In order to add more in-depth representation, information about the interaction between the end-effector and the workpiece are added. In detail, the interaction model is developed around the friction coefficient (β) between the tool and the material; this because is considered as the most significant parameter, playing a main role, in the tool wear process. Schematically, the upgrading process of the model will follow this path:

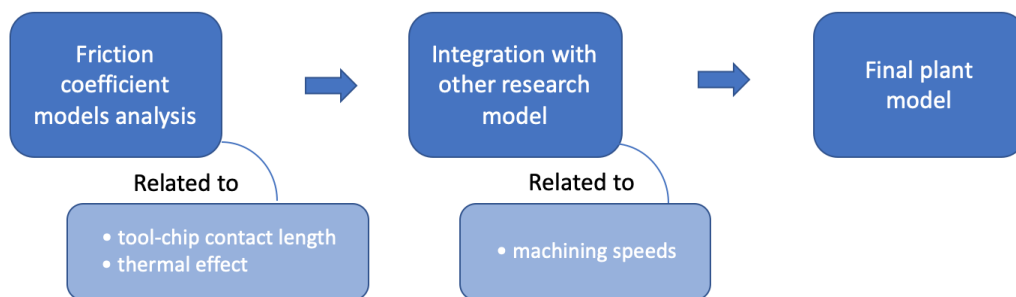


Figure 6.1 Model upgrade path

6.1 Friction coefficient models

Describe the metal cutting process is very difficult; the complexity is related to the plastic deformation that arise during the process in the metal to be shaped. The thermomechanical deformations that occur in the metal bring to microstructure modifications while one of the main causes involved in the tool wear, is the high temperature at the tool-workpiece interface.

The increasing temperature in the metal cutting, is due to the sliding contact between the metallic surfaces. This kind of motion generates high friction coefficients that dissipate huge amount of energy, that is even increased when tool wear occurs. The friction coefficients plays an important role in metal cutting process and is influenced by many factors; among the most important there are temperature, the used material of the tool and of the workpiece, the relative speed in interaction, the applied forces, the cooling media supplied in the contact interface etc etc.

One way to decrease this friction is for sure the use of cutting fluids, like lubricant oils, whose effect is more evident at low speeds. At high speeds the cooling is not able to lubricate the interaction surface with the consequence that the effect is reduced and the friction coefficients is almost the same of the dry condition. In general the whole behaviour is also related to the amount of lubricant provide to the machine.

To accurate describe and model the metal cutting procedure, detailed mathematical models are needed, able to represent how the material deform but also how the tool and the workpiece interact. The state of art related with the friction coefficient in metal machining provides a lot of model that consider different factors and provides detailed approaches considering different parameters that could influence the interactions. The goal of this study is to better understand which, between the factors, has a greater influence on the friction coefficient. Among the many approaches studied, several were discarded because of the excessive difficulty in finding the parameters, while less detailed models have been taken into consideration, which in any case give satisfactory results but are easier to deal with.

6.1.1 Friction coefficient correlated with the tool-chip contact length

The first model taken into account considers the material transfer layer thickness but also the non-linear flow velocity of the chip. The proposed model subdivides the tool-chip interaction surface into three zone that differ each other for mechanical characteristics related to the impact. In particular, is possible to define two different friction coefficients along the contact surface.

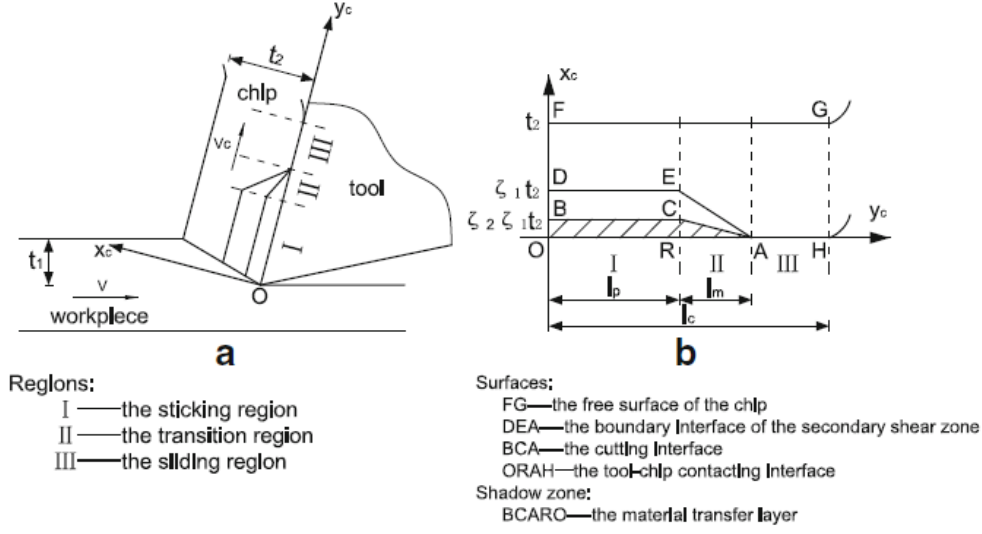


Figure 6.2 Proposed friction model.

As can be seen from figure 6.2 the three regions along the tool-chip interaction length (along the y_c axis) are respectively sticking region (l_p), transition region (l_m) and sliding region (l_c). Normally, friction coefficient models consider only the sticking and sliding region but in order to consider the chip flow continuity also a transition region is introduced. Two different kind of deformation take place: in the sticking region a plastic deformation is present ($0 \leq y_c \leq l_p$), while in both transition and sliding region there is an elastic deformation ($l_p \leq y_c \leq (l_p + l_m)$). The BCARO area represent the material transfer layer where two important zone are highlighted:

- tool-chip contacting interface (ORAH) that is the border between the tool rake face and the chip;
- cutting interface (BCA) is the one with the maximum shear strain, because this is the plane where the cutting action take place.

As mentioned before, with the proposed model it is possible to consider two different formulation of the friction coefficient and in particular in the transition and sliding areas the coefficient is defined as β_{sl} that is a local friction and is calculated as:

$$\beta_{sl} = \frac{\tau(y_c = l_p)}{P_0(1 - \frac{l_p}{l_c})\zeta} \quad (6.1)$$

where τ is the shear stress in the sticking region, P_0 the pressure at the tool tip and ζ is an empirical parameter. In order to compute the friction coefficient along

the whole tool-chip interface the following formulation can be used:

$$\begin{aligned}\beta &= \frac{\int_0^{l_p} \tau w dy_c + \int_{l_p}^{l_c} \beta_{sl} P_0 (1 - \frac{y_c}{l_c})^\zeta w dy_c}{\int_0^{l_c} P_0 (1 - \frac{y_c}{l_c})^\zeta w dy_c} \\ &= \frac{(\zeta + 1) \int_0^{l_p} \tau dy_c + \beta_{sl} P_0 l_c (1 - \frac{l_p}{l_c})^{\zeta+1}}{P_0 l_c}\end{aligned}\tag{6.2}$$

where

- ζ : empirical parameter;
- τ : shear stress;
- μ_{sl} : local coefficient in the sliding region.

Experiments carried out using a wide range of parameters show that the model works appropriately. Realizing many tests with different cutting speed, emerged that the friction coefficient value decreases with the increasing of cutting speed and that for high values of that one the sticking region disappear.

6.1.2 Thermal effect in friction coefficient

One of the main problems in metal cutting is the heat spreads from the cutting process that increase the temperature of the tool, workpiece and machine too. One of the main dangerous effect of high temperature flux in metal cutting is that they favour the tool wear and due to the high heat flux also the material of the work-piece may be subject to undesired modifications that bring to an incorrect processing.

Is very difficult to make direct measurement because of the cutting process and the main way to find some values is to use simulation models. In particular, this approach treats the whole process as a continuous flow of work material against the cutter. In this case, the interaction between the tool and material is simulate with a temperature dependent friction model. The friction coefficient remains a constant equal to $\beta_0 = 1/\sqrt{3}$ till the temperature T_0 . For temperature higher than T_0 , β decreases due the thermal softening effect depending on the melting point temperature T_m and the power m_r as described in equation 6.3.

$$\beta = \beta_0 \left[1 - \left(\frac{T - T_0}{T_m - T_0} \right)^{m_r} \right]\tag{6.3}$$

This temperature-dependent friction model is under the assumption that the effects of speeds and pressures act only on contact temperatures. The used power m_r is an empirical parameter.

6.1.3 Thermal analysis

Having chosen a temperature-dependent friction model, it now appears necessary to identify which temperatures affect the tool-workpiece interface and how the cutting parameters can impact on the temperature. The cutting temperature is not constant throughout the tool, the chip or the workpiece. It can be observed that the maximum temperature is developed not on the very cutting edge, but at the tool rake, some distance away from the cutting edge. The temperature field in the cutting zones is shown in figure 6.3.

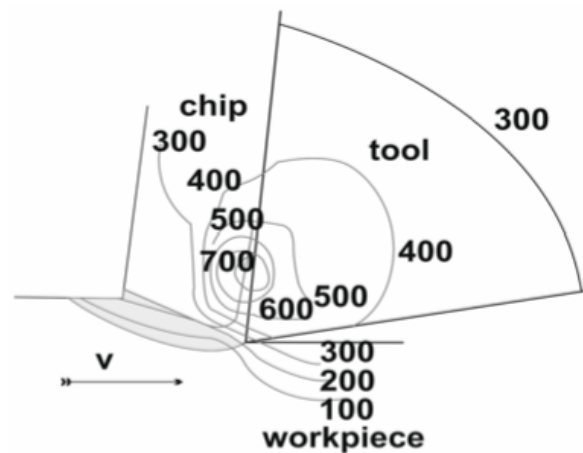


Figure 6.3 Tool-workpiece interaction temperature curves.

In order to obtain consistent temperature data, it is necessary to integrate several measurement techniques as the machine processes are complex. Therefore, two techniques from literature were analysed: K-type thermocouple and Infrared Radiation pyrometer to measure the tool-chip interface temperature [24]. The thermocouple only measures the mean temperature over the entire contact area of the tool and the workpiece. In the IR technique the surface temperature of the body is measured based on its emitted thermal energy [25] and in order to establish the temperature on the outside surfaces of these regions, the radiation from the tool, the workpiece and the chip is measured.

A total of 18 trials were analyzed in order to choose a consistent range of tool-chip interface temperature. The results are given in table 6.1.

Depth of cut [mm]	Cutting speed [m/min]	Feed rate [mm/rev]	Tool temperature [°C]	Tool-chip interface temperature [°C]
0.4	76	0.05	57	410
0.4	76	0.08	66	405
0.4	76	0.12	72	410
0.4	114	0.05	65	460
0.4	114	0.08	61	465
0.4	114	0.12	67	445
0.4	170	0.05	65	520
0.4	170	0.08	67	500
0.4	170	0.12	71	475
0.6	76	0.05	72	400
0.6	76	0.08	80	390
0.6	76	0.12	76	395
0.6	114	0.05	80	430
0.6	114	0.08	75	435
0.6	114	0.12	83	420
0.6	170	0.05	81	485
0.6	170	0.08	67	525
0.6	170	0.12	69	480

Table 6.1 Temperature values.

The influence of the cutting parameters such as cutting speed, feed rate and depth of cut on the cutting temperature was observed. The higher the cutting speed, the higher the tool-chip interface temperature increase, resulting in a significant temperature difference. The feed rate has very little effect on the temperature rise. In the figure 6.4 the influence of cutting speed on tool-chip interface temperature can be observed.

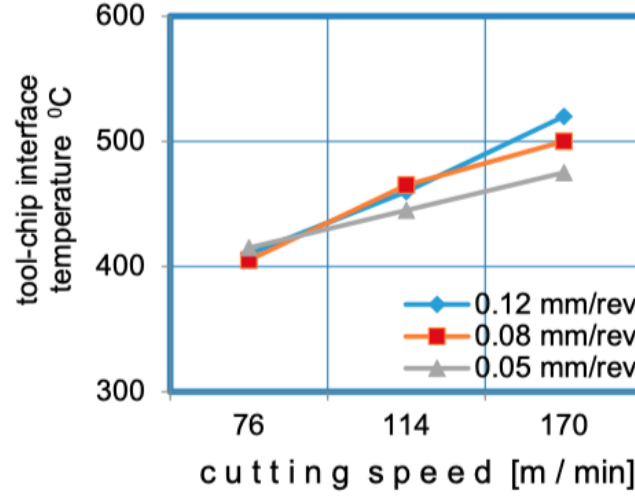


Figure 6.4 Influence of cutting speed on tool-chip interface temperature.

Thus, from the results analysed, it was chosen to simulate the model with a temperature range from 480°C to 550°C.

6.1.4 Other correlated model

Another friction coefficient modelling useful for the project development is the one related to machining speeds, analysed by Verde Antonia, a project colleague. Below are the main features for further integration in order to produce a final complete modelling.

Between the main parameters that affect the friction coefficient there are the machining speeds: cutting speed and feed rate, that have a combined effect in the cutting process. Taking into account this parameters the goal of this model is to build a friction coefficient formulation:

$$\beta = 3.32V_c^{-0.45} - 0.24f \quad (6.4)$$

where:

- $V_c = \frac{\pi DCn}{1000}$: cutting speed that is a relative velocity between the cutter of the CNC machine and the material to be processed. In particular, DC is the diameter of the cutting tool and n is the spindle speed.
- $f = n * t * cl$: feed rate that is the relative velocity at which the cutting tool is advanced along the material piece to be shaped. In particular, t is the number of teeth of the cutter and cl is the chip load.

6.1.5 Model comparison

In order to create a model as close to reality as possible, a more precise formulation of the friction coefficient was devised, taking into account two of the three model mentioned above which have a greater influence on it.

Follows a demonstration of the friction values obtained by filling the equations 6.2, 6.3, 6.4, with reliable experimental data and which will subsequently be verified with real data coming from the CNC machine used in the project.

The first model considered is the one correlated with the tool-chip contact length. Figure 6.5 shows that the range of β is between 0.255 and 0.278.

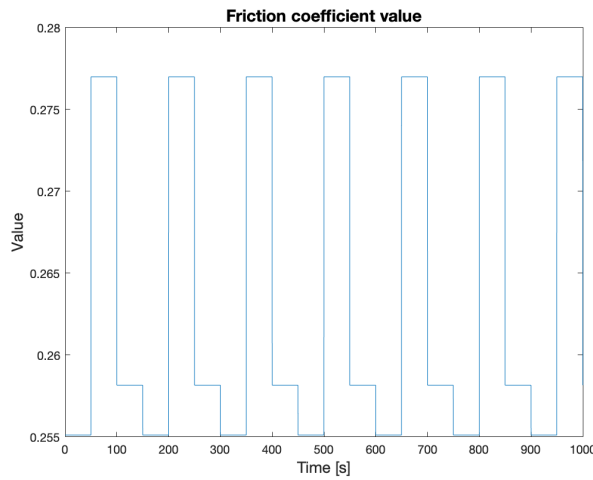


Figure 6.5 β obtained from the tool-chip contact length model

As can be seen, it is not possible to observe great variations in the friction coefficients despite the complex parameters involved in the equation.

Switching to the second considered model, the values of the coefficient, in figure 6.6, are in the range $0.38 < \beta < 0.43$.

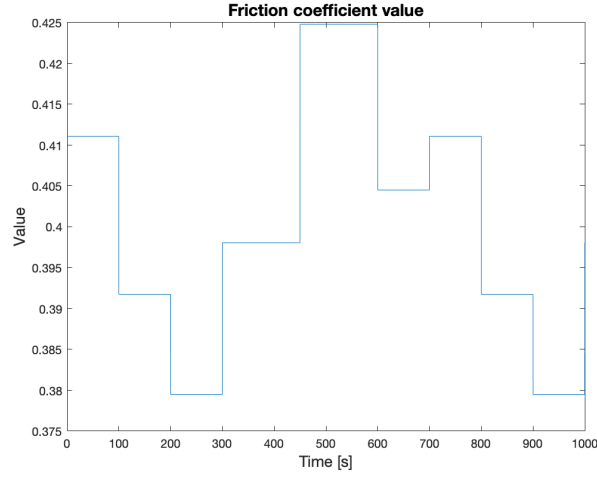


Figure 6.6 β obtained from the temperature dependent model

Compared with the first model, is possible to affirm that the parameters involved in the temperature dependent model, influence more the trend of β .

The last model is the one related with the cutting speed and the feed rate. In this case the interval is between $0.15 < \beta < 0.45$.

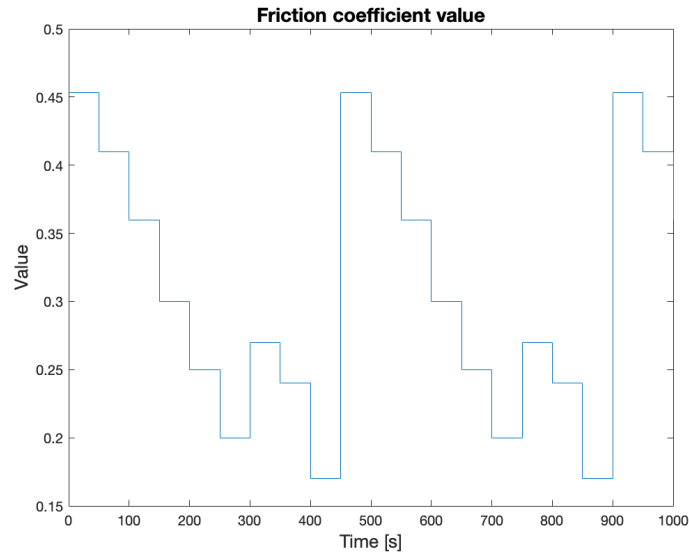


Figure 6.7 β obtained from the machining speeds model

This formulation is a function of the angular velocity input, all the other parameters are constant (DC,t) while CL (the chip load) is the only variable parameter

that impacts on the wear conditions of the cutting tool and is exactly the parameter that we want to estimate in the EKF's bank. The chip load is the theoretical length of material that is fed into each cutting edge as it moves through the work-piece. The tool wear is widely affected by the chip load as also the force to cutting the material.

6.2 Final plant model

The last step is to combine the models which have a greater incidence on the values of the friction coefficient; for this reason the model related with temperature and the one related with the machining speeds are joined. The global formulation is obtained replacing the β formulation (6.4) instead of β_0 into (6.3) with the following result:

$$\beta = \left(3.32 \left(\frac{DCn\pi}{1000} \right)^{-0.45} - 0.24(tnCL) \right) * \left[1 - \left(\frac{T - T_0}{T_m - T_0} \right)^{m_r} \right] \quad (6.5)$$

In the final equation the parameters involved are all geometric or that can be directly measured by sensors except of the chip load, which therefore becomes the parameter to be estimated because it is directly related to the tool wear. Chip Load affects five major areas of the machining process:

- Controls the required force to cut the work material
- Assists in controlling heat
- Controls tool wear
- Directly affects the metal removal rate
- Directly affects surface finish

The chip load setting is still a research topic since too much chip load increases wear, leads to premature tool failure, rough finishes and draws more torque and amperage through the machine and increased stress on the axis drives while too little chip load causes vibration and chattering that will chip the tools cutting edges and it can cause the tool to rub and wear rather than cut.

Substituting beta equation into the **system's state equations** gives:

$$\begin{cases} \ddot{\theta} = \frac{k_t i_a}{I_n} - \frac{Att_{mot} \dot{\theta}}{I_n} - \frac{(3.32(\frac{DC \dot{\theta} \pi}{1000})^{-0.45} - 0.24(t \dot{\theta} CL)) * [1 - (\frac{T - T_0}{T_m - T_0})^{m_r}] F_c \dot{\theta}}{I_n} \\ \ddot{x} = \frac{F_1}{m} - \frac{F_c(F_2 \alpha + c)}{m} \\ \dot{i}_a = \frac{V_s}{L} - \frac{R i_a}{L} - \frac{k_v \dot{\theta}}{L} \end{cases}$$

The nominal parameters of the machine are:

	Nominal value
Mass [kg]	3
Radius [m]	0.3
Resistance [kΩ]	0.6
Inductance [mH]	0.1
Torque constant	1.5
Voltage constant	0.2
Motor Inertia [kgm ²]	0.001
T_0 [°C]	400
T_m [°C]	1400
Diameter of cutter [mm]	100
Number of teeth	5

Table 6.2 Nominal CNC parameters updated

7 Multi model update

7.1 EKF integration

As far as the multimodel is concerned, the structure of the algorithm remains the same as in section 5.4. A differentiation is made in the plant equations in which the friction coefficient is no longer a dynamic independent variable but being expanded as described in equation 6.5, now depends on several variables/parameters:

- $\dot{\theta}$: angular velocity
- DC: diameter of the cutter
- t: number of cutting teeth
- CL: chip load
- T: measured contact temperature
- T_0 : thermal softening threshold
- T_m : melting point temperature
- m_r : empirical parameter

Therefore, based on research carried out on the development of the interaction model the only variable that affects tool wear is **chip load**, as all the others are either constants depending on the geometry and material of the tool or variables depending on the type of machining. Thus there is a need to include the chip load as a wear condition hypothesis within the EKF's bank and it will be used as switching parameter.

The EKF algorithm is the same as in paragraph 5.1 except for the inputs and the Jacobian of the state-space model F_k that become:

- Inputs:
 - V_a : armature voltage
 - F_1 : horizontal force that moves the cutter
 - F_c : function that define the contact with the object.
 - T : measured contact temperature
- Jacobian:

$$\hat{F} = \begin{bmatrix} -\left(-0.45 * 3.32 \left(\left(\frac{\pi DC \dot{\theta}}{1000}\right)^{-1.45}\right) \frac{\pi DC}{1000} - 0.24 CL * t * 10^{-3}\right) \left(1 - \left(\frac{T - T_0}{T_m - T_0}\right)^{m_r}\right) F_c - 0.3 & 0 & -\frac{k_t}{I_n} \\ I_n & 0 & 0 \\ 0 & -\frac{k_v}{L} & -\frac{Res}{L} \end{bmatrix}$$

Adapting the plant equations to the EKF's structure it is also decided to increase the number of filters to **N=8 EKF** in order to have a more accurate range of models and thus greater estimation accuracy. The filters works with a *CL* range values varies from 0.1 to 1.1 linearly spaced as detailed in table 7.1.

Filter	CL value
#1	0.1
#2	0.24
#3	0.38
#4	0.52
#5	0.67
#6	0.81
#7	0.95
#8	1.1

Table 7.1 Chip Load associated to each filter

7.2 Functional Tests

At this stage in order to analyze the correct behaviour of the multimodel system a series of tests were carried out. The input to the filters are collected from the plant simulation. In detail, all the range of EKF's bank has been tested according to the variation of chip load hypothesis. It is expected that when the chip load hypothesis of a certain filter is similar to that of the plant, the residual error will be the lowest as compared to the filters residual error with a hypothesis that is distant from the actual one. In the figure 7.1 is shown the Angular acceleration residual error associated to the nominal condition listed in table 7.2.

	Nominal value	Testing value
Angular velocity [$\frac{rad}{s^2}$]	210	210
Chip load	0.1	0.1
Duty cycle [%]	50	50
Number of cycles	4	4

Table 7.2 Test with nominal chip load

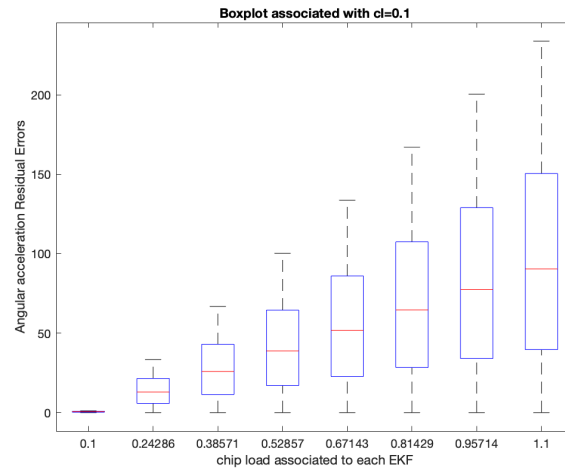


Figure 7.1 Boxplot based on chip load with nominal condition

In the following figures the chip load range from 0.2 to 1.1 has been tested. It is possible to see that when the chip load hypothesis is similar to that of the filter, the error is distinctly the smallest; whereas when the hypothesis is in the middle between two filters, the error of the two filters is so similar that it could create misunderstandings in choosing the model that best approximates the wear condition of the system.

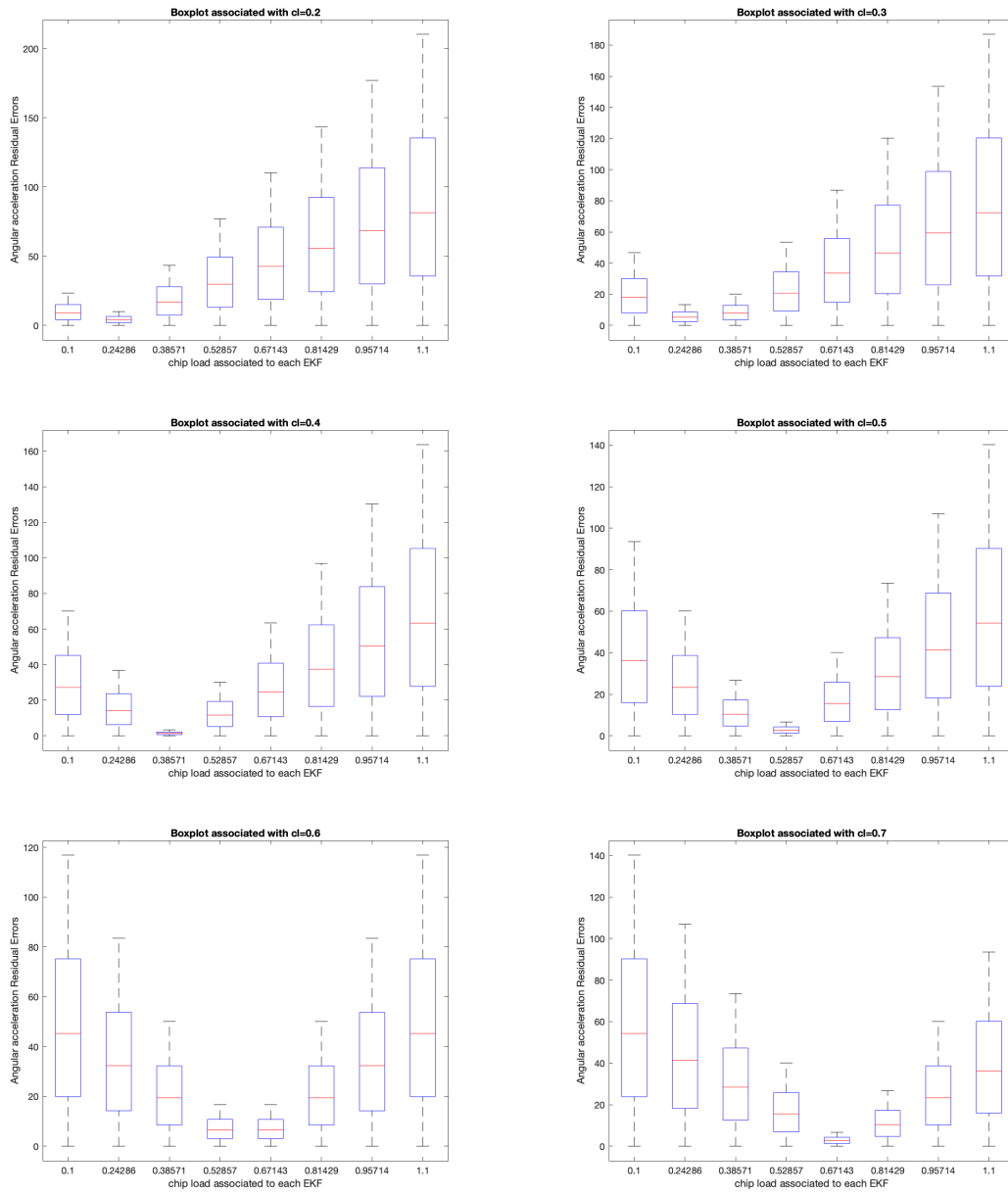


Figure 7.2 chip load analysis from 0.2 to 0.7

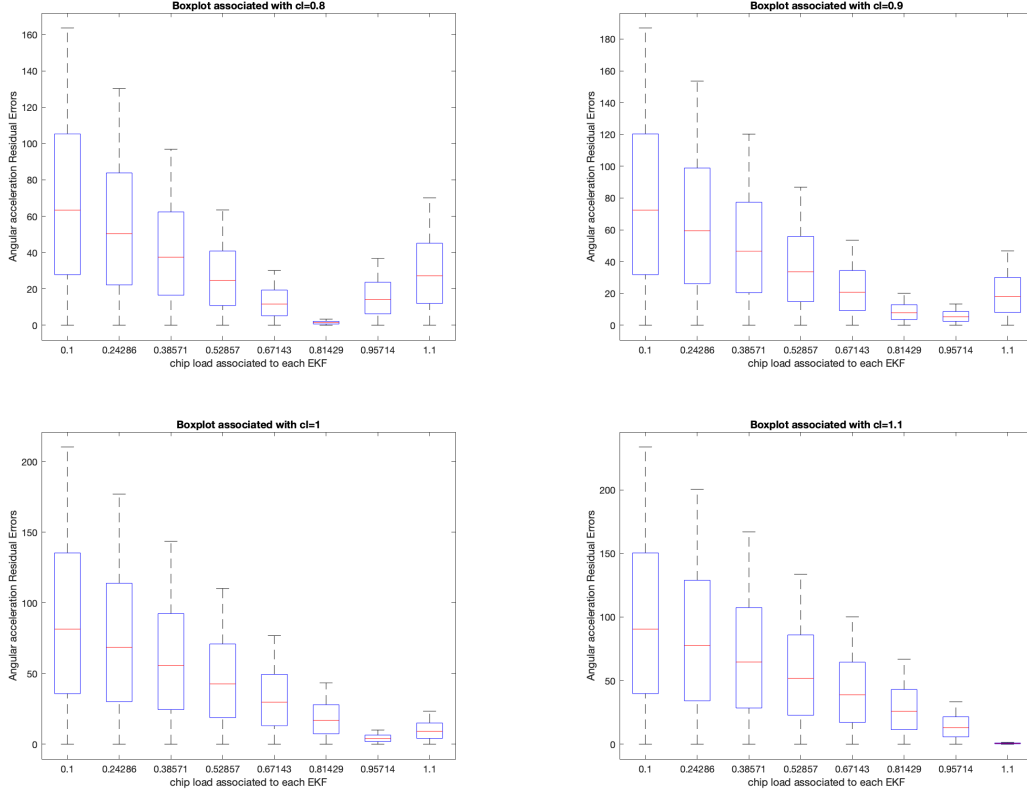
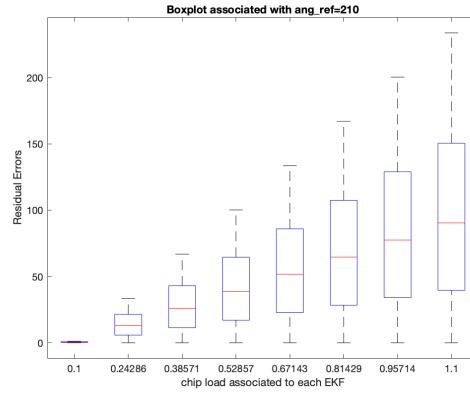


Figure 7.3 chip load analysis from 0.8 to 1.1

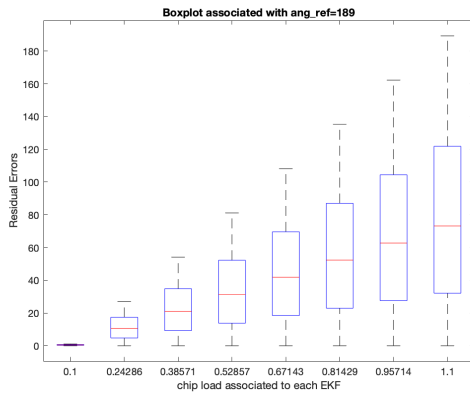
7.3 Sensitivity Tests

In an optimal situation it is always possible to generate a mathematical model that perfectly represents the system under consideration, but when this is not the case it is advisable to examine the effects of the lack of knowledge about performance in order to verify the robustness of the algorithm. Thus, a sensitivity analysis is carried out on the residual error produced by the bank of EKF with the aim of identifying, among the most significant parameters of the CNC machine model, how their possible variations affect the integral error produced and how a different input conditions affect the state observer behaviour. In particular, percentage variations with respect to the nominal value of the parameter under examination is analysed, changing one parameter at a time. The various tests are listed below, where the figure at the top centre represents the nominal condition while at the bottom left is a percentage reduction in parameter's value and at the right a percentage increase.

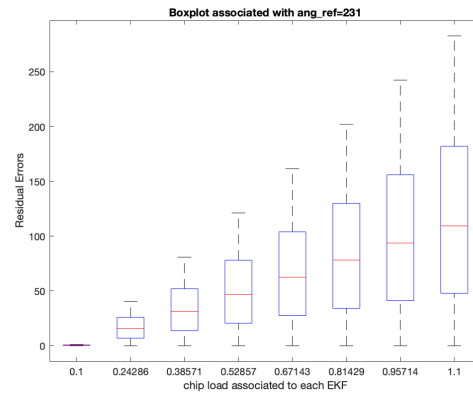
- Angular velocity:



(a) Nominal value



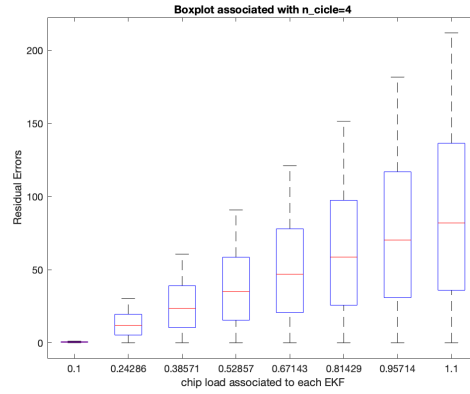
(b) -10% wrt nominal value



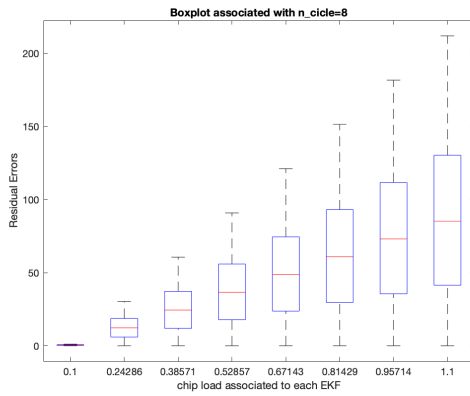
(c) +10% wrt nominal value

Figure 7.4 Test on angular velocity reference.

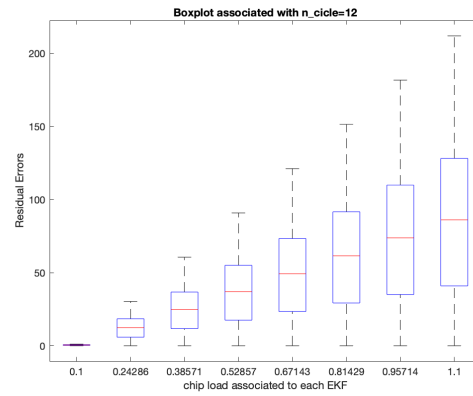
- Number of cycles:



(a) Nominal value



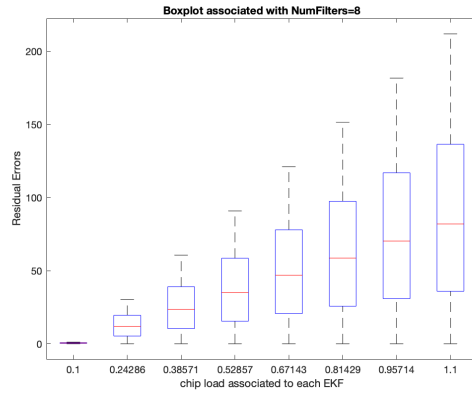
(b) +100% wrt nominal value



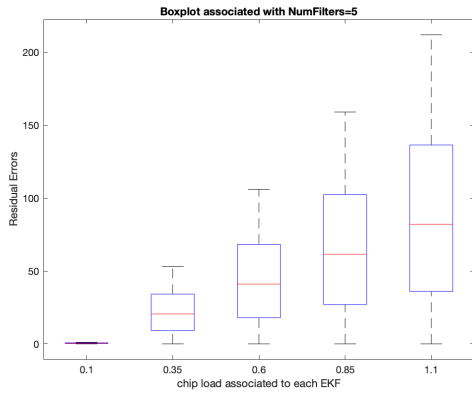
(c) +200% wrt nominal value

Figure 7.5 Test on number of cycles.

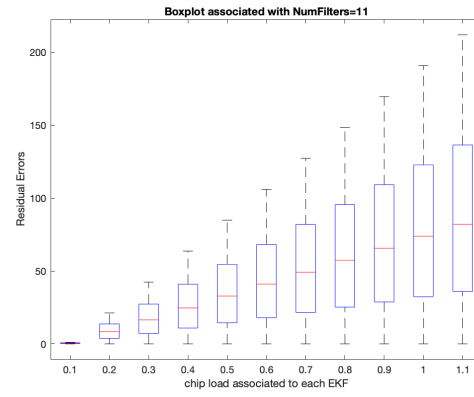
- Number of filters:



(a) 8 Filters



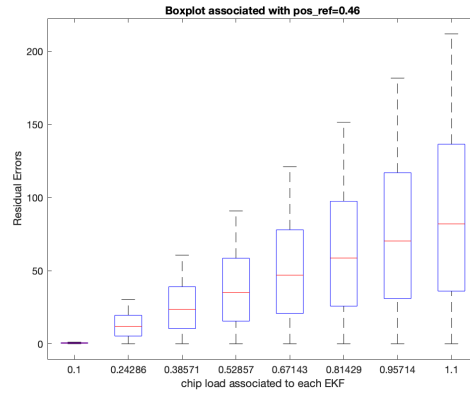
(b) 5 Filters



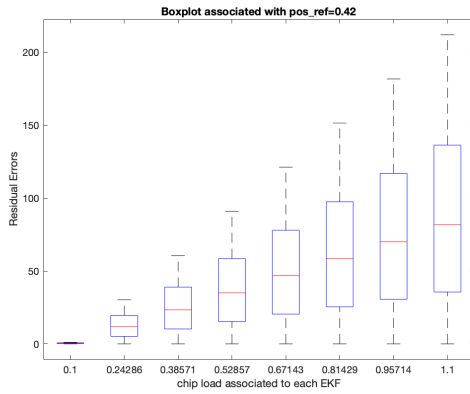
(c) 11 Filters

Figure 7.6 Test on number of Filters.

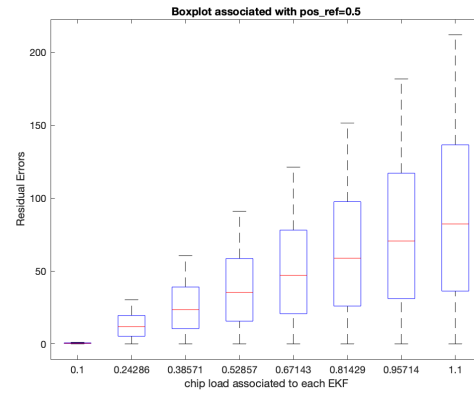
- Position reference:



(a) Nominal value



(b) -10% wrt nominal value



(c) +10% wrt nominal value

Figure 7.7 Test on position reference.

- Shift of the chip load range:

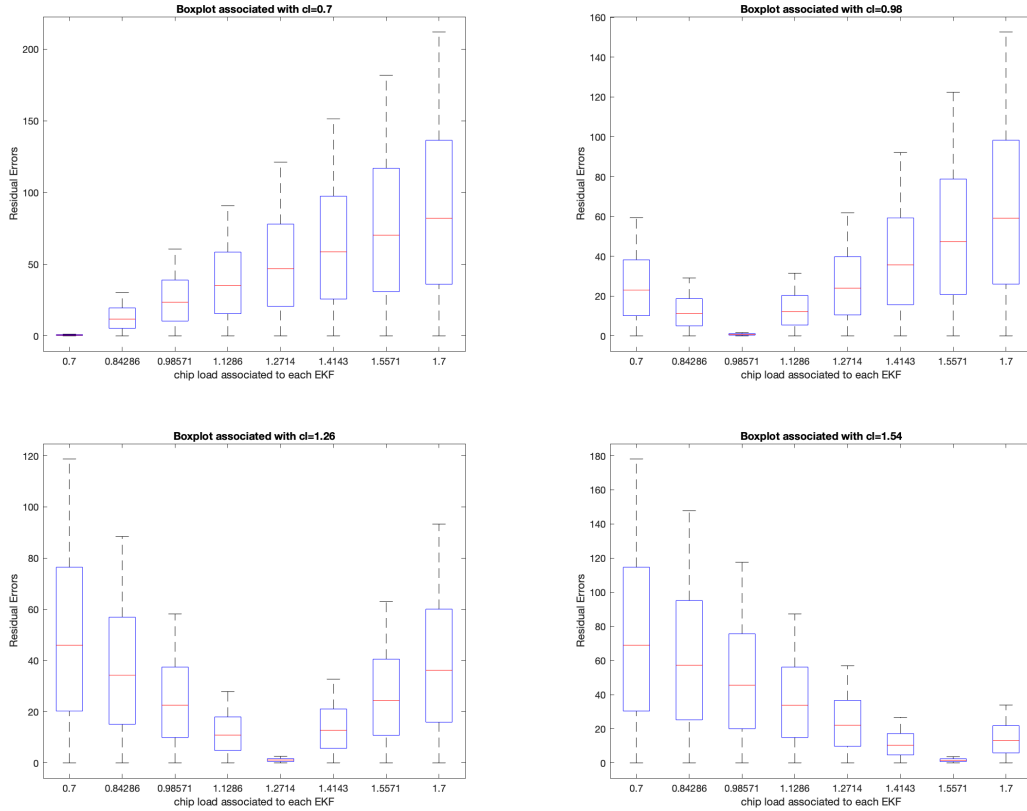


Figure 7.8 chip load range analysis from 0.7 to 1.7

Having a look at the various tests performed so far, it is possible to see that there are no sensible variation when considering different working conditions with respects to the nominal ones. Thus, the error associated with the correct model to estimate is always smaller then the other ones. Moreover, in some particular cases there is a better distinction between the boxplots, indicating a more accuracy on the estimation algorithm.

Conclusions and future works

In this work the application of multi model approach in a machine predictive maintenance domain is the main objective achieved.

- In a first stage, the development of a physical model of the CNC machine was addressed, under the hypothesis that the model should have allowed the development of a prediction algorithm as soon as possible and always keep in mind the edge-computing device goal, thus also with a relatively low computational cost. After an in-depth analysis of the literature, the friction coefficient β between the tool and the workpiece is chosen as wear hypothesis on which to base the prediction algorithm, that involves a bank of N Extended Kalman Filter. Through the residual error analysis it has been shown that a multi-model approach can be used to estimate the SoH of the machine.
- In a second stage, the need emerged to investigate the interaction between the workpiece and the tool and to explore the dependency of the friction coefficient on other parameters. Through the study and integration of more complex models, the conclusion was reached to rely on the machine's chip load as the source of the wear estimate. Finally a series of functional tests were carried out in order to verify the correct behaviour of the multi model system.

Despite the system has not been tested in a real environment, not going beyond a "Model in the loop" testing, this thesis work lays foundations for future edge-computing SoH prediction techniques and for the Morepro project progress.

- One of the possible line of action is the research of a more accurate modelling of the machine but at the same time with low computational effort in order to meet the edge-computing requirements.
- A possible future work could be the research of an alternative method to estimate the SoH through a different kind of analysis, that does not take into account the residual error but only the available measurements.
- One more change could be how to process the residual error; in this thesis work the approach adopted is an experimental one, however a possible case study could be a more formal method that links the residual error to the internal parameters of the model or a mathematical methodology in order to improve with an optimal and automatic chain the algorithm decision logic.

References

- [1] Li, Yulong; Zhang, Xiaogang; Ran, Yan; Zhang, Wei; Zhang, Genbao. *Reliability and Modal Analysis of Key Meta-Action Unit for CNC Machine Tool.*
- [2] Hashemian, H. M ; Bean, W. C *State-of-the-Art Predictive Maintenance Techniques.*
- [3] Knuth: Computers and Typesetting,
<http://www-cs-faculty.stanford.edu/~uno/abcde.html>
- [4] Preet Joy, Adel Mhamdi, Alexander Mitsos. *Optimization-based observability analysis.* Aachener Verfahrenstechnik - Process Systems Engineering (SVT), RWTH Aachen University, Forckenbeckstrasse 51, Aachen 52074, Germany
- [5] Feng, Guohu ; Huang, Xinsheng *Observability analysis of navigation system using point-based visual and inertial sensors* ISSN: 0030-4026 EISSN: 1618-1336 DOI: 10.1016/j.ijleo.2013.08.004
- [6] Gouriveau Rafael, Medjaher Kamal, Zerhouni Nouredine *From prognostic and health systems management to predictive maintenance 1: Monitoring and Prognostics* Newark: John Wiley & Sons, Incorporated, 2016
- [7] Aleksandra Marjanović, Goran Kvašćev, Predrag Tadić, Željko Đurović. *Applications of Predictive Maintenance Techniques in Industrial Systems* SERBIAN JOURNAL OF ELECTRICAL ENGINEERING Vol. 8, No. 3, November 2011, 263-279
- [8] Sayed-Mouchaweh Moamar, Lughofer Edwin. *Predictive maintenance in dynamic systems: advanced methods, decision support tools and real-world applications* Springer, 2019
- [9] Gregory L. Plett. *Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1. Background* Journal of Power Sources 134 (2004) 252–261
- [10] Daga A. P., Garibaldi L. *Machine vibration monitoring for diagnostics through hypothesis testing* MDPI AG, 2019
- [11] Bibin Pattel, Hoseinali Borhan, Sohel Anwar. *An evaluation of the moving horizon estimation algorithm for online estimation of battery state of charge and state of health* Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition, IMECE2014, November 14-20, 2014, Montreal, Quebec, Canada
- [12] Pier Giuseppe Giribone, Ottavio Caligaris, Simone Fioribello, Simone Ligato. *Implementazione della Fuzzy Logic per la gestione ottimale del portafoglio: la*

modellizzazione dell'avversione al rischio di un investitore attraverso tecniche di softcomputing September 2016, DOI: 10.47473/2020rmm0061

- [13] *Reshaping the world with fuzzy logic* WorldQuant Perspectives, April 2018
- [14] S. Khaleghi, Y. Firouz, J. Van Mierlo, P. Van den Bossche. *Developing a real-time data-driven battery health diagnosis method, using time and frequency domain condition indicators*; Department of Mobility, Logistics and Automotive Technology Research Centre, Vrije Universiteit Brussel, Pleinlaan 2, Brussels 1050, Belgium Flanders Make, 3001 Heverlee, Belgium
- [15] Davide Faverato. *Virtual Sensing for the Estimation of the State of Health of batteries* October 2020, Politecnico di Torino
- [16] J. Kallrath, ed. *Modeling Languages in Mathematical Optimization*, Applied Optimization, Vol. 88, Kluwer, Boston 2004.
- [17] Zoran Pandilov, Filip Gorski, Damian Grajewski, Damir Ciglar, Tihomir Mulc, Miho Klaic, Andrzej Milecki, Amadeusz Nowak *Virtual modelling and simulation of a CNC machine feed drive system*, TRANSACTIONS OF FA-MENA XXXIX-4 (2015), ISSN 1333-1124, eISSN 1849-1391
- [18] Dora Sabau, Mirela Dobra *System identification and control analysis for CNC machines*, 978-1-5386-2205-6/18/2018 IEEE
- [19] *MATLAB guidelines*, Application-specific guidelines for model architecture, design, and configuration R2020b. URL[<https://it.mathworks.com/help/simulink/modeling-guidelines.html>]
- [20] Harvey M. Wagne. *Global Sensitivity Analysis*; 1 Dec 1995h.
- [21] Hendrik Puls, Fritz Klocke, Drazen Veselovac. *FEM-based prediction of heat partition in dry metal cutting of AISI 1045*. Int J Adv Manuf Technol (2016) 86:737–745, DOI 10.1007/s00170-015-8190-z
- [22] Fangjuan Zhou. *A new analytical tool-chip friction model in dry cutting*. Int J Adv Manuf Technol (2014) 70:309–319, DOI 10.1007/s00170-013-5271-8
- [23] WIT GRZESIK, JOEL RECH. *Influence of machining conditions on friction in metal cutting process – A review*. DOI: <https://doi.org/10.17814/mechanik.2019.4.33>
- [24] Abdil Kus, Yahya Isik, M. Cemal Cakir, Salih Coşkun, Kadir Özdemir. *Thermocouple and Infrared Sensor-Based Measurement of Temperature Distribution in Metal Cutting*. Sensors 2015, 15, 1274-1291; doi:10.3390/s150101274
- [25] A. Fata. *Temperature measurement during machining depending on cutting conditions*. PA Sci. Technol. 2011, 1, 16–21.