Politecnico di Torino
Department of Control and Computer Engineering

Institute for Engineering and Architecture

Thesis

For the Degree of

Master of Science in Mechatronics Engineering

End-effector tools SoH prediction: parameter identification and reliability estimation

By

Luca Cecere

Submission Date: April, 2021

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

Abstract

Estimation of remaining useful life and State of Health (SoH) of machines plays a vital role in performing predictive maintenance for complex systems today. Despite the numerous researches on this topic and the huge number of requests from companies, it still remains a challenge. On the one hand, current techniques rely on machine learning algorithms not taking into account the huge amount of computational power needed and the costs of such server architectures. On the other hand, developing an edge-computing solution considering strong computational power limits can be very challenging. To address this issue, this thesis work proposes a possible edge-computing approach, based on up-to-date parameter estimations techniques such as Kalman filtering, residual error processing and global sensitivity analysis. The thesis has been carreid out in the framework of MorePRO project, owned and supervised by a collaboration between Politecnico di Torino and Brain Technologies Srl. The goals of this work are to employ a multi-model approach to estimate key parameters that can be associated to CNC machine wear, to find the best way to estimate the reliability of such predictions and then to use this reliability index to update the nonlinear predictive models for the forecast of system states. The approach adopted in this work is firstly to study the state of the art, comparing the methods and figuring out the advantages and disadvantages, secondly, to practically implement the previously mentioned approaches. Finally, various tests were carried out to empirically demonstrate the feasibility of this approach and to provide solid proofs and conclusions.

Contents

List of Figures

List of Tables

Acronyms

MPC model predictive control

SoH State of health

ERMES Extendible Range MultiModal Estimator Sensing

DOE Design of experiment

KF Kalman filter

EKF Extended Kalman filter

EM Electric Machine

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

PM Predictive Maintenance

ANN Artificial Neural Network

HMM Hidden Markov Model

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 production efficiency. 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 electronic 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) [1] machine are high precision machines which actuate manufacturing processes of material substraction 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 indipendent 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.

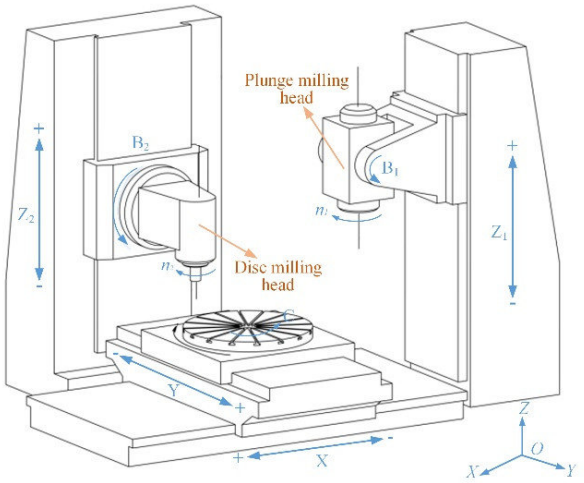


Figure 1.1. Example of CNC machine schematic diagram

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. It 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 definitly 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 cutting process of 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 microprocessors 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 uninterruptible 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 based on.

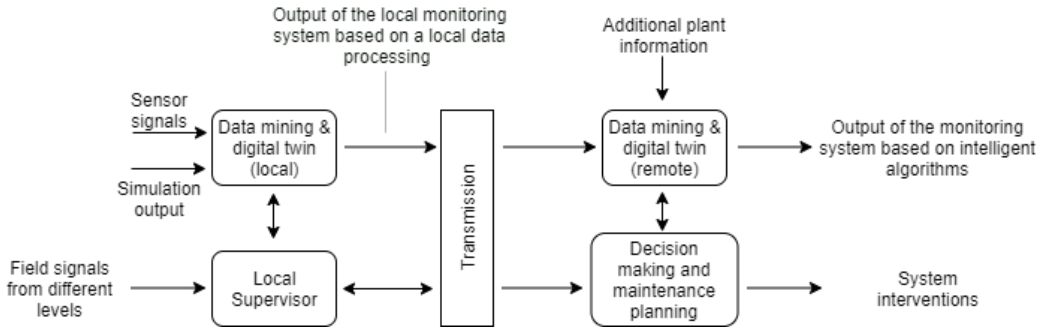


Figure 1.2. Synthetic structure of MorePRO system

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 re-configure and support the planning decisions. A possible physical implementation can be seen in the next figure.

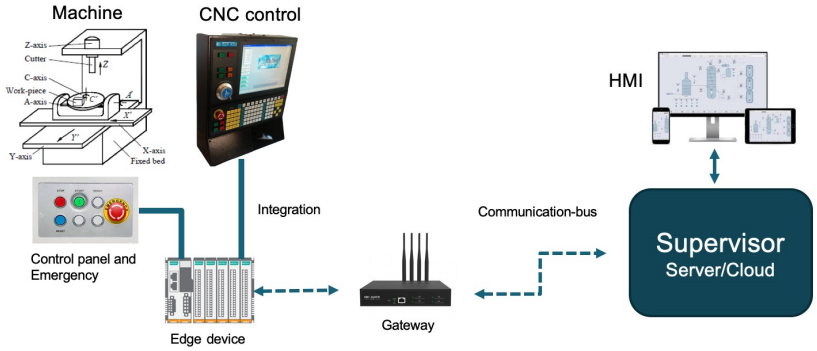


Figure 1.3. Overall system structure

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 recommend 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.

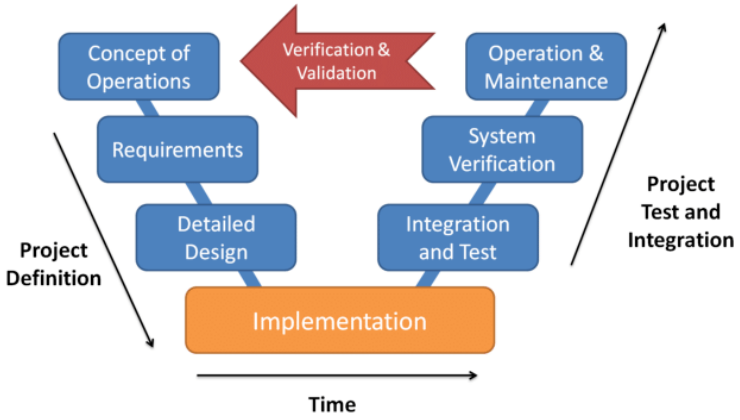


Figure 1.4. V-shape development flow.

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 aroused a team of graduating students supervised by Giovanni Guida, Innovation Manager of brain Technologies, with the aim of developing the first phases of the project. 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 that are explained in detail in 1.5.3 paragraph.

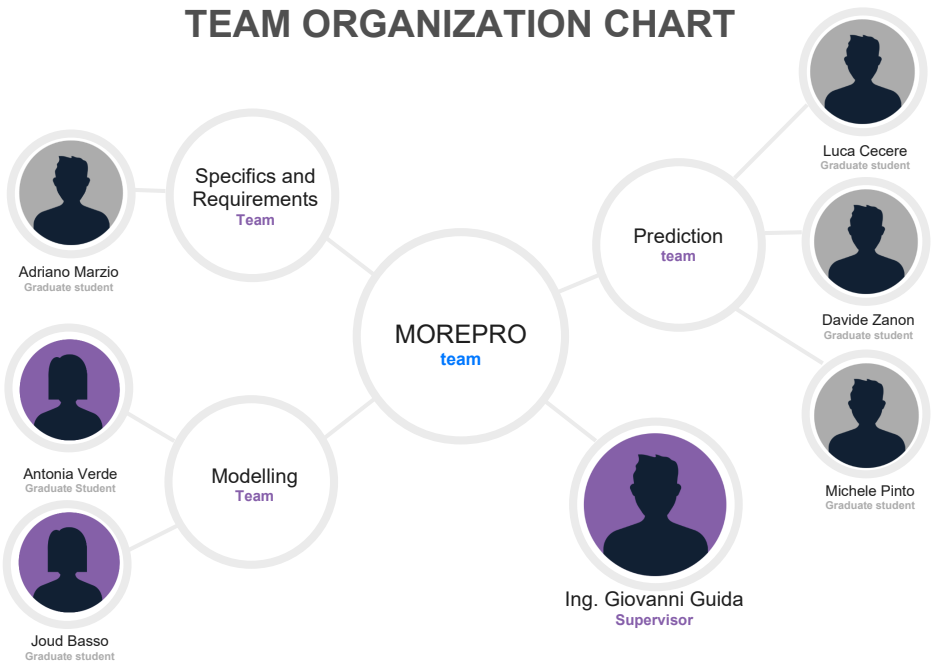


Figure 1.5. Team organization chart.

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:

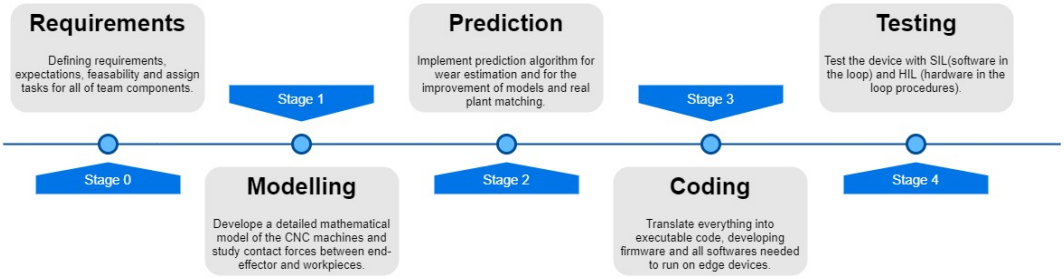


Figure 1.6. Workflow schematic.

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. Nevertheless, it is important to keep in mind a clear idea of the pre-determined work-flow.
For what regards this thesis, the work-flow is well defined and it is listed as follows:

1. State of art analysis: Comparison with all the current techniques present in the literature about parameter estimation and predictive maintenance.
2. Development of simple model for simulation scopes: In this stage the aim is to build a simulation suitable model that has to be the base for the prediction algorithm. This model is intended to be very simple in terms of physics and mathematical modelling.The idea is to test the algorithm in a simple environment as first, and then complicate the modelling until the real plant testing is ready.
3. Multi-model approach: This phase is the most crucial because it starts the actual development of the prediction algorithm. The scope of this stage is to apply the multi-model approach(only applied for Battery management systems until now) for the estimation of SoH of the end-effector.
4. Specific features: As last stage but not least, there is the implementation of specific improvements to the estimation algorithm that include machine learning techniques, reliability calculation and modelling betterment.

1.5.3 Objective of this thesis

The primary focus of this research project is to obtain a competitive knowledge about simple mechatronic modelling and parameter identification, aiming to the development of a fully operational system (starting from simulation environments) that is able to estimate the SoH of an end effector and determinate the precision and the reliability of that estimation. This thesis work arises in the scope of the MOREPRO project by brain Technologies srl, and it comes from the necessity of many companies requirements. This means that the work must match as much as possible all the needs of nowadays businesses both in terms of time and in terms of efficiency, in order to strongly overcome in the market. The first part of the study will be devoted to the application of the Batman concepts, while the second part is

the characterization work of this thesis: Reliability estimation through innovative techniques and model matching updates.

1.6 Thesis outline

The contents of this Thesis are organized as presented in the following. In Chapter 1, an introduction related to the predictive maintenance techniques is presented, starting from the CNC machine SoH and explaining the edge computing advantages with respect to the cloud computing. MOREPRO project is also introduced and presented.
In Chapter 2, an in-depth study of the state of the art literature in topics of interest for this work is carried out. In particular, the problem of estimation of the wear conditions and the parameter identification is analyzed from many possible current techniques.
In Chapter 3, physical model is carried out, starting from the mechanical and electrical part and explaining the simulink implementation.
In Chapter 4, it is explained the multi-model approach. EKF working principle is summed up and the principal application of the residual error analysis for the estimation of the SoH is introduced. Moreover, in this chapter the main blocks used in simulink are reported and explained.
Chapter 5 is core of the thesis, it explain the important concept of observability and introduce the main method for the reliability computation.
In Chapter 6, on the basis of the considerations made in the previous chapter, I want to observe the effect of the presence of exogenous disturbances on the algorithm performance. In detail, the exogenous disturbances will be considered on the most significant parameters already identified by the Sensitivity Analysis carried out previously.
In Chapter 7, the integration with the updates developed from the other team components is explained. Moreover, the tests are presented as proof of the main results of the thesis.
Chapter 8 are the conclusions of the thesis project and presents also some comments about the future works.

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. [5]

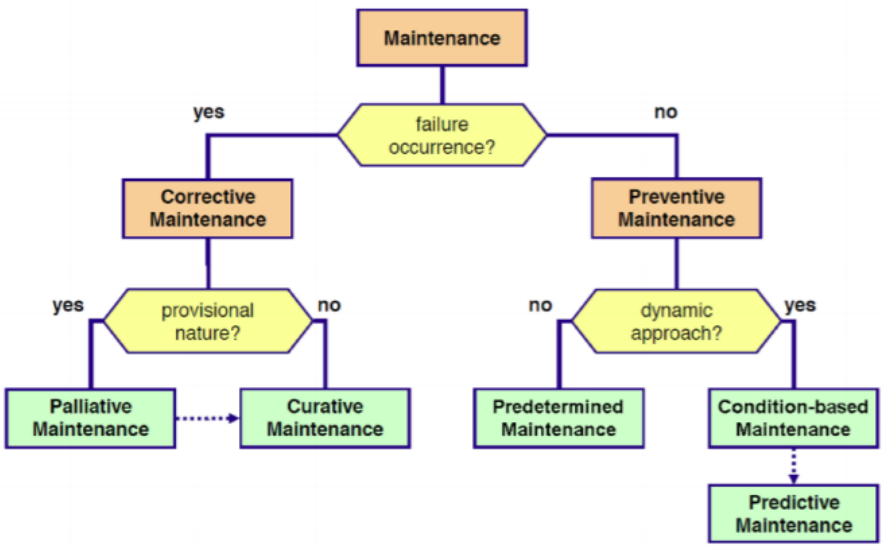


Figure 2.1. Forms of maintenance

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 adhers 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. [6]

   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. [15]

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 [7].

- ***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 [8]. 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:

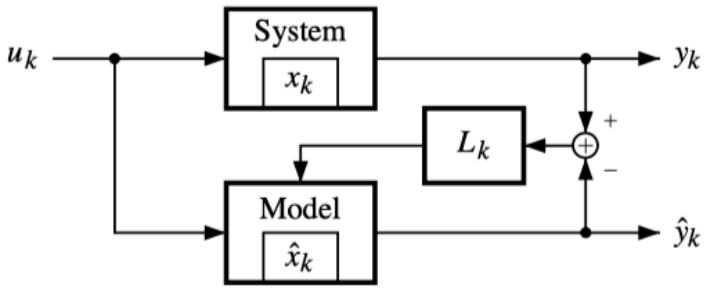


Figure 2.2. Schematic of state update

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 (LVT). 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 $(\tau + \mathrm{p})$. 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) is 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.[6, 9]

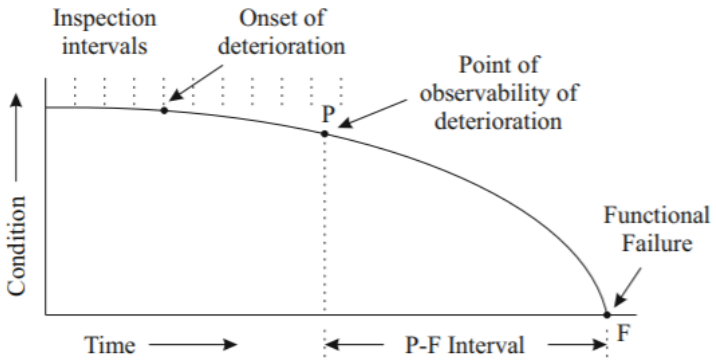


Figure 2.3. P-F curve

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 [10]. 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 nonlinear 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 back-propagation algorithm, to map the connection between the input and output. Neural networks are self-adaptive 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.

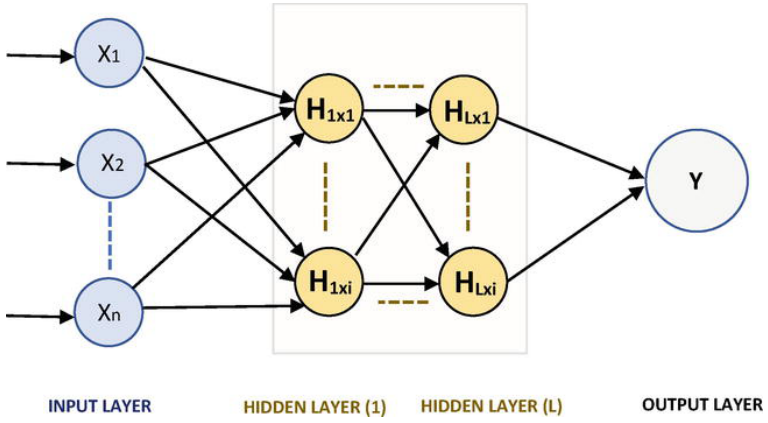


Figure 2.4. Schematic of a Artificial Neural Network

- ***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 [11]. 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.

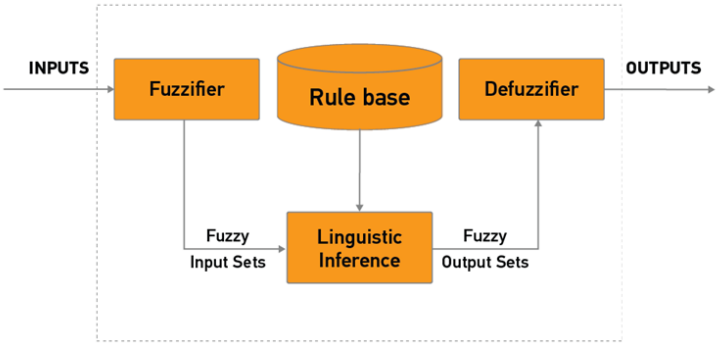


Figure 2.5. Components of a Fuzzy logic system

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 [12].

- ***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, $S = (s_1, s_2, \ldots, s_N)$ and M observation symbols, $V = (v_1, v_2, \ldots, v_M)$ is shown in the schematic below.

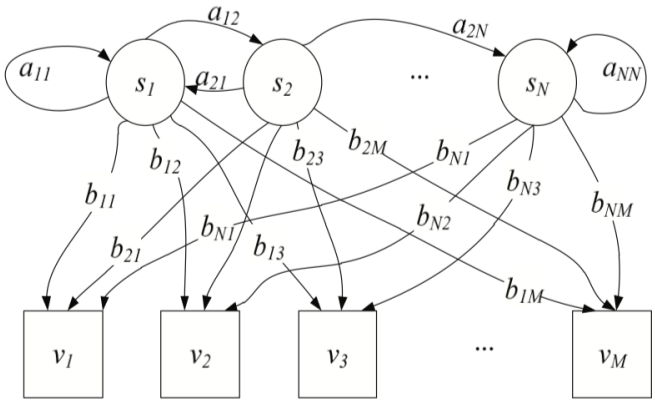


Figure 2.6. Schematic of HMM

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 "Developing a real-time data-driven battery health diagnosis method, using time and frequency domain condition indicators" [13] perfectly sum up the main features in brief. 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

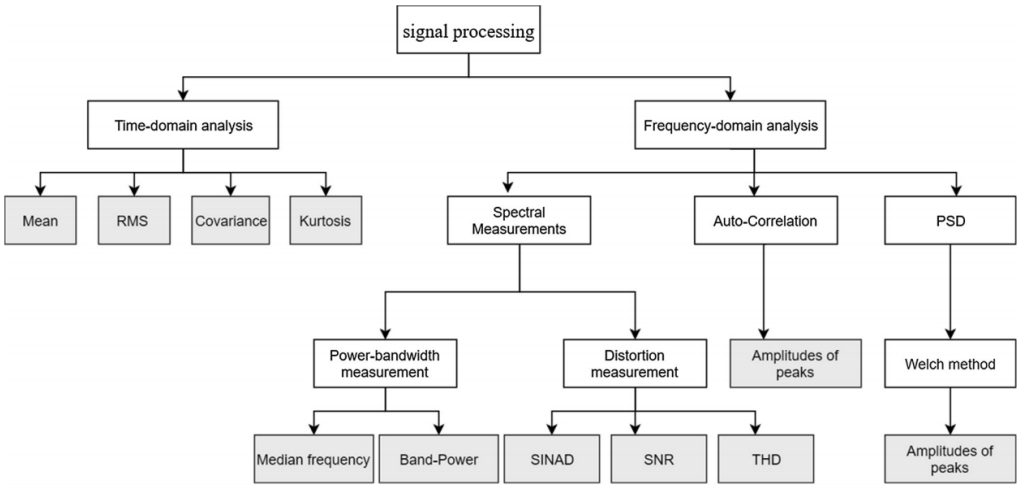


Figure 2.7. Flow diagram of condition indicator construction. The gray boxes indicate the condition indicators.[13]

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. These 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, multimodal analysis and ML.

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 an EKF bank and whose main goal is the realization 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 4, reference related to this project is *Virtual Sensing for the Estimation of the State of Health of batteries* [14]).

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 hypotesis 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 [15]. 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 technologic 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 study. However, considering the high complexity of a CNC machine, it has been decided to start from a 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 [16]. 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 elctrical 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.

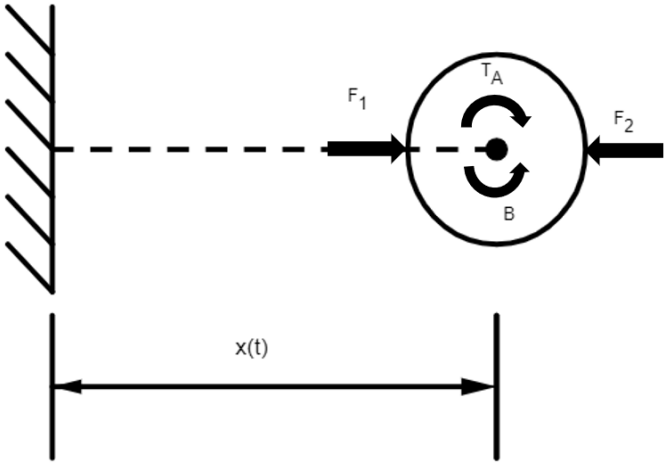


Figure 3.1. Simplified milling machine model

The state equations of the systems are obtained through a classical Newton formulation. 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 very easy to trace the two Newton equations:

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

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 [17]. 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.

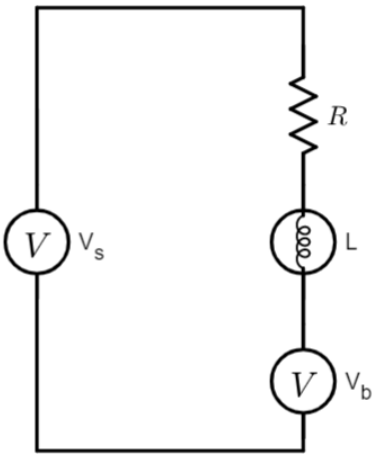


Figure 3.2. Simplified schematic of a DC motor

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.

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

Thus, playing a little bit with the equations:

$$k_t i_a(t) - Att_{mot}\dot{\theta} = I_L\dot{\theta}$$
$$Vs = Ri_a(t) + L\frac{di_a(t)}{dt} + k_b\dot{\theta}$$

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:

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

Thus, it is possible to notice that two main variables affect the speed of the motor: the supply Voltage and the Load Torque.

3.3 Plant model

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

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

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

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

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 of another research group working on the same project.

3.4.1 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 one can improve the consistency, clarity, and readability of 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 [18].

In 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.

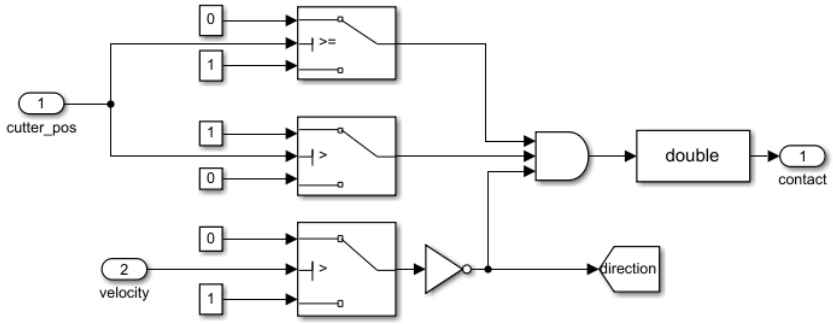


Figure 3.3. Contact logic

For the implementation of the equations described in 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:

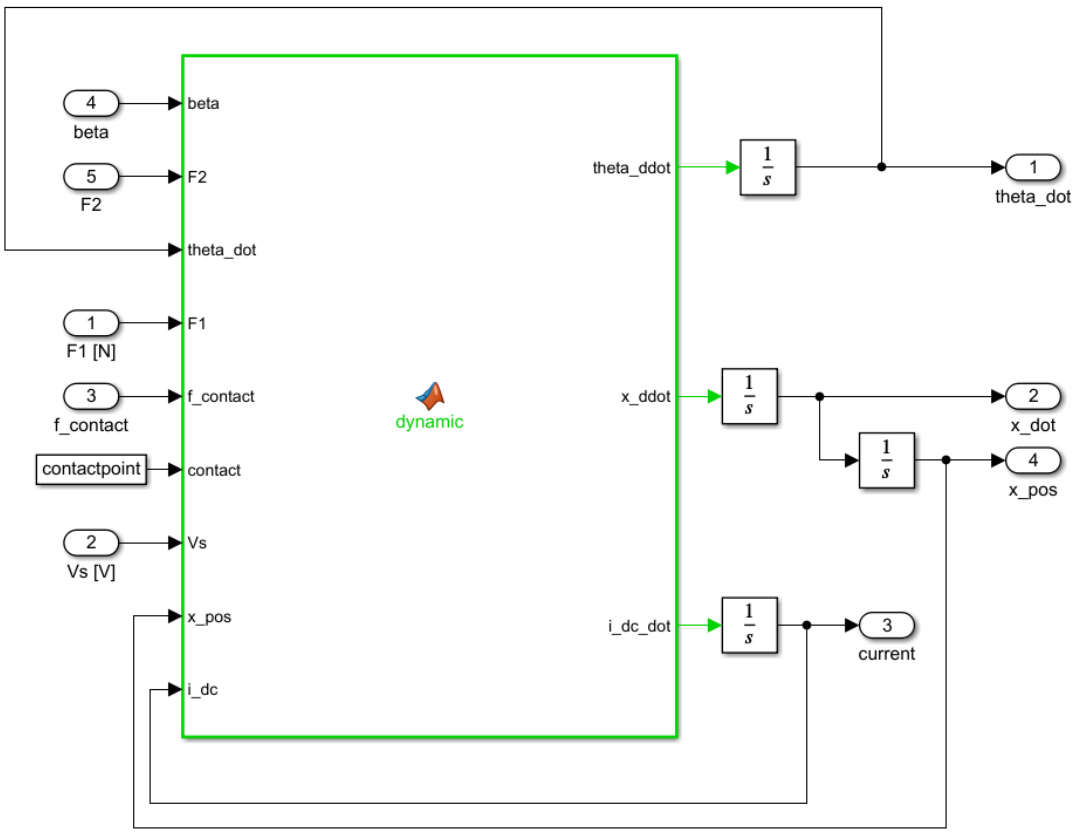


Figure 3.4. Plant Simulink.

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.

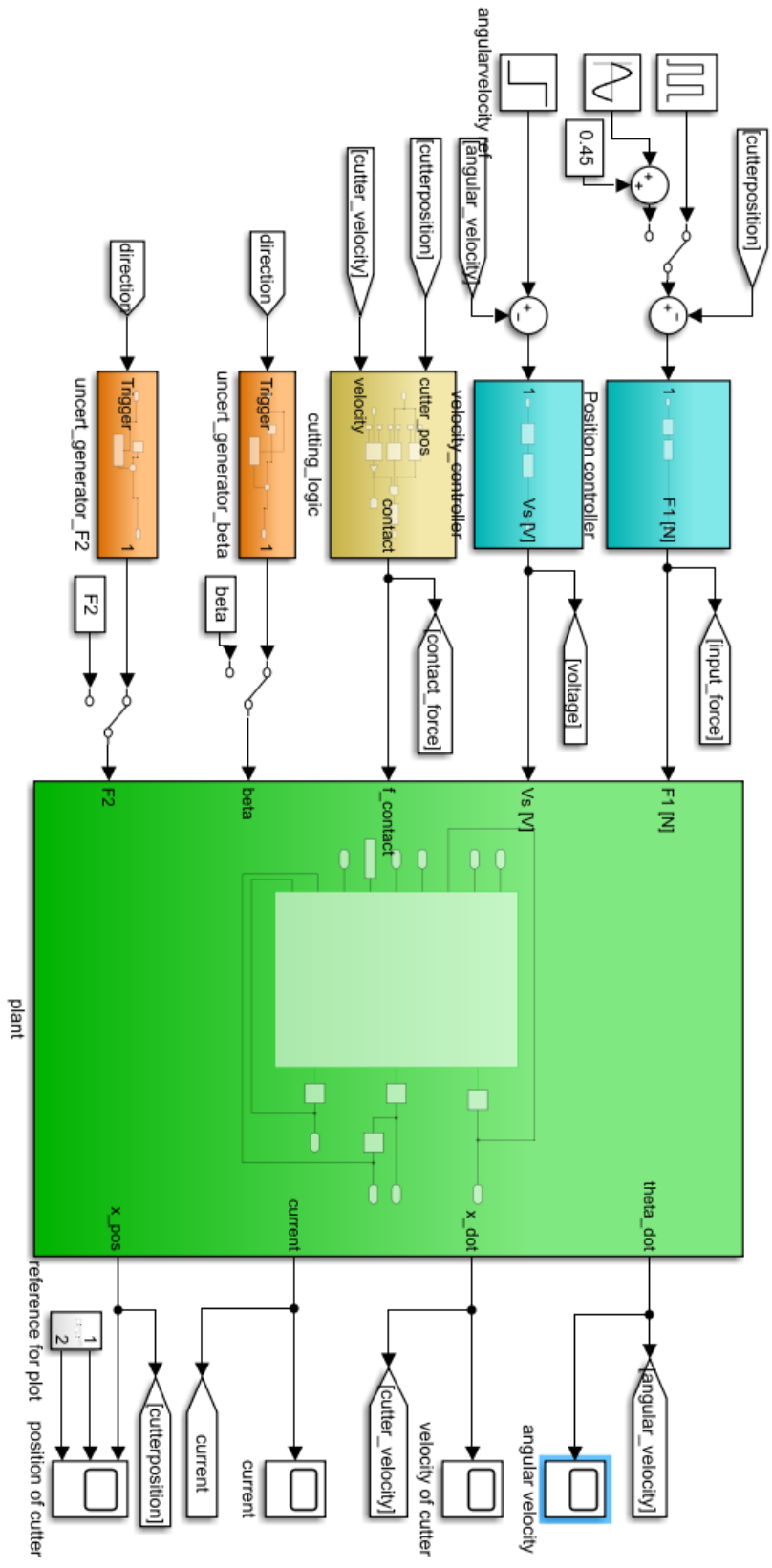


Figure 3.5. Simulink implementation of the model

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 Tooolbox) in such a way to find a good balance between robustness and efficiency. Instead, the next plot represents the contact logic output:

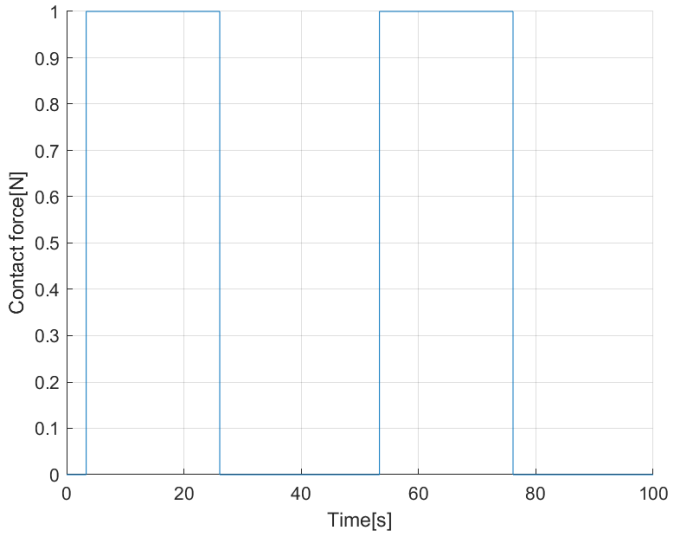
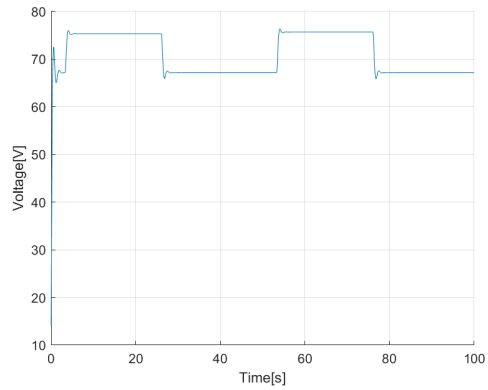
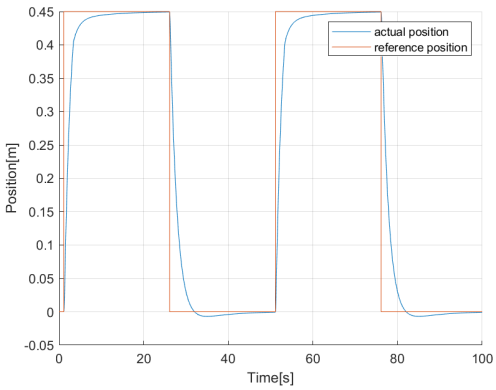


Figure 3.6. Contact force control input plot.

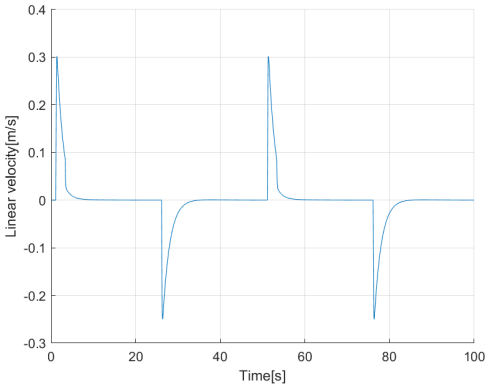
As it was expected, since the contact logic is made up so that the output is 1 when there is contact and 0 when there is no contact, the contact force input plot oscillate in a discrete way between those two values. Finally, in Figure 3.7 are depicted all outputs and main parameters of the model so that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



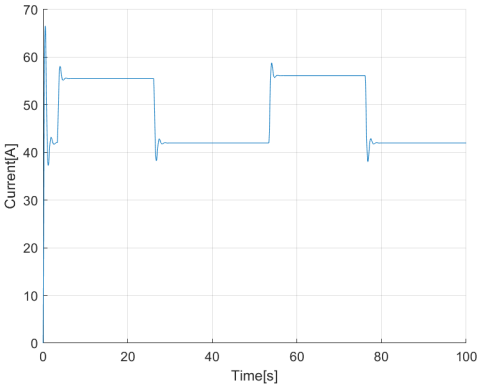
(a) Voltage plot



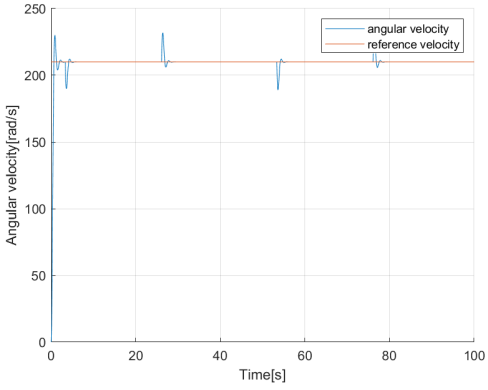
(b) Position plot



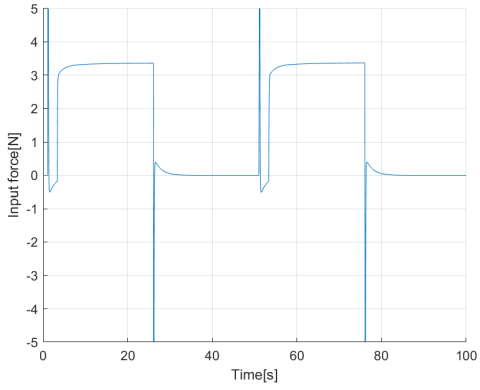
(c) Linear velocity plot



(d) Current plot



(e) Angular velocity



(f) Input force plot

Figure 3.7. Summary plots of Model's main parameters.

4 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 $\mathcal{M}(x(t))$:

$$Residual\_error = |\hat{x(t)} - x(t)| \quad \forall t$$

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[19].

4.1 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:

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

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:

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

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).

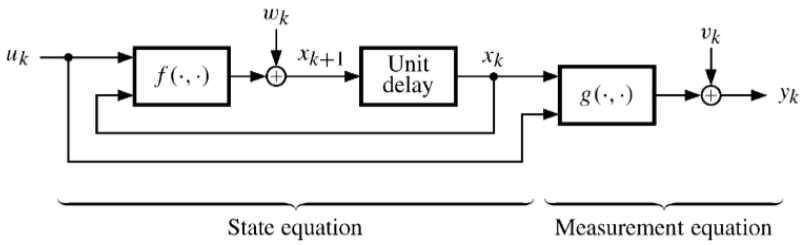


Figure 4.1. Diagram of nonlinear discrete time system in state-space form

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:

- $F_k = \dfrac{\partial f}{\partial x_k}(x_k, u_k)$ = Jacobian of f computed in (x_k, u_k)
- $H_k = \dfrac{\partial h}{\partial x_k}(x_k)$ = 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**

$$x_k^p = f(\hat{x}_{k-1}, u_{k-1})$$
$$P_k^p = F_{k-1} P_{k-1} F_{k-1}^T + Q^d$$

2. **Update**

$$S_k = H_k P_k^p H_k^T + R^d$$
$$K_k = P_k^p H_k^T S_k^{-1}$$
$$\Delta y_k = y_k - h(x_k^p)$$

$$\hat{x}_k = x_k^p + K_k \Delta y_k$$
$$P_k = (I - K_k H_k) P_k^p$$

In addition, a further step was added to the algorithm to calculate the residual error for each state variable, which we recall is the modulus of the difference between the estimates produced by EKF and the data collected from the simulation of the plant.

$$Residual\_error = |x(\hat{k}) - x(k)| \quad \forall k \tag{4.1}$$

The final output of the EKF block are therefore the residual errors that is needed to carry out an analysis and establish whether a multi-model approach may be better for the final objective. In the figure 4.2 there is the Simulink implementation of the EKF.

Figure 4.2. Simulink scheme of the EKF.

4.2 Residual error analysis

Considering the residual error as raw value, it is not significant to watch the errors amount every sample time since it does not lead to any particular conclusion. To assign a meaningful sense to the residual error, it must be conducted a signal elaboration and the discrete values of the error sampled in time must be processed. Signal theory and data processing are a widely treated in today scientific literature, so there are countless articles that can be followed in order to understand the best 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 [13]. 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 kept 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

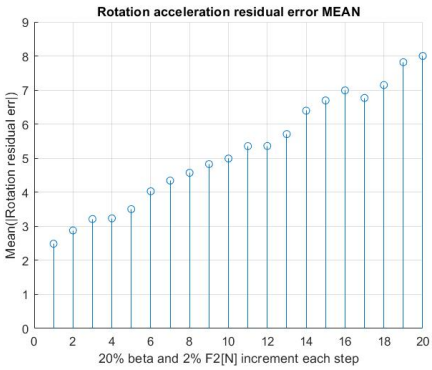
Table 4.1. Nominal values of the errors for each method.

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:

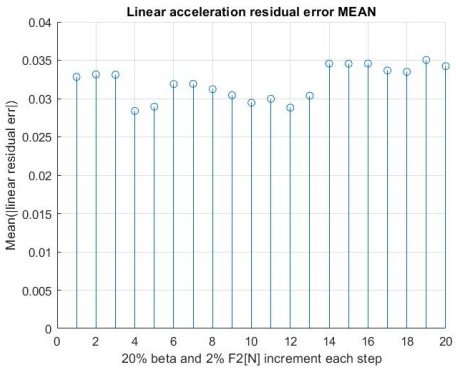
$$FOM = \frac{|\text{Residual error parameter}|}{|\text{Nominal residual error parameter}|}$$

Keeping in consideration the nominal values reported in Table 4.1, the following plot will show which signal manipulation can be considered as the most suitable. Let's start the tests from each method:

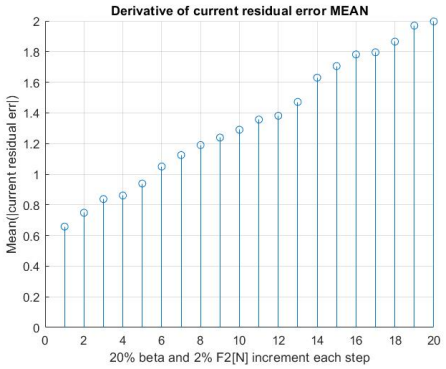
1. **MEAN:**



(a) Rot. acc.



(b) Lin. acc.



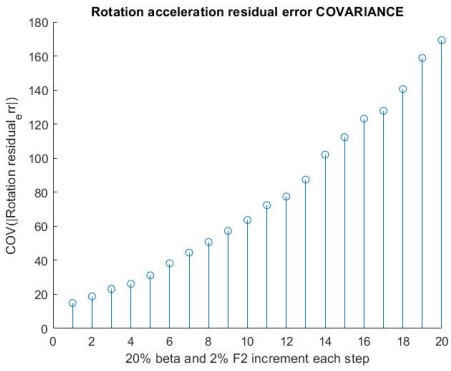
(c) Curr. der

Figure 4.3. Mean test.

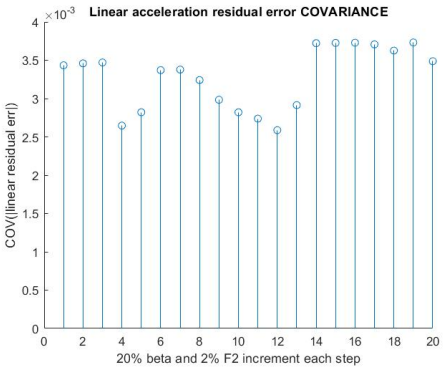
FOM angular	FOM linear	FOM current
4.6759	1.4162	4.0266

Table 4.2. FOM mean

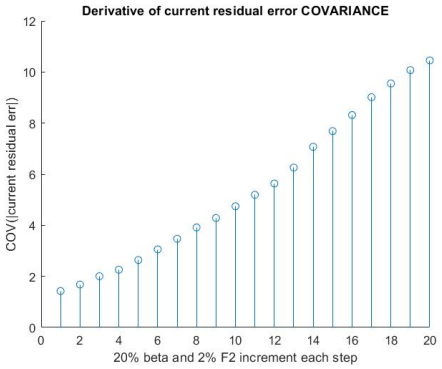
2. **Covariance:**



(a) Rot. acc.



(b) Lin. acc.



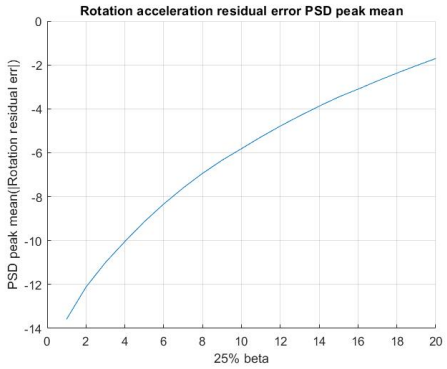
(c) Curr. der

Figure 4.4. Covariance test.

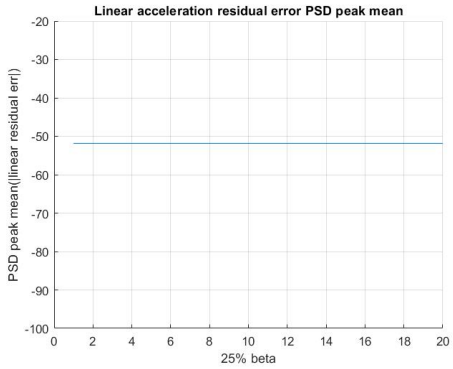
FOM angular	FOM linear	FOM current
5.9107	1.7358	7.9633

Table 4.3. FOM Covariance

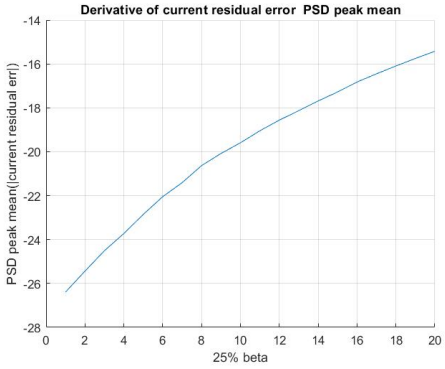
3. **PSD:** For the power spectral density, it has been considered an interpolation of the maximum peaks.



(a) Rot. acc.



(b) Lin. acc.



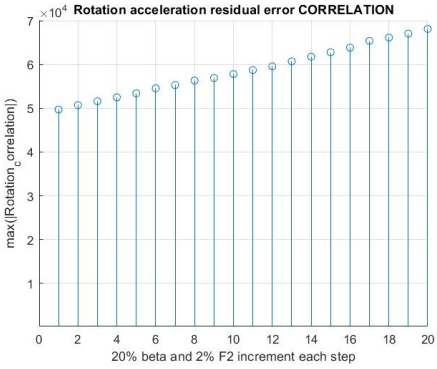
(c) Curr. der

Figure 4.5. PSD test.

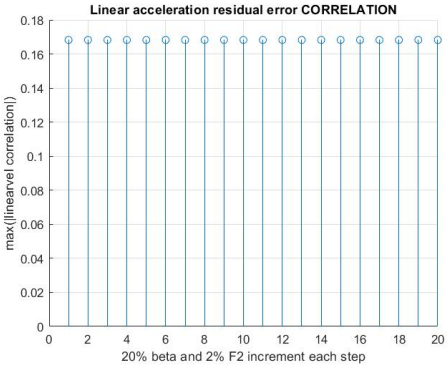
FOM angular	FOM linear	FOM current
7.527	1.001	5.743

Table 4.4. FOM PSD

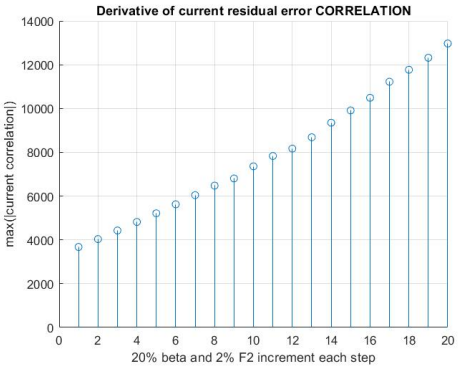
4. **Correlation:**



(a) Rot. acc.



(b) Lin. acc.



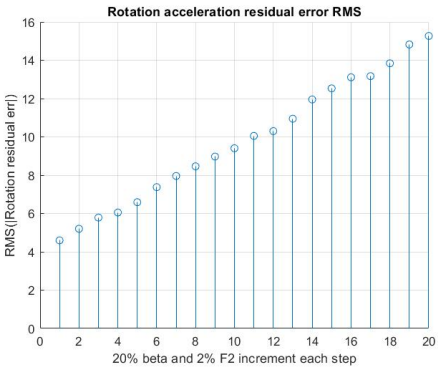
(c) Curr. der

Figure 4.6. Correlation test.

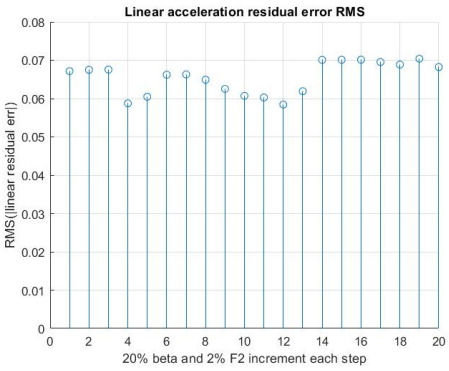
FOM angular	FOM linear	FOM current
1.3979	1.0000	3.8928

Table 4.5. FOM Correlation

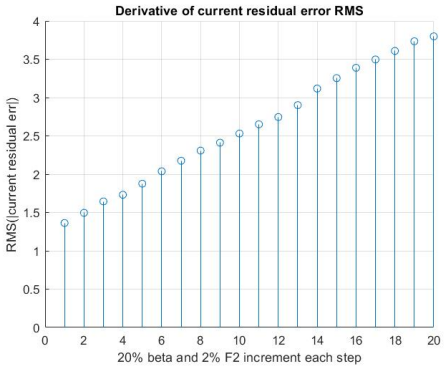
5. **RMS:**



(a) Rot. acc.



(b) Lin. acc.



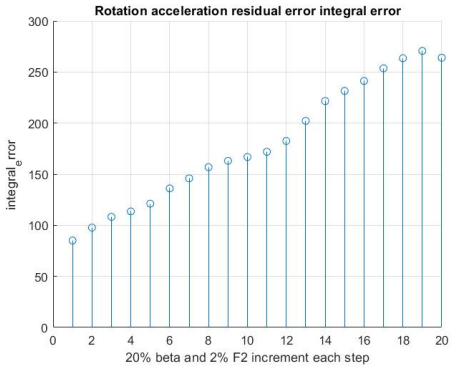
(c) Curr. der

Figure 4.7. RMS test.

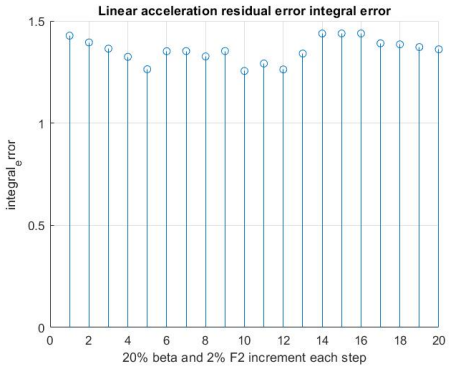
FOM angular	FOM linear	FOM current
4.1469	1.3404	3.1979

Table 4.6. FOM Correlation

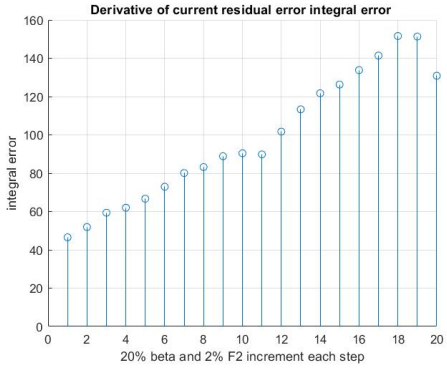
6. **Integral:**



(a) Rot. acc.



(b) Lin. acc.



(c) Curr. der

Figure 4.8. Integral test.

FOM angular	FOM linear	FOM current
4.4911	1.0886	3.8739

Table 4.7. FOM Integral

Results can be summed up in the following tables:

- Angular acceleration:

	mean	*integral*	*RMS*	*cov*	$corr_{max}$	*PSD*
	1.93	58	5.8	30	6.1	−19.9
$\beta\uparrow$	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

mean	*integral*	*RMS*	*cov*	$corr_{max}$	*PSD*
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

	mean	*integral*	*RMS*	*cov*	$corr_{max}$	*PSD*
	0.67	20	1.8	3.0	9.2	−25.9
$\beta\uparrow$	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.

4.2.1 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 4.1.
- 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 (details can be found in appendix) 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 [kΩ]	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 4.8. 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 4.9. 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.

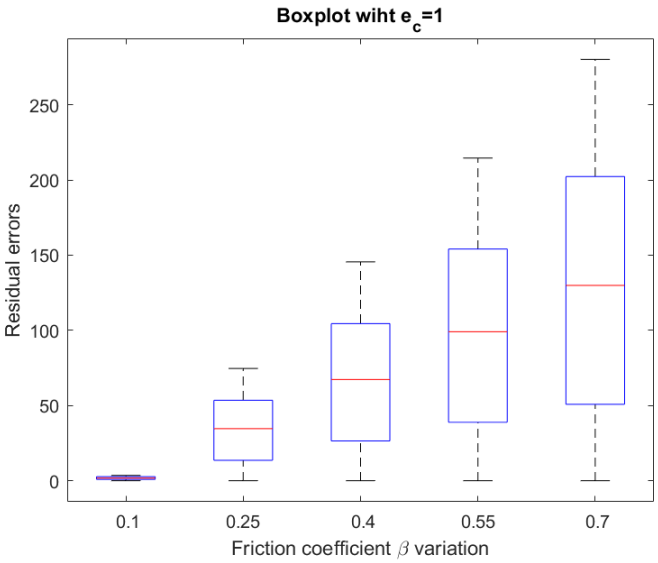


Figure 4.9. Boxplot of the angular acceleration error.

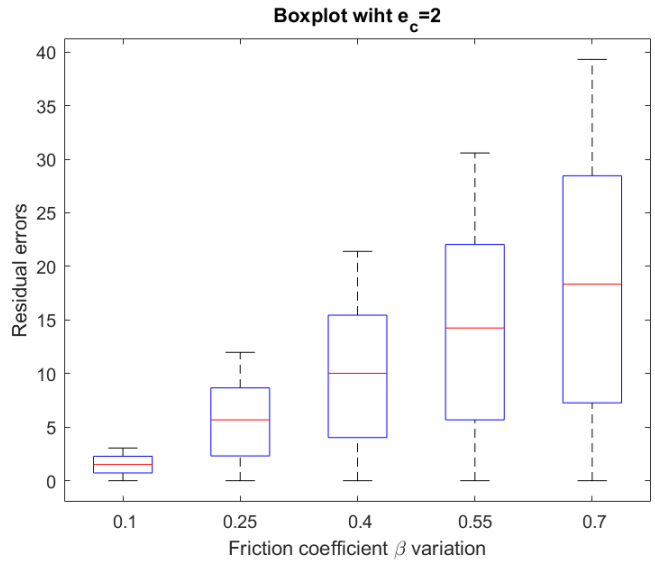


Figure 4.10. Boxplot of the current derivative error.

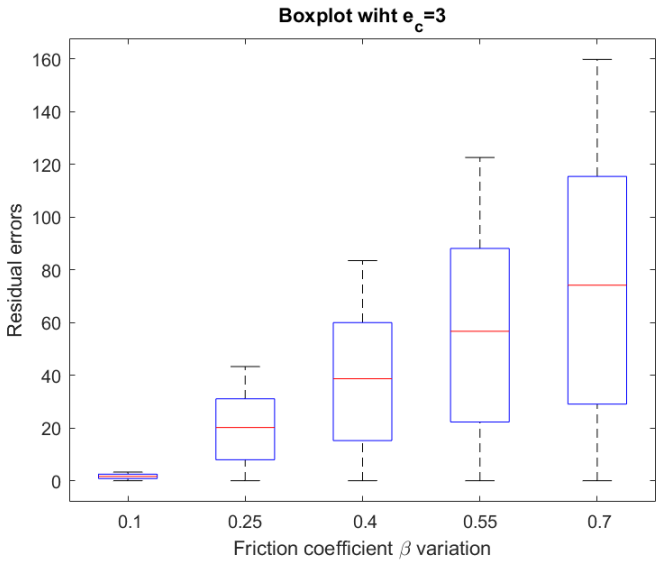


Figure 4.11. Boxplot of the average error.

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.

4.3 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:

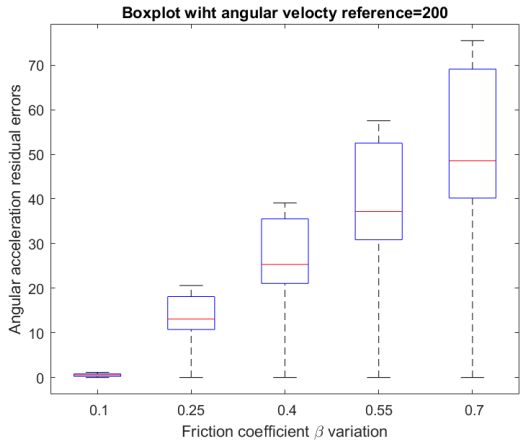


Figure 4.12. Boxplot error with 200 $[\frac{rad}{s^2}]$ angular velocity.

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

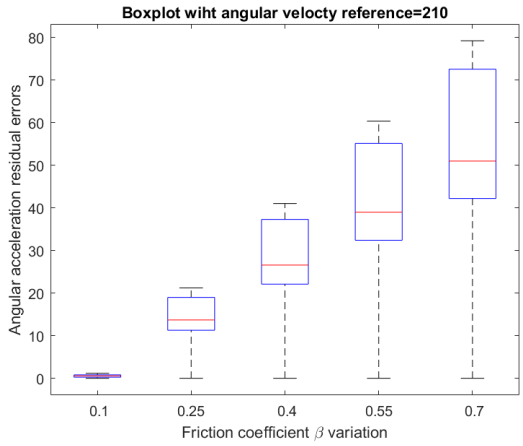


Figure 4.13. Boxplot error with 210 $[\frac{rad}{s^2}]$ angular velocity.

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

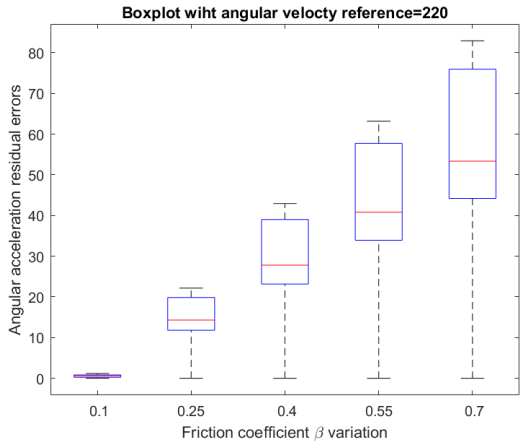


Figure 4.14. Boxplot error with 220 $\frac{rad}{s^2}$ angular velocity.

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

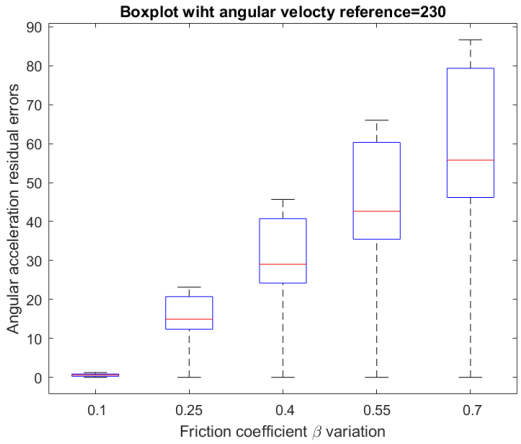


Figure 4.15. Boxplot error with 230 $[\frac{rad}{s^2}]$ angular velocity.

- **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 4.13. Test with 0.4 $[m]$ position reference.

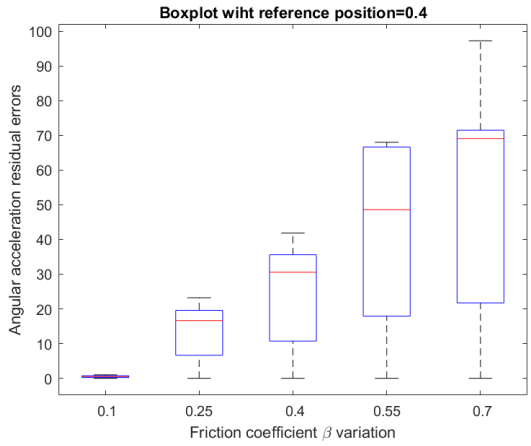


Figure 4.16. Boxplot error with 0.4 $[m]$ position reference.

	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 4.14. Test with 0.47 $[m]$ position reference.

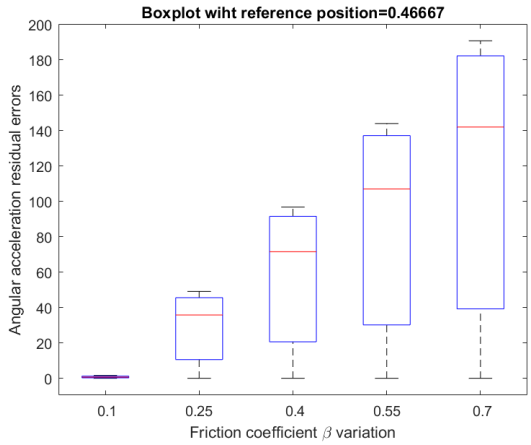


Figure 4.17. Boxplot error with 0.47 $[m]$ position reference.

	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 4.15. Test with 0.53 $[m]$ position reference.

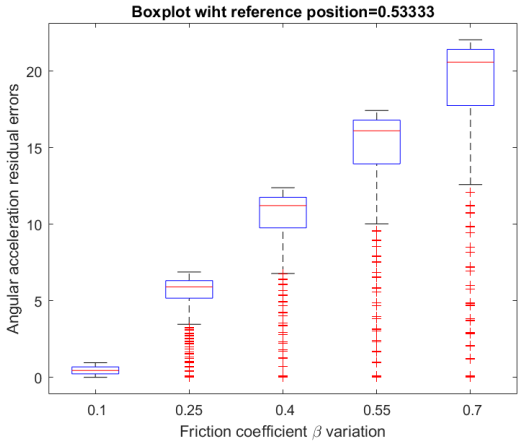


Figure 4.18. Boxplot error with 0.53 $[m]$ position reference.

	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 4.16. Test with 0.6 $[m]$ position reference.

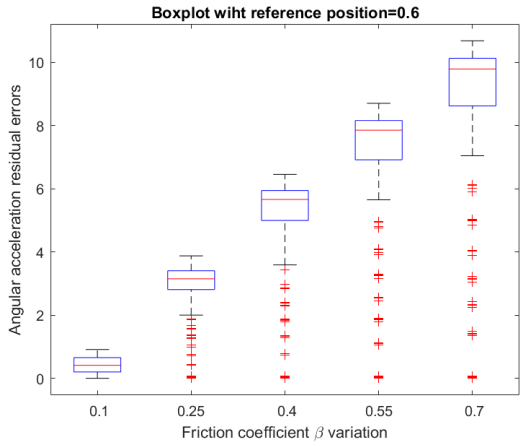


Figure 4.19. Boxplot error with 0.6 $[m]$ position reference.

- **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 4.17. Test with 20 % duty cycle.

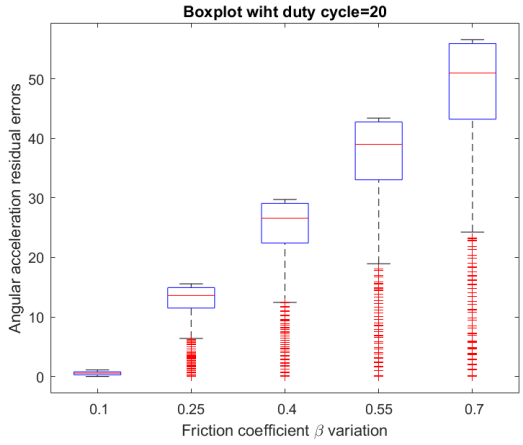


Figure 4.20. Boxplot error with 20 % duty cycle.

	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 4.18. Test with 40 % duty cycle.

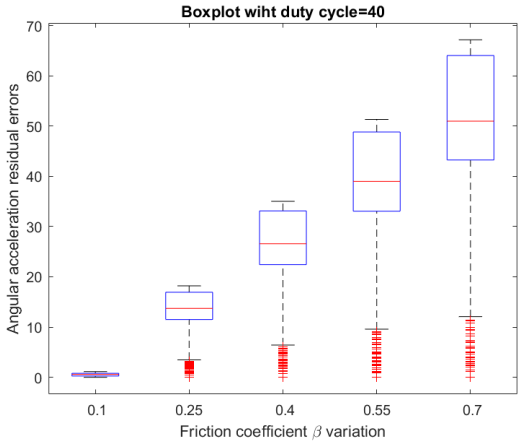


Figure 4.21. Boxplot error with 40 % duty cycle.

	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 4.19. Test with 60 % duty cycle.

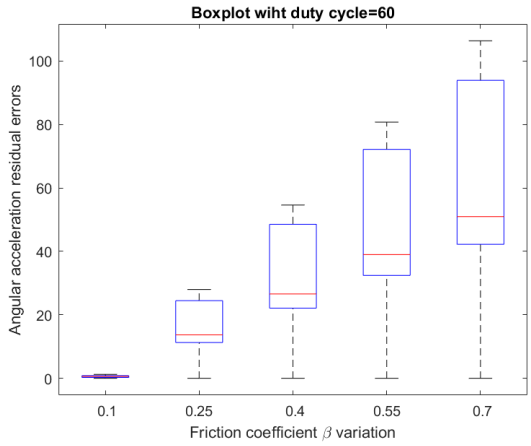


Figure 4.22. Boxplot error with 60 % duty cycle.

	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 4.20. Test with 80 % duty cycle.

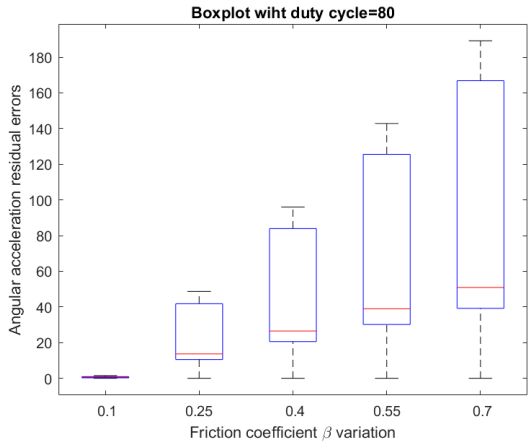


Figure 4.23. Boxplot error with 80 % duty cycle.

- **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 4.21. Test with 4 number of cycles.

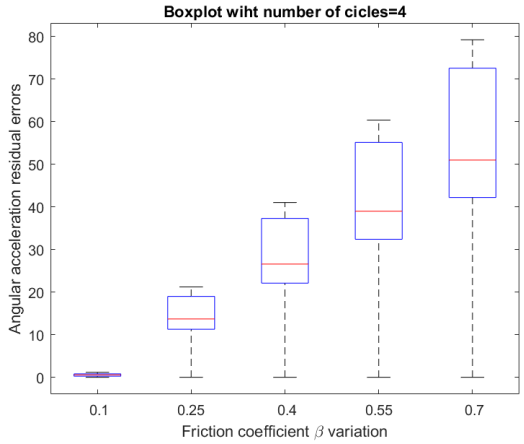


Figure 4.24. Boxplot error with 4 number of cycles.

	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 4.22. Test with 6 number of cycles.

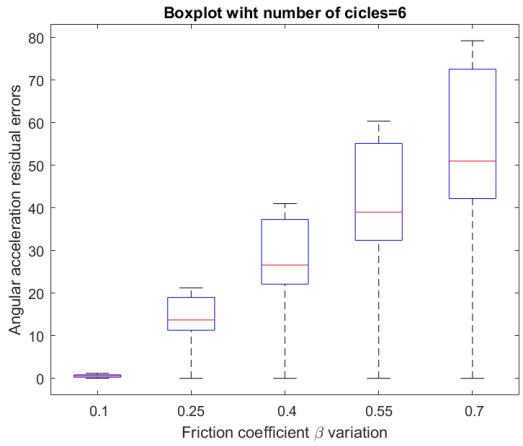


Figure 4.25. Boxplot error with 6 number of cycles.

	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 4.23. Test with 8 number of cycles.

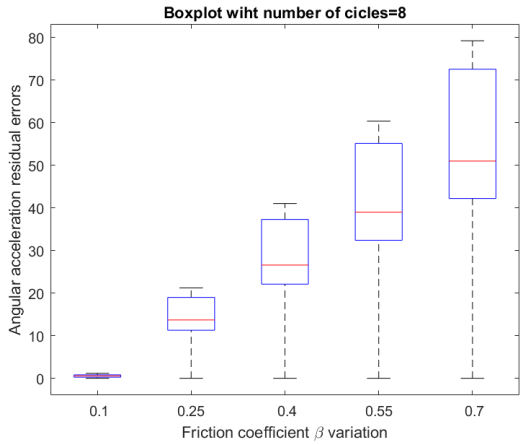


Figure 4.26. Boxplot error with 8 number of cycles.

	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 4.24. Test with 10 number of cycles.

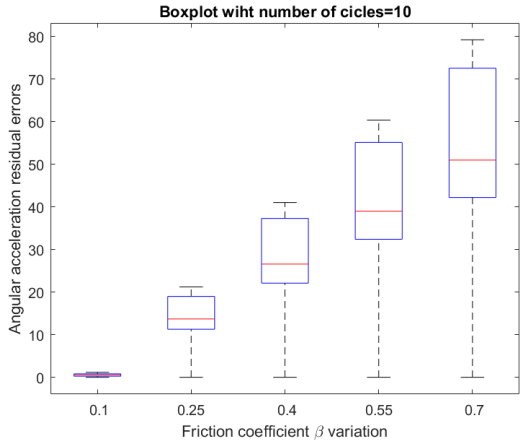


Figure 4.27. Boxplot error with 10 number of cycles.

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 than 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 4.3.1.

4.3.1 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.

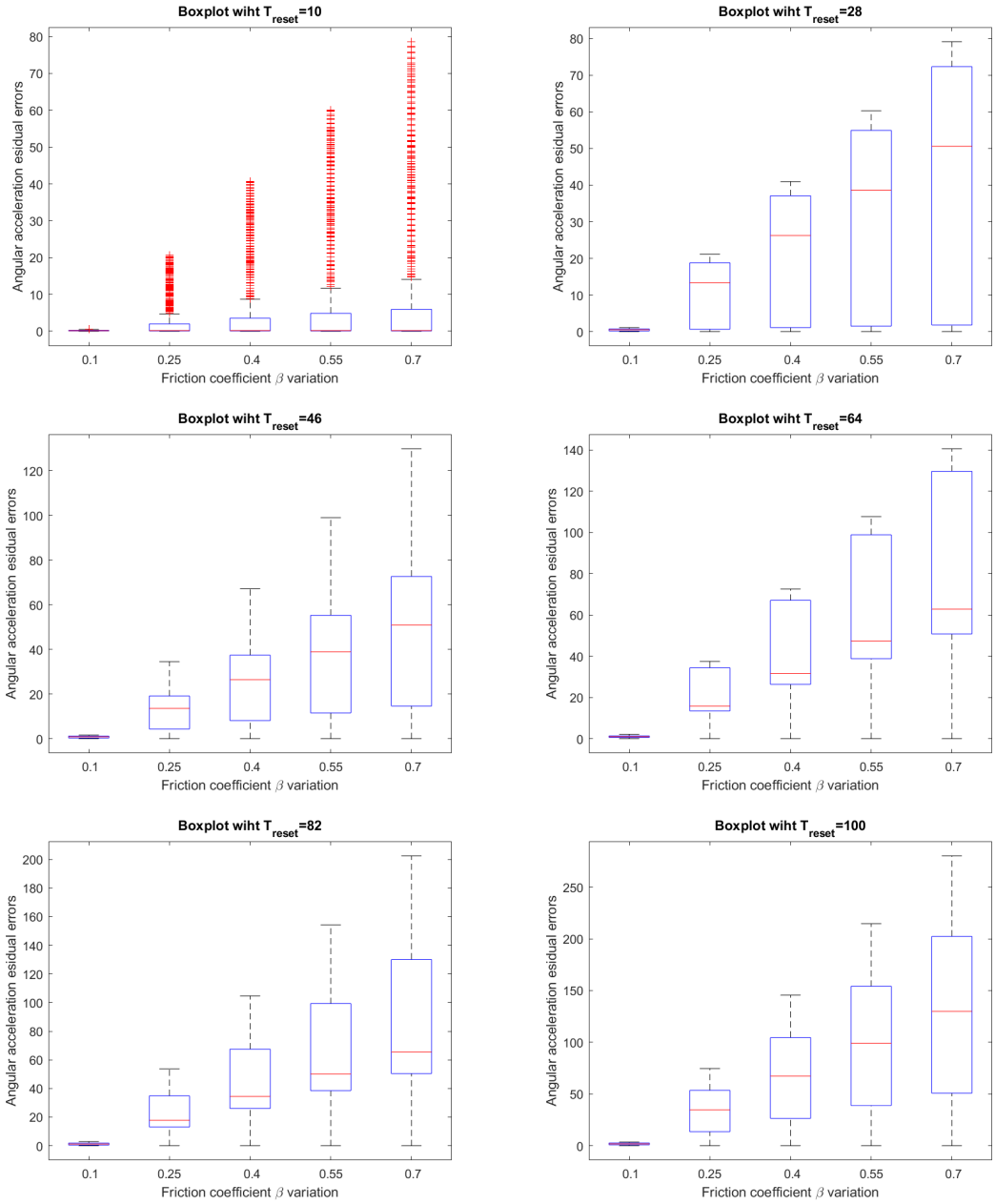


Figure 4.28. T reset analysis

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 it has been decided to investigate a narrower range right after 30s because by

increasing more and more the reset time it is possible to run into the problems listed above.

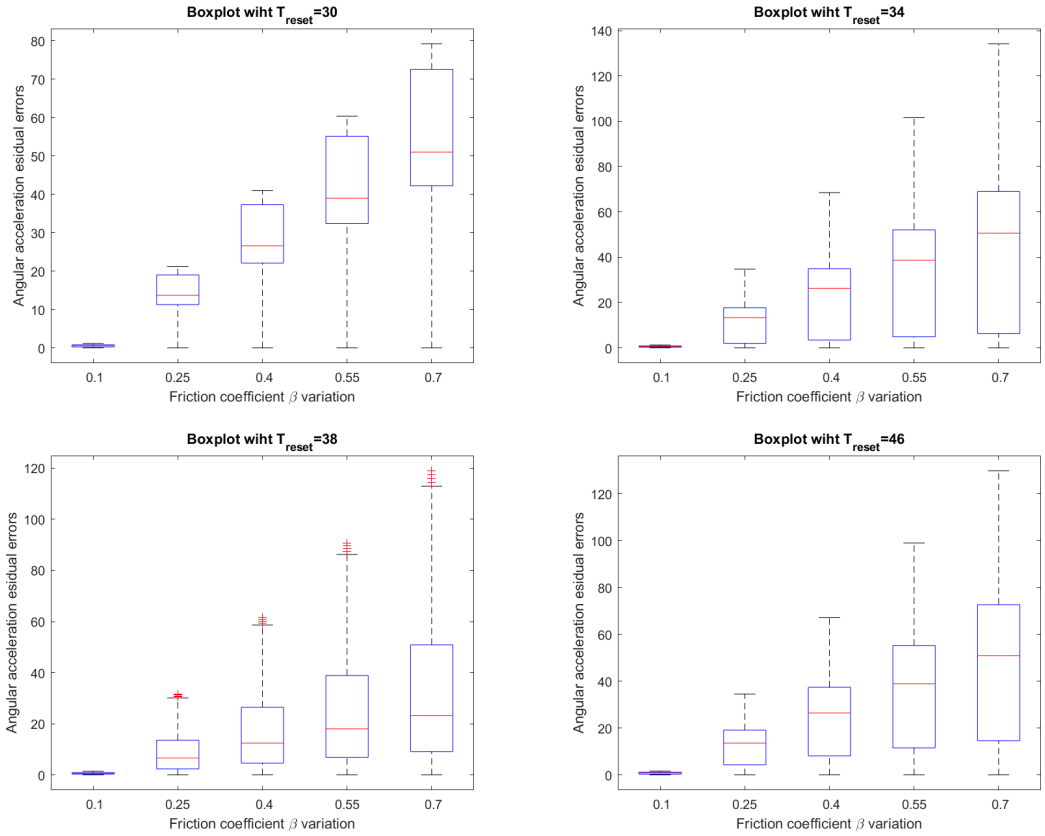


Figure 4.29. T reset choice

The results highlight that choosing a reset time of 30 seconds is the best choice for this kind of framework, because it corresponds to a processing period. In practice, since it is not possible to exactly know how long a processing period takes it is better to choose a reset time large enough to capture the dynamic variations of the system under study.
In Figure 4.30 it is possible to 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.

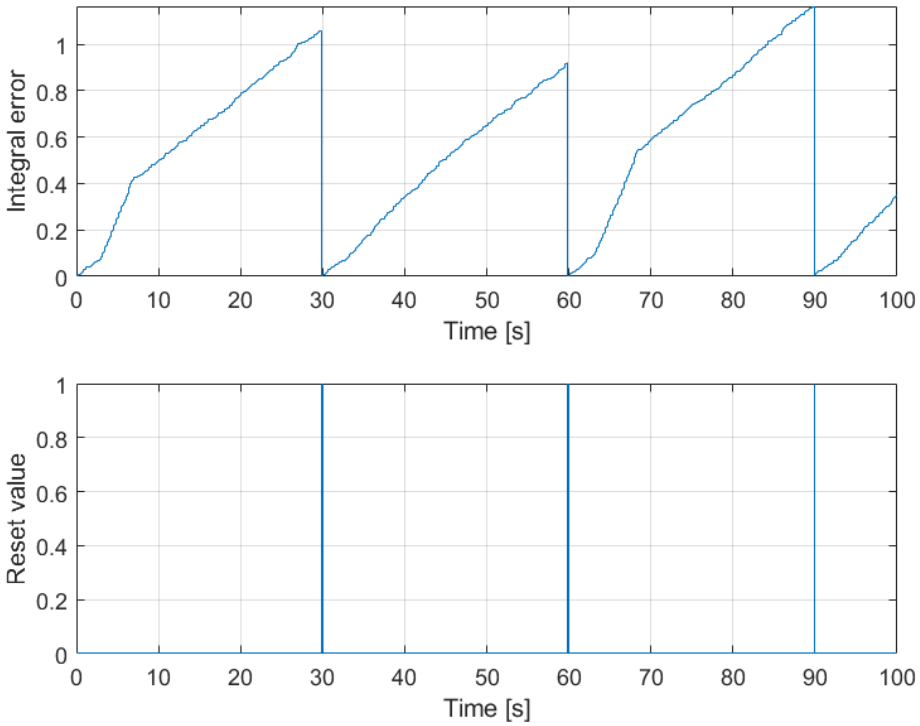


Figure 4.30. Integral error in nominal condition with Reset time of 30s.

4.4 Multi-model: Algorithm structure

With all the considerations made so far, it is possible to implement the final estimation algorithm. 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.

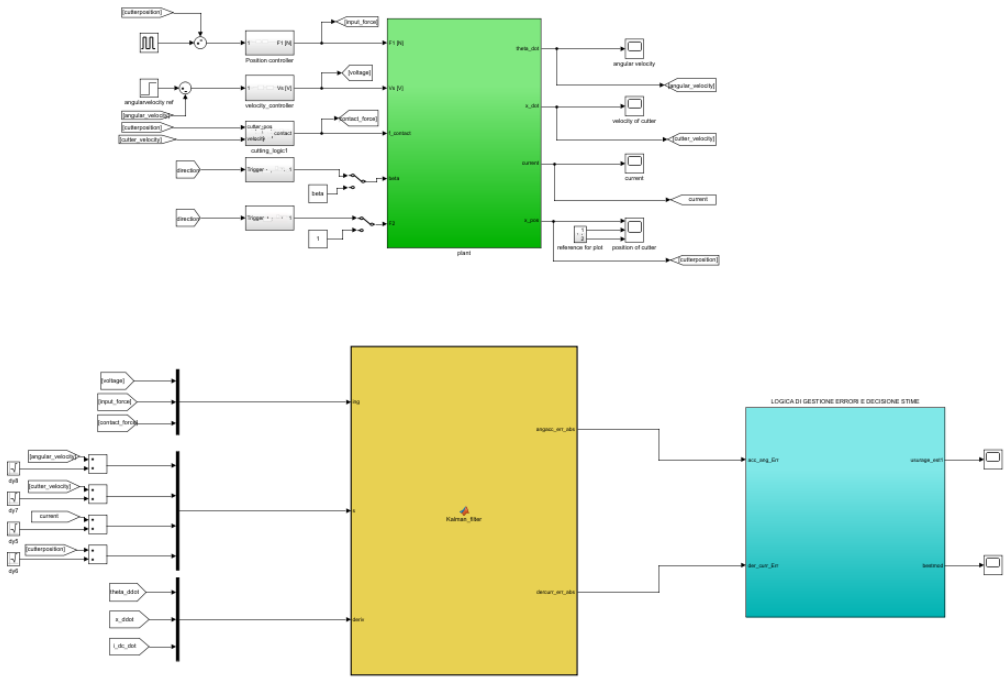


Figure 4.31. Simulink implementation of the algorithm.

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 than forwarded to the light blue block which represents the logic of error management, whose internal structure is represented in Figure 4.32.

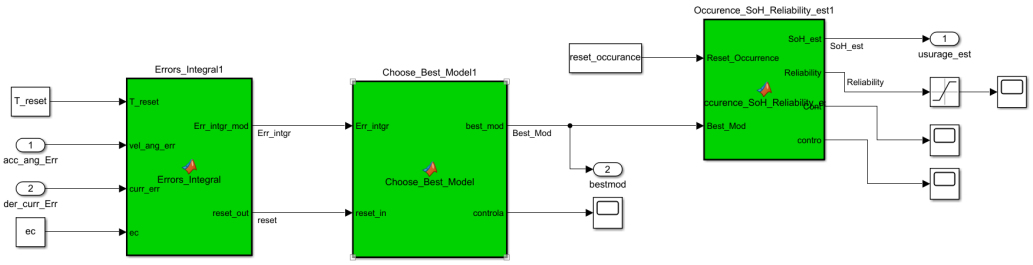


Figure 4.32. Simulink implementation of error logic.

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

4.4.1 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.

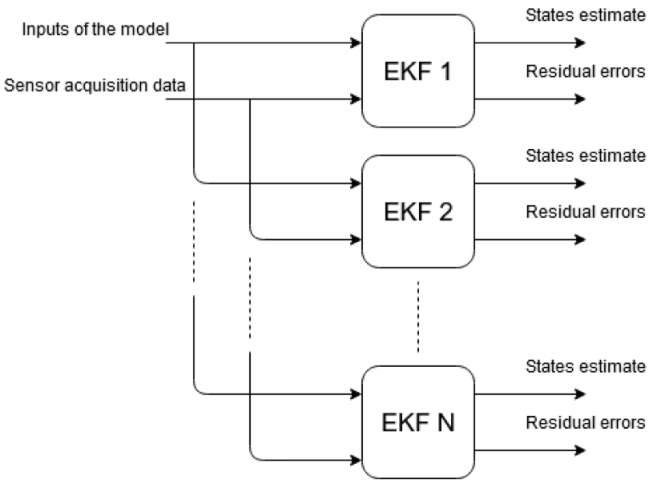


Figure 4.33. Switching estimator

The working principle of the switching estimator is:

- Each EKF will purpose its own 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 4.25. The aim is to identify the filter with the minimum residual errors, which means that filter which 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 4.25. Friction coefficient associated to each filter

4.4.2 Best model choice

A key part of the algorithm is dealing with residual errors and extrapolate useful information from the, 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 4.34 shows that with the condition listed in Table 4.26 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 4.26. Test with nominal friction coefficient

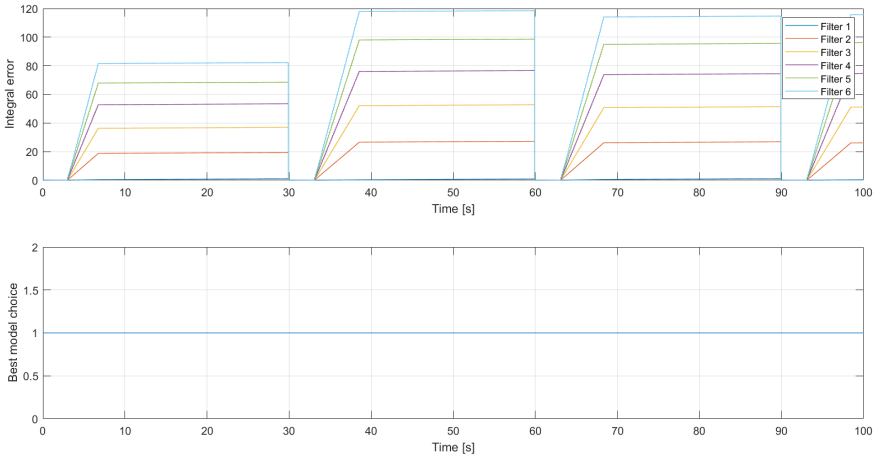


Figure 4.34. Best model choice with nominal condition

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 4.35.

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

Table 4.27. Test with friction coefficient variation

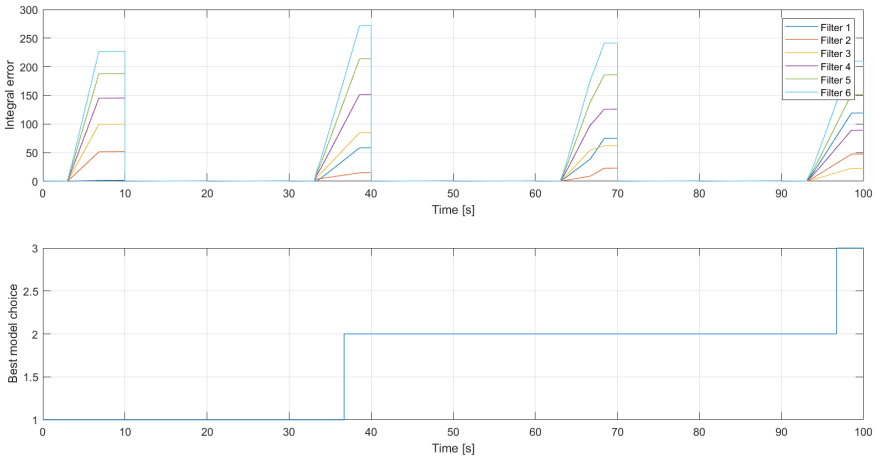


Figure 4.35. Best model choice with β variation

5 Reliability

Parameter estimation can be subject to several disturbances, such as measurement noise and propagation errors, but the correctness can be affected also by a simple mismatch between the filters (and how they are built) and the real model. Therefore, it is necessary to introduce a parameter that tries to catch this difference and indicate the correctness of the predictions. From now on, this index will be called Reliability and is expressed in percentage.

5.1 Observability concept

To understand the parameter identification and the way the reliability is intended in this thesis, some important concepts about the observability property must be introduced. Rudolf E. Kálmán introduced for the first time the concept of observability for linear systems in 1960, the aim was to find a measure of capability to deduce internal states of a system from knowledge of its external outputs. On the other side, immeasurable process states are reconstructed from measurements of other process variables and a process model. In control theory, the observability plays a fundamental role, e.g., in the model predictive control, since not all the states are directly monitored and it must be find a way to estimate them. The success of state estimation depends on three fundamental factors:

- Choice and quality of the measured variables.
- The accuracy of the model.
- Observability of the system.

The first factor is a modelling task in which the focus should be the choice of the right set of variable to consider as states and to provide an efficient way to measure the variables as much accurately as possible. Secondly, the accuracy of the model depends on the physical knowledge of the system and on the level of detail in which the system is described. Finally, observability is a system property that determines if the states of the process can be uniquely determined at any time instant from measurements of the system inputs and outputs over a finite time period. Supposing to have no model-plant mismatch, determining the state at any time is equivalent to understand if the initial state is unique. If all the system state can be estimated, the system is said to be completely observable, otherwise it is called "unobservable". Nevertheless, one possibility is that only some of the states are partially observable. For this particular case, providing that the system is detectable, it is possible to achieve a good state estimation. A detectable system is a special unobservable case when the state modes have negative real parts. In this case, the estimated state converges to the true value asymptotically. On the other hand, an observable system state estimation always converge to the true state over a finite time period. Mathematically, the strict definition of observability is:

The system $\mathcal{M}(x)$ is observable if, and only if, for any $x_0, x_o \in \mathcal{U}_x$

$$\exists \boldsymbol{u} \in U_u : \quad \boldsymbol{y}\left(t, \boldsymbol{u}(t), \boldsymbol{x}_0\right) = \boldsymbol{y}\left(t, \boldsymbol{u}(t), \boldsymbol{x}_0'\right) \quad \forall t \in [0, t_f] \Rightarrow \boldsymbol{x}_0' = \boldsymbol{x}_0 \tag{5.1}$$

According to this definition, the system is observable if, for any possible evolution of state and control vectors, the current state can be estimated using only the information from outputs . Nevertheless, there is a weaker definition: A system is locally weakly observable if the initial states determined from measurements are unique in its neighborhood. Observability and local weak observability are equivalent for linear systems (Hermann and Krener, 1977). An advantage of local weak observability is that it can be verified by algebraic tests e.g., using differential algebra and Lie derivatives [21].
As it was shown until now, the observability is a global concept that requires a very rigorous analysis, that become even more difficult when the system to analyze is non-linear. However, Singh and Hahn (2005) use empirical observability grammians to analyse observability of non-linear systems and Rumschinski et al. (2014) present the formulation of a finite-time output energy measure which they use in a computational method to analyze the observability of uncertain systems[20]. For the analysis of the physical model presented in this thesis, since the model has been developed only for simulation purposes and the objective of this thesis is not to demonstrate the observability properties, this analysis will be done only empirically, meaning that the observability properties are seen by looking at the behaviour of the outputs in the simulation. However, it is important to have in mind that the obsevability property of a state can have a crucial role in the estimation of the parameters.

5.2 Preliminary topics

In order to understand how to calculate the reliability, some preliminary concepts must be introduced. A first definition of Reliability is *the probability that a product, system, or service will perform its intended function adequately for a specified period of time, or will operate in a defined environment without failure.* In this study case, reliability refers to how consistently a method measures something. A fundamental completing an instrument meant to measure motivation should have approximately the same responses each time the test is completed. Although it is not possible to give a precise calculation of reliability, an estimation of this index can be achieved through different measures.[22] In this thesis, reliability is expressed as the percentage of precision of the estimations. As it was mentioned before, is not possible to achieve a precise calculation of it because it depends from many factors. To reduce the number of factors that must be taken into account, it can be done a differentiation between the ones related to the physical model, and the ones due to the algorithm implementation. For this thesis, reliability calculation has been chosen to be related strictly related to the algorithm factors. In the next chapters, it will be also empirically established that this kind of calculation

based on the algorithm results, is also affected by the variation of physical model parameters with respect to the nominal ones. Finally, one important algorithm results that directly influence the reliability (beyond the residual error) is the result of the best choice model represented in an occurrence barplot, that is a very understandable plot that shows the number of times that a Model has been chosen as the best one normalized with the total number of choices.

5.3 Reliability related issues

Having a single number (or percentage) that clearly and immediately shows the reliability of the obtained results can lead to huge advantages. At first, it is possible to have a direct feedback of the results, moreover, this index allows to implement a closed loop that in some way can update the models to make the prediction always more accurate. On one side, in order to achieve a closed loop (5.1, it is necessary that the index is updated as fast as possible, on the other side, the system is designed to update the parameters and the range every 250 samples because it is mandatory to wait a certain time to observe the response of the system to the updated parameters. Thus, it must be found a good trade off for the setting of the update time and the reliability must be interpolated in order to predict the progress of the next updates.

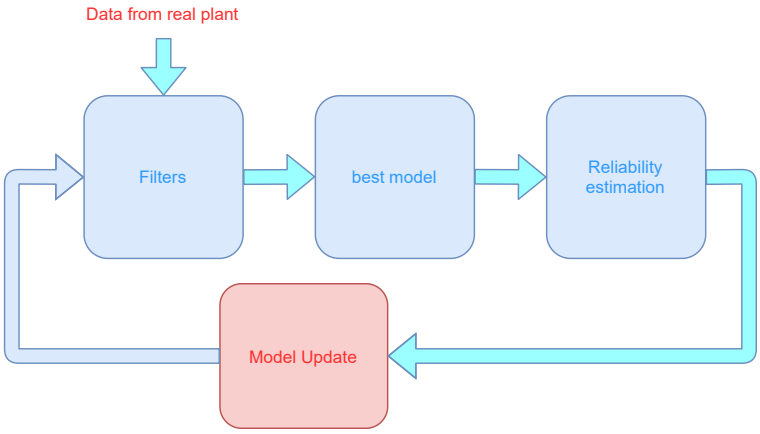


Figure 5.1. Block scheme of parameters update.

A more technical but relevant problem related to the computation of a trustworthy reliability index regards the countless configurations of predicted outputs.
In particular, the reliability index is mainly related to three important factor:

- Accuracy of the match between real plant and models: The most important factor is that the model must match (in terms of parameter modelling) the real plant behaviour as much as possible. This requirement is also the most difficult to check, because there are many cases where the prediction can

appear very good while the model is completely wrong. In this thesis, it is proposed a solution involving an analysis of the residual errors and the distance among them combined to some possible calculation looking at the best model occurrence plot.

- Update time: As it was mentioned before, a too small update time could lead to inaccuracy because the shortness of the observed period could be not enough to catch parameter update results. In opposition, a too large update time could lead to mistakes on the interpretation of the reliability behavior.
- Choice of reliability calculation method and shape of the "best model occurrence" barplots: This point is strictly related to the prediction algorithm, there is indeed a strong correlation between the shape of the best model occurrence plots and the type of algorithm that we want to choice for the predictions.

5.4 Computation methods

There are several ways that allows the computation of reliability, some of the most relevant and efficiency aimed are presented in this thesis. In particular, three methods are resulted to be the most suitable:

1. Gaussian mask: this method comes from the first approach of reliability computation of the BAT-MAN approach.It utilizes statistical pdfs such as the Gaussian distribution.
2. Regression tree: Application of simple machine learning methods to classificate the output of the algorithm.
3. Relative error: on-line computation of the relative errors among the residuals.

5.4.1 Gaussian mask

This method wants to extract the reliability percentage through the usage of a specific gaussian function built in order to fit the vector of best model occurrances. This vector has N elements as the number of filters and each element represent the percentage of occurrances. For example:

$$BMO_{vector} = \begin{pmatrix} 0 \\ 0 \\ 5 \\ 75 \\ 20 \end{pmatrix}$$

This vector means that there are 5 filters, and every element of the vector is the relative percentage of occurrence. The method wants to check the shape of this vector which must be similar to a gaussian curve, or it must be contained inside it. The gaussian must be built properly choosing the σ which represents the width

of the curve, the μ which is the center of the function that matches the most high occurrence. Then, this function can be compared with the occurrence vectors by using many techniques, the simplest one is to make a subtraction point by point and pull out a Reliability percentage from the number of subtraction that results greater than 0.

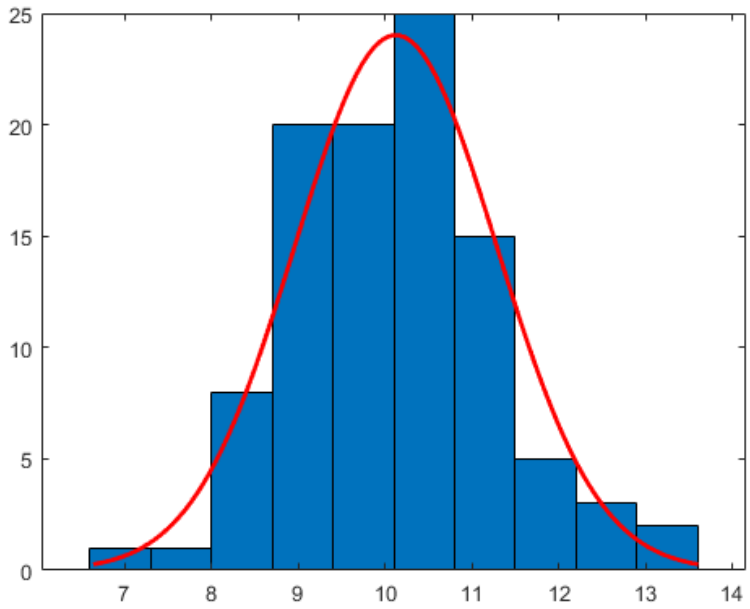


Figure 5.2. Normal distribution fitting example.

This method works with systems that works around a defined range of values. For example, in the estimation of the residual charge of a battery, where the voltage and the current works around prefixed values and cannot overcome this range. In this study case, the domain of the parameters of a mechanical system can assume very different values with respect to the nominal conditions. Therefore this method is not suitable for mechanical systems because it doesn't provide enough information about the plausibility of the parameters. For instance, if we consider a friction parameter beta which in the real model is way out the range where the filters are built (considering the filters as ideally coherent with the model except from the beta parameter) the best choice of the model would be the filter with the nearest beta, and the reliability would be high, but there is no information about how far we are from the real model, so the reliability calculation could be totally wrong.

5.4.2 Regression tree

Another method is provided by simple machine learning algorithms such as linear regression and classification. The idea behind is very simple: Train a regression

tree with an hand-made database which consider most of the possibilities that the occurrence of the model can achieve and then use this tree to evaluate the reliability. In this way it is implemented a simple supervised machine learning algorithm. The database must include all the most common configuration of occurrences. Matlab also provides a learnerForCoder which trains the tree in order to be compatible with the code generation process, which is fundamental for the edge-computing requirements of the whole project. The function that matlab provides is fitrtree(X,Y) that returns a regression tree based on the input variables X and the output Y. The returned tree is a binary tree where each branching node is split based on the values of a column of X. An example of the database built can be:

$$X = \begin{pmatrix} 12.5 & 12.5 & 12.5 & 12.5 & 12.5 & 12.5 & 12.5 & 12.5 \\ 50.0 & 0 & 0 & 0 & 0 & 0 & 0 & 50.0 \\ 25.0 & 25.0 & 0 & 0 & 0 & 0 & 25.0 & 25.0 \\ 0 & 50.0 & 0 & 0 & & 0 & 0 & 50.0 \\ 100.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100.0 \end{pmatrix}$$

And the relative output reliability:

$$Y = \begin{pmatrix} 0 \\ 5 \\ 3 \\ 15 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{pmatrix}$$

The database has been extended with more in order to include more possibilities, and can always be modified in order to add particular situations that must be evaluated differently.

5.4.3 Relative error

This method has been introduced in order to compensate the lack of functionality introduced in the Reliability computation of BAT-MAN project. In the computation of the reliability of the multimodel applied to BMS(battery management systems) there is no need to look outside the working range of voltages of the batteries, so the possibility of the prediction to go totally outside the range it was not considered. Now, since the beta can go outside the set range of action of the filters, this possibility must be considered. The idea behind the calculation of the Relative error is to find a way to validate the range of the models looking at the residual error of each model If the residual errors are too similar among them it would means that the models are too similar between them or otherwise that the distance between the models and the filters is too much. That would result in two possibilities: The Noise is too high and must be reduced (if possible) or the filter parameters must be updated in order to match the Real model as much as possible. An ideal set-up is shown in the following figure:

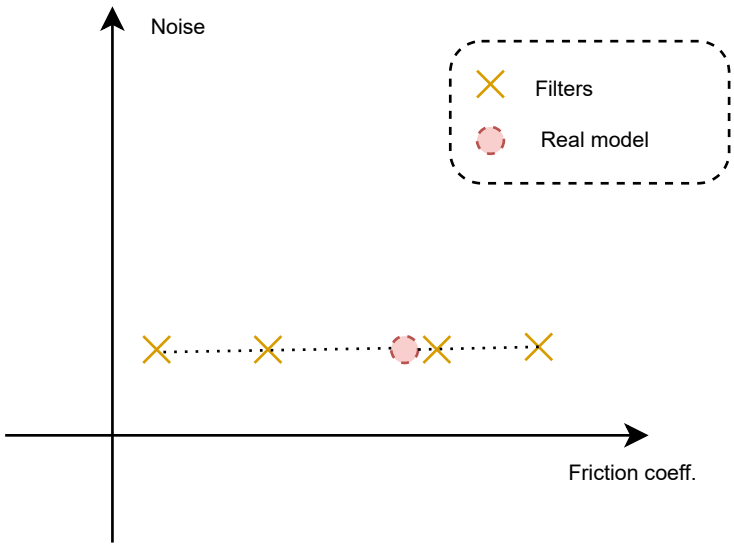


Figure 5.3. Noise-friction plane models representation: ideal situation

When the noise is too high, the distances are very similar:

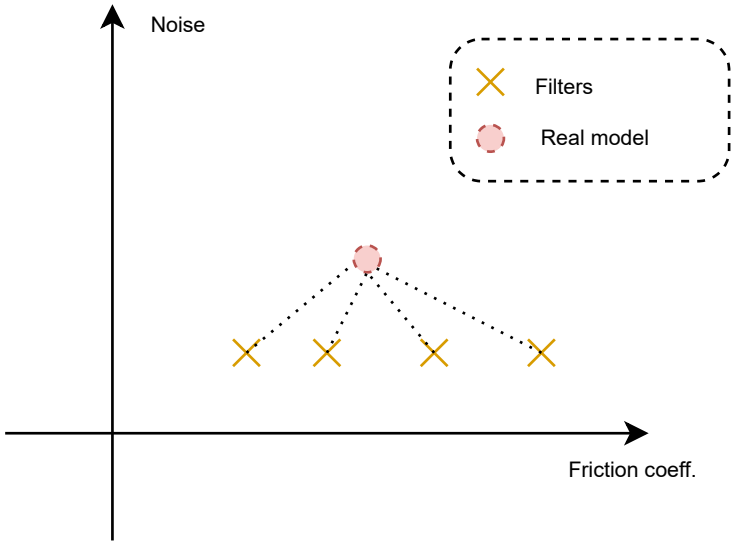


Figure 5.4. Noise-friction plane models representation: higher noise

Finally, when the Filters are too far from the physical model and they need to be updated the situation appears as in the figure below. Also in this case the distances are very high but similar between them.

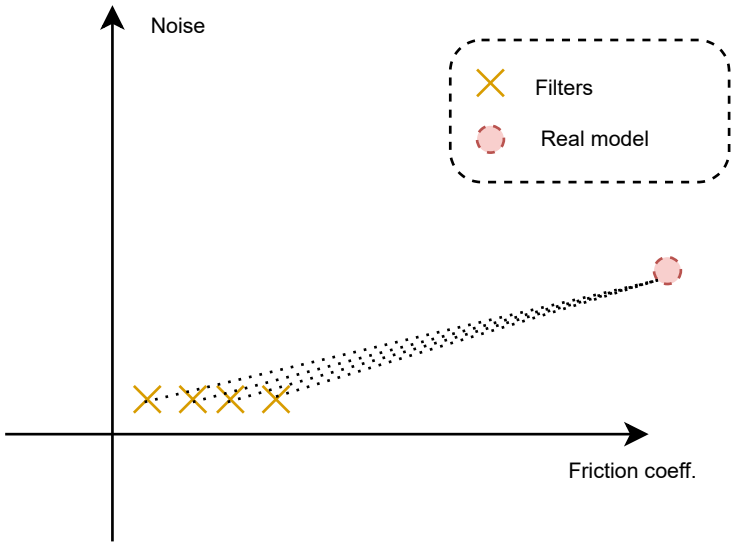


Figure 5.5. Noise-friction plane models representation: high noise and bad filter calibration

Thus, the relative error is intended to be a parameter which goes from 0 to 1

depending on how much the errors are similar among them and it is defined as:

$$err_{rel} = \frac{|\max(\text{ error }_{\text{vec}}) - \min(\text{ error }_{\text{vec}})|}{|\max(\text{ error }_{\text{vec}})|} \tag{5.2}$$

which is implemented in simulink as in the figure below:

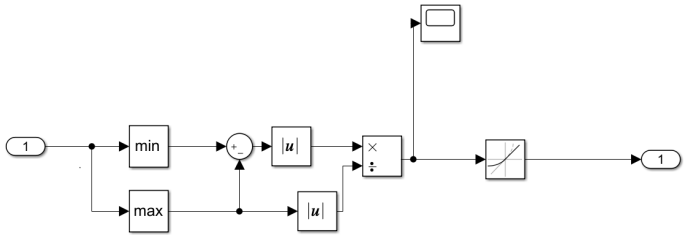


Figure 5.6. Simulink implementation of relative error computation.

5.5 Method comparison

At this stage, the methods must be compared in order to understand which is the most suitable. In this phase, is fundamental to try to understand which are the situations and the condition which could lead the algorithm to fail, thus, some "stress" tests are set and the most relevant results are shown in the next paragraphs.

5.5.1 Gaussian mask

In this test, there are 8 filters with beta equally spaced from 0.1 to 1, the first plot is a simulation with real β equal to 0.46. The vector of occurrances is:

$$BMO_{vector} = \begin{pmatrix} 0.2522 \\ 0.0111 \\ 19.7469 \\ 79.9898 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{5.3}$$

the barplot of the BMO vector and the gaussian are graphically shown in the next figure.

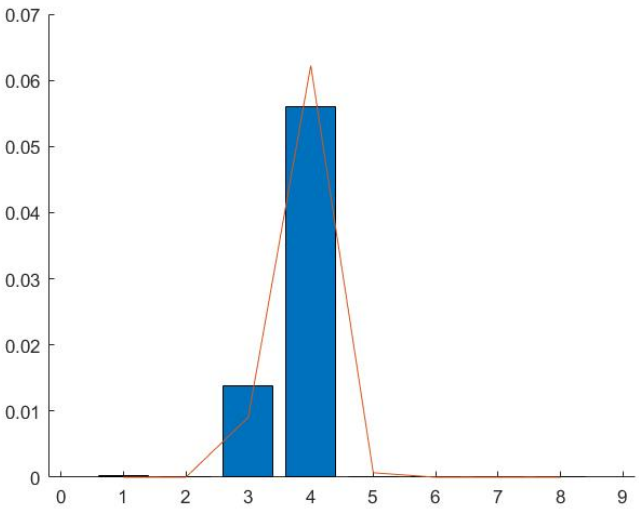


Figure 5.7. Bar plot of the best model occurrences with Gaussian mask fitting.

The obtained reliability is 95.0354% which is a good estimation considering the shape of the barplot. As it was said previously, this method is not suitable in two cases: The first case, is when the occurrences are equally allocated. To test this case, it is possible to simulate a β input that changes in time from the first filter to the last one as shown in the following figure:

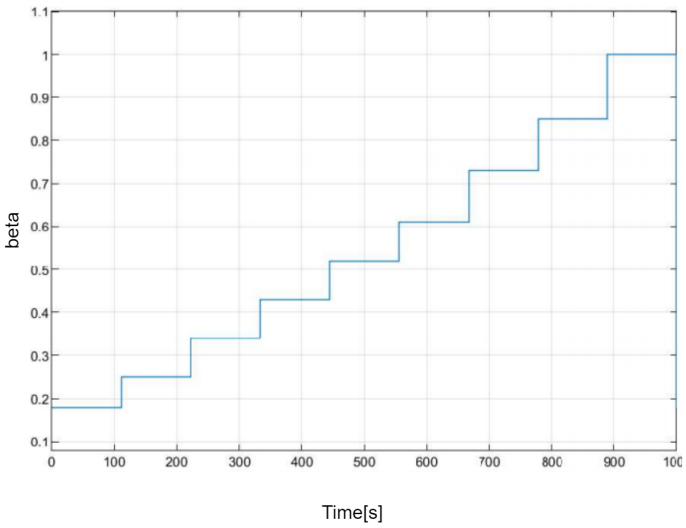


Figure 5.8. Fake stair input for simulation.

The relative barplot and the gaussian distribution appears as in the figure below:

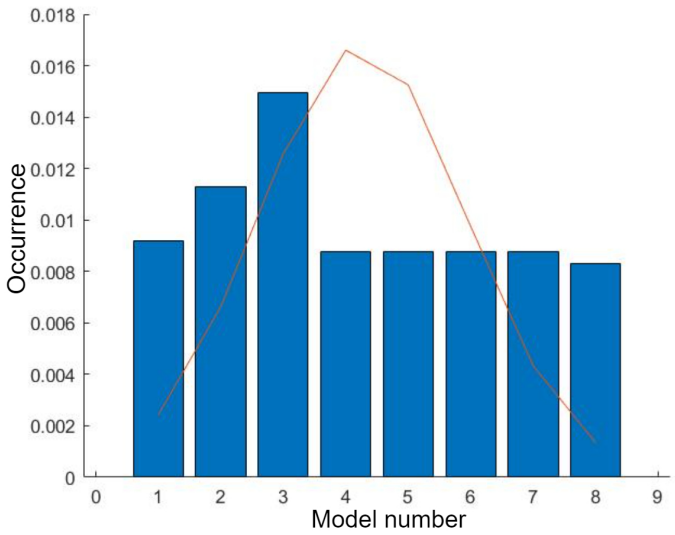


Figure 5.9. Barplot of best model occurance output.

And the Reliability obtained is 74.8158% which is too high for this barplot shape. What can be done to improve this estimation is to try modifying the shape of the gaussian curve, but there is not a unique shape that works in every condition, therefore, this method has limited possibilities of application. Moreover, this tests shows how the reliability, instead of going to 0, remains around 50% when real beta goes outside the working range of the filters. This happens because there is no way that this method can understand that the real beta is actually outside the range of the filters. The red dotted line shows delimits the working area of the filters and the blue line is the reliability tracking:

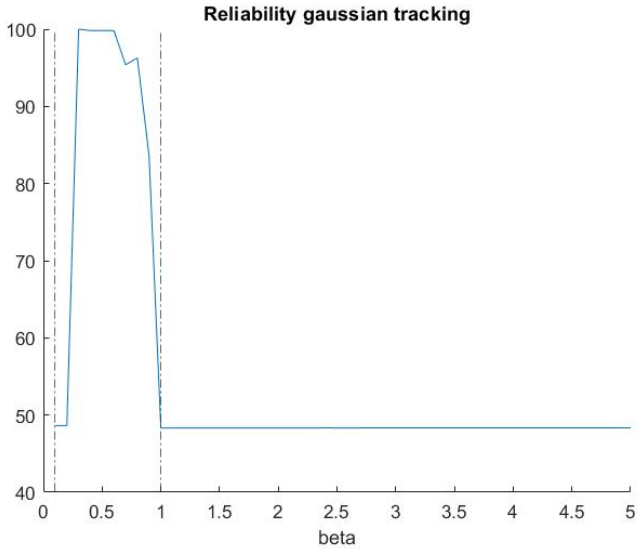


Figure 5.10. Reliability tracking with Gaussian method.

5.5.2 Regression Tree

The regression tree method, has no particular faults detection when beta is inside the range, but has the same defects of the gaussian mask method when beta goes outside the range:

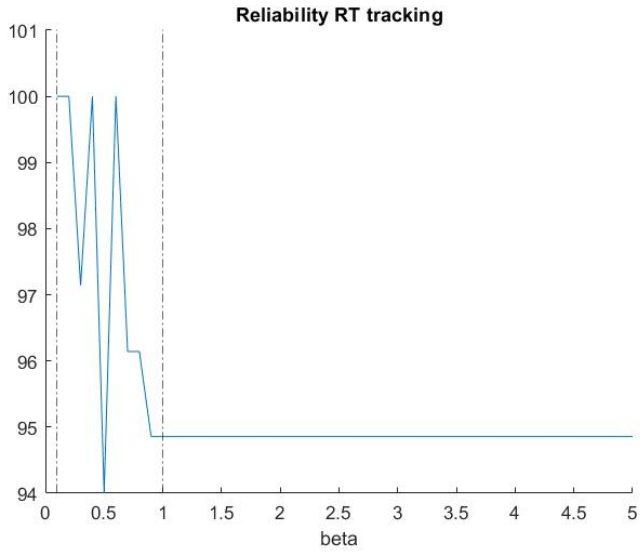


Figure 5.11. Reliability tracking with Regression tree method

As it was expected, the reliability outside the range remains wrongly high, this happens because the best model choice is always the closest to the highest beta filter, and this is translated with a 100% occurrence of the last filter.

5.5.3 Relative error

Finally, with the introduction of the relative error in the calculation of the reliability as in the equation (1), it is established how the reliability decreases as long as beta goes outside the range of the filters.

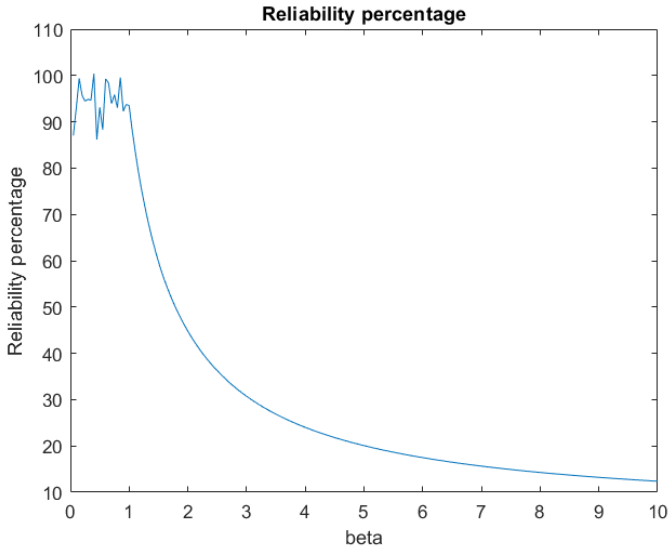


Figure 5.12. Reliability tracking with combination of regression tree and Relative error method.

This result is fundamental for the development of the thesis work, because it allows to understand if the set of beta is correctly set. From now on, the reliability index will be used not only to understand the quality of the predictions, but also as a **feedback** used to **update** the filter's beta. With "Range update" it is meant to change the value of the beta in each filter equally spaced from the first value of the range to the last one. Obviously, if the parameter that we want to estimate is not beta, or is the combination of two or more parameters, the "range update" expression will refer to the change of each of those parameters singularly. It is important to recall that beta is considered as the representative parameter of the SoH in this thesis hyphotesis, nevertheless, it is always possible to consider other parameters or a combination of them.

5.5.4 Final reliability computation

The reliability parameter, for simplicity reasons, is a singular variable expressed in percentage that combine two method mentioned in the paragraphs 5.4.2 and 5.4.3 and it is computed as follows:

$$Rel = (RT_{prediction} - (1 - err_{rel}^2) * 100) \tag{5.4}$$

Where $\mathrm{err_{rel}}$ is the relative error and $\mathrm{RT_{prediction}}$ is the reliability percentage that results from the Regression tree. The final formula 5.4 is obtained from the idea of checking the correctness of the algorithm choice trough the regression tree, while checking also the correctness of the residuals trough the relative error.
As it is explained in the paragraph 5.5.3, the relative error plays an important role in this formula, since it allows to understand how much the residual errors

are relatively far from each other. In other words, the relative error is directly proportional to the Reliability. The more the relative error decreases, the more the prediction are considered to be unreliable. This concept could go against intuition, because it can be very natural to look for a small residual error and a small difference between them, but in this case, the relative error represents the relative position in the 2-D plane of the real model with respect to each filter. Therefore, the ideal condition to achieve would be that the filters had residual error equal to 0, while the other filters had high residual errors, because this would mean that the one of the filters perfectly match up the reality. In this ideal case, the relative error would be 0, and the reliability would be consequently 100%.

6 Robustness and parameter update

Once the index for the reliability is established, the goal becomes to update the filters parameters, starting from the friction coefficient, to make the EKF bank match the real system. To do this, many methods based on machine learning theory and statisical consideration are suitable. One of the simplest and most utilized is the gradient descent algorithm.

6.1 Gradient descent based optimization algorithm

Supposing that the models are represented in a N-dimensional plane, where N is the number of parameters that are supposed to be updated, the update is completely "blind" as long as the direction and the module of the update-vector is not regardless known. Furthermore, A relative N-dimensional Function can be defined as the Function to be optimized:

$$Rel\_function = f(\beta_{range}, R, kt...) \tag{6.1}$$

The figure below is a 2-D simplified representation of the problem, where the Reliability function only depends on the range of action of the model.

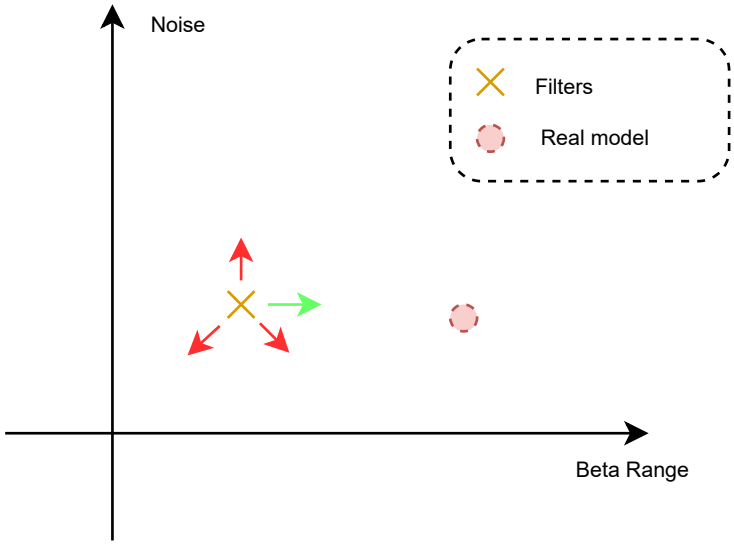


Figure 6.1. Noise-Range representation of filters and models.

In this case, considering the optimization problem that it is described, the Gradient descent algorithm is very suitable. Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. This algorithm can be the base for the Reliability optimization algorithm, with the difference that the aim is to maximize the reliability instead of minimizing a function.

6.1.1 Statement of the problem

Let f : $\mathbb{R}^{\mathbb{N}} \longrightarrow \mathbb{R}$ be the objective function one intends to minimize. Given an initial point $x^{(0)} \in \mathbb{R}^{\mathbb{N}}$, one iteratively updates this point using the steepest descent of the objective function at the current point,

$$x^{(t+1)} = x^{(t)} - \eta H^{-1} \nabla f(x^{(t)}) \tag{6.2}$$

where $\eta > 0$ is a hyperparameter (called the learning rate in a machine learning context)which may in general be step-dependent. The iterative update therefore includes H of the objective function, which is the hessian matrix of the function.

6.2 Range update

The algorithm defined in the previous paragraphs can be implemented, for example, to find the best range in which the filters should change. The range of beta is intended to be the scale in which the filters change. Supposing that the

first filter is set to the nominal value of beta, which is the "new" condition of the machine, the difference between the friction coefficient of the last filter and the nominal friction is the range that must be updated. Supposing that the noise is very low, to have good reliability, the range must obviously include the friction coefficient of the nominal model. Regarding the implementation of the algorithm, since the response of the system is not immediate, it can be implemented a discrete version, where every T_u of time the Reliability is checked out and if it is below a predetermined threshold a counter is increased, otherwise is set to 0. Once the counter reaches an established value, which means that the Reliability stays below the threshold for enough time, there is a direction of update tentative. The update direction is chosen following the Gradient descent criteria, as well as H (see equation 6.2) which is the hessian matrix of the function in the original gradient descent algorithm. In this case, H is intended to be the step of the update, and it is chosen proportional to the distance of the Reliability from 100%, so that there will be stronger update when the reliability is very low.

6.2.1 Tests

The first test shows how the Range update is fundamental when the model works with a friction coefficient which is totally outside of the Real plant friction. In this simulation, the real plant friction is set to $\beta = 10$ and the initial range is 1, while the nominal friction is $\beta = 0.1$, consequently, the last filter has a friction coefficient of $\beta = 1.1$;

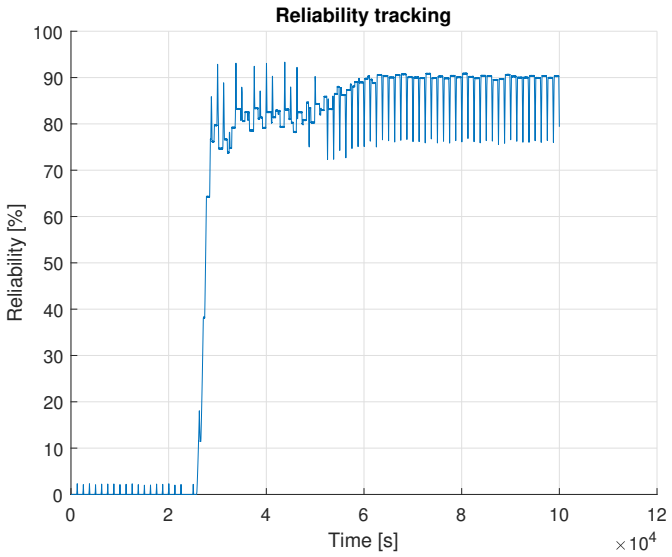


Figure 6.2. Reliability tracking with range update and $\beta = 15$

As we the figure above shows, the Reliability increases after the updates until it is stabilized to 90-95%

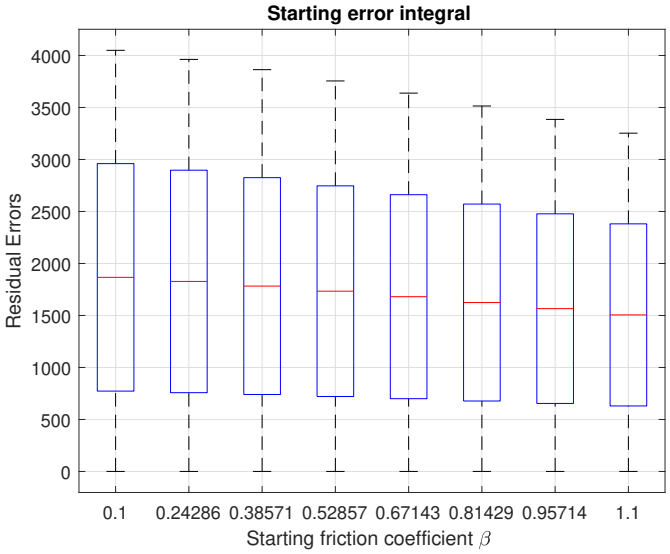


Figure 6.3. Residual error boxplot of the first 800 s of simulation with $\beta = 15$.

The boxplot above, shows how initially the integral errors of the filters are almost equal.(it is the boxplot of the error in the first 800 s of simulation). After the update, in the last 800 s of simulation, the boxplot of the integral error appears as follows:

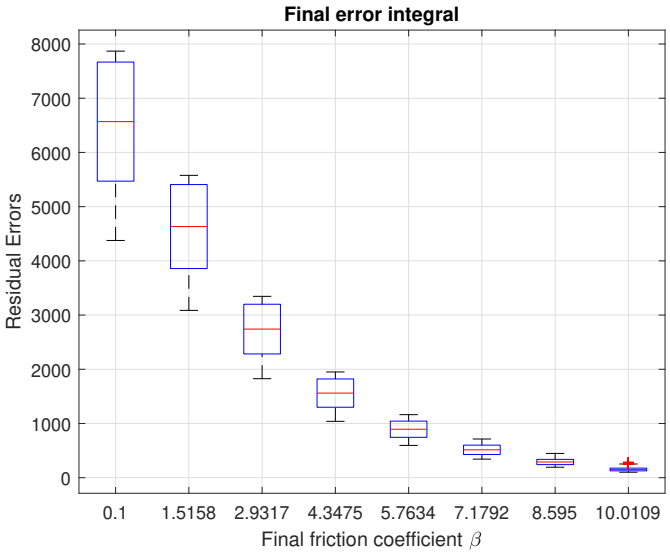


Figure 6.4. Residual error boxplot of the first 800 s of simulation with $\beta = 15$.

Finally, the next plot shows how the range is updated until the real friction coefficient is included in the range.

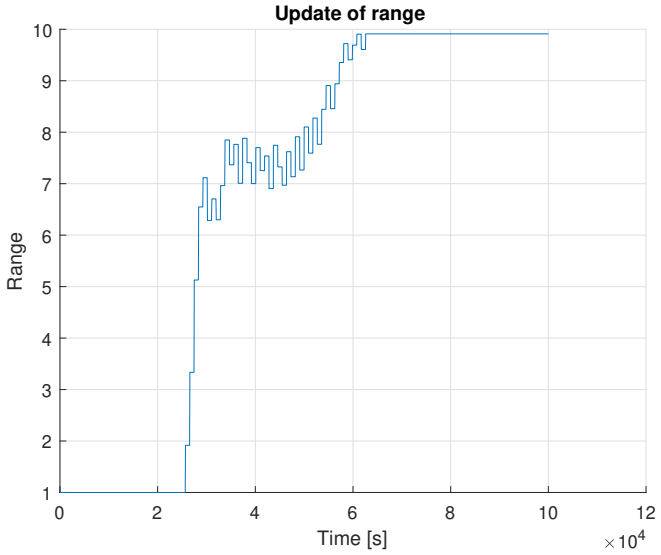


Figure 6.5. Range update tracking with $\beta = 15$.

When the real plant friction coefficient is already included in the initial range, there are two possibilities: if the initial reliability is higher than the predetermined threshold, there will be an update, otherwise the range will remain fixed. For example, with beta=0.6 we have an high initial reliability, but not enough for the threshold, which in this case is set to 90%. Initially, the reliability is around 80%, so there are little updates until the reliability reaches at leas 90. The next plots shows, the reliability tracking and the boxplots before and after the updates. Reliability:

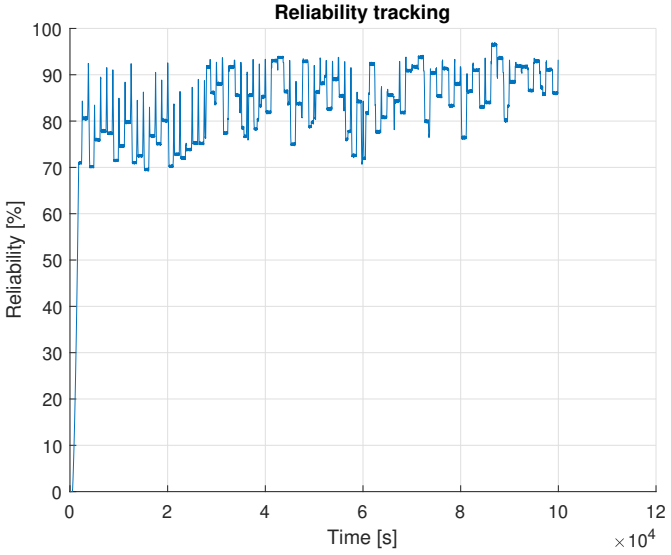


Figure 6.6. Reliability tracking with range update and $\beta = 0.6$.

Initial integral error:

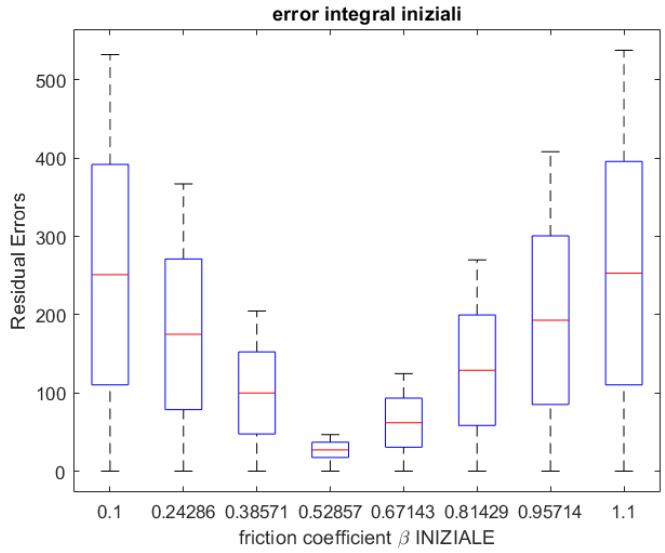


Figure 6.7. Residual error boxplot of the initial 800 s of simulation with $\beta = 0.6$.

after the updates:

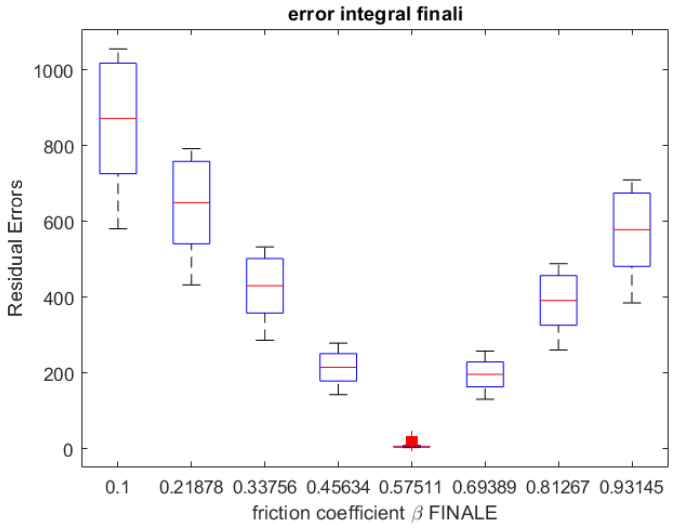


Figure 6.8. Residual error boxplot of the final 800 s of simulation(after the update) with $\beta = 0.6$.

6.3 Range Translation

Another possibility regarding the update of the filters is the translation of the range of action of the filters, which can be combined with the range update previously described. One the first hand, this technique has the advantage of keeping the range of action smaller, so that the estimation of beta and the SOH is more accurate, but translating the the range of action of the filters also means losing a direct relation with the nominal filter, which can be good for the estimation of the beta, but not necessarily useful for the machine SOH estimation. The algorithm is implemented with the Gradient descent basic ideas, and it is very similar to the previous one, but in this case there is also the update translation of the first filter beta until the range is smaller than a threshold and the reliability is high. subsectionTests For instance, in this case real plant beta is fixed to 3.1, it is imposed that the range of action must be smaller than 2, the simulation time is 40000 and the reliability threshold is 90%. Initial box-plot of the errors:

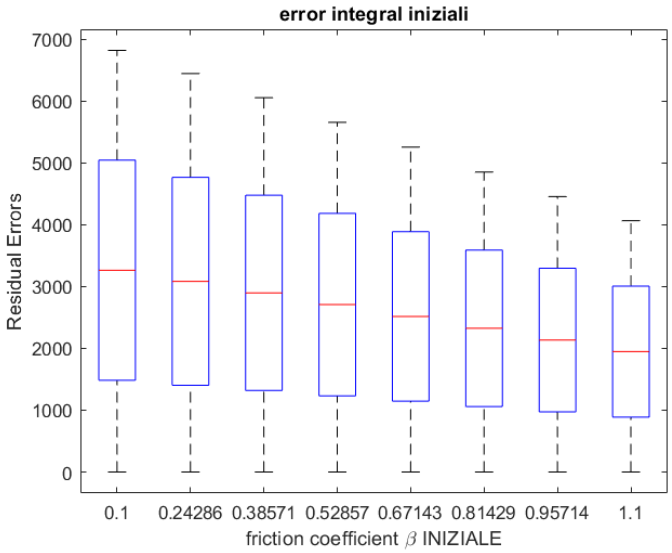


Figure 6.9. Residual error boxplot of the initial 800 s of simulation with $\beta = 3.1$.

Final box-plot of the errors:

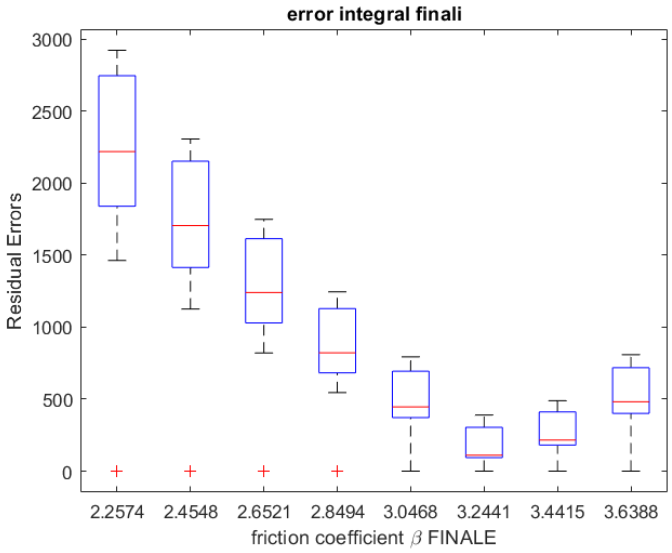


Figure 6.10. Residual error boxplot of the final 800 s of simulation(after the update) with $\beta = 3.1$.

Reliability:

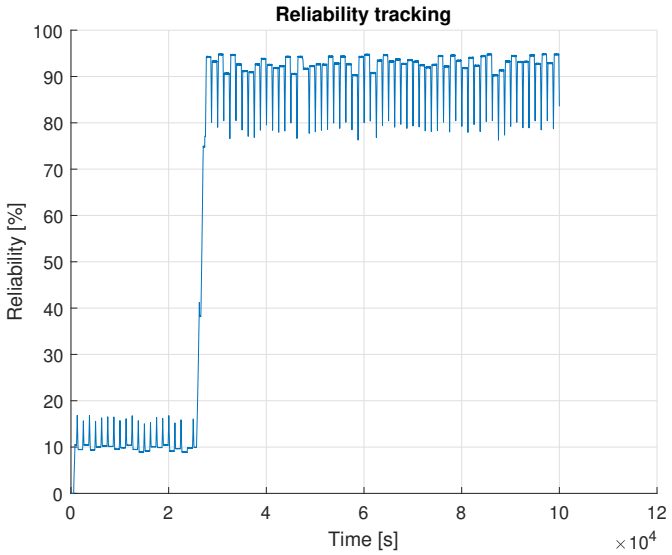
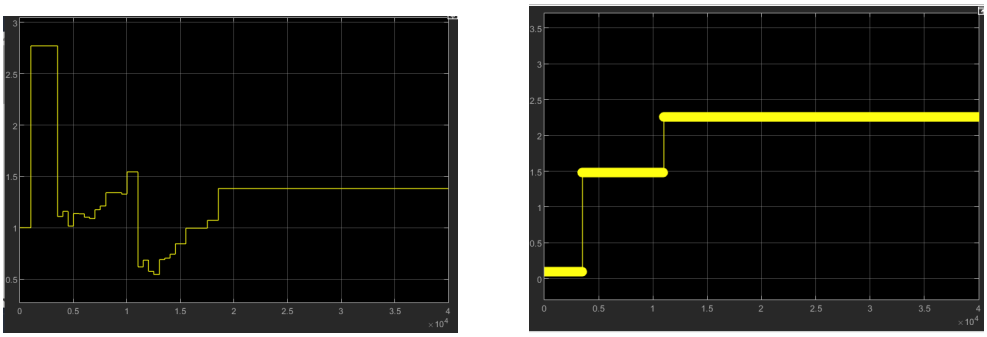


Figure 6.11. Reliability tracking with range update and $\beta = 3.1$. x-axys: time[s]; y-axys:Reliability percentage[%].



(a) Range tracking **(b)** First beta tracking

Figure 6.12. Beta first filter and range update with $\beta = 3.1$. x-axys: time[s].

This last figure represents the tracking of the range update in the panel (a) and the tracking of the initial value of the range of the panel (b). This method has not been investigated further because of the loss of direct relation with the nominal friction. In fact, the first value of the range of the filters shall be set to the nominal conditions, therefore, changing the value of the first beta means to lose it for the computation of SoH. Although is always possible to keep the value of the nominal friction and compare it with the estimation, the goal is to provide the SoH and not to estimate the friction coefficient and even if the results obtained with this method are good, it is way more valuable to keep a direct reference to the nominal friction to provide real-time computation of SoH.

6.4 Robustness

System robustness is defined as a system's ability to remain functioning under disturbances. This implies that information is needed on how the system responds to different degrees of disturbance. All the simulation made so far, didn't consider any noise except a small measurement one as input of the filters states, but it was considered that the filters perfectly matched the real model apart from the beta friction coefficient. The absence of noise is a very strong supposition and it doesn't match the real conditions, therefore, an uncertainty must be considered to make the simulation more realistic and to test the robustness.

6.5 Parameters uncertainty

There are many disturbances that can be considered, but for simplicity reasons it is possible to introduce an uncertainty that can change the filters parameters of a fixed percentage. Considering an uncertainty percentage fixed to 20%, for example, this uncertainty can change the value of the kt (constant of the electric motor) of plus or minus 20% of the nominal value in the real plant. A robustness analysis of the system can be performed by imposing some uncertainties and look at the system response to them. For example, what happens at the reliability when

the uncertainty grows up? The next figure shows how the reliability decreases as the uncertainty percentage grows (in the y axis there is the reliability while the x axis is the uncertainty).

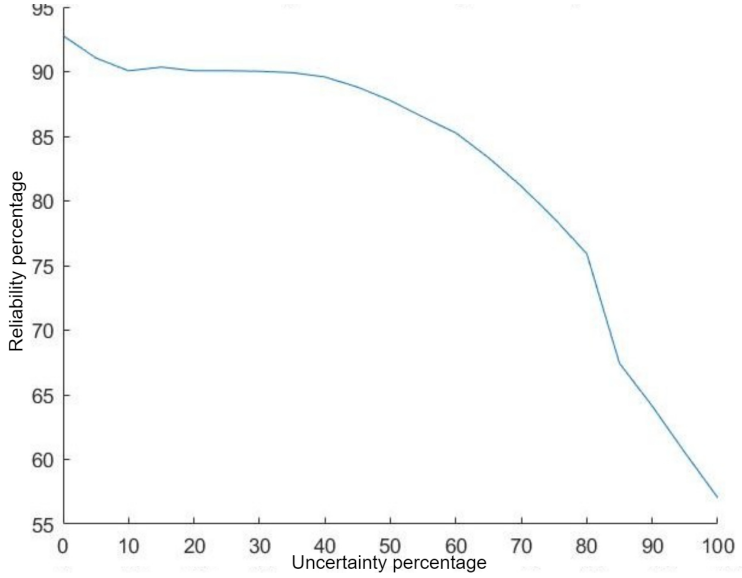


Figure 6.13. Reliability tracking with uncertainty percentage increase.

As it was expected, by increasing the percentage of uncertainty the reliability decreases. As well as the beta estimation become more inaccurate:

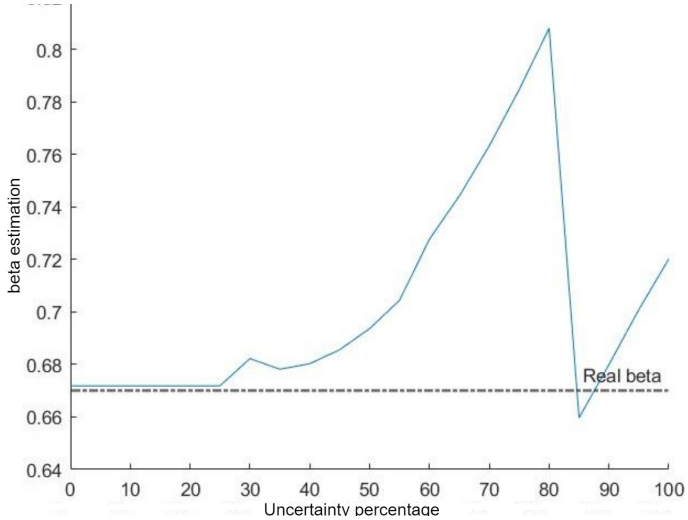


Figure 6.14. β estimation with uncertainty percentage increase.

6.5.1 Residual error behaviour

Finally, the last figure is the behaviour of the residual error average with the uncertainty increase:

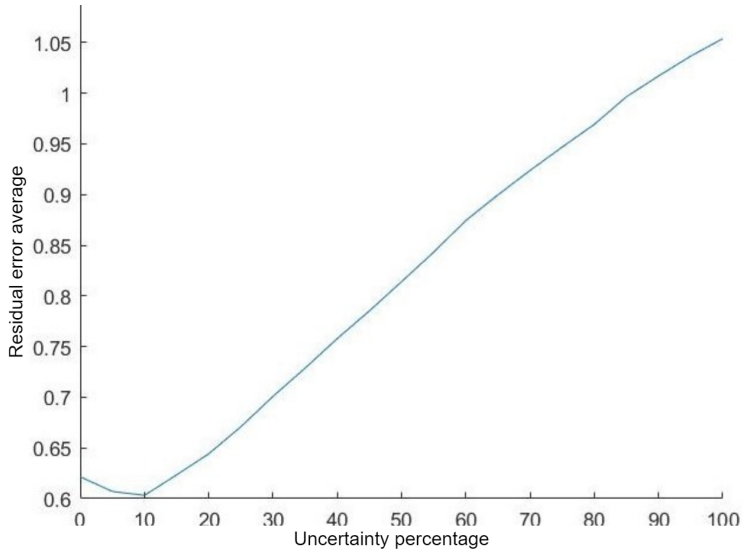


Figure 6.15. Residual error behaviour with uncertainty percentage increase.

The behaviour of the residual error is important to understand how the noise and the uncertainty affects the estimation error. The three panels reported above (6.13, 6.14,6.15) perform a robustness analysis to the uncertainty parameters. What it is established is that an uncertainty on the parameters lower than around 50% keep the estimation to acceptable performances, while adding disturbances of more than 50% could lead to big estimation errors.

6.5.2 Model parameters update

The improvement of the filter's model parameter range is fundamental to make the estimation of the SoH more accurate. The next step consists in the update of the model "fixed parameters", that is the parameters which should be not dependent from the change of the SoH of the machine. Since there is not always a direct correlation of the parameters with the residual error, and moreover it is not proven that all the parameters are observable, this operation is not simple. The first possibility is to attempt by changing only few parameters, one idea is to change the most influential parameters and see the system response. This parameters variation is strictly correlated with the error in the estimation and come from the analysis carried out by the team of the MorePRO project [27]. The idea is to try a gradient descent optimization algorithm one parameter at a time and see the effect on the estimation. Since there is no way to define the direction of the update, it is very difficult to implement a random algorithm that can lead to good optimization. However, this work can be simplified a lot if the working modes of the systems need to be chosen from a finite number of models. Therefore, the model parameter update has not been investigated further, since an analysis of the possibility to reduce the number of models to a finite number must be an input

of this study. Nevertheless, the MorePRO team also include as objective a deeper machine learning solution which could lead to this important result.This solution has been investigated by the Morepro team in parallel with the development of the work presented in this thesis. Thus, the model parameter update is still an available option to improve the estimation, but is considered a future step to work on.

7 Integration with CL friction model and final results

The final result of the MorePro project, wants to obtain a model that represents the physics in the best way. Therefore, the modelling team of the project developed a more accurate description of the end-effector tool contact with the workpiece. thus, the global formulation proposed is:

$$\beta = \left(3.32\left(\frac{DCn\pi}{1000}\right)^{-0.45} - 0.24(tnCL)\right) * \left[1 - \left(\frac{T - T_0}{T_m - T_0}\right)^{m_r}\right] \tag{7.1}$$

In this formulation 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. This model includes more detail about the physical environment such as the temperature T, the diameter of the cutting tool DC, the cutting speed $V_c = \frac{\pi DCn}{1000}$ and the fleed rate $fr = nTcl$.
This model can be simply included in this thesis work changing the filters equation including the equation 7.1. The new representative parameters for the SoH is the chip load CL that is defined as the theoretical length of material that is fed into each cutting edge as it moves through the work material. This modification to the model used until now has been implemented to the final model used for the tests and for the validation of the functional behavior of the system.
The final integrated model structure is shown in the figure 7.1:

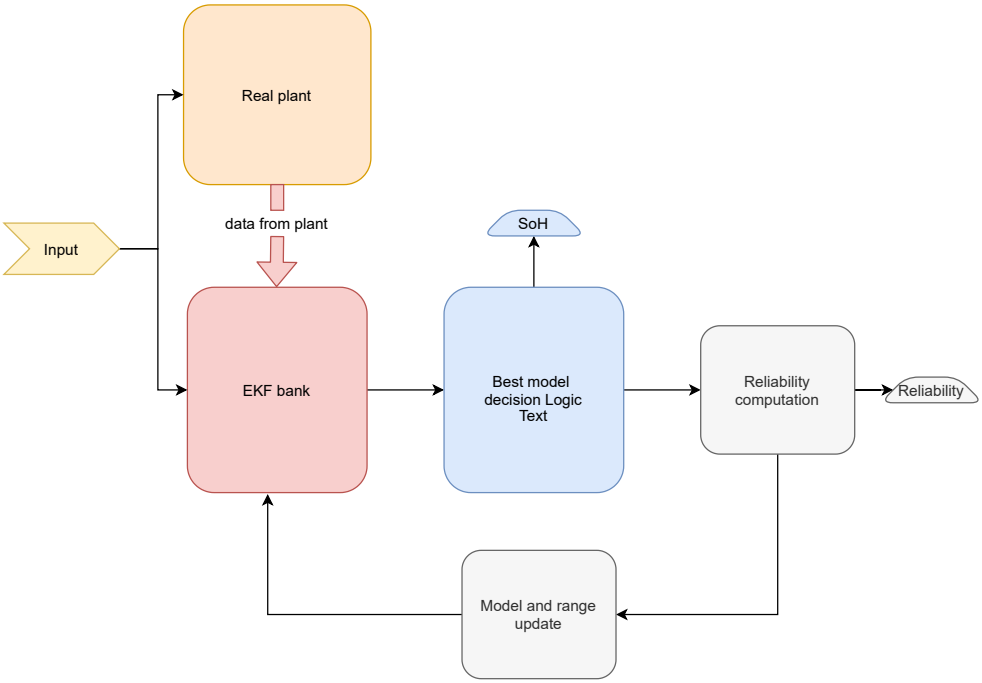


Figure 7.1. Block diagram of the final integration

The functioning of this block diagram is very similar to the one explained in the previous chapter. The inputs are the ones related to the plant:

- Reference of the cutter position.
- Velocity reference.

The ouputs of interest are:

- SoH.
- Reliability.

The blocks are:

- The light-orange block is the real plant, that represents the real working condition of the cutting process.
- The Red block is the EKF bank, that contains the Filters and is in charge to compute the integral residual errors for each filter. This block receives as input the data from the real plant and the input of the plant.
- The blue block is the Logic related to the best model decision. This block is needed to choose the best mode, keep track of the occurrences and to compute the SoH. The input to this block are the data from the EKF bank related to the residual errors.
- The light grey blocks are in charge of the computation of the reliability based on the best model occurence output received from the blue block and to compute the updates based on the calculated reliability

The main functional behaviour of the whole system used for the testing should be clear, but it is necessary to clarify also the most important parameters that needs to be set in each simulation:

- Simulation time: Duration of simulation expressed in seconds. It cannot be too high due limited computation power.
- Uncertainty: This parameter express the difference between the filter parameters and the real model parameters. The value is expressed in percentage and can be summed or subtracted from the real plant parameter value. For example uncertainty of 10% means that the filter parameter differ of plus or minus 10% with respect to the real plant equivalent parameter.
- Real CL: This is the value of the CL in the real plant that has to be estimated and represents the SoH.
- Number of filters.
- Inital filter range: Difference between the value of the last filter value of CL and the nominal CL. For example, if the number of filters is 8 and the range is 1, the 8 filters value of CL will be equally spaced from 0.1 to 1.1.

- Sample time:Simulation step that is set fixed to 0.1 in order to make the simulation computationally lighter
- Number of cycles: Number of cut repetitions. The frequency of a cut depends on the simulation time divided by the number of cycles.

Now, an important consideration regards the SoH calculation. SoH represent the wear condition of the system, nevertheless, the real variation of the CL in function of the SoH should come from experimental prooves in a real environment. Thus, in this thesis the SoH is considered to be 100% when the CL is equal to the nominal value of 0.1, at the opposite side, SoH is considered to be 0% when CL is equal to 1.5. That consideration comes from the possible accepted values of CL in industrial machines, and it could not match the reality. Nevertheless, this thesis wants to demonstrate the applicability of the multi-model approach on each possible situations, as a consequence, there is no need to be precise on the definition of the variation of CL. For instance, if from experimental results it is derived that CL value in nominal and in degraded condition is very different, it is always possible to adapt the algorithm in order to make it work with different CL. In the 7.2 section of this chapter will be reported some of the most significant situation in order to demonstrate the robustness with respect to uncertainties and to show the behavior of the system with respect to different input and conditions.

7.1 Tests organization

The test are organized following a DoE procedure, in order to reduce process development time and maximize the process reliability. The preliminary hypothesis of the reported tests are:

- The model is supposed to represent the system in a real environment with the nominal condition considered as "new" machine.
- The tests are performed using limited computational environment, so the sample time and the duration of the simulation are constraints.

The organization of the tests can be summed up in the following table:

Simulation time	**uncert**	**Real CL**	**# of filters**	**Cutter speed**	**Initial filter range**	**Position reference**
10000-40000s	0-50%	0.1-1.6 mm	4-8	210-350	0.5-1.5	Sinusoidal, Step

Table 7.1. Test main parameters

All of these parameters were combined and randomly set in order to simulate as much conditions as possible. Nevertheless, in the next paragraph only some of

the most relevant results are reported, in particular, variation of CL and some condition with increasing uncertainties.

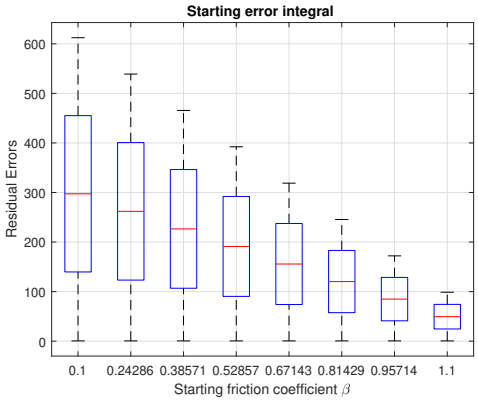
7.2 Results

In order to understand the main settings used in tests and the plots shown for each test, the main settings are reported in the following table 7.2 and in the figure 7.2.

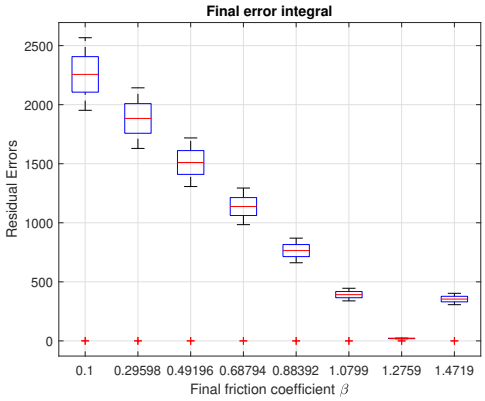
Simulation time	**uncert**	**Real CL**	**Real SoH**	**# of filters**	**Initial filter range**
10000s	10%	1.3 mm	$\sim$ 15%	8	1

Table 7.2. Test main settings.

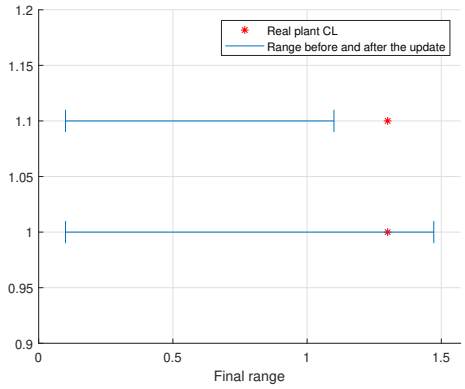
In particular, the panels (a) and (b) represent the residual error boxplots of each filter before and after the update of the filter's range of action; In the x axis are shown the 8 different values of CL for the filters, while in the y axis there is the residual error integral. The panel (c) graphically shows the initial and the final range of the filters. The panel (d) is the tracking of the filter's range update. Finally, the panel (e) is the reliability behavior, while the panel (f) depicts the final wear estimation to be around 85%, which means 15% of remaining useful life (SoH) of the end-effector tool.



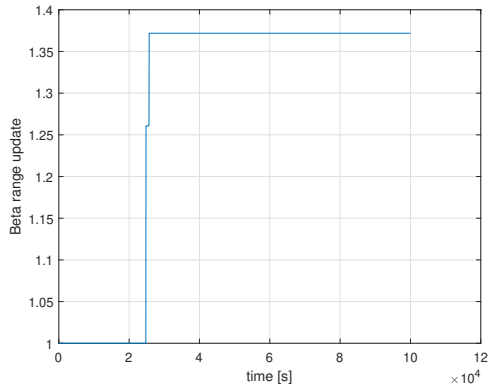
(a) Boxplot of the models residual error before the update



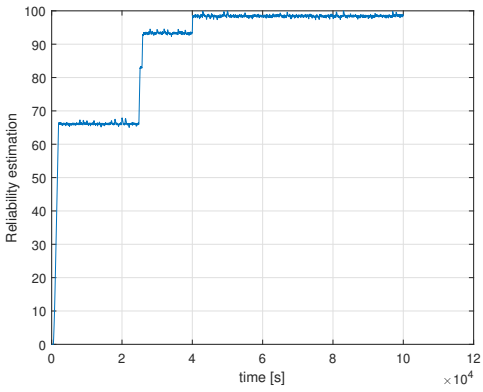
(b) Boxplot of the models residual error after the update



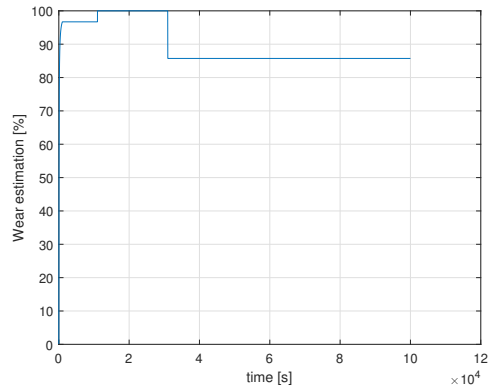
(c) Range before and after the update



(d) Update of the range



(e) Reliability



(f) Wear estimation

Figure 7.2. Significant plots

7.2.1 Test on Nominal conditions with CL=0.1

The first test reported is a simulation in nominal conditions, in order to show the behavior of the system when the SoH of the real system is equal to the nominal(new conditions). The uncertainty imposed is equal to 0, meaning that there is perfect match between the model and the filters. The only disturbance is the one related to the EKF inputs, that it is imposed to have mean equal to 0 and variance equal to 5% of the value.

cl	uncert	t_sim
0.1	0	10000

Table 7.3. Test settings

In the following table are Reported the estimation error before the update of filter's range of action.

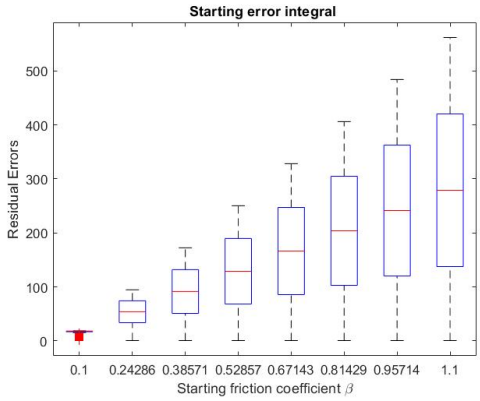
cl	cl_estimated_before_update	err_before_update
0.1	0.102	0,02

Table 7.4. Estimation errors before the update

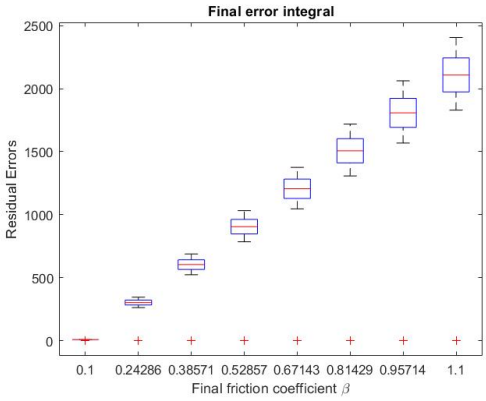
However, as it was expected, since this is a simulation in nominal conditions and the CL=0.1 is naturally already in the range of action of the filters, the update does not lead to improvement in the estimation of CL.

cl	cl_estimated_after_update	err_after_update
0.1	0.102	0,018737592

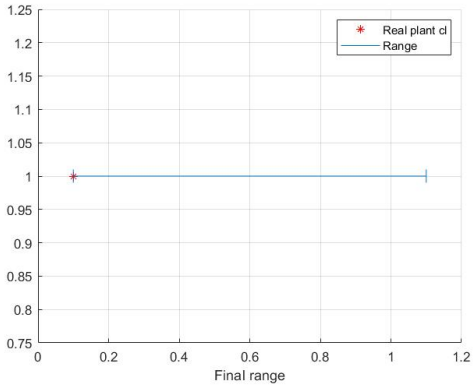
Table 7.5. Estimation errors after the update



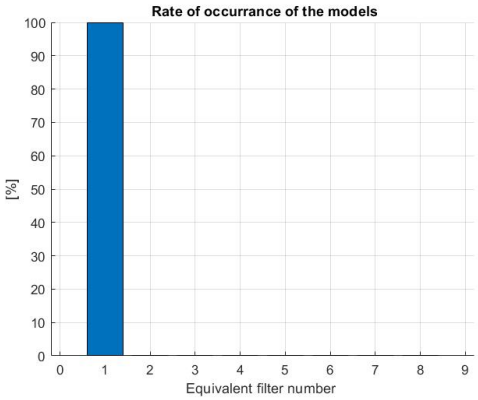
(a) Boxplot of the models residual error before the update



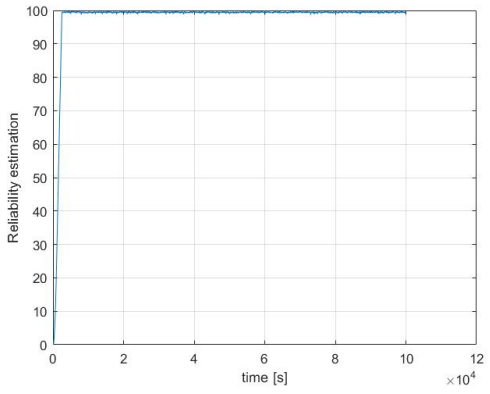
(b) Boxplot of the models residual error after the update



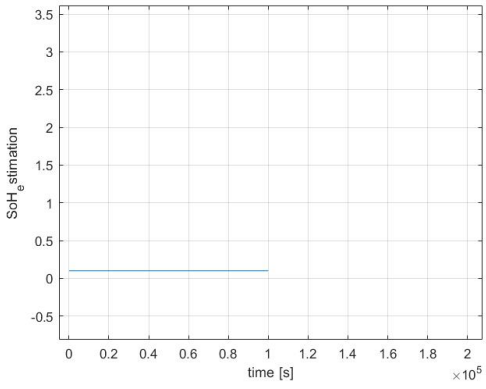
(c) Range representation



(d) Occurrence rate



(e) Reliability



(f) Wear estimation

Figure 7.3. Test on Nominal conditions with CL=0.1

7.2.2 Test on high wear conditions with CL=1.4 outside the initial range

The second test reported wants to show response of the system when the CL of the real system is outside the initial filter's range. The uncertainty imposed is equal to 0 so that there is no parameters difference between the real model and the plant except of CL.

cl	uncert	t_sim
1.4	0	10000

Table 7.6. Test settings

The following table shows the estimation error before the update of filter's range of action.

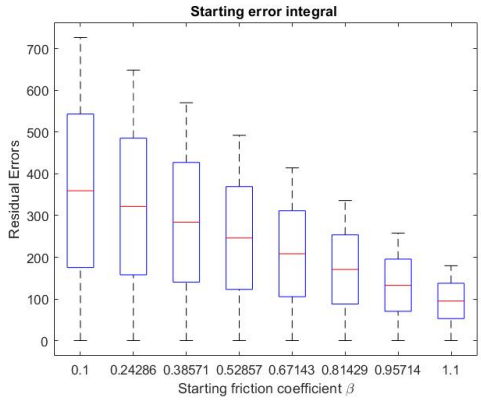
cl	cl_estimated_before_update	err_before_update
1.4	1,067	0,333

Table 7.7. Estimation errors before the update

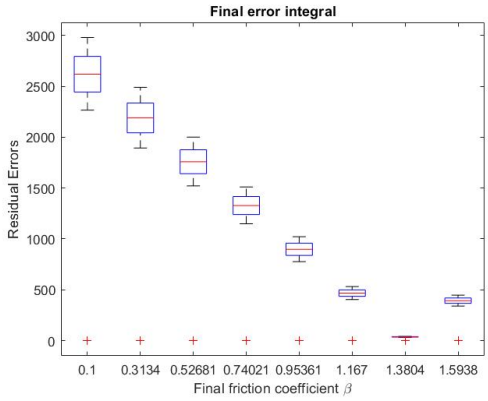
In this case, it is clear that the upgrade of the range which initially didn't contain the the real plant CL, is fundamental to significantly reduce the estimation error:

cl	cl_estimated_after_update	err_after_update
1.4	1,380635367	0,019364633

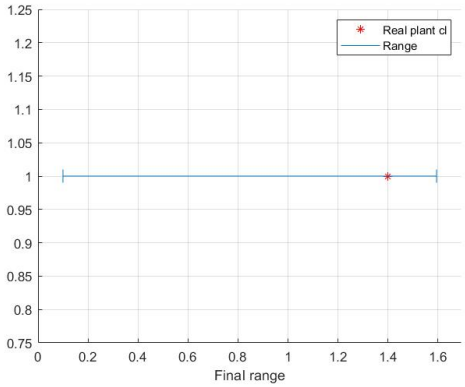
Table 7.8. Estimation errors after the update



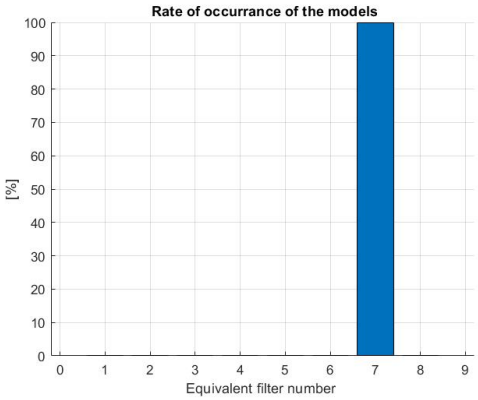
(a) Boxplot of the models residual error before the update



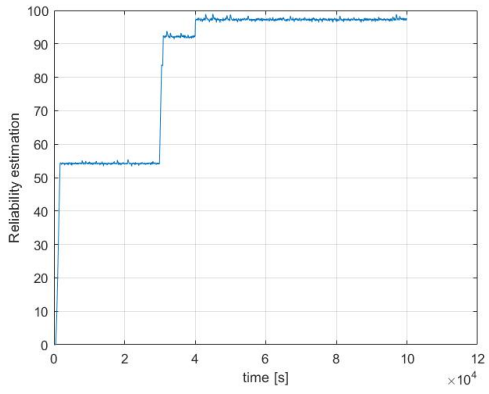
(b) Boxplot of the models residual error after the update



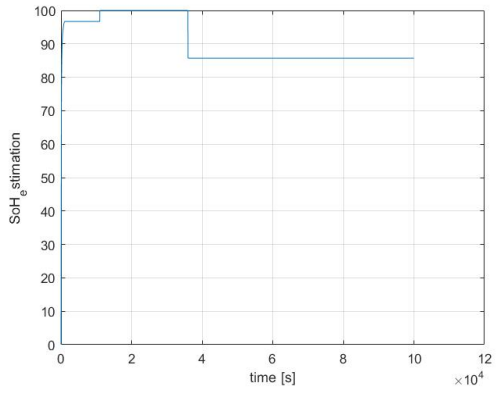
(c) Range representation



(d) Occurrence rate



(e) Reliability



(f) Wear estimation

Figure 7.4. Test on high wear conditions with CL=1.4 outside the initial range

7.2.3 Test on uncertainty 10% and CL 1.3

This test wants to show response of the system when the CL of the real system to uncertainties. The uncertainty imposed is equal to 10%.

cl	uncert	t_sim
1.3	10%	10000

Table 7.9. Test settings

The following table shows the estimation error before the update of filter's range of action.

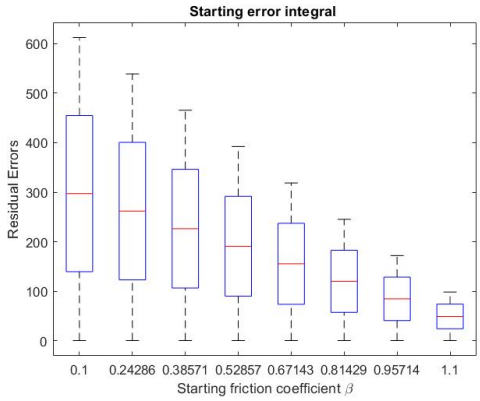
cl	cl_estimated_before_update	err_before_update
1.3	1,067	0,233

Table 7.10. Estimation errors before the update

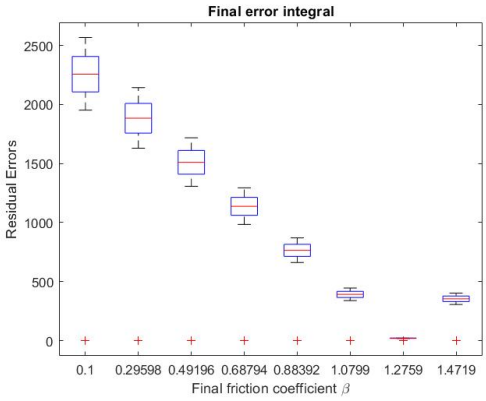
Since in this case the initial range is not enough to include the CL, the update is again significant to reduce the estimation error. However, the estimation error increases as long as the uncertainty percentage increases.

cl	cl_estimated_after_update	err_after_update
1.3	1,276069035	0,023930965

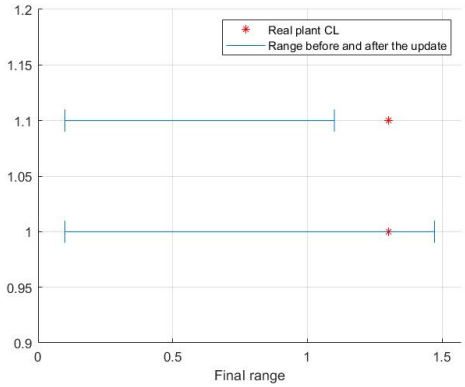
Table 7.11. Estimation errors after the update



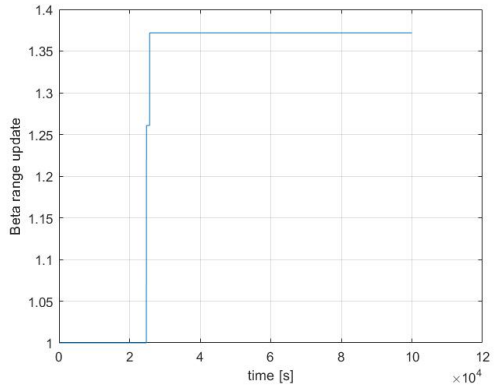
(a) Boxplot of the models residual error before the update



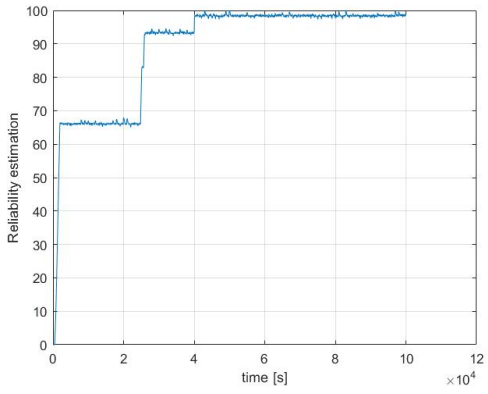
(b) Boxplot of the models residual error after the update



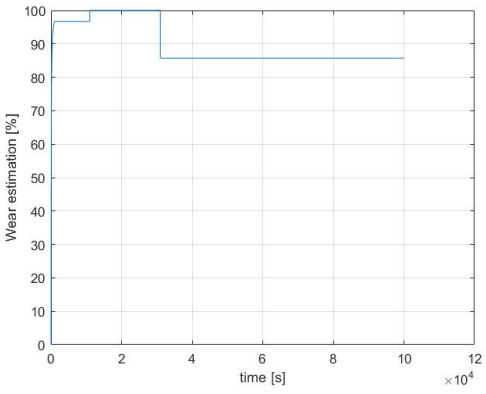
(c) Range representation



(d) Range update



(e) Reliability



(f) Wear estimation

Figure 7.5. Test on uncertainty 10% and CL 1.3

7.2.4 Test on uncertainty 50% and CL 0.5

The last reported test wants to show the system response to 50% uncertainty.

cl	uncert	t_sim
0.5	50%	10000

Table 7.12. Test settings

The following table shows the estimation error before the update of filter's range of action.

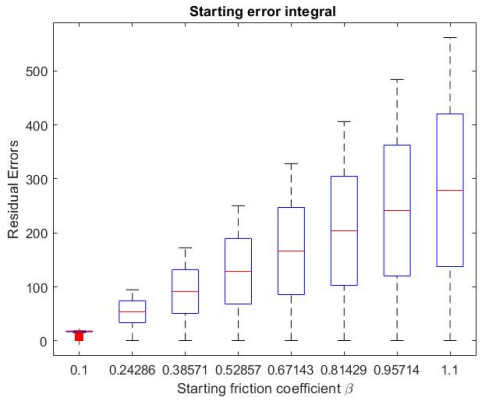
cl	cl_estimated_before_update	err_before_update
0.5	0.39	0,12

Table 7.13. Estimation errors before the update

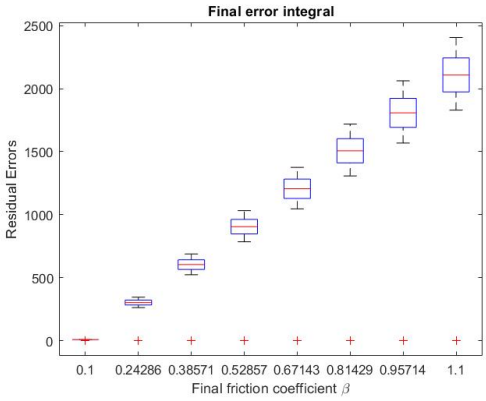
As it is shown in the 6.13, the estimation error significantly increases after 50% of uncertainties. In fact, also in this test it around 0.09.

cl	cl_estimated_after_update	err_after_update
0.5	0.41	0,09

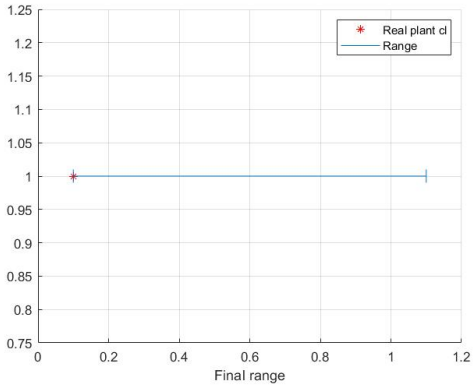
Table 7.14. Estimation errors after the update



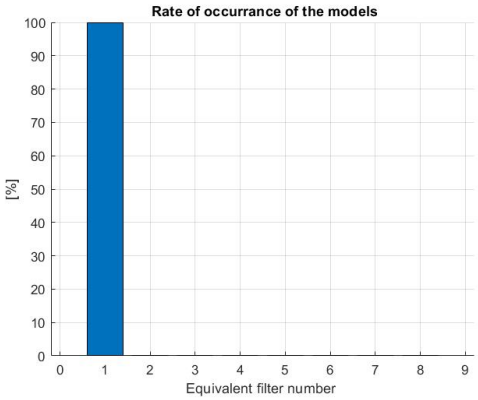
(a) Boxplot of the models residual error before the update



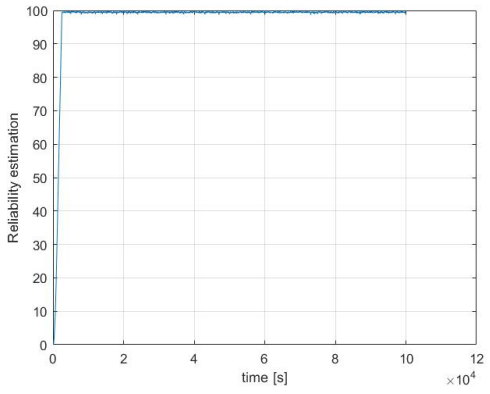
(b) Boxplot of the models residual error after the update



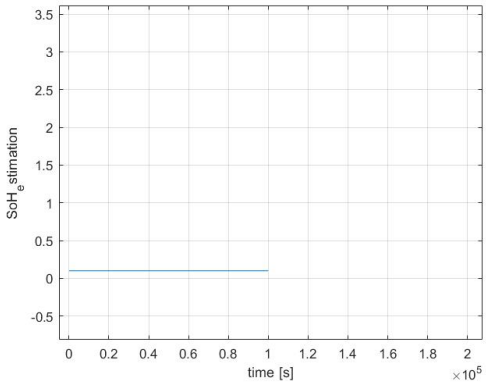
(c) Range representation



(d) Occurrence rate



(e) Reliability



(f) Wear estimation

Figure 7.6. Test on uncertainty 50% and CL 0.5

8 Conclusion and future works

This thesis analyzes several aspects related to the operation and development of a new estimation technique based on a multimodel approach called multi-model approach and produced by Brain technologies and part of the MorePro project. The starting principle that characterized the whole development of this predictive technique is the edge-computing. This characteristic imposes strong computational limitations that lead consequently to an approach that is completely different from the majority of the state of art proposals. Another important principle on which the project is based on is the huge potential field of application of the system. In fact, even if the system proposed in this thesis is limited to a simplified model of CNC machines, this work lays the foundation for future application to more complex systems. In particular, following the same path described in the chapters the following contributions and conclusions were obtained: Firstly, it has been seen how the application of the algorithm already developed by brain Technologies in the ERMES project for the SoH estimation of battery system by adopting a new switching multi-model estimation technique is also suitable for a mechanical applications, which concerns the wear prediction of a dynamical model resembling the work carried out by a CNC machine. This result, comes from the choice of a friction coefficient as main representative for discriminating different wear conditions of the plant under assumption and from the analysis on the residual errors of the Extended Kalman Filters bank implemented. Secondly, a reliability analysis has been carried out, developing an index that has two main functions:

- Check the correctness of the SoH estimation in terms of parameters range and best model occurrence validation.
- Give the necessary feedback for the filters range of action and parameters update.

Despite the system has not been tested in a real environment, not going beyond a MIL (Model in the loop) testing, this thesis work lays foundations for future edge-computing SoH prediction techniques.

8.1 Future works

This thesis should act as foundation for the application of this approach for reliable and low-cost predictive maintenance techniques for industrial machines. Once the applicability of this approach has been proven, the next steps concerns to bring improvements to the systems and perform the next steps of testing: Software in the loop and hardware in the loop. First of all, the first possible improvement regards the possibility to include machine learning techniques to reduce the possible models to a finite number of possibilities. This study has been already started and a first draft has been carried out by the MorePro team [24], in particular, K-Neares and self-organizing maps has been implemented in order to reduce the possible models among all. This work could lead to huge steps forward regarding the model's

parameters update because the algorithm should choose among finite number of possible models.
Another possibility regards the modelling of the State of health as function of more than a single parameter. To do this, data from observation of real machine are necessary. Once the data are available, a deeper modelling of the cutting process of new and old machinery can be carried out, and the prediction algorithm could be tested with this new data. The final step to do is without doubts the testing with a real system, planning a detailed DoE in order to finally test the behavior in all possible scenarios and definitively proof the feasibility of this approach.

Appendix A

Boxplot

The boxplot, otherwise called the exterior and quartiles diagram, is a graphic representation adopted to describe the distribution of a sample by means of dispersion and position indices.

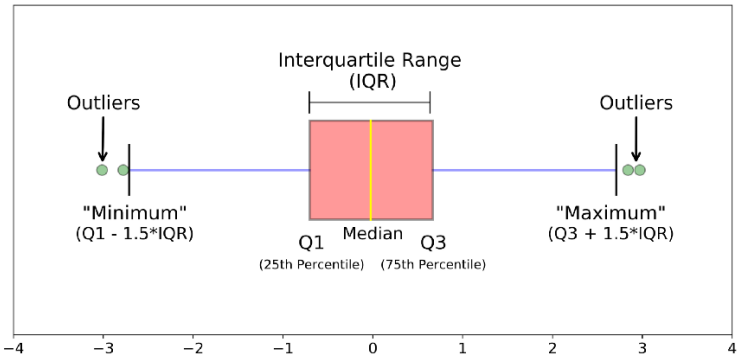


Figure 8.1. Boxplot representation

It can be oriented both vertically and horizontally by a rectangle and it is divided into two parts from which two segments come out. The box is delimited by the first and third quartile q1/4 and q3/4 and divided inside by the median q1/2. So that in the red rectangle are represented the majority of the values, while segments delimits the minimum and maximum values also called outliers.

Appendix B

DoE

Design of experiments (DOE) is defined as the branch of applied statistics that deals with planning, conducting, analyzing, and interpreting controlled tests to evaluate the factors that control the value of a parameter or group of parameters [14]. DOE is a powerful data collection and analysis tool that can be very useful in a variety of experimental situations. Statistical methods are often used at the end of the testing phase in order to summarize data and extract additional information about the process. Nevertheless, incorporating statistical considerations into the design of the experiments can lead to many advantages such as:

- Reduction of process development time;
- efficient use of resources;
- Increased process reliability;

The DOE method consists of two main phases:

- Screening phase where the objective is the identification of the significant factors and their correlation.
- Optimization phase during which the objective is the identification of the response.

Some important concepts of the DoE are[23]:

- Controllable input factors, or x factors, are those input parameters that can be modified in an experiment or process.
- Uncontrollable input factors are those parameters that cannot be changed.
- Responses, or output measures, are the elements of the process outcome that gage the desired effect.
- Hypothesis testing helps determine the significant factors using statistical methods. There are two possibilities in a hypothesis statement: the null and the alternative.
- Blocking and replication: Blocking is an experimental technique to avoid any unwanted variations in the input or experimental process. For example, an experiment may be conducted with the same equipment to avoid any equipment variations.
- Interaction: When an experiment has three or more variables, an interaction is a situation in which the simultaneous influence of two variables on a third is not additive.

The DoE process can be summed up in the following figure:

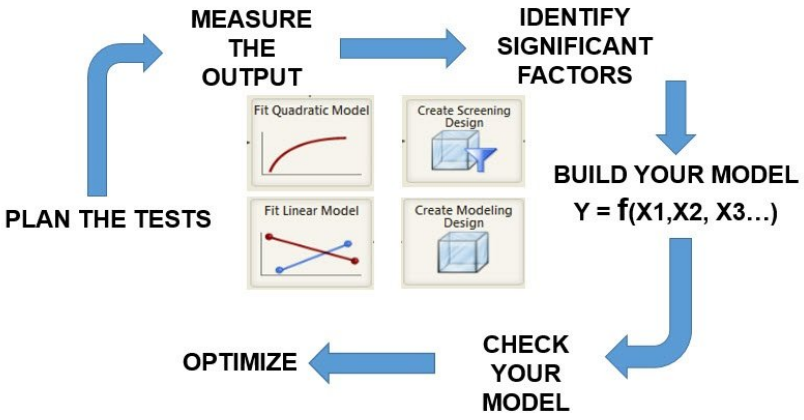


Figure 8.2. DoE

Appendix C

Gradient Descent algorithm

Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. The pseudocode of a gradient descent algorithm is:

Input:

Training sample $S = \{(\vec{x}, y)\}^m$, where $(\vec{x}, y) \in X \times \{r_q \succ r_{q-1} \succ \cdots \succ r_1\}$

Learning rate $\eta > 0$, cost parameters $\{\tau_{k(i)}\}$ and $\{\mu_{q(i)}\}$, penalty weight λ

Make $S' = \{((\vec{x}_i^{(1)}, \vec{x}_i^{(2)}), z_i)\}_{i=1}^{\ell}$ from S;

$\vec{w} = \vec{0}$;

while (stop_condition isn't met){

$\Delta\vec{w} = \vec{0}$;

for$(i = 0;\ i < \ell;\ i{+}{+})$\{

if $(z_i \left\langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \right\rangle < 1)$ $\quad \Delta\vec{w} = \Delta\vec{w} + z_i \tau_{k(i)} \mu_{q(i)} (\vec{x}_i^{(1)} - \vec{x}_i^{(2)})$;

}

$\Delta\vec{w} = \Delta\vec{w} - 2\lambda w$;

$\vec{w} = \vec{w} + \eta\Delta\vec{w}$;

}

return $\vec{w}$;

Figure 8.3. Gradient descent pseudocode

On each iteration, the parameters are updated in the opposite direction of the gradient of the objective function J(w) w.r.t the parameters where the gradient gives the direction of the steepest ascent. The size of the step we take on each iteration to reach the local minimum is determined by the learning rate. Therefore, the direction of the slope downhill until we reach a local minimum is followed.

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] Preet Joy, Adel Mhamdi, Alexander Mitsos. *Optimization-based observability analysis.* Aachener Verfahrenstechnik - Process Systems Engineering (SVT), RWTH Aachen University, Forckenbeckstrasse 51, Aachen 52074, Germany.

[4] 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

[5] Gouriveau Rafael, Medjaher Kamal, Zerhouni Noureddine *From prognostic and health systems management to predictive maintenance 1: Monitoring and Prognostics* Newark: John Wiley & Sons, Incorporated, 2016.

[6] Aleksandra Marjanović, Goran Kvaščev, 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.

[7] Sayed-Mouchaweh Moamar, Lughofer Edwin. *Predictive maintenance in dynamic systems: advanced methods, decision support tools and real-world applications* Springer, 2019.

[8] 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.

[9] Daga A. P., Garibaldi L. *Machine vibration monitoring for diagnostics through hypothesis testing* MDPI AG, 2019.

[10] 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.

[11] 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.

[12] *Reshaping the world with fuzzy logic* WorldQuant Perspectives, April 2018.

[13] 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.

[14] Davide Faverato. *Virtual Sensing for the Estimation of the State of Health of batteries* October 2020, Politecnico di Torino.

[15] J. Kallrath, ed. *Modeling Languages in Mathematical Optimization*, Applied Optimization, Vol. 88, Kluwer, Boston 2004.

[16] 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 FAMENA XXXIX-4 (2015),ISSN 1333-1124, eISSN 1849-1391.

[17] Dora Sabau, Mirela Dobra *System identification and control analysis for CNC machines*, 978-1-5386-2205-6/18/2018 IEEE.

[18] *MATLAB guidelines*,Application-specific guidelines for model architecture, design, and configuration R2020b. URL[https://it.mathworks.com/help/simulink/modeling-guidelines.html].

[19] Harvey M. Wagne. *Global Sensitivity Analysis*; 1 Dec 1995h.

[20] Preet Joy, Adel Mhamdi, Alexander Mitsos. *Optimization-based observability analysis.* Aachener Verfahrenstechnik - Process Systems Engineering (SVT), RWTH Aachen University, Forckenbeckstrasse 51, Aachen 52074, Germany.

[21] 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.

[22] Roberta Heale, Alison Twycross *Validity and reliability in quantitative studies* doi: 10.1136/eb-2015-1021292015 18: 66-67 originally published online May 15, 2015

[23] K. Sundararajan; *DESIGN OF EXPERIMENTS – A PRIMER*

[24] Davide Zanon; *End-effector tools State of Health estimation: a data driven approach*, April 2021; Politecnico di Torino.

[25] Marjolein J.P.Mensab Frans KlijnaKarin M.de Bruijna Eelcovan Beekab; *The meaning of system robustness for flood risk management*

[26] Cao, Yunbo Xu, Jun Liu, Tie-Yan Li, Hang Huang, Yalou Hon, Hsiao-Wuen. *Adapting Ranking SVM to Document Retrieval*; 10.1145/1148170.1148205 JO - Proc. ACM SIGIR Int. Conf. Information Retrieval (SIGIR'06).

[27] Adriano Di Marzio; *End-effector wear monitoring system of milling macchine 5-axis: System Design and Validation*, April 2021; Politecnico di Torino.