# Politecnico di Torino

Department of Mechanical and Aerospace Engineering

Master's Degree Thesis

# Prognostics of on-board electromechanical actuators: Bio-Inspired Metaheuristic Algorithms

**Supervisors**

Prof. Paolo **MAGGIORE**
Eng. Matteo D.L. **DALLA VEDOVA**
Eng. Pier Carlo **BERRI**

**Candidate**

Ivana **QUERQUES**

A.Y. 2020-2021

*Alla prima riga*
*e seconda riga*

# Abstract

The aerospace sector is increasingly looking to the advent of the more electrical philosophy, but from a practical point of view the technologies available are neither sufficient nor suitable for aeronautical use, especially for primary flight controls, for which the levels of safety and reliability required are very high. For this reason, great care is needed in fault management, not only through redundancies, which are not always feasible on an aircraft for weight reasons, but also through suitably designed diagnostic and prognostic software.

Given the importance that these problems have assumed today, it was therefore considered important to try to develop algorithms capable of detecting faults before they can irreversibly compromise the aircraft.

In particular, the choice of the field of application of these algorithms (inspired by various biological species) fell on an electromechanical actuator, aimed at the implementation of primary flight commands.

The actuator and its failure modes were modelled in Matlab Simulink. After studying the dynamic response of the system and reporting the results, several algorithms were implemented in order to use them for prognostic purposes.

# Contents

**List of Figures** I

**List of Tables** III

**1 Overview** 1

1.1 More-electric philosophy . . . . . . . . . . 1
1.2 EMA vs EHA . . . . . . . . . . 2
1.3 Faults and reliability . . . . . . . . . . 5
1.4 Prognostics as a solution . . . . . . . . . . 9
1.4.1 Model-based prognostics . . . . . . . . . . 10
1.5 Methaeuristic bio-inspired algorithms . . . . . . . . . . 11
1.5.1 Evolutionary Algorithms (EA) . . . . . . . . . . 12
1.5.2 Swarm Intelligence algorithms (SI) . . . . . . . . . . 17

**2 Electro-mechanical actuator model** **23**

2.1 Reference Model . . . . . . . . . . 24
2.1.1 Command block . . . . . . . . . . 26
2.1.2 Load (Resistant Torque) block . . . . . . . . . . 27
2.1.3 Controller block . . . . . . . . . . 28
2.1.4 Inverter block . . . . . . . . . . 30
2.1.5 Motor block . . . . . . . . . . 34
2.1.6 Motor-transmission dynamics block . . . . . . . . . . 40
2.1.7 Signal acquisition block . . . . . . . . . . 42
2.2 Monitor Model . . . . . . . . . . 43
2.2.1 Controller block . . . . . . . . . . 43
2.2.2 Electrical model block . . . . . . . . . . 44
2.2.3 Mechanical model block . . . . . . . . . . 46
2.3 Models output and comparison . . . . . . . . . . 47

**3 Reliability and fault implementation** **53**

3.1 Failure Analysis . . . . . . . . . . 53
3.2 Fault implementation . . . . . . . . . . 58
3.2.1 Dry friction . . . . . . . . . . 58
3.2.2 Backlash . . . . . . . . . . 65
3.2.3 Short Circuit . . . . . . . . . . 70

3.2.4 Eccentricity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
3.2.5 Proportional Gain . . . . . . . . . . . . . . . . . . . . . . . . . . . 84

**4 Algorithms 89**

**Bibliography 91**

# List of Figures

1.1 *Fly-by-wire system* . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2 *Electro-hydraulical actuator* . . . . . . . . . . . . . . . . . . . . . . . . 3
1.3 *Electro-mechanical actuator* . . . . . . . . . . . . . . . . . . . . . . . . 4
1.4 *Control functions* . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
1.5 *Evolutive Algorithm flow-chart* . . . . . . . . . . . . . . . . . . . . . . 14
1.6 *Genetic algorithm scheme* . . . . . . . . . . . . . . . . . . . . . . . . 16
1.7 *Differential Evolution scheme* . . . . . . . . . . . . . . . . . . . . . . 17
1.8 *Generic function and the three displacements of the particles* . . . . . . . . . 19
1.9 *Typical PSO flowchart* . . . . . . . . . . . . . . . . . . . . . . . . . 20
1.10 *Generic initial and final particles condition* . . . . . . . . . . . . . . . . 20
1.11 *GWO algorithm flowchart* . . . . . . . . . . . . . . . . . . . . . . . . 22
1.12 *GWO update position scheme and hierarchy* . . . . . . . . . . . . . . . . 22

2.1 *Schematic of the general high-fidelity model* . . . . . . . . . . . . . . . . 24
2.2 *Command Block Simulink* . . . . . . . . . . . . . . . . . . . . . . . . 27
2.3 *Load Block Simulink* . . . . . . . . . . . . . . . . . . . . . . . . . . 27
2.4 *Control Electronics block diagram* . . . . . . . . . . . . . . . . . . . . 29
2.5 *PID Controller block diagram* . . . . . . . . . . . . . . . . . . . . . . 30
2.6 *Resolver block* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
2.7 *Park antitransformate block* . . . . . . . . . . . . . . . . . . . . . . . 32
2.8 *Clarke antitransformate block* . . . . . . . . . . . . . . . . . . . . . . 32
2.9 *Hysteresis PWM block* . . . . . . . . . . . . . . . . . . . . . . . . . 33
2.10 *Three-phase H-bridge block* . . . . . . . . . . . . . . . . . . . . . . . 34
2.11 *Three-phase Permanent Magnet Synchronous Motor* . . . . . . . . . . . . . 35
2.12 *Counter-eletromotive force constant implementation* . . . . . . . . . . . . . 36
2.13 *Stator-inverter connection* . . . . . . . . . . . . . . . . . . . . . . . . 37
2.14 *Equivalent circuit for a single phase* . . . . . . . . . . . . . . . . . . . 37
2.15 *RL circuit modeling* . . . . . . . . . . . . . . . . . . . . . . . . . . 38
2.16 *Motor torque implementation* . . . . . . . . . . . . . . . . . . . . . . 39
2.17 *PMSM motor modeling* . . . . . . . . . . . . . . . . . . . . . . . . . 39
2.18 *Motor-transmission dynamics block* . . . . . . . . . . . . . . . . . . . 40
2.19 *Comparison between the actuation torque sign and the saturation port value* . 41
2.20 *Signal acquisition block* . . . . . . . . . . . . . . . . . . . . . . . . . 42
2.21 *Controller block for the monitor model* . . . . . . . . . . . . . . . . . . 44
2.22 *Electrical block for the monitor model* . . . . . . . . . . . . . . . . . . 45

2.23 *Mechanical block for the monitor model* . . . . . . . . . . . . . . . . . . . . 47
2.24 *Motor position* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
2.25 *Motor position zoom* . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
2.26 *Motor position error* . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
2.27 *User position* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
2.28 *User position zoom* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
2.29 *User position error* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
2.30 *Motor speed* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
2.31 *Motor speed zoom* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
2.32 *Motor angular speed error* . . . . . . . . . . . . . . . . . . . . . . . . 50
2.33 *Three-phase equivalent current* . . . . . . . . . . . . . . . . . . . . . 50
2.34 *Three-phase equivalent current error* . . . . . . . . . . . . . . . . . . 51

3.1 *Stribeck curve* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
3.2 *Borello dry friction model representation* . . . . . . . . . . . . . . . . 61
3.3 *Borello dry friction Simulink implementation* . . . . . . . . . . . . . . 61
3.4 *User position for a step command - Dry Friction* . . . . . . . . . . . . 62
3.5 *Motor velocity for a step command - Dry Friction* . . . . . . . . . . . . 63
3.6 *Motor velocity for a step command - Zoom - Dry Friction* . . . . . . . . 63
3.7 *Equivalent current for a step command - Dry Friction* . . . . . . . . . . 63
3.8 *User position for a chirp command - Dry Friction* . . . . . . . . . . . . 64
3.9 *User position for a chirp command - Zoom - Dry Friction* . . . . . . . . 64
3.10 *User position for a chirp command - Dry Friction* . . . . . . . . . . . . 64
3.11 *User position for a chirp command - Zoom - Dry Friction* . . . . . . . . 64
3.12 *Equivalent current for a chirp command - Dry Friction* . . . . . . . . . 65
3.13 *User position for a step command - Backlash* . . . . . . . . . . . . . . 67
3.14 *User position for a step command - Zoom - Backlash* . . . . . . . . . . 67
3.15 *Motor velocity for a step command - Backlash* . . . . . . . . . . . . . . 67
3.16 *Equivalent current for a step command - Backlash* . . . . . . . . . . . 68
3.17 *User position for a chirp command - Backlash* . . . . . . . . . . . . . . 68
3.18 *User position for a chirp command - Zoom - Backlash* . . . . . . . . . . 68
3.19 *Motor velocity for a chirp command - Backlash* . . . . . . . . . . . . . 69
3.20 *Motor velocity for a chirp command - Zoom - Backlash* . . . . . . . . . 69
3.21 *Equivalent current for a chirp command - Backlash* . . . . . . . . . . . 69
3.22 *Equivalent current for a chirp command - Zoom - Backlash* . . . . . . . 69
3.23 *Short-circuit correction block - Monitor Model* . . . . . . . . . . . . . . 73
3.24 *User position for a step command - Short Circuit* . . . . . . . . . . . . 74
3.25 *User position for a step command - Zoom - Short Circuit* . . . . . . . . 74
3.26 *Motor velocity for a step command - Short Circuit* . . . . . . . . . . . . 74
3.27 *Motor velocity for a step command - Zoom - Short Circuit* . . . . . . . 74
3.28 *Equivalent current for a step command - Short Circuit* . . . . . . . . . 75
3.29 *Equivalent current for a step command - Zoom - Short Circuit* . . . . . 75
3.30 *User position for a chirp command - Short Circuit* . . . . . . . . . . . . 75
3.31 *User position for a chirp command - Zoom - Short Circuit* . . . . . . . 75
3.32 *Motor velocity for a chirp command - Short Circuit* . . . . . . . . . . . 76

3.33 *Motor velocity for a chirp command - Zoom - Short Circuit* . . . . . . . . . . 76
3.34 *Equivalent current for a chirp command - Short Circuit* . . . . . . . . . . . 76
3.35 *Nominal condition - Eccentricity* . . . . . . . . . . . . . . . . . . . . . . . 77
3.36 *Static eccentricity* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
3.37 *Dynamic eccentricity* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
3.38 *Eccentricity correction block - Monitor Model* . . . . . . . . . . . . . . . . 80
3.39 *User position for a step command - Eccentricity* . . . . . . . . . . . . . . . 81
3.40 *Motor velocity for a step command - Eccentricity* . . . . . . . . . . . . . . 81
3.41 *Equivalent current for a step command - Eccentricity* . . . . . . . . . . . . 82
3.42 *Equivalent current for a step command - Zoom - Eccentricity* . . . . . . . . 82
3.43 *User position for a chirp command - Eccentricity* . . . . . . . . . . . . . . 82
3.44 *User position for a step command - Zoom - Eccentricity* . . . . . . . . . . 82
3.45 *Motor velocity for a chirp command - Eccentricity* . . . . . . . . . . . . . 83
3.46 *Motor velocity for a step command - Zoom - Eccentricity* . . . . . . . . . . 83
3.47 *Equivalent current for a chirp command - Eccentricity* . . . . . . . . . . . 84
3.48 *User position for a step command - Proportional Gain* . . . . . . . . . . . 85
3.49 *Motor velocity for a step command - Proportional Gain* . . . . . . . . . . . 85
3.50 *Equivalent current for a step command - Proportional Gain* . . . . . . . . . 85
3.51 *User position for a chirp command - Proportional Gain* . . . . . . . . . . . 86
3.52 *User position for a chirp command - Zoom - Proportional Gain* . . . . . . . 86
3.53 *Motor velocity for a chirp command - Proportional Gain* . . . . . . . . . . . 86
3.54 *Motor velocity for a chirp command - Zoom - Proportional Gain* . . . . . . 86
3.55 *Equivalent current for a chirp command - Proportional Gain* . . . . . . . . 87
3.56 *Equivalent current for a chirp command -Zoom - Proportional Gain* . . . . 87

# List of Tables

1.1 *Advantages and disadvantages of EMA and EHA systems* . . . . . . . . . . . . 5

2.1 *EMA systems parameters - High Fidelity Model* . . . . . . . . . . . . . . . . . . 25
2.2 *Command settings - High Fidelity Model* . . . . . . . . . . . . . . . . . . . . . 26
2.3 *Mechanical parameters - Low Fidelity Model* . . . . . . . . . . . . . . . . . . . 46

3.1 *Mechanical/structural faults* . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
3.2 *Motor faults* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
3.3 *Electrical/Electronics faults* . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
3.4 *Failure mode effects* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
3.5 *Fault Tree Analysis Requirements* . . . . . . . . . . . . . . . . . . . . . . . . 58

# Chapter 1

# Overview

## 1.1 More-electric philosophy

In the aerospace sector, the development of technology generally aims to achieve a reduction in weight and operating and/or maintenance costs and an increase in performance, reliability and safety. This evolution is combined with that of electrical systems, so that heavy and complex hydraulic power systems can be replaced by lighter, less bulky and less expensive electrical systems during their operating life.
For this reason, the *more electric* philosophy is becoming increasingly widespread, tending more and more towards the *all electric* one [1].
The current application field of this philosophy involves flight commands, power generation and propulsion control. Here we will focus on the flight controls.

At present, the implementation of flight controls in medium and large aircraft is done by *fly-by-wire* (FBW), a technology created to meet the need for increasingly faster modern aircraft.
Specifically, in the cockpit the control column allows roll and pitch maneuvers to be performed, while the pedals allow the rudder to be controlled for yawing. This is done by means of a computer that translates pilot-induced command movements into low-power electrical signals. The system comprises electric cables, which have already replaced the old steel ones, to transmit the electric signals to a hydraulic pump position transducer, which drives the relevant moving surface. The *fly-by-wire* system also allows signals to be introduced not from pilot input, but from the computer itself, to automatically stabilise the aircraft, acting as a closed-loop system [2].
As a result, the system is electrically controlled but hydraulically powered.
A diagram of the typical operation of the *fly-by-wire* system is shown in the figure 1.1.

Technology is moving in the direction of *fly-by-light*, where electric cables are replaced by fibre optic cables. This system is lighter and has a higher bandwidth and a lower risk of electromagnetic interference.

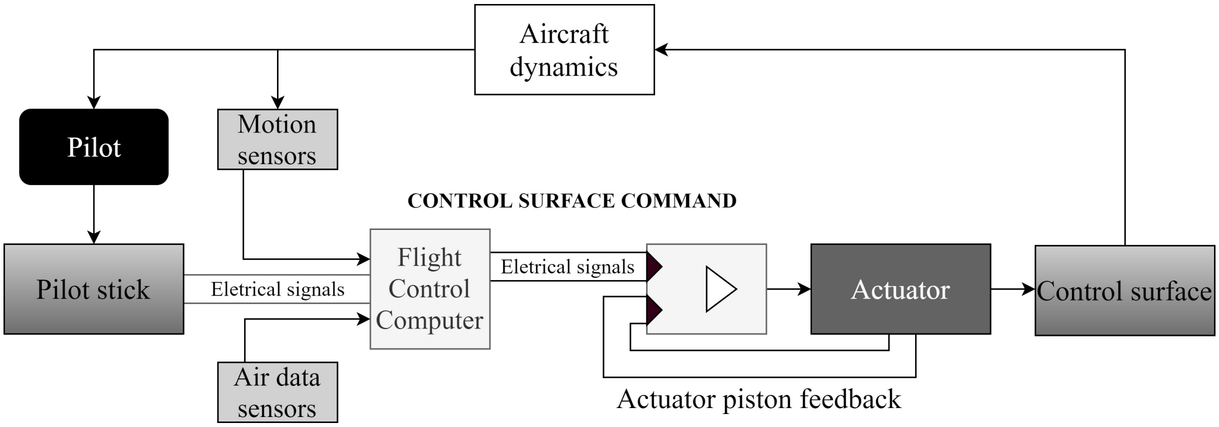


Figure 1.1: *Fly-by-wire system*

However, the challenge is to completely eliminate hydraulic connections and the associated risks. Currently, this type of technology is slowly finding use mainly for secondary flight controls as they are less critical to flight safety than primary ones. It is also being developed for primary flight controls for small aircraft.
The development in this sense of an all-electric actuation - which finds application in the electromechanical actuator (EMA) that we will analyze in the section 1.2 - is due to the different criticality of flight controls.

In fact, primary flight controls have the task of locally modifying the aerodynamic forces to generate aerodynamic moments around the three body axes of the aircraft. On the other hand, the secondary flight controls have the objective of altering the coefficients of drag and maximum lift without modifying the aircraft's attitude; this makes it possibile to obtain greater and better control of the aircraft in any flight phase and condition.

Due to the crucial nature of the former, the mere replacement of the hydraulic system by the electrical one is insufficient. In fact, a higher level of safety is required, which can be achieved by the installation of additional devices that increase the reliability of the system. However, this process makes the system itself havier, more complicated and more expensive; for these reasons the solution described still needs further development.

## 1.2 EMA vs EHA

In order to better understand the issues related to hydraulic or electric actuation technologies, the main features of electro-hydraulic (EHA) and electro-mechanical actuators (EMA) are presented in this section.

The most commonly used electro-hydraulic actuator consists of a bidirectional variable speed pump driven by appropriate electronics. It provides switching or actuation by hydraulic force, which is constantly regulated by servo valves. These regulate the hydraulic flow of the actuator through the relationship between pressure and flow rate, allowing the conversion of hydraulic energy into mechanical one. The fluid pressure pushes a piston, which is located inside a specific chamber, back and forth.

The electro-hydraulic actuator provides movement by constantly adapting to the demands of the system in commensurate increments. This is made possible by a set of valves and a servo amplifier in the actuator circuit. The amplifier receives input signals from the sensors, interprets the demands and processes them into appropriate signals, which are sent to a series of servo valves. Each servo valve is used to control the duration of the hydraulic actuator's movement, the speed at which it moves, and the amount of torque or power it exerts.
An electro-hydraulic actuator can be linear, i.e. perform the actuating movement in a straight line, or rotative. In the first case, the actuator is called a jack and is mainly used for primary flight controls; in the second case, the actuator is a motor, currently used for secondary flight controls [3].

At present, aircraft such as the A380 and B787 already use the EHA system as a back-up to the hydraulic system.

The figure 1.2 shows a generic example of an electro-hydraulic actuator.

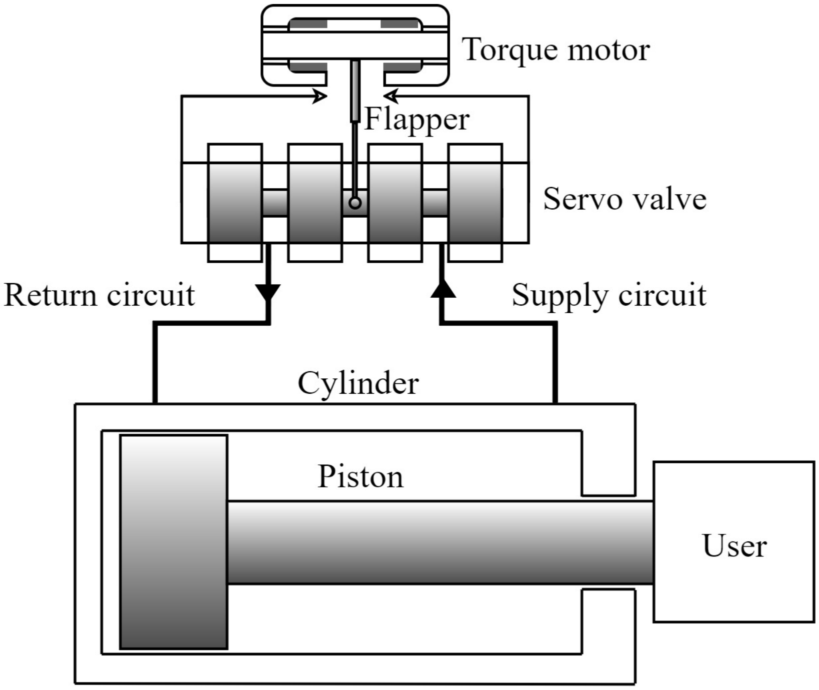


Figure 1.2: *Electro-hydraulical actuator*

The advantages and disadvantages of an electro-hydraulic actuation system are summarised below [4]:

- long service life;

- large actuation forces;
- overload protection;
- reliability due to a well-designed and guaranteed pump;
- developed and validated active/stand-by and active/active configurations;
- hydraulic maintenance requirements;
- required weight improvement;
- required cost improvement.

The electromechanical actuator (EMA) is now described in general terms.

It consists of an electric motor (DC, in aviation) which takes electrical power from the transmission bar of the electrical system and converts it into mechanical power. In particular, the angular rate of the electric motor is regulated by a power converter which changes the magnitude and frequency of the phase voltages applied to the stator windings. The torque and speed developed by the motor are managed by a reducer to reach values that meet the downstream power requirements, making the system more efficient in terms of volume and weight. The system ends with a screw drive element or, in general, a gear system [5].
In fact, for an electric system the actuator can only be rotative, so if translational actuation is required, a component is needed that is able to convert rotational motion into translational one. The system is connected to a control logic that receives the command and compares it with the actual position or speed. These measurements are acquired by means of appropriate transducers that collect the signals and convey them to an amplifier before reaching the motor.

The figure 1.3 below shows a generic example of an electro-mechanical actuator.

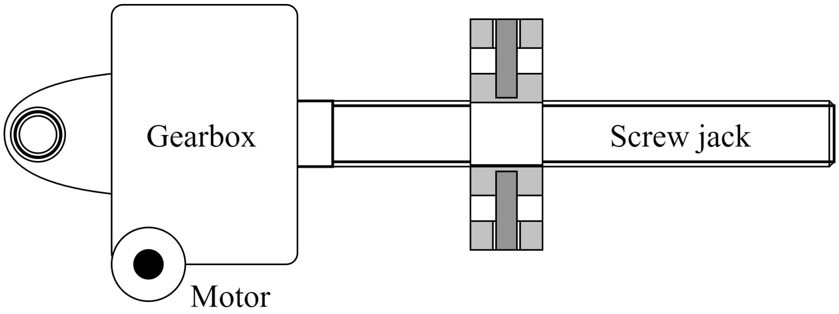


Figure 1.3: *Electro-mechanical actuator*

An electromechanical actuator, on the other hand, has the following characteristics [4]

- low weight;

- high efficiency;
- maintenance cost reduction;
- potentially higher reliability, but technology still under development;
- possible use of the actuators in stand-by;
- sensitive and critical screw elements because they increase mechanical play (can be replaced by ball or roller screws);
- flutter due to mechanical backlash;
- susceptibility to jamming.

The following table (1.1) summarises the advantages and disadvantages of both implemented systems, taking into account the current state of the art. In white are the pros, in black the cons, and in grey the possible advantages that could be obtained following further development of the current technology.

Table 1.1: *Advantages and disadvantages of EMA and EHA systems*

| **EHA** | | **EMA** | |
|---|---|---|---|
| □ | Long service life | □ | High efficiency |
| □ | Ability to predict failures | ■ | Inability to predict failures without diagnostic and prognostic tools |
| □ | Reliability driven by a well-designed and guaranteed pump | ▒ | Potentially higher reliability but under development technology |
| □ | Developed and validated active/stand-by and active/active configuration | □ | Possible use of actuators in stand-by configuration |
| ■ | Hydraulic maintenance requirements | ■ | Maintenance requirements due to the critical screw elements risulting in possible backlash and jamming |
| ■ | Required cost improvement | □ | Maintenance cost reduction |
| ■ | Required weight improvement | □ | Low weight |

## 1.3 Faults and reliability

The advent of the *more electric*, as we have already seen, is directing the development of technology towards an increasing use of electric devices to replace hydraulic ones. However, in the field of actuators, this does not imply a direct switch from an electro-hydraulic actuator to an electromechanical one. In fact, it is necessary to evaluate what is involved in replacing a hydraulic device with an electric one, in terms of use, implementation, monitoring and tendency of the device to fail.

We can assess the problems related to the transition from an actuation system working with hydraulic devices to a fully electric one.

An electric system does not use any hydraulic power, so it favors greater reliability, associated with a reduction in losses, and a lowering of maintenance costs as there are no components subject to wear such as, for example, the gaskets. Furthermore, it is a *position sensitive* device, that is it's only used for maintaining and movement of the implemented utilities. The hydraulic system, on the other hand, produces continuous pressure, i.e. it is a *load sensitive* system, consuming a lot of energy and generating a lot of unwanted heat. In this case the oil temperature must be monitored because the leakage gives heat to the fluid with the risk of loss of its viscous properties and possible polymerization [3].

It must be considered that it is possible to replace the old hydraulic system with an electric actuation system as long as it meets at least the same safety requirements [6]. This translates into appropriate fault management, should they occur.
A failure is an unwanted event consisting of the interruption of a control action, the generation of sensor measurement errors or the modification of the input or output dynamics expected for a defined system. The effect is a degradation of performance, damage, or in the most catastrophic case, a collapse of the entire equipment [7].

For hydraulic systems there is a tendency to easily isolate an actuator leak that could compromise flight safety and reliability. Similarly, it must be ensured that a critical failure of the electromechanical actuator can be readily identified and isolated. However, in the former case a possible fault can be warned before a load is required from the user, by virtue of its continuous operation; conversely, this is not the case for the electromechanical system. It follows that occurrence of a fault in the electrical system is risky and introduces a new safety problem, because no preventive corrective action can be taken to mitigate the effect of the fault itself if no additional auxiliary system is provided.

For this purpose, it is necessary to design an appropriate control system that performs two main functions: recognition and removal. The first function encompasses the two sub-functions of detection, diagnosis and prognosis [8].

The following describes the intrinsic characteristics of the EMA that make prevention system necessary.

The safety of the EMA is often tied to the reliability of the motor and electrical component. In particular, possible electromagnetic and electronic failures of an actuator include open circuit of windings or power devices, closed circuit of windings, motor terminals or power devices, failures of DC link capacitor and sensors malfunction. In order to detect the fault, and therefore make a diagnosis in real time, the power modules must have a high short-circuit capacity. Furthermore, it is necessary that the two functions of fault isolation and fault identification can be implemented. In fact, if one of these malfunctions occurs, it is necessary that an ongoing fault does not affect the operations of adjacent phases, so that the service is always ensured without interruption [4].

For this reason, it is essential to have devices capable of stopping the flow of current in the event of a short circuit and a configuration of the system able to maintain the functionality in case of fault. Therefore, it is important that these modules can be replaced individually. Moreover, a modular and distributed control system is needed to meet these requirements. The electrical machine must have a high efficiency during operation, high winding inductance to reduce short-circuit currents in the coils, high torque for the same current to reduce losses and finally have insulation between phases to prevent fault propagation and reduce thermal transmission. In addition, attention must be paid to electronic devices already installed on the aircraft, which induce electromagnetic compatibility (EMC) problems for the electromechanical actuator.

This system is supported by the sensors that provide important informations on the actuator's health status by giving out values of specific measured quantities, thus supporting the fault identification function. Among the sensors commonly used on an electromechanical actuator we find: phase current sensors to analyze the signal to detect faults in the windings and in the mechanics; DC voltage sensors to obtain information on possible short circuits and capacitor overvoltages; sensors at the output of the capacitor to verify the correct power modulation and detect faults in the power switches or motor phases; position or rate sensors to close the control loop. The latter are also useful for detecting mechanical or electromagnetic faults.
Unfortunately, sensors are also prone to faults. It may happen that the failure is detected and reported via the communication bus. If the fault is not detected the value provided by the sensor is incorrect, so a malfunction occurs. In this case it is necessary to evaluate which sensor is not working and, if necessary and possible, proceed with replacement [4].

In addition to electrical failures, mechanical faults must also be considered. In fact, the loads continuously applied on the actuator induce fatigue and wear of the structures that can produce problems of mechanical backlash or eccentricity of the motor rotor that, in general, lead to a deterioration of the performance of the whole system. This type of failure must also be properly detected.

Currently in the aviation industry, it is necessary to ensure the detection of at least 99% of the failure cases that occur . This means increasing the reliability of the actuator by applying a *fault-tolerant approach* to the system. Fault tolerance must be applied appropriately to each device in the system so that it, as a whole, can be as efficient and safe as possible.

To understand how this approach is generally applied, examples are given for different parts of the system.
For instance, the motor in use in aviation can in most cases be a *Switched Reluctance Motor* (SRM) or *Permanent Magnet* (PM) motor. The former is inherently fault-tolerant but is not suitable for the electromechanical actuator, for which reduced weight and higher efficiency are desirable. The latter, on the other hand, is efficient over a wide rate range, has a high power ratio, and allows for easy cooling.
Another type of choice, conversely, is made for power electronics, for which the fault tolerance approach is applied through the choice of appropriate redundancies. In this case a multiphase

drive is applied so that each phase can be considered as a single part, which has only minimal influence on the others that continue to operate normally. Doubling the power electronics allows the nominal voltage to be reduced, resulting in lower inverter losses, lower demands on the dissipator and, unexpectedly, a reduction in weight. In addition, redundant components generally only operate when a fault occurs in the main components.
Sensors also make use of redundancy, at least in part. In this way, in fact, permanent failures can be managed by the duplicated hardware components, while temporary ones can still be detected, analyzed and processed by properly designed software [4].

It is clear that it is not possible to perform the fault diagnosis method through the simple use of redundancies for the whole system. In fact, hardware redundancy is often expensive and requires volumes and weights that are not always available or possible on board.

From these examples, we can see that the system of an electromechanical actuator is composed of several interconnected hardware and software components, which make the system itself critical and complex to monitor and possibly protect. In fact, it is not enough to analyse the components individually, but it is necessary to study their cross interactions so that the control system can correctly perform its function of monitoring, fault detection and calculation of a possible divergence path.

For this reason, appropriate monitoring technology must be added to a fault-tolerant system to increase its reliability.
This is where prognostics comes in. Prognostics allows faults to be identified in their early stage, before they induce catastrophic consequences through their propagation. The two main functions of prognosis are, in fact: prediction, i.e. predicting the effect of a failure and studying its possible propagation at several levels in the operative environment; extrapolation, i.e. predicting the trend of a fault over time [8]. It allows, therefore, to plan maintenance interventions before the actuator, or the aircraft in general, completely loses its functionality.

The figure 1.4 is a summary diagram of the control system functions specified above.

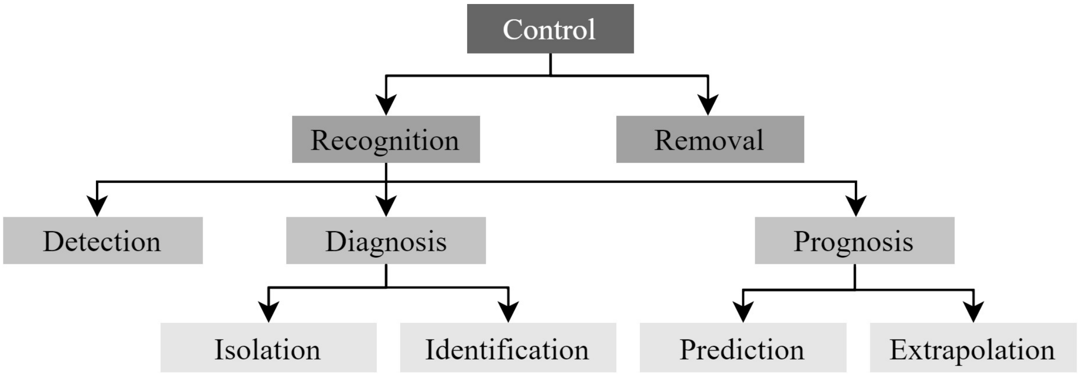


Figure 1.4: *Control functions*

## 1.4 Prognostics as a solution

Being able to monitor the system before a fatal failure occurs is crucial to ensure greater system reliability. Indeed, a malfunction of an electromechanical actuator can induce a mechanical jam, which could be fatal in terms of safety. However, through prognostics a failure can be detected quite early, increasing the reliability of the engine and electronics, it may be possible to use electromechanical actuator technology also on primary flight controls.

In fact, prognostics is based on the observation of changes in operating parameters of a given system during its normal operating cycle. The objective is to predict an incipient failure before it causes a shutdown that would have disastrous conseguences on the production cycle and the integrity of the system itself. In other words, it must be ensured that the components always conform to the required performance and specifications [9].

Until now, this need has been met by standard maintenance procedures, i.e., so-called *predictive maintenance*. Initially, that is, the machine was physically inspected, when not in operation, at regular time intervals. This method, however, cannot handle faults that occur before the end of the operating cycle, thus before the periodic inspection, risking to induce the loss of the component or of the whole aircraft. Conversely, if premature failures do not occur, intervention is often not justified by the actual conditions of the system, since maintenance actions are performed following the definition of very stringent reliability requirements [10], thus the costs are higher than necessary. In addition, this activity also implies a decrease in the readiness of the different components during the mission, thus generally requiring some non-use time of the aircraft, implying further costs.

Another solution adopted was to opt for the addition of redundancies in order to increase the reliability of the system, as seen in the previous paragraph. This solution, apart from not being the most economical, also complicates control operations and it is not a solution that can be adopted for all system components.

Current health monitoring techniques make an accurate diagnosis of the actual state of the component, allowing *Condition Based Maintenance* (CBM), through which maintenance plans are organised according to the existing condition. The evolution of this diagnostic system tends to continue towards the prognostics systems of our interest, which are able to predict the degradation of the component starting from the current health state of the system continuously updated, thus allowing the definition of advanced maintenance plans and the definition of the useful life of the different components [9]. This is why we speak of *Prognostic & Health Monitoring* (PHM) Systems.

A more advanced stage of this technology could be the one constituted by the *proactive prognostics*, which has the objective, not only to foresee the evolution of a failure, but also to suggest to the pilot actions or maneuvers that, in the long run, may lead to a reduction of the damage induced by a specific failure.

The operations of a prognostic activity can be summarized as follows:

- identification of a failure case history, i.e. how the failure may occur and how it affects the production process (statistical analysis and FMECA);
- determination of a failure evidence through the study of significant signals or their combination with respect to an incipient failure: measurements and acquisitions are carried out and a data base is compiled to highlight temporal divergences of faults;
- development of algorithms, i.e. elaboration of the acquired signals to reveal significant variations from a normal condition to a fault condition and, if necessary, report them;
- prediction of *Remaining Useful Life* (RUL).

### 1.4.1 Model-based prognostics

In this work we discuss a model-based approach to prognostics. It is based on an analytical and dynamic model consisting of algebraic or differential equations that are able to represent the real behavior of the analyzed system, including degradation phenomena, in a simulation environment.

This environment, in our case Matlab-Simulink, is a kind of virtual test bench that allows the input and analysis of faults to direct the development of a specific implemented algorithm, which represents the system degradation and its effects on the performance of the actuator components. In particular, the parameters of the system are estimated through the comparison between the actual response simulated by the control command and the response provided by the algorithm [9].

A typical procedure for the model-based approach is as follows [9]:

- automatic analysis of faults and their propagation;
- design and verification of the PHM algorithm(s);
- dynamic simulations;
- code generation;
- validation tests (not performed during this thesis work).

There are several advantages of using the model-based approach. First of all, they are easy to understand because the mathematical relationships are closely related to the physical mechanisms that affect the health of the system. In addition, the failure modes are correlated to significant physical parameters of the system, which provide the maintainer with immediate diagnostic information. Finally, the presence of a model is useful when there is no off-the-shelf data and it is not possible to collect it, but also when the physical system is not present.
On the other hand, one has to consider the difficulty of generating an accurate model of a system characterized by complex physical behaviors under both nominal and non nominal operation. Furthermore, the model can be used and be deemed accurate to the physical system if the operating condition considered reflects the specific conditions for which it

was created; therefore more experimentation may be required, which can be costly and time consuming. Finally, it must be taken into account that the model must be suitably simplified so that it can provide results in real time and be useful, therefore, for prognostic and diagnostic purposes.

For example, during this work, as we will see in the chapter 2, we first consider a high-fidelity electromechanical actuator model, then, through appropriate and researched simplifications, we obtain a low-fidelity but computationally faster model.

In the next paragraph we will describe in a general way the type of algorithms that have been implemented to support the development of the prognostics software.

## 1.5 Methaeuristic bio-inspired algorithms

In general, the optimization process is a kind of computational manipulation of a structure to achieve specific goals. Such a system is expressed through mathematical functions and the goal is to find, depending on the field of application, the minimum or maximum. These functions are called *objective functions* and are described by specific variables called *decision variables*, limited by specific constraints [11].

For the resolution of this problem the application of exact methods is not always possible for two main reasons: such methods take unreasonable computation time to generate a solution or the system is too complex. Because of this, an alternative way is reached through *heuristic methods* that involve two apparently opposite characteristics: the empirical intuition related to a specific context and the rigor and systematicity of an algorithm capable of verifying that the solution is close to the one sought or providing a stopping criterion[12].
The presence of a stochastic component implies that these methods rely mostly on random numbers, so if the algorithm is run multiple times, the result can be, to certain extent, different. In contrast, an algorithm is deterministic [11].

In particular, the task of heuristic methods is to generate good solutions, i.e. close to the optimum, at a reasonable computational cost but they are unable to guarantee the optimum itself [13]. This approach, for example, is suitable for model-based prognostics. In fact, many of the parameters involved are determined from assumptions on which depend estimates or approximate values inevitably affected by error. Consequently, it is not efficient to use an algorithm with high computational cost to find the exact optimum of an approximate solution.

Heuristic methods can be classified into three main categories [12]:

- *constructive heuristics*: starts from an empty set to search for at least one admissible solution;

- *ameliorative heuristics*: attempts to improve an admissible solution;

- *metaheuristics*: uses several procedures to obtain good solutions to optimization problems within a large set of admissible solutions.

The method that will be considered here is the metaheuristic method. This must have components and their respective interactions defined in order to obtain a good solution for that specific problem. In fact, metaheuristics is based on local search applied to a particular field of application, among many possible ones. What allows the good fitting of the method to the problem under consideration is the correct selection of the type of optimization algorithm.

The algorithms that will be considered here do not have primary mathematical origins but have biological ones.
The interest in nature is related to the fact that it is mainly based on the phenomenon of *adaptation* that can be interpreted as a form of optimization.
Adaptation is defined as a gradual change in properties and behaviors in response to changes in the environment or surrounding conditions. Biological organisms also, in some cases, exhibit the ability to cooperate with each other.
The goal of bio-inspired algorithms, however, is not to reproduce biological behavior but to select relevant aspects of it for each specific problem to be solved. Some algorithms, in fact, tend to exasperate the analogy with biological processes, some eliminate some features, others add properties that are not pertinent to the phenomenon being studied [11].

Here two types of algorithms will be treated: the evolutionary and swarm intelligence ones. This type of approach is becoming increasingly popular in the development of new optimization techniques because they are based on simple ideas, consequently they are easy to implement, and at the same time they succeed in approaching complex problems.

### 1.5.1 Evolutionary Algorithms (EA)

Evolutionary algorithms take their name phenomena of biological evolution, which is where they owe their terminology.

In fact, one must distinguish between natural and artificial evolution. The former, as we have seen, has only the purpose of long-term adaptation and has no second goal. Vice versa, the second one performs a specific optimization process suitable for a particular problem.
It follows that an individual in the first case is considered to be as good as its ability to reproduce; in the second one, it is considered valid an individual who is able to solve better than others a specific objective. For this reason, as much as an algorithm can be implemented to ensure diversity among individuals through the introduction of elements of randomness, certainly the biodiversity present in nature can not be compared in artificial evolution algorithms, because all solutions will aim to ensure the resolution of a particular problem.

The distinguishing features to evolutionary algorithms, which directly conform to the properties of natural evolution, are:

- *population*: there can be no evolution with only one organism: in fact, during the evolution is not the individual to change but the population; it is initialized at the beginning of the cycle;

- *variety*: the different characteristics of individuals ensure adaptation to external conditions and environment and allows to discover new features or behaviors of the species; this ensures that the search for solutions does not rely solely on stochastic effects;

- *heredity*: ability to pass on a trait to their offspring in order to confer better coping skills than their peers;

- *selection*: not all of the population is able to procreate: the best elements of the population will have larger offspring, the less suitable or those who possess undesirable characteristics have low or no probability of reproduction.

We need to pay attention to the variety property. One must make sure that there are no individuals that have too high a fitness score compared to the rest of the population. In fact, the algorithm would tend to evolve only such individuals, without considering other solutions, and in practice would prevent the same requirement of variety. In this case, moreover, the algorithm would tend to converge too early, probably without having explored other valid genotypes.
Variety is held during algorithm execution through the process of *crossover*, which allows recombination of parental features, or through random *mutation* of the genotype to look for solutions not yet explored. Mutation prevents local minima and is especially useful when the population is strongly convergent so that genetic recombination has limited effect.

With regard to selection property, the ability of a species to be selected in nature is influenced exclusively by the environment in which individuals survive, so it is not said that future generations are better than the parents because the boundary conditions are different. On the contrary, for artificial algorithms selection must occur only in the desired direction, that is, towards the minimum or maximum condition. This implies, once again, the necessary presence of a final goal or optimal state of a system.

At the computational level, the objective to be pursued during the execution of the algorithm is determined by the so-called *fitness function*. The individuals represent the various solutions to which a score called *fitness value* is attributed, determined by evaluating the phenotype of the subjects in question. In addition, a reproduction operator selects the phenotypes with higher score and generates a determined number of copies of the corresponding genotypes. Some of these genotypes are then modified and reintroduced into the initial population. These steps are performed iteratively until either a solution near the minimum or maximum point is reached or the stopping condition occurs .

In the figure 1.5 below it is represented the generic diagram of flow of an evolutionary algorithm. In chapter 4 each will be described in detail.

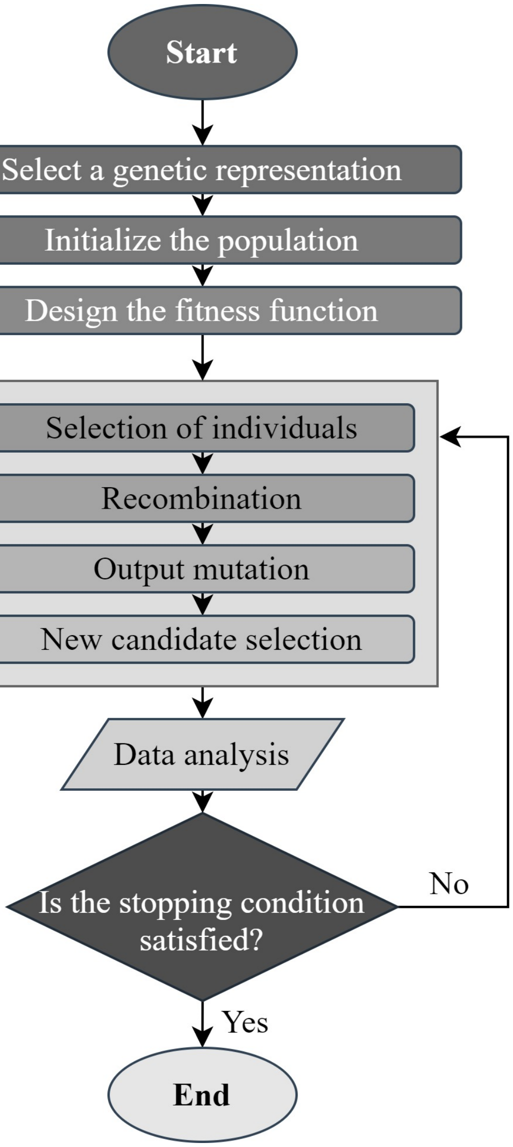

Start
Select a genetic representation
Initialize the population
Design the fitness function
Selection of individuals
Recombination
Output mutation
New candidate selection
Data analysis
Is the stopping condition satisfied?
No
Yes
End

Figure 1.5: *Evolutive Algorithm flow-chart*

Evolutionary algorithms can be applied to many problems as long as a consistent genetic representation can be defined. In fact, first it is necessary to define one or more discrete neighbourhoods so that a particular subset of genotypes can be associated with each solution. These neighbourhoods must be appropriately selected depending on the problem and the goals to be achieved. For example, small neighbourhoods allow for faster solution convergence, but discrete local minima result in worse solutions. Indeed, on the one hand it must be considered that, in most cases, the algorithm must examine all the admissible solutions in order to select the best one; on the other hand, not always small neighbourhoods include a sufficient number of admissible solutions so that the calculation could be concluded in an insufficient time. For example, if most individuals have a similar fitness value, the mutations and variations that may occur have a very low probability of producing an improvement, so a larger initial population is needed [14][15].

The evolutionary algorithms implemented and applied to the electromechanical actuator model are the *Genetic Algorithm* (GA) and *Differential Evolution* (DE). The latter belongs to the subgroup of *Evolutionary Strategies* (ES).

### Genetic Algorithms (GA)

Genetic algorithm is a combinatorial optimization technique based on search, natural selection and the principles of genetics. The individuals on which the algorithm operates consist of binary representations (strings of 0s and 1s).

In this case, the set of possible solutions is called the *search space*, in which there is a defined set of solutions to the problem, the *chromosomes*, constituting the *population.* Each chromosome consists of the randomly arranged *genes*, which correspond to the individual values of the string and identify the decision variables of the problem. The variation of a gene determines a different chromosome, consequently a different solution. Finally, each gene corresponds to an *allele*, which is the value defined by a gene.

A graphical representation of the constituent parts of the genetic algorithm is shown in the figure 1.6.

The algorithm starts with an initial population of randomly generated solutions that evolve iteratively. At each iteration these are evaluated by a fitness function, i.e. the quantitative measure of an individual's fitness and, consequently, chosen. Those selected, are then recombined to generate new solutions, the offspring, which tend to transmit the good characteristics of the parent solutions in subsequent generations. Recombination of parents is referred to as *crossover* and is often associated with *mutation* (in which case the process of recombination and mutation is referred to as *mating*), which is the random modification of certain genes that prevents premature population convergence and similarity between individuals.

The iteration ends when the arrest condition is satisfied, thus the best generated solution is provided.
Possible stopping criteria can be:

- maximum number of iterations;
- limit on the execution time;
- maximum number of iterations reached without achieving improvement;
- low variance of fitness values between chromosomes.

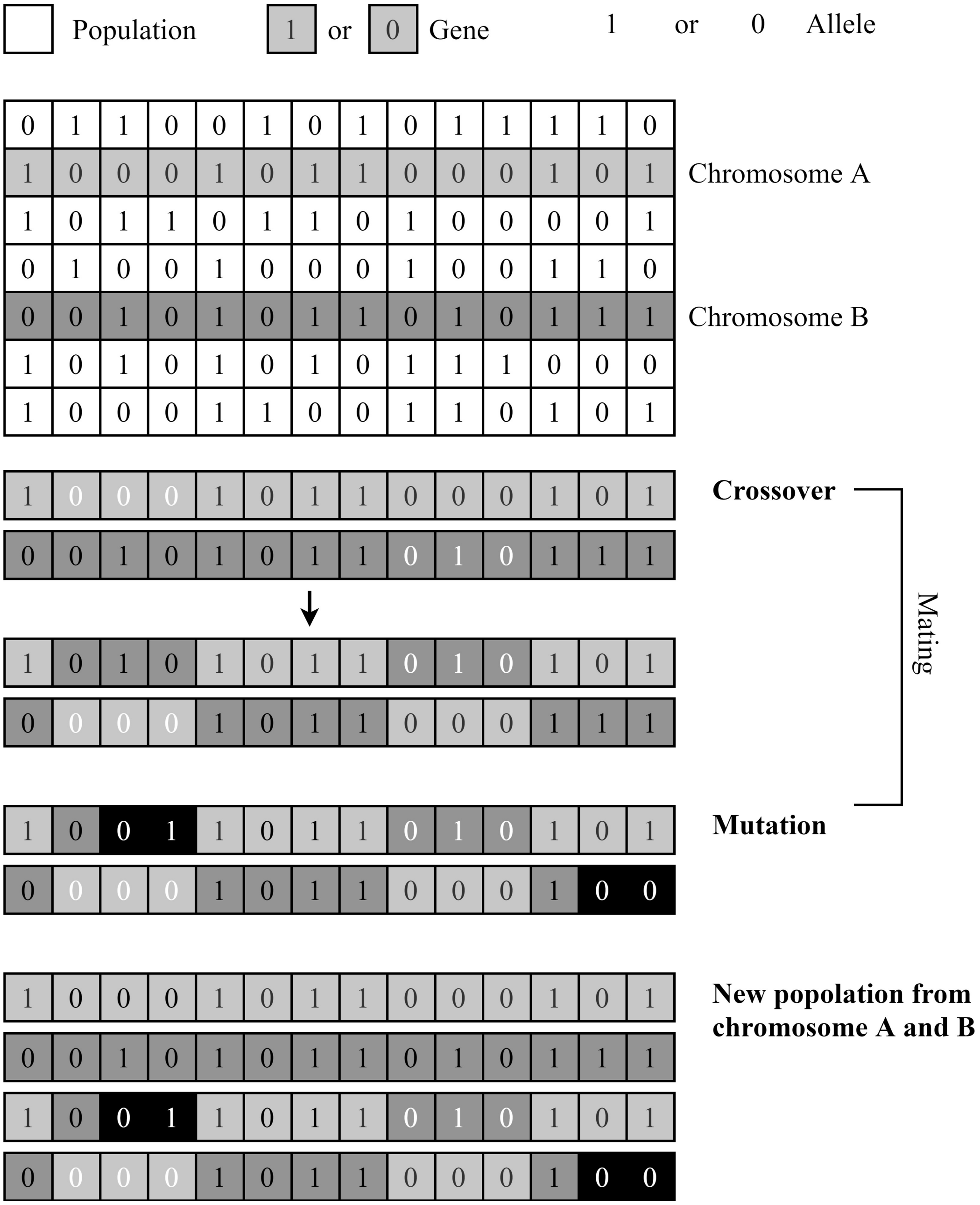


Figure 1.6: *Genetic algorithm scheme*

### Differential Evolution (DE)

The algorithm of Differential Evolution is a method of solving a mathematical process of optimization based on the concept of a differential vector that has the task of perturbing the initial population vector in order to obtain new species that improve the fitness value.

In particular, there is a single vector related to the whole population, afferent to the single

first generation.
The development of the algorithm begins through the process of *mutation*. This occurs when two individuals, then two parts of the population vector, are compared by difference, and the resulting gradient is added to a third vector. At this point the *recombination* or *crossover phase* begins: the mutated parameters of the initial vector are mixed with those of the so-called *target vector* to form the *trial vector*. Through the selection process the trial and target vectors are compared. The vector with the best fitness value will remain in the future generation: for instance, if the trial has a higher score than the target, it will be the future target, otherwise it is simply discarded.
In this case the stopping criterion is defined by the algorithm itself as the described procedure is performed for each vector belonging to the first generation [12].

A graphical representation of the steps of the differential evolution algorithm is shown in the figure 1.7.

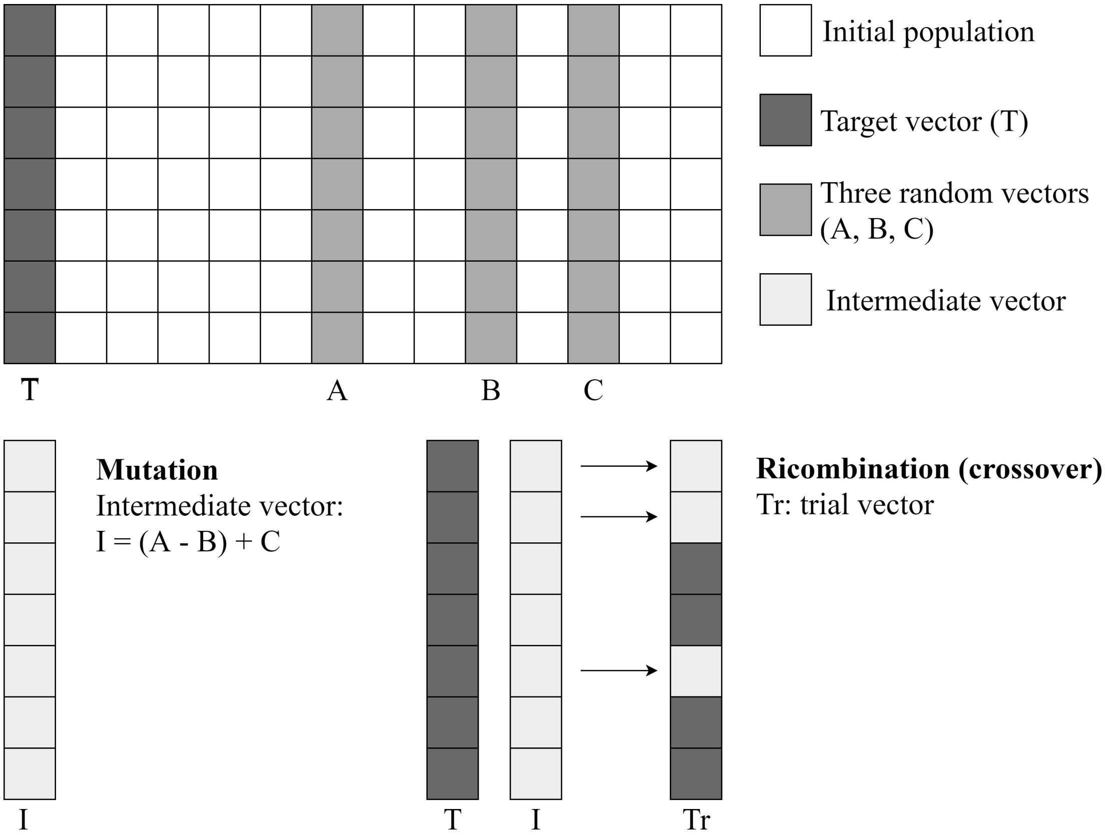


Figure 1.7: *Differential Evolution scheme*

### 1.5.2 Swarm Intelligence algorithms (SI)

Swarm intelligence refers to the intelligent behavior of biological swarms through the interaction of two or more individuals in certain environments. These methods are used to find a solution

to real problems through simulation of biological behaviors.
These collective phenomena have an adaptive function that would not be satisfiable if individuals operated alone.

The main features of this type of algorithm are *self-organization* and *division of labor*. In fact, the interactions of individuals generate a higher level structure. This is characterized by a complexity and an improvement in the general capabilities of perception and action, resulting in an increase in efficiency, which an individual alone could never achieve. For example, in nature everyone in the group is individually responsible for a specific task in order to achieve a specific common goal such as searching for food, defense, hunting, leadership, etc. This type of phenomenon is not the result of a genetic process, but a collective one, driven by interaction.

Another important feature of collective intelligence is the ability to voluntarily or involuntarily provide *feedback* or signal to individuals in the group, so as to signal the increased presence of food or the presence of danger by adjusting the intensity of the signal itself. When the signal is involuntary it is called a *cue*; an example of this is the paw mark left on the ground. In this case it is up to the observer to determine whether or not to follow the cue. A *signal*, on the other hand, is left voluntarily by individuals with the goal that it be perceived and followed by the other components.

In order that the optimal solution to a specific problem is found, it is necessary that the algorithm is opportunely selected, therefore the collective phenomenon suitable to explain the dynamics of a specific problem is chosen [16].

Two algorithms inspired by collective phenomena will be described below: *Particle Swarm Optimisation* (PSO) and Gray Wolf Optimisation (GWO).

### Particle swarm (PSO)

This optimisation method is derived from the flocking of birds in search of food. When a bird finds insects in a specific region, it emits a signal whose intensity is proportional to the amount of food found. An individual bird then moves towards the nearest companion, which emits a sound of greater intensity. In this case, birds are defined as generic particles, each with specific position and properties, which collectively move towards a point of optimum.

The initial position of the particles within a parametric space is random; these are updated by taking into account three displacements: in the current direction, in the direction towards the loudest signal, and in the direction of the nearest companion with the highest immediate signal. The final position is the resultant of these three shifts and it represents the level of adaptation to the specific objective [16]. Possible uncertainties should be taken into account, if necessary.

For purely descriptive purposes, a generic function $F(x)$ and the three displacements described are shown in the figure 1.8.

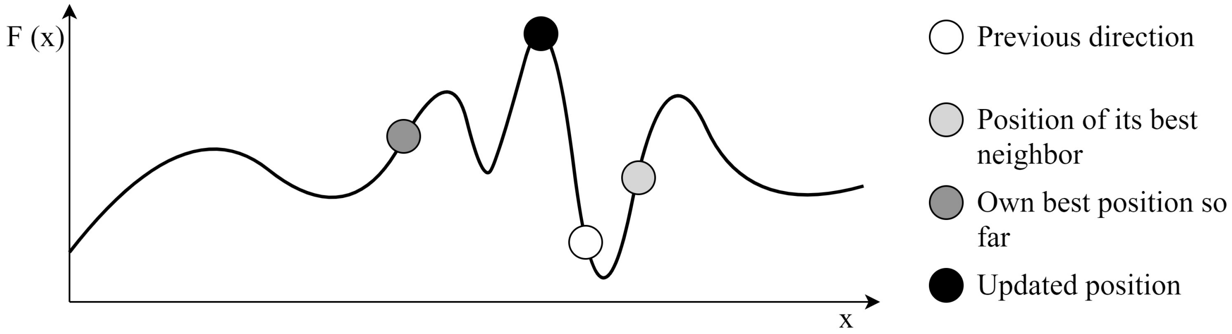


Figure 1.8: *Generic function and the three displacements of the particles*

It is interesting to note that each individual can update their position at the same time or at different times. In addition, each individual particle is able to determine the goodness of its current position based on both the space it has explored and the space known to the other particles, thanks to the sharing principle.

There is still a stochastic component to the velocities as each particle is able to explore unknown areas. This allows extensive investigation of the parametric space and enables the best solutions to be found efficiently. It does not guarantee the absolute best solution, but it ensures that the global optimum solution is approached without running into local optimum solutions [12].

As in the case of evolutionary algorithms particle swarm optimisation, one has an initial population of individuals to refer to. However, whereas in the first case the selection of individuals was the result of a genetic solution, in this case the result is provided by updating the position and velocity of each individual within a collective behaviour [16].
However, it is not enough to define these two variable parameters, as constants must also be determined to represent the social and cognitive components that do not change during the process and ensure that the algorithm itself works well.

This method is capable of optimising non-linear and multi-dimensional problems and achieving good solutions with minimal parameterisation.

A generic flowchart for a particle swarm optimisation algorithm is shown in the figure 1.9 and a diagram representing the initial and final state of the particle group iteration is shown in the figure 1.10.

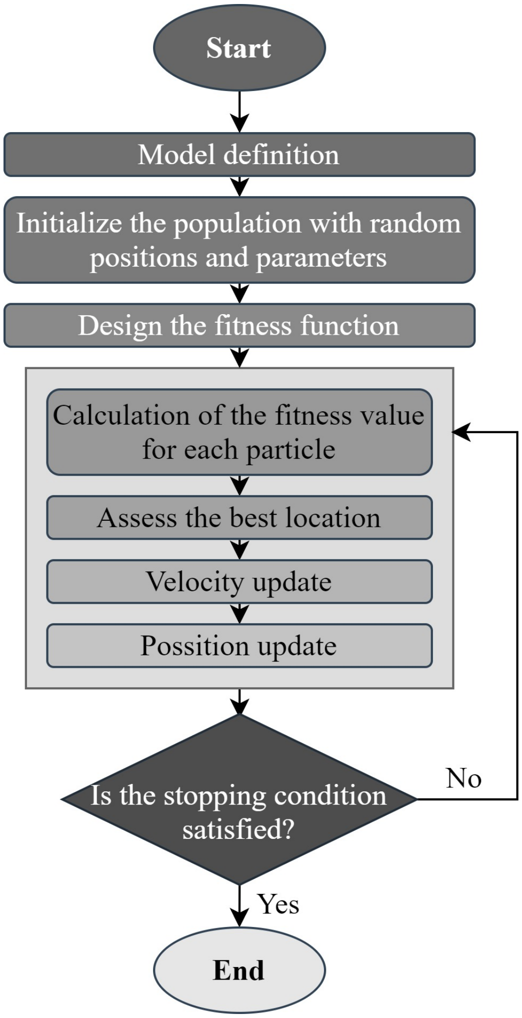

Start
Model definition
Initialize the population with random positions and parameters
Design the fitness function
Calculation of the fitness value for each particle
Assess the best location
Velocity update
Possition update
Is the stopping condition satisfied?
No
Yes
End

Figure 1.9: *Typical PSO flowchart*

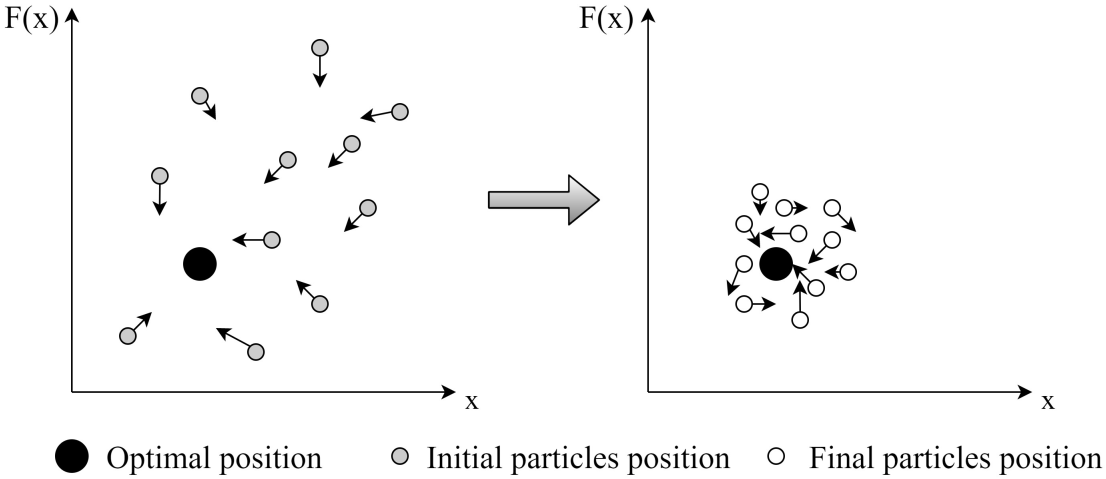


Figure 1.10: *Generic initial and final particles condition*

### Gray Wolf Optimisation (GWO)

The grey wolf optimisation method is inspired by the hierarchy and leadership typical of these animals, which manifests itself, in especial, during the prey hunting phase. In particular, the pack is divided into the following categories of wolves, in order of importance:

- $\alpha$: leaders of the group (generally a male and a female), they give orders and controls the subordinate categories;
- $\beta$: they assist the $\alpha$ in decision-making and give orders to the $\delta$ and $\omega$ categories; they also provide information on other wolves; when the $\alpha$ wolves die or become old, the $\beta$ become $\alpha$; subordinate categories;
- $\delta$: they determine pack action strategies based on commands received and pass information to higher wolf categories;
- $\omega$: the rest of the wolves obey orders and act.

In the algorithm, each wolf is a solution to a specific problem. In particular, to each category of wolves corresponds a different value of fitness of the optimisation algorithm, respectively the highest assigned to the $\alpha$ wolves and the lowest to the $\omega$ wolves.
Each subgroup has a specific function and fulfils a specific hunting phase:

- identifying and tracking prey ($\omega$ wolves): in nature this is done through visual and olfactory sensory perceptions; in the algorithm this step is performed by the omega wolves which calculate the distance between the solutions and the prey, determining the hierarchy of $\alpha$, $\beta$ and $\delta$ wolves according to their proximity to the prey itself;
- circling the prey ($\alpha$, $\beta$ and $\delta$ wolves): the position of the wolves is updated in the direction closest to the $\alpha$ wolf, which is closest to the catch, thus to the best solution;
- striking and target the prey ($\alpha$ wolves): determines the ability of the $\alpha$ wolf to move towards prey.

The subdivision into categories and the determination of a strongly hierarchical scale translates on the algorithm level into greater proximity to the prey, i.e. the desired solution [17].

From a computational point of view, it is important to define the number of wolves belonging to the pack. In fact, the accuracy and execution time of the algorithm depend on this factor. However, we have to consider that, unlike what we have seen for the PSO, in this case the different individuals of the population do not have the possibility to communicate their position to the other members, so we could explore again solutions already discarded and this implies a small variance among the different individuals that could lead to the quick convergence of the solution. For this reason, an appropriate strategy of population initialisation and search for the pseudo-optimal solution must be selected [18].
Again, as this optimisation strategy may incur local minima, the ability to explore or store positions between individuals should be improved in some way.

A generic flowchart for a GWO algorithm is shown in the figure 1.11 and a diagram of the updated position of the wolves is shown in the figure 1.12.

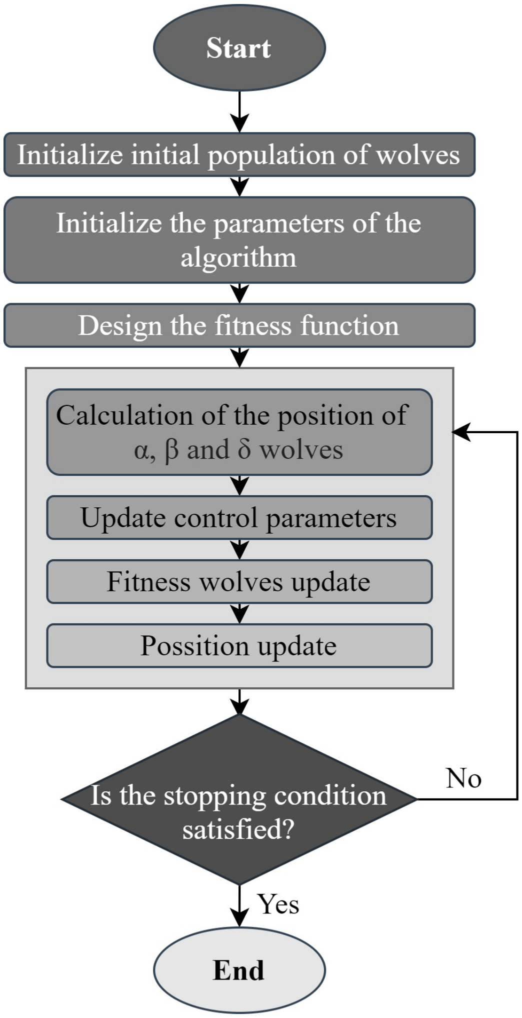

Start
Initialize initial population of wolves
Initialize the parameters of the algorithm
Design the fitness function
Calculation of the position of α, β and δ wolves
Update control parameters
Fitness wolves update
Possition update
Is the stopping condition satisfied?
No
Yes
End

Figure 1.11: *GWO algorithm flowchart*

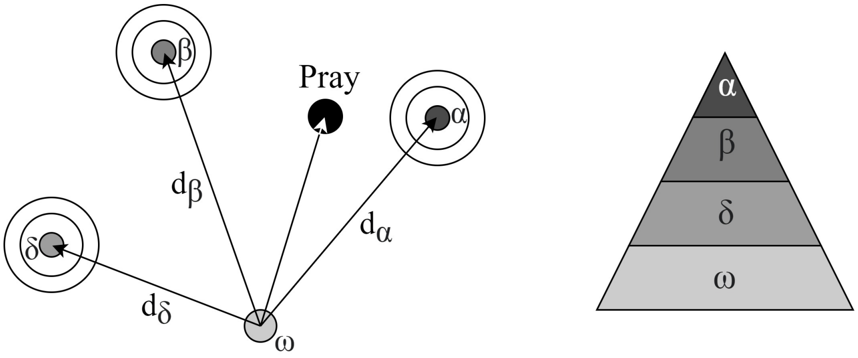


Figure 1.12: *GWO update position scheme and hierarchy*

# Chapter 2

# Electro-mechanical actuator model

Model-based prognostics, as the name implies, is based on the definition and elaboration of a model. This is centred on physics which is able to describe by means of parameters not only the dynamics of the system, but also the recognition and prediction of different failure modes. The model must be appropriately defined for the system to be monitored and must simulate the response of the system when it receives a command [10].

In particular, two different fidelity models are used for this purpose: *high fidelity* (HF) and *low fidelity* (LW). Both have been implemented on Matlab-Simulink.
The high-fidelity model is the *reference model* that simulates the real behaviour of an electromechanical system and, de-facto, constitutes a virtual test bench. It is characterised by high performance so that the modelled physics is accurate on temporal and spatial scales. The behaviour of the individual components of the EMA is described by appropriate governing equations which are integrated into the modular elements of the system (Simulink blocks) which can be evaluated individually. The results produced must be comparable with those that would be obtained in a real context in which the actuator will be operating. In fact, the objective of the high-fidelity model is to verify the correspondence of the simulated phenomena with the real ones and to reproduce their dynamics. The high accuracy of this model is generally obtained with small integration steps, but this is paid for in terms of computational effort, hence in the time needed to perform the simulation and, consequently, in costs. Additionally, this issue could limit the amount of usable data over time, as it requires more computer memory.
The low fidelity model is called, instead, *monitor model* which makes use of approximations of the parameters constituting the physics of the system in order to reduce the computational cost and allow a real time evaluation of the health status of the real system. In this case, exhaustive modelling of the system is dispensed with, but it must be able to accurately reflect the different failure modes.

Using both models one is able to obtain maximum model accuracy while minimising the cost associated with parameterisation, increasing the possibility of obtaining valid estimates [19][20].

The sections 2.1 e 2.2 below describe these two models in detail.

## 2.1 Reference Model

A high-level introductory diagram of the high-fidelity model is shown in the figure 2.1. It is characterised by four main blocks [20]:

- *controller*: a PID controller compares the position signal *Com* with the position and angular speed user sensors by means of a feedback loop and it outputs the reference current $I_{REF}$ to be managed by the inverter;
- *inverter*: it translates the reference current $I_{REF}$ into the tripartite current and voltage between the active motor phases and performs the corresponding PWM modulation, trying to leave the input current unchanged;
- *sinusoidal BLDC motor*: it receives as input position $\theta$ and angular velocity $\dot{\theta}$ updated iteratively by the dynamics module and supplies the required motor torque $T_M$; the model is described taking into account electromagnetic interactions between stator and rotor;
- *motor-transmission dynamics*: it receives as input the motor torque and the resistant torque $T_R$ and calculates the dynamics of the motor-transmission module instant by instant in terms of angular velocity $\dot{\theta}$ and position $\theta$; the model was obtained by taking into account second-order dynamics and multiple non-linearities, which are described in the following.

Simulink's high-fidelity model also includes a block, not shown in the figure, specifically for managing the currents of the individual motor phases in order to generate a single equivalent current which will then be useful, as we shall see, in defining the monitor model.

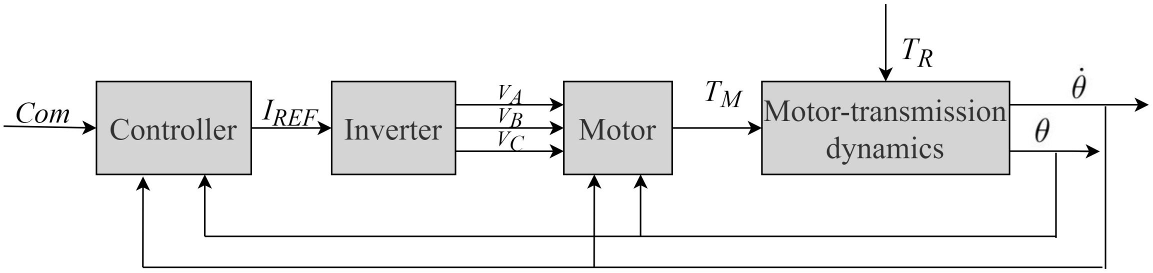


Figure 2.1: *Schematic of the general high-fidelity model*

In the following paragraphs these parameters and blocks will be explained in detail.

The table 2.1 shows the initial reference data characteristic of each main component of the high-fidelity model.

Table 2.1: *EMA systems parameters - High Fidelity Model*

| **Controller** | | | |
|---|---|---|---|
| *Parameter* | *Symbol* | *Value* | *Units of measurement* |
| Error proportional gain | $G_{prop}$ | $1 \cdot 10^5$ | $\frac{1}{s}$ |
| PID proportional gain | $K_P$ | $5 \cdot 10^{-2}$ | $\frac{N \cdot m \cdot s}{rad}$ |
| PID integrative gain | $K_I$ | 0 | $\frac{N \cdot m}{rad}$ |
| PID derivative gain | $K_D$ | 0 | $\frac{N \cdot m \cdots^2}{rad}$ |
| Maximum current | $I_{max}$ | 22.5 | $A$ |
| **Inverter** | | | |
| *Parameter* | *Symbol* | *Value* | *Units of measurement* |
| Maximum supply voltage (DC) | $V_{max}$ | 48 | $V$ |
| **Motor** | | | |
| *Parameter* | *Symbol* | *Value* | *Units of measurement* |
| Maximum motor torque | $T_{m,max}$ | 1.689 | $N \cdot m$ |
| Counter-electromotive force constant | $k_{cemf}$ | $3.76 \cdot 10^{-2}$ | $\frac{N \cdot m}{A}$ |
| Nominal phase-to-phase resistance | $R_S$ | 2.13 | $\Omega$ |
| Phase-to-phase inductance | $L_S$ | $7.2 \cdot 10^{-4}$ | $H$ |
| Number of pole pairs per phase | $N_P$ | 2 | - |
| Nominal rotor static eccentricity | $\zeta$ | 0.5 | - |
| Nominal rotor static eccentricity phase | $\phi$ | 1 | - |
| **Motor-Transmission dynamics** | | | |
| *Parameter* | *Symbol* | *Value* | *Units of measurement* |
| Moment of inertia of the motor | $J_m$ | $1.3 \cdot 10^{-5}$ | $kg \cdot m^2$ |
| Motor viscous damping coefficient | $C_m$ | $9.549 \cdot 10^{-6}$ | $\frac{N \cdot m \cdot s}{rad}$ |
| User viscous damping coefficient | $C_u$ | $4.507 \cdot 10^{-7}$ | $\frac{N \cdot m \cdot s}{rad}$ |
| Static motor friction | $f_{sm}$ | $0.06 \cdot T_{m,max}$ | $N \cdot m$ |
| Dynamic motor friction | $f_{dm}$ | $0.03 \cdot T_{m,max}$ | $N \cdot m$ |
| Nominal backlash | $BLK$ | $1 \cdot 10^{-5}$ | $rad$ |

### 2.1.1 Command block

In Simulink, the command is given to the control system by means of a block that allows the user to choose the type of input to be submitted to the actuator.
The inputs that can be implemented are shown in the table 2.2 ; the settings with which the simulations were carried out are also shown.

Table 2.2: *Command settings - High Fidelity Model*

| **Step command** | | |
|---|---|---|
| *Parameter* | *Value* | *Units of measurement* |
| Initial amplitude | 0 | $rad$ |
| Final amplitude | 1 | $rad$ |
| Instant of application | 0.01 | $s$ |
| **Ramp command** | | |
| *Parameter* | *Value* | *Units of measurement* |
| Ramp gradient | $1 \cdot 10^{-3}$ | $\frac{rad}{s}$ |
| Initial ramp output | 0 | $rad$ |
| Instant of application | 0 | $s$ |
| **Sine command** | | |
| *Parameter* | *Value* | *Units of measurement* |
| Semi-amplitude input | $5 \cdot 10^{-3}$ | $rad$ |
| Bias input | 0 | $rad$ |
| Input frequency | $15 \cdot 2\pi$ | $\frac{rad}{s}$ |
| **Chirp command** | | |
| *Parameter* | *Value* | *Units of measurement* |
| Initial amplitude | $5 \cdot 10^{-3}$ | $rad$ |
| Initial frequency | 0 | $Hz$ |
| Target frequency | 15 | $Hz$ |
| Instant when the target frequency is reached | 0.5 | $s$ |

In addition to these commands, it is possible to issue the command set directly by the user or to run the simulation on a free system, without any command implemented.
The Simulink *Com*-block interface is shown in the figure 2.2.

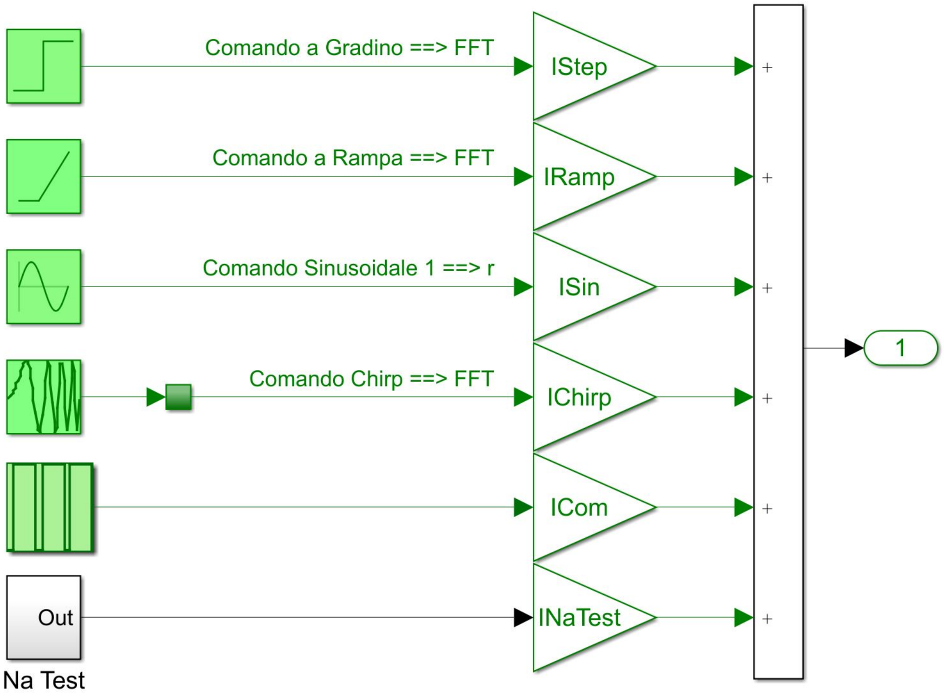


Figure 2.2: *Command Block Simulink*

### 2.1.2 Load (Resistant Torque) block

The load is implemented in a similar way to the command block. The three possible load profiles are step, ramp, sine wave signal or no signal.

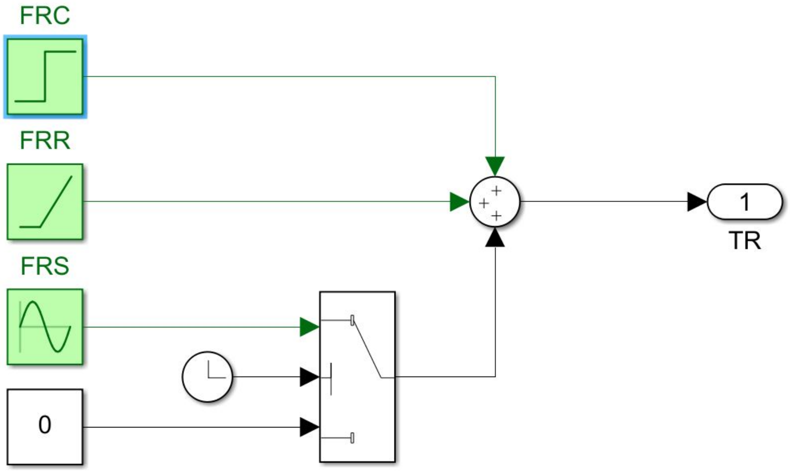


Figure 2.3: *Load Block Simulink*

### 2.1.3 Controller block

The control electronics receive as input the position defined by the selected command, the user feedback position and the angular speed of the motor. Comparing the two positions generates a position error defined as 2.1. In order to be able to evaluate the speed error (eq. 2.2) as well, the position error is multiplied by a proportional gain and saturated within the limits of $\pm\,8\,000\,rpm$. The value obtained is compared with the motor speed, and a speed error is then generated which will be provided as input to the PID controller.

$$Err_{pos} = Com - \theta_u \tag{2.1}$$

$$Err_{vel} = \omega_{ref} - \dot{\theta}_m \tag{2.2}$$

The PID controller receives the two speed and position errors and the motor speed as input. It performs the role of dynamic compensation through three basic branches:

- *proportional* to the reference error: it increases the variable to be controlled when it is too low and lowers it when it is too high; it is defined by the constant $K_P$;
- proportional to the *integrative* of the reference error: if the error does not tend towards zero it still allows to be within a certain range defined by the steady-state error; it is defined by the constant $K_I$;
- proportional to the *derivative* of the reference error: it anticipates the correction before the error increases or decreases too quickly; it is defined by the constant $K_D$.

The general control law for the PID controller is

$$u(t) = K_P e(t) + K_I \int_0^t \epsilon(\tau) d\tau + K_D \frac{de(t)}{dt} = K_P \left( e(t) + \frac{1}{t_I} \int_0^t \epsilon(\tau) d\tau + T_D \frac{de(t)}{dt} \right) \tag{2.3}$$

where $T_I = \frac{K_P}{K_I}$ is the characteristic time of the integrative branch set to $10\,000\,s$ and $T_D = K_D$ is the characteristic time of the derivative branch set to $0\,s$. Applying the Laplace Transform to the formula, the control law 2.3 becomes

$$U(s) = K_P \left( 1 + \frac{1}{T_I s} + T_D s \right) E(s) = K_P \left( \frac{T_D T_I s^2 + T_I s + 1}{T_I s} \right) E(s) \tag{2.4}$$

By filtering out the derivative action, a component that cannot be physically realised, we obtain the real control law

$$U(s) = K_P \left( 1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \frac{T_D s}{N}} \right) \tag{2.5}$$

where $N$ is the bandwidth of the derivative filter set at $1\,000\,\frac{1}{s}$. This law has been implemented on Simulink.

Modules were also added to the integrative branch to counteract *windup* phenomena. In particular, three anti-windup sections were implemented:

- *velocity* anti-windup: if the velocity is zero for more than one integration step, the integrative branch is deactivated;

- *position* anti-windup: if the position error is greater than a defined threshold, the integration branch is disabled;

- *saturation* anti-windup: if the command is saturated, the integrative branch offsets for this.

The error taken into account by the PID controller can be either speed or position. The option can be selected manually by means of a switch. If the speed error is considered, the model distinguishes between an error greater or less than 0. In the first case, the input speed error of the integrative branch is null, while in the second case the input error is multiplied by a proportional gain expressed in $\frac{N \cdot m \cdot s}{rad}$. On the other hand, if the position error is considered, the proportional gain is multiplied to a tolerance of $10^{-3}\,rad$ before entering the integration branch.

In both cases the output provided by the PID controller is a motor reference torque. Dividing this torque by the counter-electromotive force constant of the motor, the reference current is obtained. This value is saturated up to the maximum value of $22.5A$ so that damage to the motor can be excluded. This current has a purely control-related meaning and has no relation to the current flowing through the stator coils.

Finally, there is a *white-noise disturbance* block in the model which generates random numbers; however, the noise gain was set to 0 as it was found to have a negligible effect on the system [21].

The following figures 2.4 and 2.5 show the block diagrams of the control electronics and the PID controller.

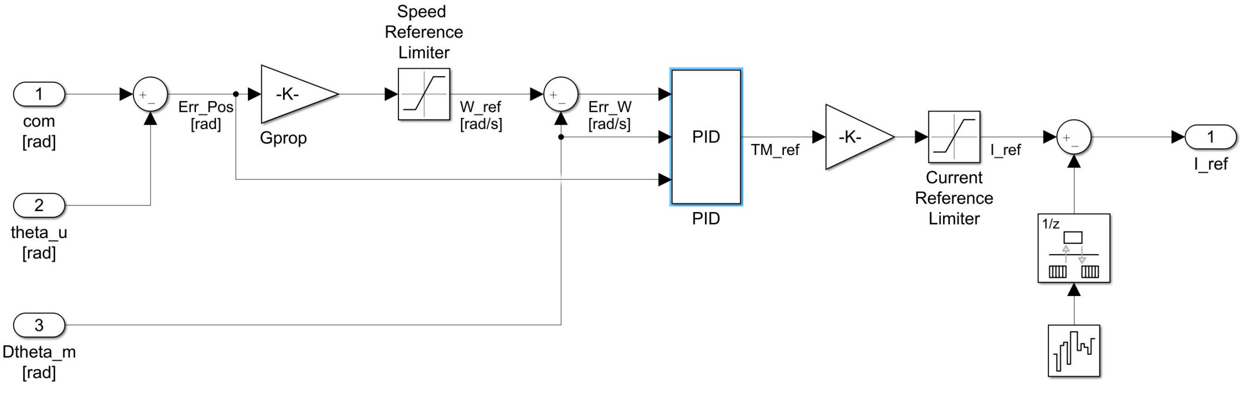


Figure 2.4: *Control Electronics block diagram*

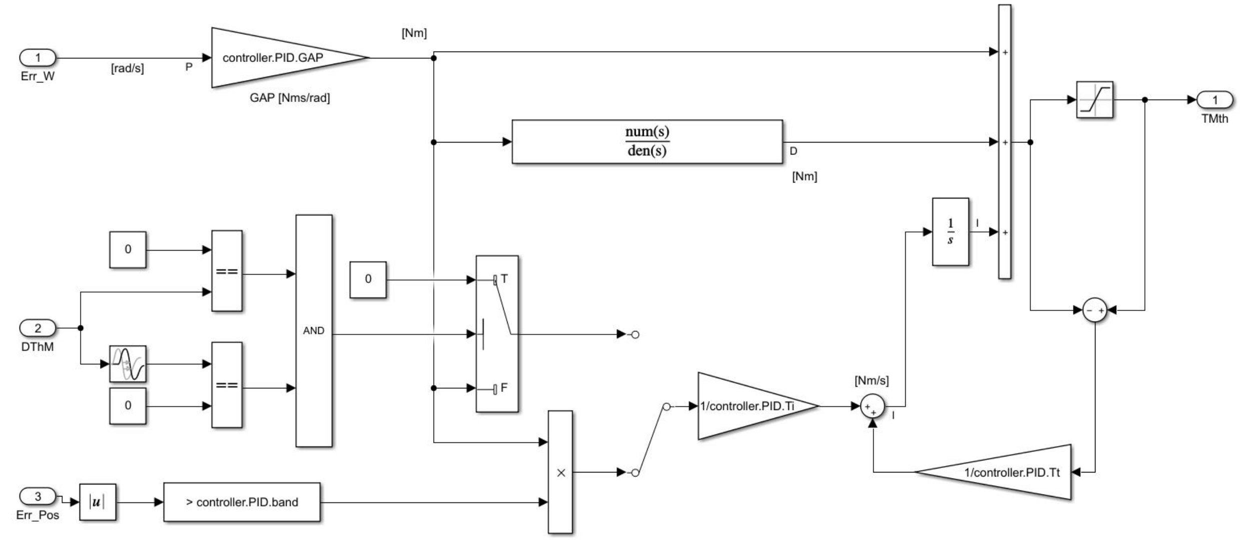


Figure 2.5: *PID Controller block diagram*

### 2.1.4 Inverter block

The inverter block takes as input the reference current $I_{REF}$ provided by the control system, the phase currents $I_A$, $I_B$ and $I_C$ provided in feedback from the motor block and the electrical position of the motor $\theta_e$.

The latter is calculated by means of an appropriate function by a special block prior to the inverter representing the *resolver*, i.e. an inductive displacement transducer which makes it possible to detect the change in magnetic induction flux as a function of the position of the solenoid(s) in the circuit. The resolver makes it possible to transform the mechanical angle output from the motor-transmission dynamics $\theta_m$ into an electrical angle $\theta_e$.
In fact, the electric motor refers to an abstract reference system in which the segments depicted are representative of specific electrical quantities. The motor is an electromagnetic rotating machine, whereby a sinusoidal voltage is produced. The instantaneous value of this voltage depends on the physical position assumed by the conductors during rotation within the magnetic field and, therefore, on the number of pole pairs present in the machine itself.
In our case, a relationship has been found between the angle of the physical model of the electric motor and the angle between the vectors representing the electrical quantities produced by the machine, described by the relationship 2.6

$$\theta_e = 2\pi \left( N_p * \frac{\theta_m}{2\pi} - floor\left( N_P \frac{\theta_m}{2\pi} \right) \right) \tag{2.6}$$

where the function $floor(x)$ rounds each element of $x$ to the nearest integer less than or equal to that element (in our case $\theta_m$) [22].

The same function is described in Matlab using the block resolver shown in the figure 2.6.

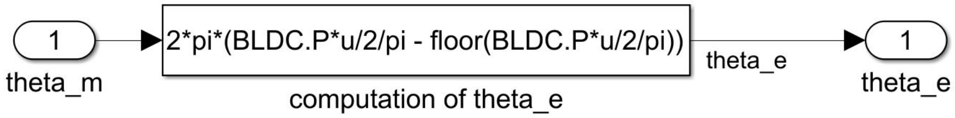


Figure 2.6: *Resolver block*

Once the electrical angle has been defined, the modules constituting the inverter model are now described.
It consists of three main parts:

- valuation of the three currents $I_{REF,A}$, $I_{REF,B}$ and $I_{REF,C}$ in the stator phases from the reference current $I_{REF}$;
- hysteresis PWM modulation of the currents;
- modelling of the H-bridge for static power conversion.

In the first block, Park and Clarke's anti-transformations are implemented to change the current from rotoric reference to phase current.
The input current $I_{REF}$ can be defined as the real part of a complex variable. The real part coincides with the component of the *direct* current $i_d$, i.e., concordant with the magnetic flux, responsible for the generation of the magnetic field; on the other hand, the imaginary component, set equal to 0, defines the *quadrature* current $i_q$, i.e., perpendicular to the magnetic field and responsible for the generation of the torque.

Starting from these two components, the Park antitransformate allows to shift the view of the currents from the rotor with velocity $\dot{\theta}_m$ to the stator, passing from a rotating biaxial system to a static biaxial system, characterized by the currents $i_\alpha$ and $i_\beta$, synchronous with the stator [23]. The Park antitransformate is defined by the relation 2.7.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta_m & -\sin\theta_m \\ \sin\theta_m & \cos\theta_n \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{2.7}$$

Clarke's anti-transform, on the other hand, makes it possible to pass from a stationary two-phase system for the stator to a generic three-phase system, where the triad of instantaneous phase values is determined [24]. Clarke's anti-transform is defined by the relation 2.8.

$$\begin{bmatrix} I_{REF,A} \\ I_{REF,B} \\ I_{REF,C} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \sqrt{\frac{3}{2}} \\ -\frac{1}{2} & -\sqrt{\frac{3}{2}} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2.8}$$

The Simulink representation of the two anti-transformations are shown in the figures 2.7 and 2.8.

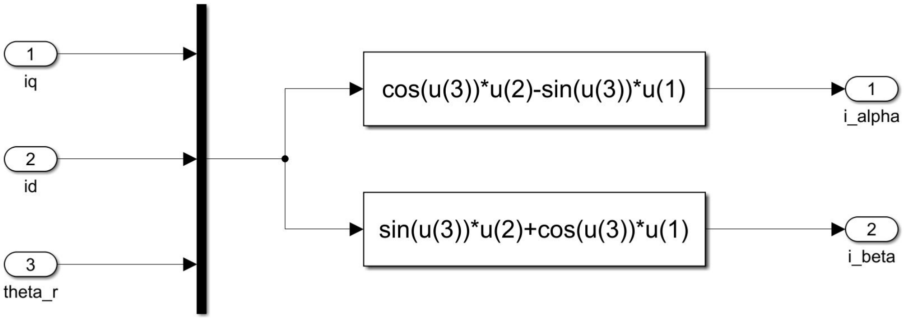


Figure 2.7: *Park antitransformate block*

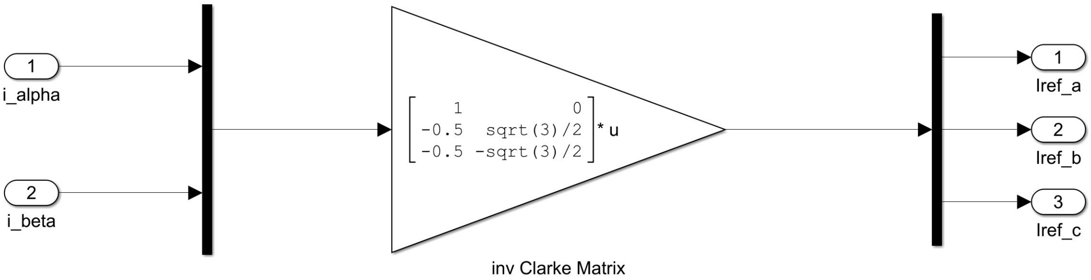


Figure 2.8: *Clarke antitransformate block*

Then the current values evaluated in the electromagnetic model of the motor $I_A$, $I_B$ and $I_C$ are subtracted from the current values $I_{REF,A}$, $I_{REF,B}$ and $I_{REF,C}$ in order to obtain a current error $I_{err,A}$, $I_{err,B}$ and $I_{err,C}$ for the three phases, according to the generic relationship 2.9.

$$I_{err,i} = I_{REF,i} - I_i \tag{2.9}$$

These three values will be given as input to the hysteresis PWM (Pulse Width Modulation) block.

In this module the current of each phase is evaluated and modulated by means of pulse control defined by multiple switching, in order to obtain continuous regulation of the motor speed when reversing the direction of rotation in the inverter. The commutation takes place around the input signal of each phase within upper and lower limits that define the *hysteresis band* (in our case $0.5\,A$). If the input current tends to be higher than the upper limit, switching towards the lower limit is activated, vice versa if the current tends to exceed the bottom limit of the hysteresis band. Due to the dependence of the switching on the value of the input current error and the width of the hysteresis band, it follows that the switching frequency is not constant, but varies according to the shape of the current. This type of commutation is electronic and eliminates the creeping contacts typical of brush motors. In fact, as we shall see in the paragraph 3.2.3, the motor taken into consideration is of the brushless type.

The hysteresis PWM block is shown in the figure 2.9.

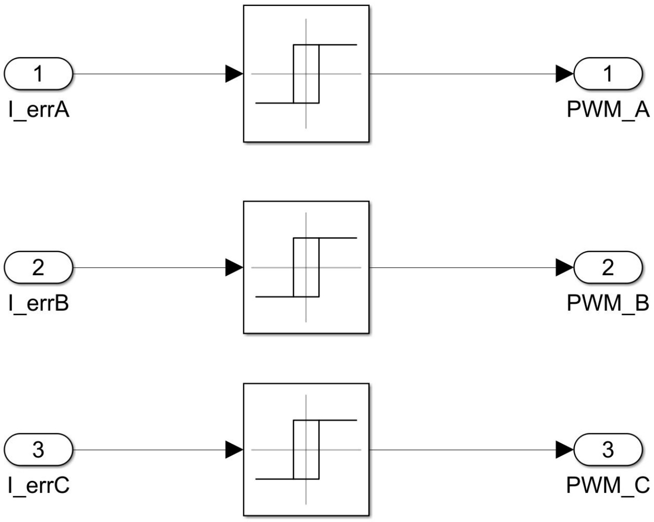


Figure 2.9: *Hysteresis PWM block*

The outputs provided by this block are given as input to the static power converter consisting of an three-phase H-bridge.

In reality, this consists of six transistors supported by six protective diodes.
The transistors work two by two, allowing to vary as needed the voltages of the three phases driving the motor.
Diodes, as mentioned, have a protective function. In fact, when the transistor is switched off, then the circuit is opened, but the current does not cancel instantaneously because energy is stored in the ferromagnetic material of the rotor. The discontinuity formed by the open circuit, however, induces the formation of an electric arc near the junction, where the electric field has a lower intensity, which causes a strong thermal dispersion. As a result, the transistor gets very hot and can burn out, damaging the entire system. Through a protective diode, the potential in the junction increases, so that current circulates in the internal circuit which, due to the Joule effect, will lead to a gradual cancellation of the current.

In Simulink the three-phase H-bridge is modelled using the *Universal Bridge* block from the *Simscape* library. This is connected to a DC voltage source which corresponds to the 48 V supply. Each motor phase is connected to two MOSFET-type transistors, one of which is electrical grounded. The mechanism for the activation and deactivation of the transistors is managed by Boolean logic which associates each PWM value with its corresponding Boolean one and its negation. These values represent whether or not current is flowing in a particular phase. This excludes the possibility of short circuits inside the H-bridge [25].
The bridge generates three output voltages which supply the motor in the *RL Model* block.

The simulink block of the three-phase bridge is shown in the figure 2.10.

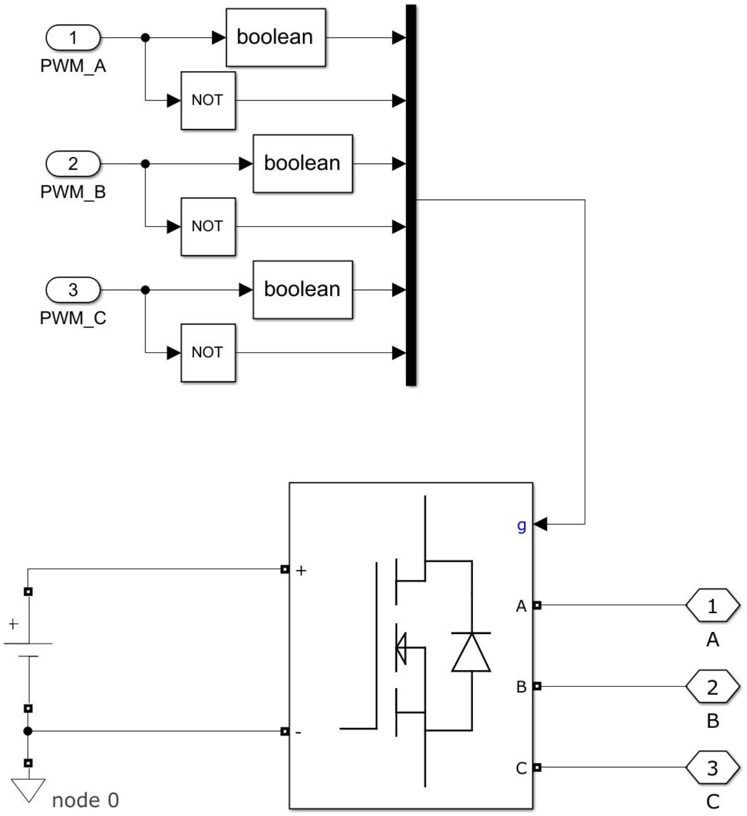


Figure 2.10: *Three-phase H-bridge block*

### 2.1.5 Motor block

The electric motor taken into consideration for modelling the electromechanical actuator is a three-phase sinusoidal *Permanent Magnet Synchronous Motor* (PMSM).

The motor consists of two main parts: the *stator*, the fixed external part, and the *rotor*, the rotating internal part. On the stator are the electrical windings, i.e. the phases that constitute the armature circuit, which, by remaining in a fixed mutual position, allow their own power supply without the need for sliding contacts. This is why the motor is called brushless. The rotor, on the other hand, accommodates the permanent magnets and forms the induction circuit. The stator and rotor are both made of ferromagnetic material and are separated by a thin air gap.
A representation of the motor is shown in the figure 2.11.

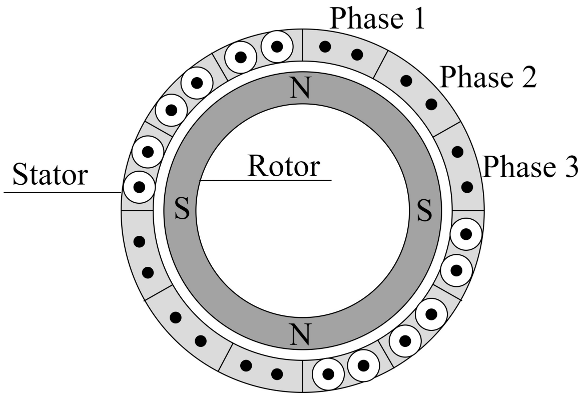


Figure 2.11: *Three-phase Permanent Magnet Synchronous Motor*

The three phases are each connected to a pair of terminals through which power can be supplied from an external source. When current flows through a phase on the stator, a magnetic field is produced which interacts with the magnetic field developed by the permanent magnets in the rotor, generating a torque which tends to align the two fields with a certain angular velocity.

The aim is to seek a distribution of the stator currents that changes in *synchrony* with the position of the rotor and that, under all operating conditions, the rotating magnetic field of the stator is always electric 90° ahead of the rotating magnetic field of the rotor. This phase shift between the two fields corresponds to the maximum achievable torque [26].

If the three phases of the stator are supplied with three appropriately phased alternating currents, an alternating magnetic field will be obtained, the resultant of which will be a rotating magnetic field constant in modulus. The three phases are powered by the inverter which controls the switching of the current and thus the rotation of the magnetic field. In order for the appropriate switching to take place while the machine is rotating, it must be dependent on the rotor position, which must be determined with great precision. The switching circuit detects the angular position of the rotor and the magnetic field attached to it and, by closing or opening the various transistors, determines at any given moment which phases to activate in order to provide the desired torque during rotation [27].

The motor under consideration is *sinusoidal*, i.e. the waveform of the counter-electromotive force generated in the stator is sinusoidal. This depends on the distribution of the conducting filaments within the stator structure and the permanent magnets. In fact, the arrangement of the windings modifies the way in which the rotor magnetic field lines concatenate with the current paths [28].

Three main blocks were used to model the motor just described with the objectives of:

- calculate the coefficients of the counter-electromotive force for each phase;
- implement the motor's RL circuit for calculating the voltage and current of each phase;

– calculate the motor torque which will be given as input to the motor-transmission coupling dynamics.

The calculation of the counter-electromotive force is based on a *form function* which depends mainly on the angle $\theta_m$. In particular, the special arrangement of the permanent magnets in the rotor ensures that this dependence is sinusoidal. Furthermore, since the motor may incur in the failure condition of eccentricity deviation from the nominal value (see chapter 3), we find the relation of the counter-electromotive force constant as a function of the modulus and phase of the static eccentricity, $\zeta$ and $\phi$ respectively.

For completeness we anticipate in this section the term *static eccentricity.* Its modulus is defined as

$$\zeta = \frac{x_0}{g_0} \tag{2.10}$$

where $x_0$ is the ratio of the offset between the rotor axis and the stator axis, while $g_0$ is the nominal value of the space between rotor and stator.

The calculation of the electromotive force constant is carried out for the three phases of the stator, so that the electric reference angle of the first phase, obtained as described in the paragraph 2.1.4, must be shifted by $\frac{2}{3}\pi$ and $\frac{4}{3}\pi$ for the second and third phases. Below are the relations for the calculation performed and in the figure 2.12 the respective model used in Simulink for the three phases.

$$k_{A,cemf} = N_A k_{fcem} \left(-\sin\left(\theta\right)\right) \left(N_p + 1\right) \zeta \cos\left(\theta_e - \phi\right) \tag{2.11}$$

$$k_{B,cemf} = N_B k_{fcem} \left(-\sin\left(\theta - \frac{2}{3}\pi\right)\right) \left(N_p + 1\right) \zeta \cos\left(\theta_e - \phi + \frac{2}{3}\pi\right) \tag{2.12}$$

$$k_{C,cemf} = N_C k_{fcem} \left(-\sin\left(\theta - \frac{4}{3}\pi\right)\right) \left(N_p + 1\right) \zeta \cos\left(\theta_e - \phi + \frac{4}{3}\pi\right) \tag{2.13}$$

In the equations $N_i$ is 1 if the corresponding phase is active, 0 if it is not.

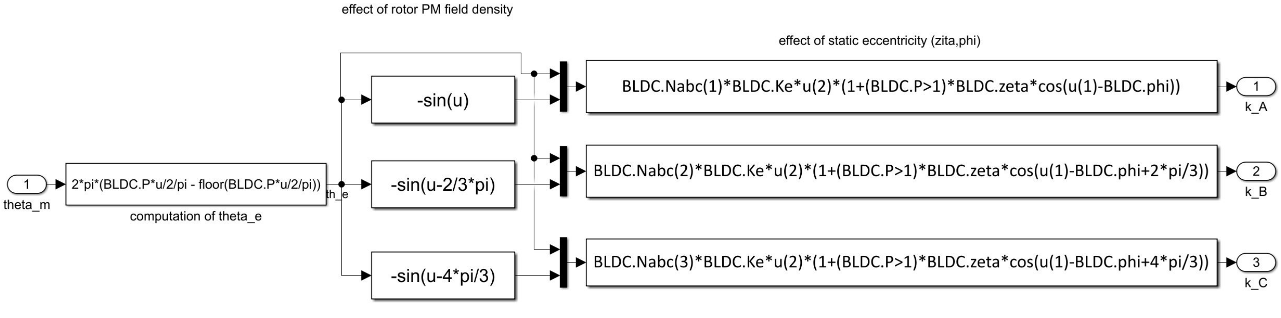

effect of rotor PM field density
effect of static eccentricity (zita,phi)
1
theta_m
2*pi*(BLDC.P*u/2/pi - floor(BLDC.P*u/2/pi))
computation of theta_e
th_e
-sin(u)
-sin(u-2/3*pi)
-sin(u-4*pi/3)
BLDC.Nabc(1)*BLDC.Ke*u(2)*(1+(BLDC.P>1)*BLDC.zeta*cos(u(1)-BLDC.phi))
BLDC.Nabc(2)*BLDC.Ke*u(2)*(1+(BLDC.P>1)*BLDC.zeta*cos(u(1)-BLDC.phi+2*pi/3))
BLDC.Nabc(3)*BLDC.Ke*u(2)*(1+(BLDC.P>1)*BLDC.zeta*cos(u(1)-BLDC.phi+4*pi/3))
1
k_A
2
k_B
3
k_C

Figure 2.12: *Counter-eletromotive force constant implementation*

From the calculated coefficients, we can determine the counter-electromotive force for the three phases as a product of these constants and the angular velocity of the motor, provided at each instant by the dynamic module. For each phase one has

$$E = k_{i,cfem} \cdot \omega_m \tag{2.14}$$

The calculated counter-electromotive forces and voltages from the inverter module are then provided as input to the RL circuit.
The figure 2.13 shows the stator circuit connected to the transistors circuit that makes up the inverter. The three phases are arranged in a star pattern, each switched by a pair of transistors.

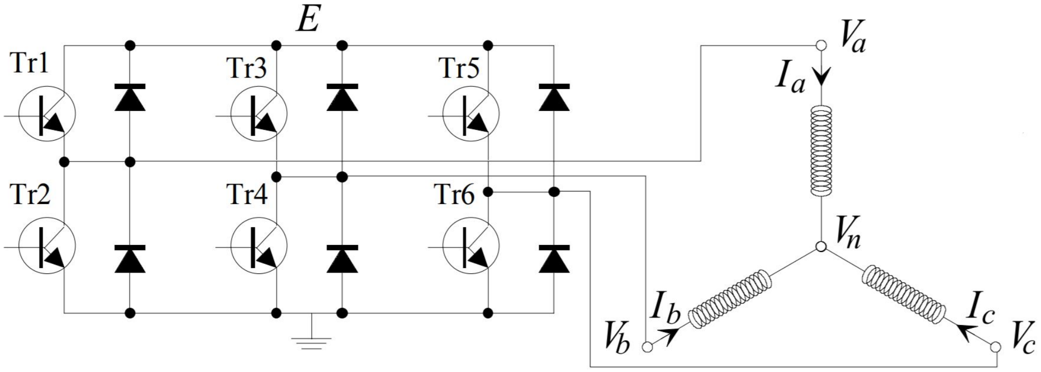


Figure 2.13: *Stator-inverter connection*

In particular, for each phase an equivalent circuit can be represented as in the figure 2.14,

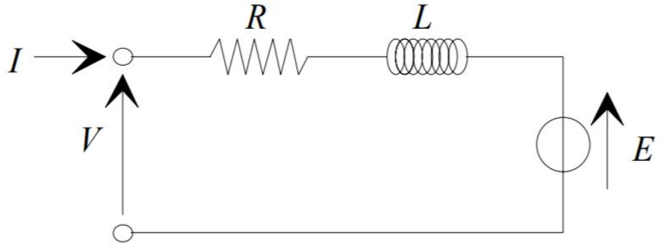


Figure 2.14: *Equivalent circuit for a single phase*

where $L$ is the inductance of the circuit, $R$ is the resistance, $I$ and $V$ are the current and voltage respectively applied to the terminals and $E$ is the electromotive force induced by the rotor flux in the stator phase.

The equations for the represented circuit in figure 2.14, taking into account the three-phase star pattern, are summarised by the formula 2.15. The voltage is equal to the sum of the contributions of resistive and inductive drops and of counter-electromotive force produced by the permanent magnets.

$$\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = R \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} + L\frac{d}{dt} \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} + \begin{bmatrix} E_A \\ E_B \\ E_C \end{bmatrix} + \begin{bmatrix} V_n \\ V_n \\ V_n \end{bmatrix} \tag{2.15}$$

In Simulink, resistance and inductance are implemented using *Simscape Electrical*, a tool that provides libraries capable of building models of physical components, characterised by physical connections, which integrate directly with block diagrams. This results in a multi-domain subsystem. In fact, this method makes it possible to calculate the exact current in the phases and therefore the correct voltage at the centre of the star circuit, without having to arithmetically average out the phases and therefore giving up the accuracy of the model [21].

The RL circuit block implemented in Simulink is shown in figure 2.15.

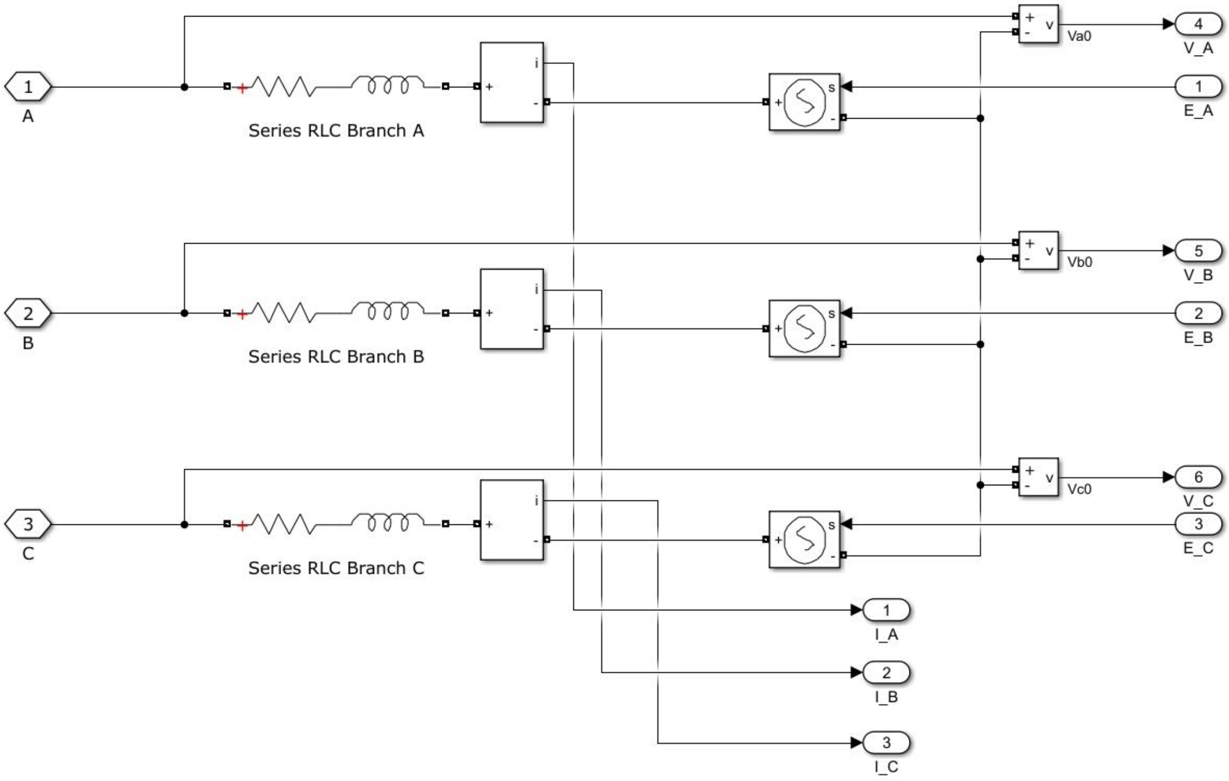


Figure 2.15: *RL circuit modeling*

The last module defining the motor block allows the rotor torque to be defined from the currents flowing in the phases and the counter-electromotive force constants. It is

$$T_m = \sum_{i=1}^{3} [k_{cfem_i} \cdot I_i]\sqrt{2} \tag{2.16}$$

To ensure that the torque does not exceed the maximum permissible value for the motor in question, a saturation block with limits $T_{m,max}$ and $-T_{m,max}$ has been implemented in the Simulink block, as shown in the figure 2.16.

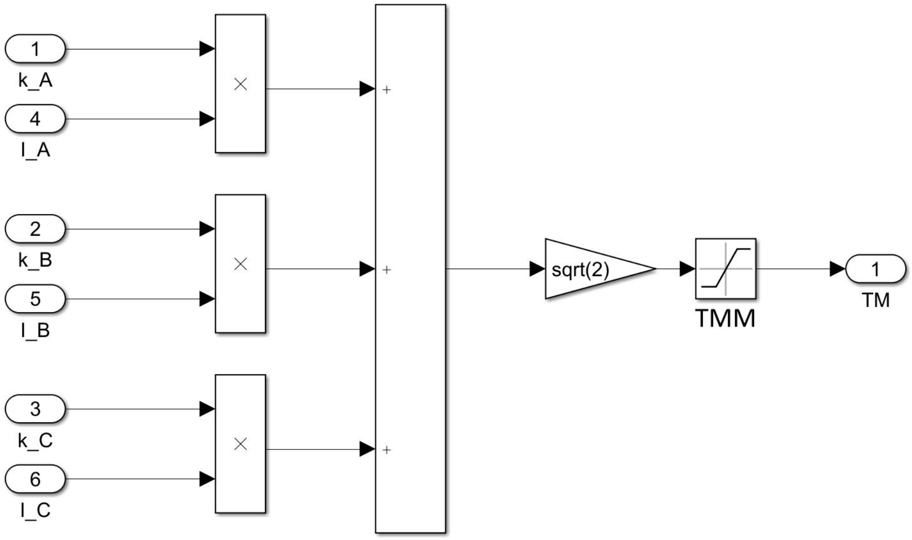


Figure 2.16: *Motor torque implementation*

The torque calculated in this way will be the input for the motor-transmission dynamics block.

Finally, in order to have a complete view of the motor module, a representation of the blocks just described and their connections in Simulink is shown in the figure 2.17.

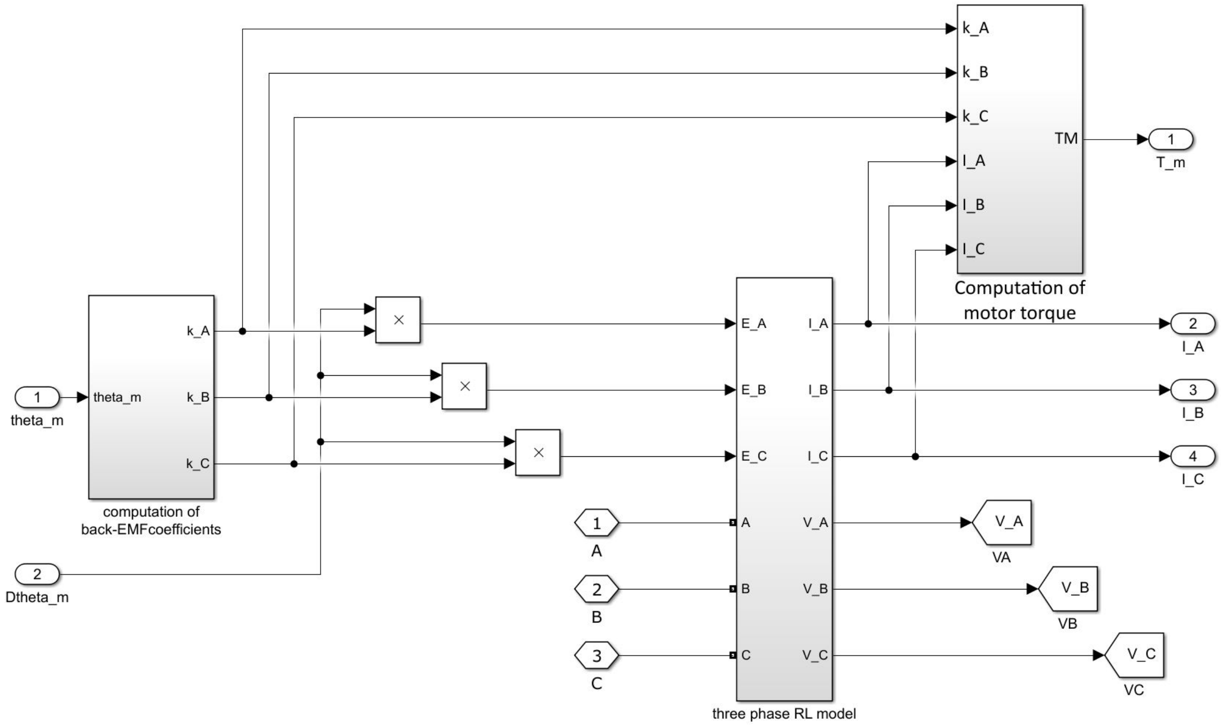


Figure 2.17: *PMSM motor modeling*

### 2.1.6 Motor-transmission dynamics block

The dynamics block implements the actuator dynamics on the basis of the motor torque and the external torque in order to identify the torque required for the user to reach a specific commanded position or velocity. The associated block diagram is represented in the figure 2.18.

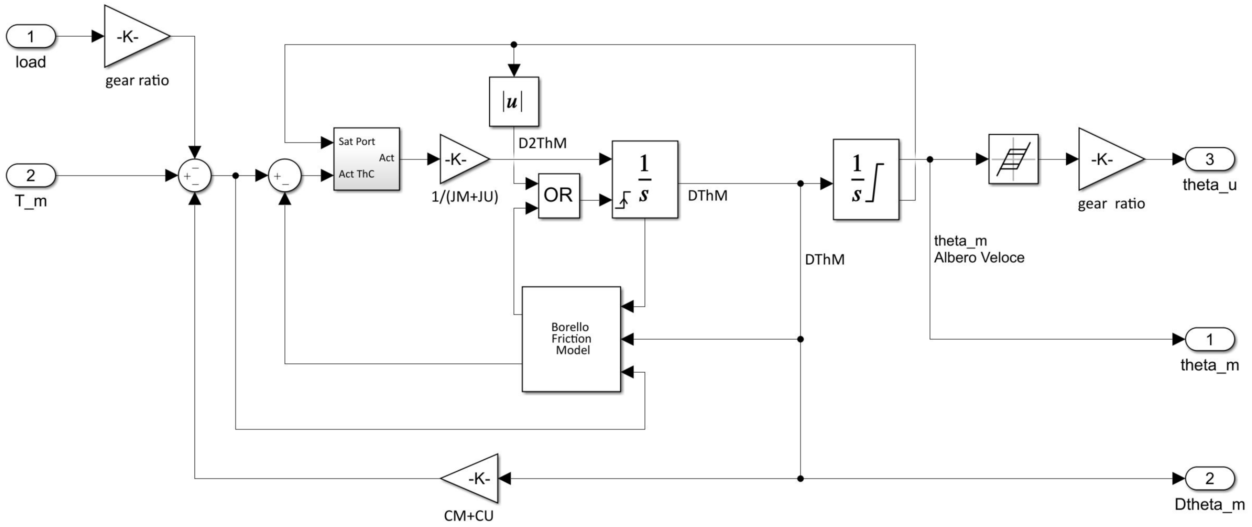


Figure 2.18: *Motor-transmission dynamics block*

The input parameters are the torque provided by the motor and the external torque, e.g. generated by an aerodynamic force on the moving surface. The latter is multiplied by a *gear ratio*, i.e. the ratio of the angular velocity of the initial gear element to that of the final gear element, equal to $\tau = \frac{1}{500}$.

The load contribution and the viscous friction factor are subtracted from the torque provided by the electrical model. The viscous friction factor is defined as the product of the angular speed of the motor and the sum of the viscous damping coefficients of the motor and the user, as in the formula 2.17.

$$T_{vf} = (C_u + C_m)\,\dot{\theta}_m = C_{tot}\,\dot{\theta}_m \tag{2.17}$$

The resulting torque is equal to

$$T = T_m - \tau \cdot Load - C_{tot}\,\dot{\theta}_m \tag{2.18}$$

The contribution of the friction force must be further subtracted from the motor torque, which was calculated using a dedicated block implementing Borello's friction model. This selects the static and dynamic friction coefficients based on the variation of the sign of the motor's angular speed.

The resulting force is called the *actuation force*, which is given as input to a suitable block able to evaluate whether to assign the torque itself or provide a null output. This decision is made on the basis of a comparison of the sign of the actuation torque with the value defined by a saturation port declaring whether or not a mechanical end-stop has been reached. In particular, the saturation gate assumes the value 1 if the upper limit is reached, 0 if there is no limit and −1 if the lower limit is achieved. If the saturation port and the actuation torque have the same sign, and in particular the latter assumes a value greater than or equal to 0.5, then the output is null. Conversely, if the signs are opposite, the output is the actuation torque itself. This mechanism is represented by the block diagram in the figure 2.19.

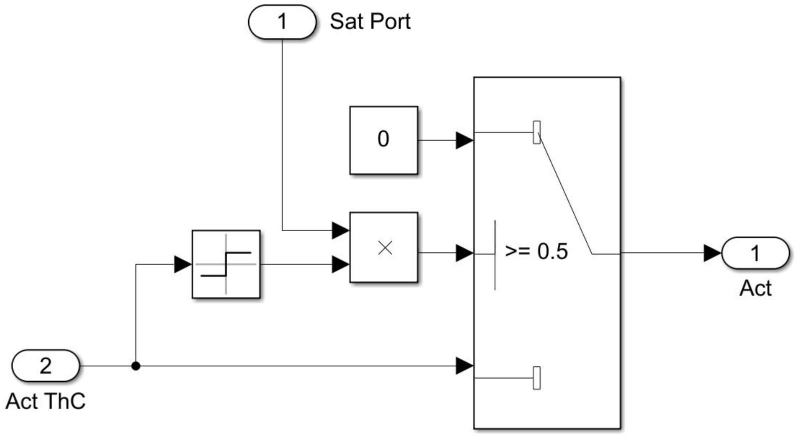


Figure 2.19: *Comparison between the actuation torque sign and the saturation port value*

If the output is the actuation torque, this is divided by the total inertia of the motor and user, according to the formula 2.19. This gives the acceleration to be applied to the motor $\ddot{\theta}_m$.

$$T_{act} = J \cdot \ddot{\theta}_m \Longrightarrow \ddot{\theta}_m = \frac{T_{act}}{J} \tag{2.19}$$

Integrating this relationship once gives the motor speed, and integrating it twice gives the motor position.

These two integrators also act as reset conditions.
In particular, the first resets the speed output when it receives a sign change as input; this condition occurs in the following two situations:

- the upper or lower end-stop is reached: the saturation block provides an output of ±1. As shown in the figure 2.18, the modulus of this value is taken into account, so in fact the first integrator sees as input the passage from the value 0 to 1 upon reaching any end-stop condition.
- the model is not able to evaluate the sign of the force when the velocity is close to 0. In fact, it could result that the mass that should have zero velocity, has velocities of opposite sign due to the friction force. This condition is not physically admissible, so a reset condition is necessary when the velocity assumes a value lower than a certain threshold [21].

- the Borello friction model block is capable of evaluating the change in sign of the velocity evaluated in two successive integration steps: if this occurs a reset signal is given.

It is sufficient for only one of these conditions to occur (established by the $OR$ logic gate) for the reset signal for the first integrator to be activated.

Finally, a *backlash* block is also inserted to shape the mechanical clearance of the gearbox, defined by a dead band centred on the output.
Then multiplying the result obtained by the transmission ratio $\tau$, the user position $\theta_u$ is obtained directly.

### 2.1.7 Signal acquisition block

As mentioned, a further block is added to the ones described above, defined as the *signal acquisition block*.

This receives as input the currents of the three phases $I_A$, $I_B$ and $I_C$ evaluated in the motor model and the value of the electrical angle $\theta_e$ to process a signal corresponding to a single equivalent current $I_{3,eq}$ that will be used in the monitor model.

In a reverse step to that seen for inverter blocking, described in the paragraph 2.1.4, the two direct and quadrature $I_d$ and $I_q$ currents are evaluated via the Clarke and Park transforms. The current values obtained are characterised by high frequencies due to PWM switching and unwanted noise. For this reason, a filter is required for the two currents, as shown in the figure 2.20.

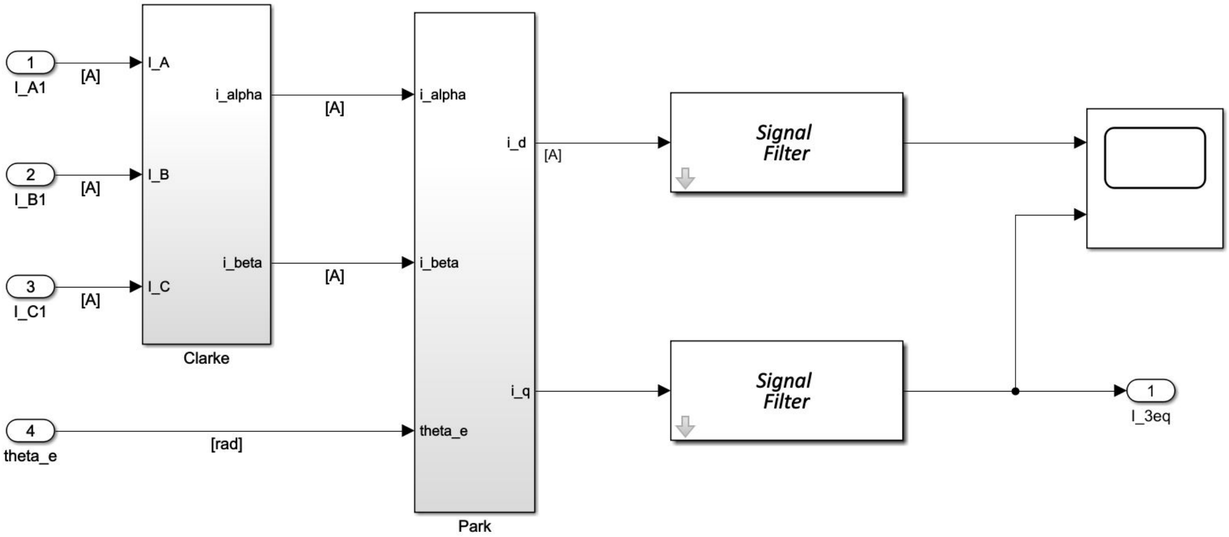


Figure 2.20: *Signal acquisition block*

From the current values obtained, the theoretical equivalent current is evaluated using the relationship 2.20.

$$I_{3,eq} = I_q \cdot \hat{q} + I_d \cdot \hat{d} \tag{2.20}$$

Since the resulting torque must always be orthogonal to the rotor's magnetic field, the direct current is zero and the equivalent current corresponds to the quadrature current. Since the resulting torque must always be orthogonal to the rotor's magnetic field, the direct current is zero and the equivalent current corresponds to the quadrature current, secondo la relazione 2.21 [25].

$$I_{3,eq} = I_q \cdot \hat{q} = \left[ -I_A + \frac{1}{2}\left(I_B + I_C\right) \right] \sin\theta_e + \frac{\sqrt{3}}{2}\left(I_B - I_C\right)\cos\theta_e \tag{2.21}$$

## 2.2 Monitor Model

The monitor model is the simplified model of the one described in the paragraph 2.1 and has the main objective of reducing the computational cost without losing too much information due to a change in model accuracy.

Although this has the same logical and functional architecture as the high-fidelity model, its applicability is limited to particular nominal or non-nominal operating conditions that must be defined specifically in advance [29]. In particular, the monitor model was compared to the reference model for nominal conditions, while the fault conditions were obtained through parameter optimisation.

The main difference between the two models is the elimination of the three-phase inverter switching module. In fact, in the reference model, as we have seen, a single equivalent current value is taken into account. This is compared in the monitor model with the current value flowing in the permanent magnet motor consisting of a single phase.
In the following paragraphs more details will be given on what this implies.

### 2.2.1 Controller block

The control block receives as input the command *Com*, the angular position of the user $\theta_u$ sent in feedback from the dynamic model and the angular speed $\dot{\theta}_m$ of the motor.

As in the high-fidelity model, both position and speed error are evaluated. In particular, the former is multiplied at a gain of $10^{-5}\,\frac{1}{s}$ to obtain a speed, while the latter is multiplied at a gain of $5 \cdot 10^{-2}\,\frac{N \cdot m \cdot s}{rad}$ and divided by the torque constant $k_v$ in order to output the reference current $I_{REF}$.
Both speed and current are limited by dedicated saturation blocks that bound the quantities to their respective maximum permissible values.

The controller block reflects the structure of the high-fidelity model. From the figure 2.21 it can be seen that the only discriminating factor is the absence of the block implementing the disturbances, which is not present in the monitor model and the replacement of a PID controller with a gain proportional to the error.

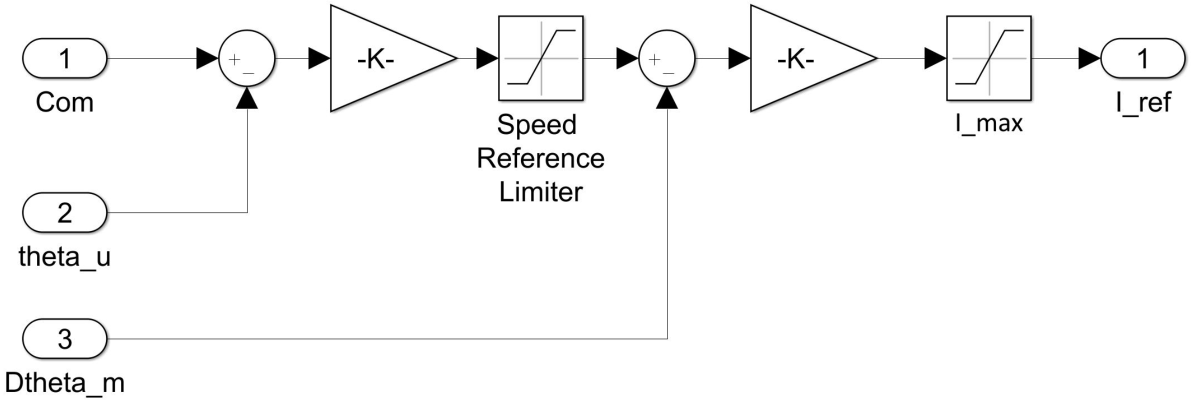


Figure 2.21: *Controller block for the monitor model*

### 2.2.2 Electrical model block

As mentioned above, the monitor model considers a simplified motor, consisting of a single phase and therefore characterised by a single current.

In particular, the simplified motor model takes as input the reference current $I_{REF}$ at the controller's output, which is compared with the current value $I_m$ supplied in feedback and processed by a suitable transfer function. The resulting error is given as input to a *sign block* that detects the sign of the current flowing. This block simulates and simplifies PWM current control. In fact, the sign of the detected current is multiplied to the supply voltage ($V_{max}$) and the result indicates whether to provide a positive, negative or zero voltage [20]. In this way, three conditions can be verified:

- the motor current is greater than the reference current: the error is negative, so the voltage to be supplied will be negative;

$$I_m > I_{REF} \Longrightarrow V = -V_{max} \tag{2.22}$$

- the motor current is lower than the reference current: the error is positive, so the voltage to be supplied will be positive;

$$I_m < I_{REF} \Longrightarrow V = V_{max} \tag{2.23}$$

- the motor current is equal to the reference current: the error is zero, so the voltage to be supplied will be zero.

$$I_m = I_{REF} \Longrightarrow V = 0\,V \tag{2.24}$$

The branch just described implements the nominal operation of the motor.
In order to take into account the different types of faults, corrections to the voltage itself due

to the presence of a short circuit and the effect of motor eccentricity must also be modelled. In addition, changing from a three-phase to a single-phase architecture involves a change in the counter-electromotive force constant due to the different current waveform and a change in torque gain. The latter influences the angular speed of the motor, consequently the counter-electromotive force coefficient and in turn the current.
For this reason, the contributions of the counter-electromotive force and the two corrections are subtracted from the nominal value calculated on the first branch, as shown in the figure 2.22.

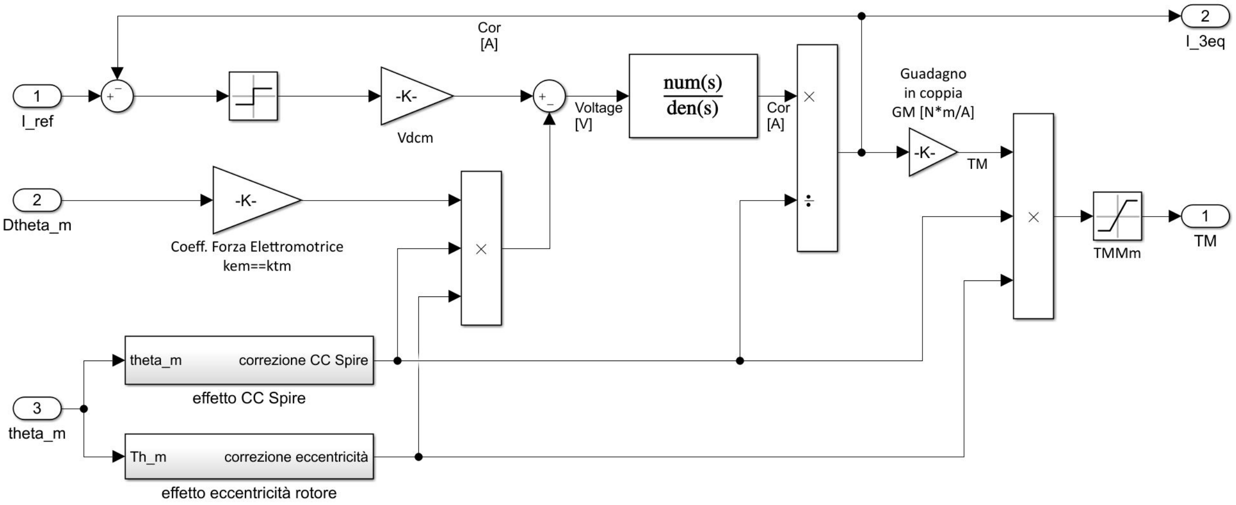


Figure 2.22: *Electrical block for the monitor model*

Furthermore, the implementation of the single-phase motor model, together with the simplification carried out on the current, makes it possible to reduce the integration step by an order of magnitude compared to the reference model and consequently requires a lower computational cost.

Once the actual voltage on the motor is defined, it is supplied as input to the motor, which is modelled with the first-order transfer function described by the equation 2.25

$$\frac{I}{V} = \frac{\frac{1}{R_m}}{\frac{N_A+N_B+N_C}{3}\tau_{RL,m}s+1} \tag{2.25}$$

where $R_m = 2.13\,\Omega$ is the windings resistance, $\tau_{RL,m} = \frac{R_m}{L_m}$ is the time constant of the system; $L_m = 7.2 \cdot 10^{-4}\,H$ is the circuit inductance, calculated as the average of the inductances of the three phases. This simplification does not affect the correspondence of dynamic response of the monitor model to the reference model one.
Note that the transfer function is dependent on the number of active phases. This value depends on the amount of short-circuit fault affecting the individual phases.

This gives the equivalent current $I_{3,eq}$, which is corrected by windings compensation. The resulting value is multiplied by a torque gain of $0.0752\,\frac{N{\cdot}m}{A}$ to give the motor torque $T_m$. The latter receives coil and eccentricity corrections and than it is saturated within the minimum and maximum limits $\pm T_{m,max}$.

The presence of the two corrections mentioned is necessary since the torque gain is actually affected by the number of active phases and the gap thickness due to eccentricity. This contribution, while in the reference model it is explicit, in the monitor model, instead, is absent, so corrective functions capable of considering the angular position of the motor have been identified to distinguish the active phases and the eccentricity characteristics.

The blocks relating to short-circuit and eccentricity corrections will be discussed in more detail in the respective sections 3.2.3 and 3.2.4. In any case, as a general conclusion, we can say that both corrections generate dimensionless corrective multiplicative coefficients which influence the electromechanical characteristics of the motor. If we identify these two coefficients as $\phi_{SC}$ and $\phi_E$, we can write the resulting motor voltage and torque values as

$$V_{corr} = \phi_{SC} \cdot \phi_E \cdot E \tag{2.26}$$

$$TM_{corr} = \phi_{SC} \cdot \phi_E \cdot TM \tag{2.27}$$

### 2.2.3 Mechanical model block

The mechanical model of the monitor model faithfully reflects the motor-transmission coupling dynamics described for the high-fidelity reference model in section 2.1.6.

The only difference is the use of the equivalent viscous damping coefficient values for the motor only; the table 2.3 shows the values used in the mechanical model.

Table 2.3: *Mechanical parameters - Low Fidelity Model*

| **Mechanical block** | | | |
|---|---|---|---|
| *Parameter* | *Symbol* | *Value* | *Units of measurement* |
| Total moment of inertia | $J_{M,m}$ | $2.5 \cdot 10^{-5}$ | $Kg \cdot m^2$ |
| Total viscous friction coefficient | $C_{M,m}$ | $5.172 \cdot 10^{-5}$ | $-$ |
| Equivalent dynamic friction | $f_{sm}$ | $0.1 \cdot T_{m,max}$ | $N \cdot m$ |
| Equivalent static friction | $f_{dm}$ | $0.05 \cdot T_{m,max}$ | $N \cdot m$ |

The figure 2.23 shows the mechanical model implementation in Simulink, which is essentially identical to that shown for the reference model.

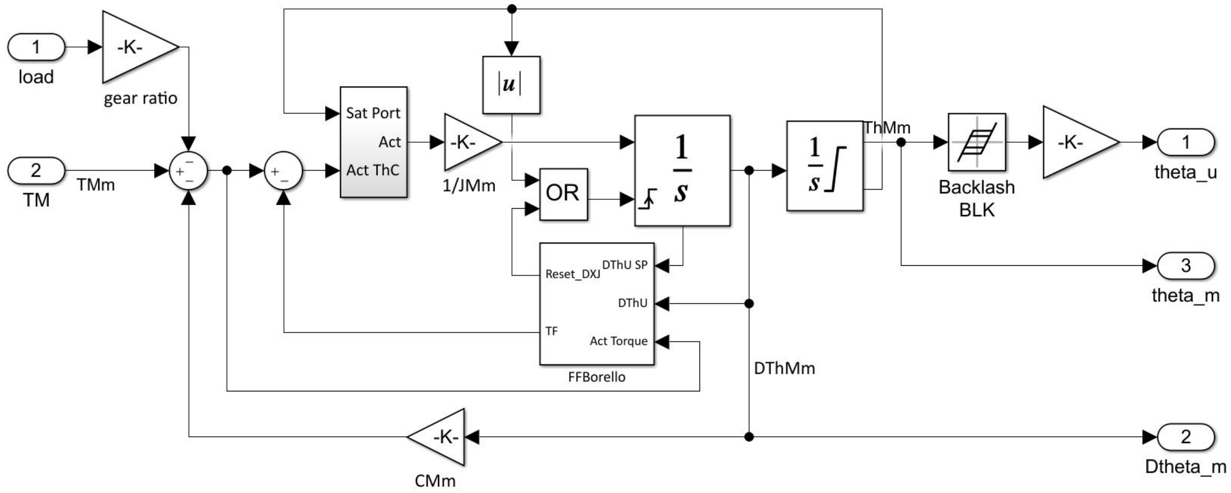


Figure 2.23: *Mechanical block for the monitor model*

The current calculated $I_{3,eq}$ in the mechanical model is then subjected to a low-pass filter so that the reference and monitor models have the same delay. The filter, in fact, is described by the same characteristic time $\tau_f = 5 \cdot 10^{-5}\, s$.

## 2.3 Models output and comparison

Before describing how the different faults were implemented, it is necessary to assess how faithful the monitor model is to the reference model under nominal conditions. For this it is useful to observe the response of the system to the different command and load profiles.
In particular, the one shown refers to a chirp command and a null load. These results are reported because the chirp command is the most variable and allows to observe the difference, in terms of response speed of the monitor model compared to the reference one, in a significant way.
The response of the system is assessed by observing over time the angular positions of the motor $\theta_m$ and user $\theta_u$, the angular speed of the motor $\dot{\theta}_m$ and the equivalent current value $I_{3,eq}$. The duration of the simulation is $0.5\, s$.
The figure 2.24 shows the evolution of the motor position in the two models. It can be seen that the system described by the monitor model faithfully follows the commanded profile. The error between the two positions, as can be seen from the figures 2.25 and 2.26, is confined to the order of $10^{-3}\, rad$ for the simulation time specified. It is also noted that as the frequency of the command increases, the error tends to rise.

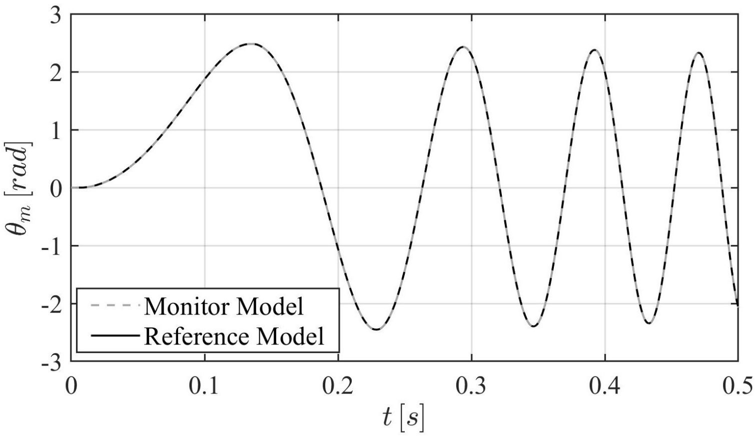


Figure 2.24: *Motor position*

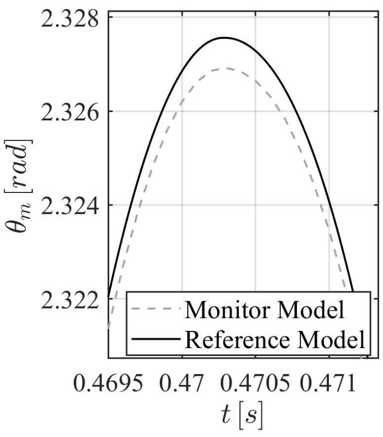


Figure 2.25: *Motor position zoom*

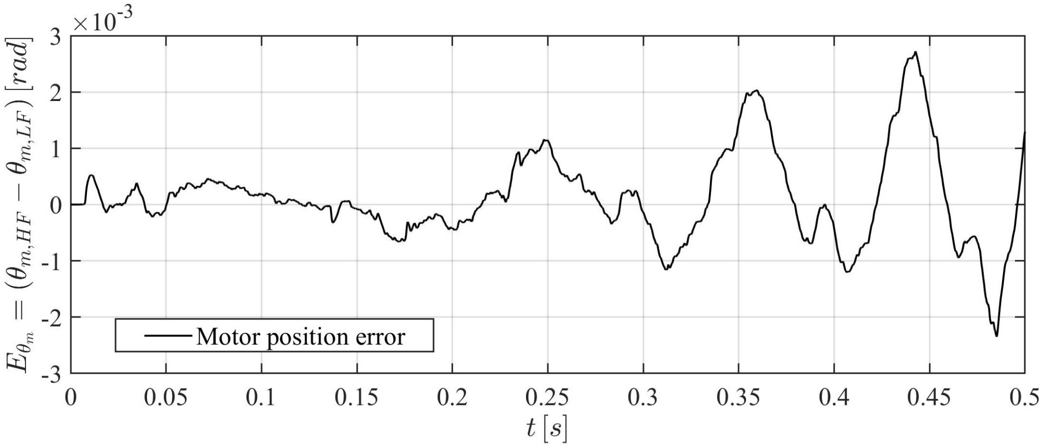


Figure 2.26: *Motor position error*

The following shows the development of the user position. As can be seen from the figure 2.27, this also follows, with good approximation, that of the reference model. In this case, the error recorded between the two models is smaller than that of the motor and is of the order of $10^{-6}\,rad$.This can be seen from the figures 2.28 and 2.29. Again, the error tends to increase as the frequency of the chirp command grows.

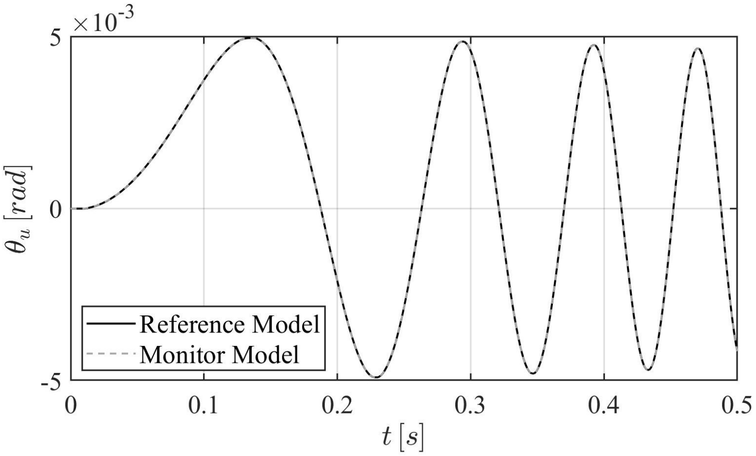


Figure 2.27: *User position*

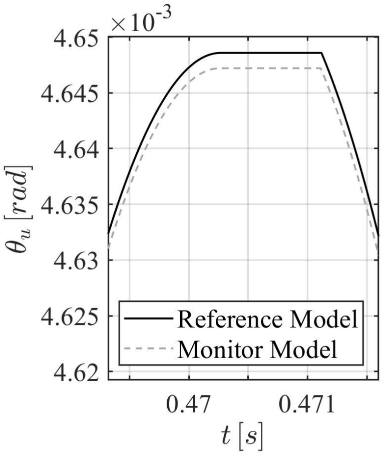


Figure 2.28: *User position zoom*

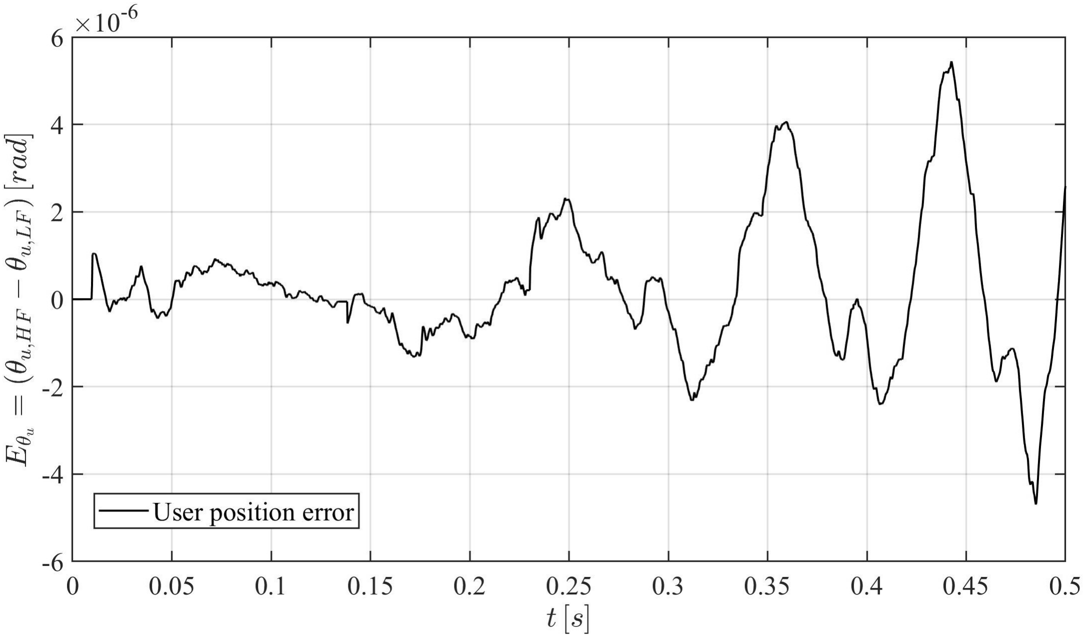


Figure 2.29: *User position error*

The following figures show the speed and speed error trends of the motor. Also in this case the monitor model seems to follow the trend of the reference model, as shown in the figure 2.30. However, the error trend is oscillating due to the curve of the motor speed in the reference model, which can be seen in the figure 2.31. The error is greater than those assessed previously and it is of the order of $10^{-1}\,\frac{rad}{s}$, as shown in the figure 2.32.

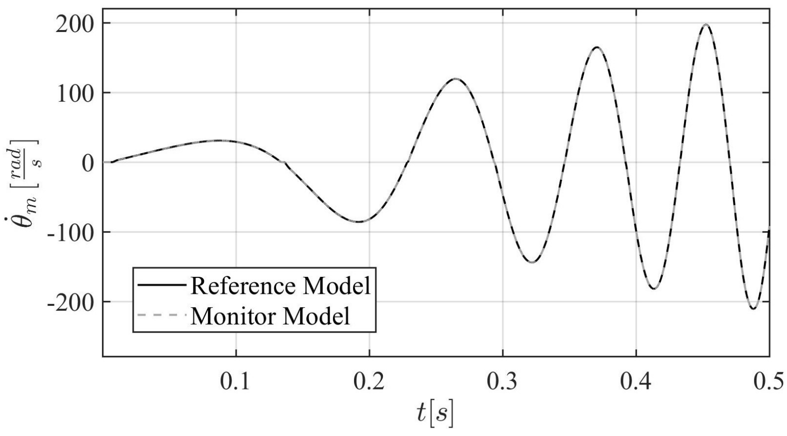


Figure 2.30: *Motor speed*

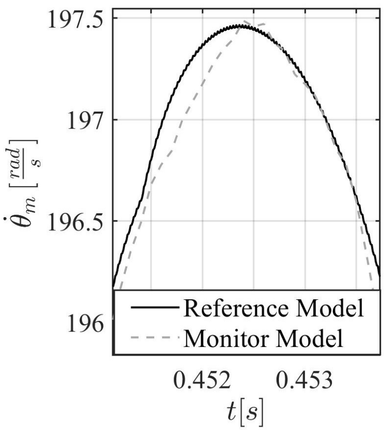


Figure 2.31: *Motor speed zoom*

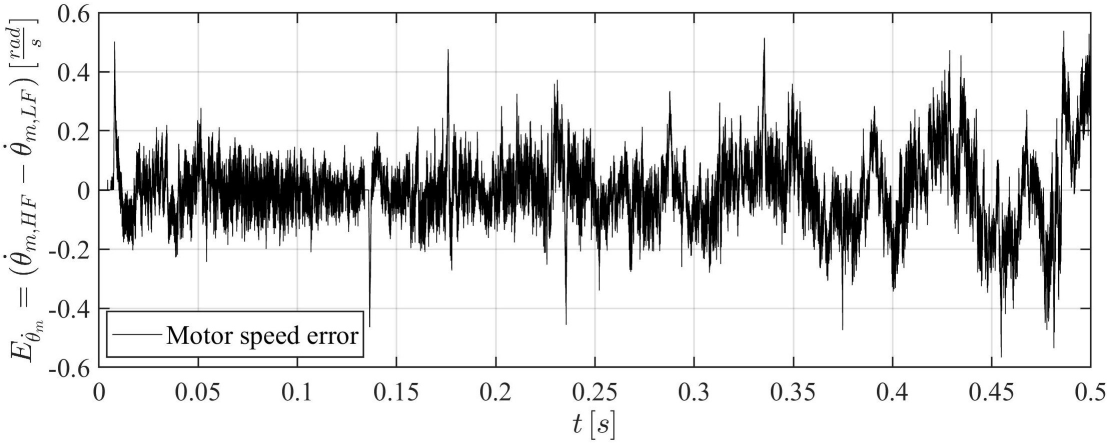


Figure 2.32: *Motor angular speed error*

Finally, the current and error trend is shown in figures 2.33 and 2.34. It can be seen that the current, as well as its error, has a trend dependent on the frequency of the command.

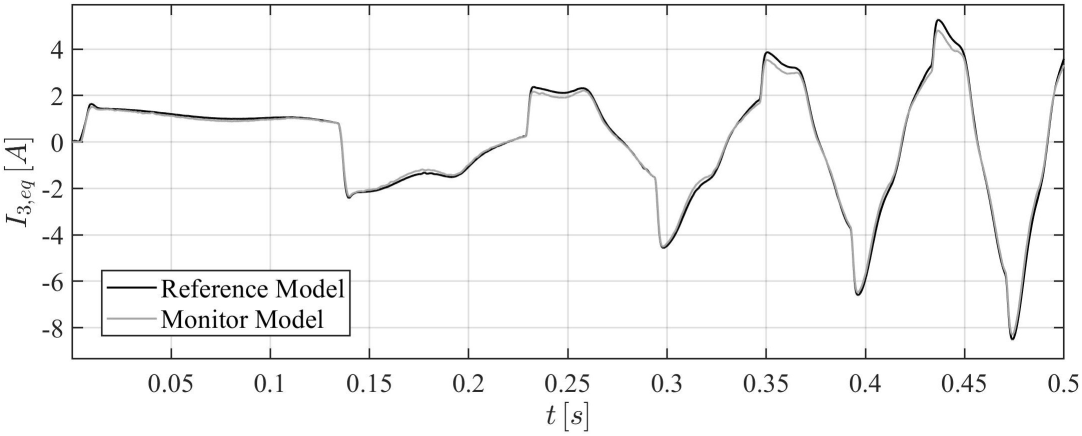


Figure 2.33: *Three-phase equivalent current*

This dependence can be explained by the fact that inertial forces increase with frequency as greater readiness to reverse direction is required. This results in a greater demand for actuation speed, therefore acceleration, and consequently greater torque with each change of direction of the drive. Since the current is proportional to the motor torque, the necessary dependence on the drive frequency follows.
The error between the two is of the order of $10^{-1}\,A$ and tends to increase as the frequency of the chirp command is higher.

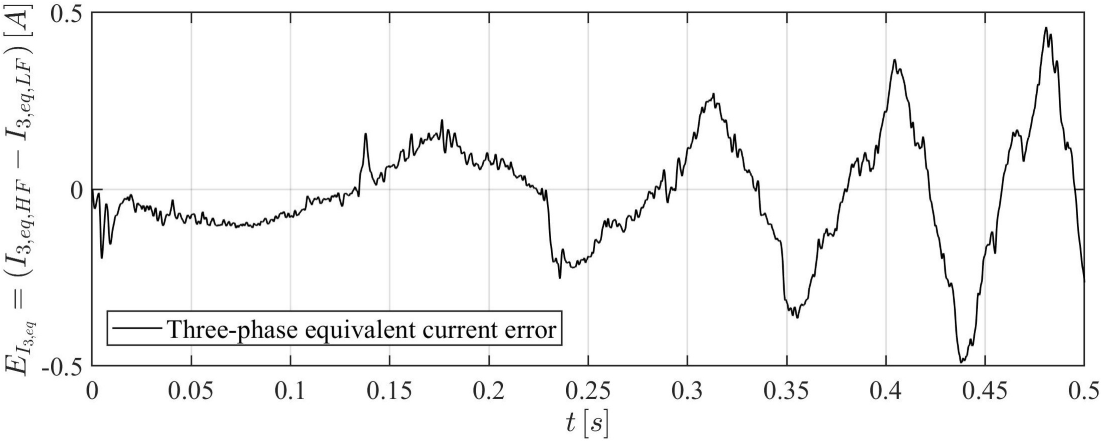


Figure 2.34: *Three-phase equivalent current error*

The analysis just carried out for the nominal conditions can be performed in the reverse process when defining the monitor model in order to determine the parameters, for instance torque gain or the counter-electromotive force coefficient, to be close to the reference model. In this way, the characteristics of the motor are defined and so evaluations can be performed on changes in parameters depending on the fault or combination of faults implemented [30].

# Chapter 3

# Reliability and fault implementation

As introduced in chapter 1, actuators are critical to the safety of an aerospace system and an undetected failure can lead to serious consequences. For this reason, a health management system must allow enough time for the management of unforeseen events but, for prognostic purposes in particular, fault detection must be early enough [31].

The problem with these systems, however, is that the use of electromechanical actuators in aeronautics is rather recent, so that consistent statistical data are not yet available due to the fact that they have not yet been used long enough and on a large scale [31][32].

For this reason, the used model-based approach is helpful, as it allows the physical properties of the system to be identified and, consequently, failure modes to be determined efficiently by varying model parameters to simulate different levels of performance degradation. In this way it is possible to assess a failure by evaluating the output response and verifying that it has a pattern that deviates from the expected one in reaction to a specific input command. It must be considered that the system output in this case is the superposition of the effects of faults and uncertainties on the nominal response [10].

In this chapter, a failure analysis of an electromechanical actuator will be carried out and then the failures implemented in the Matlab-Simulink model will be explained.

## 3.1 Failure Analysis

Failure analysis uses two fundamental tools: *Failure Mode Effect and Criticality Analysis* (FMECA) and *Fault Tree Analysis* (FTA).

FMECA is a commonly used reliability analysis tool that aims to identify potential failures of a system or process, assess the effects of the failure of each component at several levels, classify the different failure modes identified according to their criticality and, finally, define failure mitigation actions [33].

According to a thorough FMECA carried out by [34], 1950 failure modes were distinguished for the individual electromechanical actuator, of each severity and probability of occurrence. However, for the sake of simplicity, the faults that can generally occur in an electromechanical actuator can be divided into four macro categories:

- *Mechanical or structural faults*: they are caused by excessive loads, environmental factors, lubrication problems or manufacturing defects. For instance, they may involve the gearbox, which is responsible for transferring rotational energy and reducing motor speed; in this case the device may be subject to fatigue cracking or gear chipping. A further example is the screw that performs the linear displacement to the actuation system; it may be subject to bearing seizure, due to a lack of lubrication, or jamming [10][34].

- *Motor faults*: due to the high speed of rotation, the system may experience an increase in temperature and large mechanical stresses. The thermal effects can compromise the insulation of the windings by generating short circuits first between the coils of the same phase, then between the windings themselves due to the reduction in resistance and the increase in current, which leads to further overheating. The high mechanical stresses, in addition to the tolerances and construction imperfections of the motor, induce an offset between the axes of the rotor and stator, i.e. they introduce the eccentricity factor, which causes the magnetic flux to be periodically variable according to the position of the rotor [32][34].

- *Electrical/Electronic faults*: they are caused by overheating, wear, overcurrents or loss of power. The failure modes of the electronics in the electromechanical actuator are entirely identical to those possible for any electronic device on board the aircraft. A failure of the electronics, for example, can result in a malfunction of the controller, which manages the motor output on the basis of the command signal and position feedback; if communication is suspended, the output cannot be generated correctly [10][34].

- *Sensor faults*: this type of fault also affects the controller which, as already seen, bases its action on the feedback obtained from the sensors. In this way, evaluations of the system's physics are incorrect and do not correspond to the actual state of the device being considered. The faults that can occur in this case are bias, drift, scaling, noise and intermittent dropout [21][31]. For instance, the electrical connection may be interrupted due to a short or open circuit, faults on transistors or capacitors, but also faulty wiring or soldering.

For each subgroup in the tables 3.1, 3.2 and 3.3 from [31] it is detailed which types of failure can occur and at what level of probability and criticality. The level of probability $P$ and criticality $C$ of failure is expressed by a scale of 1 to 10 from least likely and least critical to most frequent and catastrophic respectively. It is clear that it is desirable for a catastrophic event to have a near-zero chance of occurring.

Table 3.1: *Mechanical/structural faults*

| **Fault** | **Failure** | **P** | **C** |
|---|---|---|---|
| *Screw* | | | |
| Spalling | Severe vibrations, metal flakes separating | 5 | 3 |
| Wear/backlash | Severe backlash | 7 | 3 |
| *Nut* | | | |
| Spalling (mild) | Severe vibrations, metal flakes separating | 5 | 3 |
| Backlash | Severe backlash | 7 | 3 |
| Degrated operation | Seizure/disintegration | 3 | 5 |
| Binding/sticking | Seizure/disintegration | 3 | 5 |
| Bent/dented/warped | Seizure/disintegration | 1 | 5 |
| *Ball returns* | | | |
| Jam | Seizure/disintegration | 5 | 8 |
| *Bearings* | | | |
| Spalling | Severe vibrations, metal flakes separating | 5 | 3 |
| Binding/sticking | Seizure/disintegration | 2 | 4 |
| Corroded | Severe vibrations, metal flakes separating, seizure disintegration | 2 | 5 |
| Backlash | Severe backlash, vibrations, disintegration | 7 | 3 |
| *Piston* | | | |
| Cracks, slop/play | Structural failure | 1 | 10 |
| *Dynamics seals* | | | |
| Wear | Structural failure | 4 | 6 |
| Structural failure | Structural failure | 3 | 8 |
| *Static seals* | | | |
| Structural failure | Structural failure | 2 | 8 |
| *Balls* | | | |
| Spalling/deformation | Severe vibrations, metal flakes separating | 5 | 3 |
| Excessive wear | Backlash | 7 | 5 |
| *Mountings* | | | |
| Cracks, slop/play | Complete failure | 1 | 7 |
| *Lubricant* | | | |
| Contamination | Seizure/disintegration | 8 | 5 |
| Chemical breakdown | Seizure/disintegration | 4 | 5 |
| Run-dry | Seizure/disintegration | 3 | 10 |

Table 3.2: *Motor faults*

| **Fault** | **Failure** | **P** | **C** |
|---|---|---|---|
| *Connectors* | | | |
| Degraded operation (increase of resistance) | Disconnect | 5 | 6 |
| Intermittent contact | Disconnect | 3 | 7 |
| *Stator* | | | |
| Stator coil fails open (degraded EMA performance) | Opening failure | 4 | 4 |
| Insulation deterioration/wire chafing (reduced or intermittent current) | Short cricuit | 5 | 5 |
| *Resolver* | | | |
| Coil fails open | Opening failure | 4 | 10 |
| Intermittent coil failures | Permanent coil failure | 5 | 7 |
| Insulation deterioration/wire chefing | Short circuit | 5 | 7 |
| *Rotor and magnets* | | | |
| Chemical bond deterioration | Complete magnet separation | 2 | 10 |
| *Dynamics seals* | | | |
| Rotor eccentricity | Bearing support failure | 3 | 6 |

Table 3.3: *Electrical/Electronics faults*

| **Fault** | **Failure** | **P** | **C** |
|---|---|---|---|
| *Power supply* | | | |
| Short circuit | Short circuit | 5 | 10 |
| Open circuit | Open circuit | 5 | 10 |
| Intermittant performance | Short circuit or open circuit | 5 | 8 |
| Thermal runaway | Dielectric breakdown of components leading to open or short circuit | 6 | 10 |
| *Controller capacitors* | | | |
| Dielectric breakdown | Short circuit or open circuit | 4 | 8 |
| *Controller transistors* | | | |
| Dielectric breakdown | Short circuit or open circuit | 4 | 8 |
| *Wirings* | | | |
| Short circuit | Short circuit | 5 | 10 |
| Open circuit | Open circuit | 5 | 10 |
| Insulation deterioration/wire chafing | Short circuit or open circuit | 5 | 8 |
| *Solder joints* | | | |
| Intermittent contact | Disconnect | 5 | 8 |

Not all faults in the tables above lead to actuator-level effects, defined as *end effects*. For this reason it is important to report the effects that these faults have on the actuator, grouped according to the *item failure rate* expressed in failure per million hour ($fpmh$). This is reported in the table 3.4 from [34].

Table 3.4: *Failure mode effects*

| **End effect** | **Item failure rate ($fpmh$)** |
|---|---|
| *Actuator jam* | $3.647 \cdot 10^{-2}$ |
| *Actuator runaway* | $6.0 \cdot 10^{-6}$ |
| *False alarm signal* | $7.859 \cdot 10^{-2}$ |
| *Loss of actuator* | 6.152 |
| *Loss of capability to engage the static brake* | $3.664 \cdot 10^{-1}$ |
| *Loss of service communication* | $5.748 \cdot 10^{-2}$ |
| *No functional effect* | $1.039 \cdot 10^{1}$ |
| *No functional effect (critical in presence of other failures)* | $1.777 \cdot 10^{-1}$ |
| *No significant effect* | $4.366 \cdot 10^{-2}$ |
| *Possible loss of actuator* | $1.550 \cdot 10^{-4}$ |
| *Static brake always engaged* | $6.660 \cdot 10^{-2}$ |

The second tool for feasibility analysis is the *Fault Tree Analysis* (FTA). Its objectives are: to identify the causes of a specific failure (especially safety-critical failures), to evaluate the design of a system for its safety (or reliability), to quantify the probability of failure. A fault tree is a detailed logic model describing the combinations of faults that can produce a specific fault in the system of interest (*top event*), developed according to applicable standards, such as ECSS-Q-ST-40-12C [33].

According to the study undertaken by [34], this tool led to the definition of four failure modes that allowed the development of four main security requirements for the electromechanical actuator:

- *Loss of control/function*: the actuator can no longer be controlled and can therefore no longer be used;
- *Free floating*: the actuator is moving freely or, due to extensive structural damage, there is excessive mechanical play;
- *Runaway*: the actuator has little or no controlled movement and oscillations, it is in a free-floating or hardover condition;
- *Jam*: the actuator becomes inoperative due to a fault in the mechanisms that drive the actuator.

Failure rates were associated with each of these failure modes in accordance with FMECA, which enabled the requirements summarised in the table 3.5 to be developed. These are expressed as a function of flight hours ($FH$).

Table 3.5: *Fault Tree Analysis Requirements*

| **Top event (actuator)** | **Failure probability** $\frac{1}{FH}$ | **Requirement** $\frac{1}{FH}$ |
|---|---|---|
| *Loss of control/function* | $6.218 \cdot 10^{-6}$ | $< 10^{-7}$ |
| *Free floating* | $6.000 \cdot 10^{-9}$ | $< 10^{-7}$ |
| *Runaway* | $2.071 \cdot 10^{-12}$ | $< 10^{-8}$ |
| *Jam* | $3.648 \cdot 10^{-8}$ | $< 10^{-9}$ |

It is noted that actuator jamming, being able to occur at only one point, requires a more stringent reliability requirement; conversely, runaway, which is possible at several points, has a less rigorous specification [34].

## 3.2 Fault implementation

Of all the possible faults highlighted in the paragraph 3.1, only the faults implemented in Matlab-Simulink will be described in detail in this section: dry friction, backlash, short circuit, eccentricity, proportional gain and noise.

### 3.2.1 Dry friction

Friction is a force that occurs between two components in contact moving in relative motion. It acts opposite to the motion and therefore to the speed, in a direction parallel to the surfaces, with characteristics that depend on the properties of the two sliding surfaces.
Friction is defined as *dry* when the surfaces are not separated by any layer of lubrication. In this condition the two surfaces come into direct contact. Each of them is characterised by a roughness index that depends on the type of machining carried out on the materials. When the two surfaces are brought into contact, assuming that the surfaces have identical properties, friction occurs not over the entire theoretical contact area, but in a limited number of regions, where the roughness of one meets the other. Mechanical interactions are generated at these points and micro-junctions are formed due to the high local pressures [35][36].

Friction between the moving parts of actuators, electric motors, transmissions or bearings is a phenomenon that limits the performance of control systems. It is a dissipative force, always linked to a loss of energy, a rise in temperature but above all to wear of the parts in contact. In addition, the nature of friction force is highly non-linear so it is not easy for a conventional controller to achieve high accuracy in tracking the commanded trajectory.

However, through appropriate modelling of the phenomenon, it can provide the correct compensation to improve the accuracy of position control in electrical drives.

If the friction conditions were to change while the electromechanical actuator is in use, this could lead to a reduction in system performance in terms of accuracy or the introduction of unforeseen and undesirable behaviour.
Inaccurate modelling of the friction model in the presence of proportional-integrative control logic could result in the emergence of limit cycles due to the interaction of the friction force and the integrative action of the controller.

For example, at low speeds a transient can be observed experimentally that is responsible for a system instability defined as the *stick-slip* phenomenon. This dynamic instability is linked to the oscillation of the force which is interpreted by the controller as an acceleration of the body, which induces a variation in speed, and consequently in position; under these conditions the phenomenon is amplified rather than compensated for.
The reason why this instability occurs, as already mentioned, derives from the typical trend of the friction force as a function of speed, called the Stribeck curve, shown in the figure 3.1.

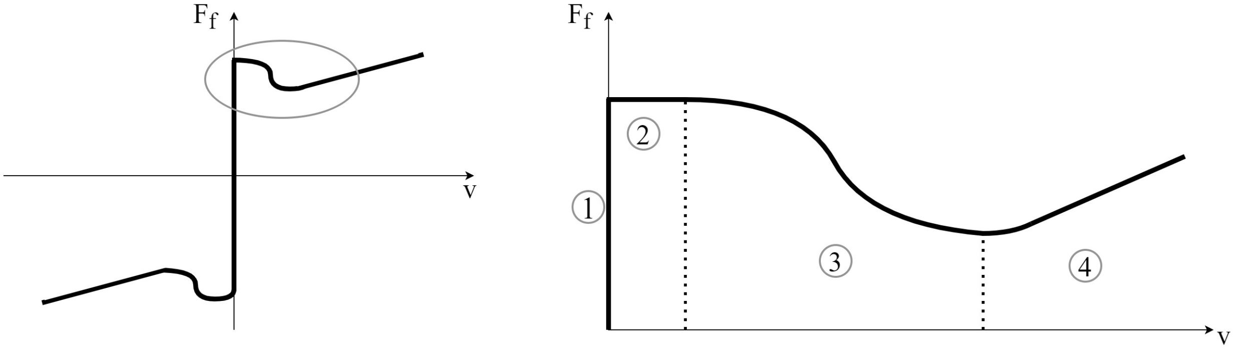


Figure 3.1: *Stribeck curve*

In the zone 1 the bodies are in *static regime*, that is to say there is no relative displacement, even if an elastic deformation occurs between the asperities, which constitute real constraints. In the 2 regime there is a *lubrication boundary*, which occurs when some of the asperities begin to break drastically in a brittle way, depending on the hardness of the material or on the presence of oxides that are not perfectly bound to the body, so that a short-lived sliding starts, which is however interrupted by the presence of further asperities. In the 3 regime of *partial lubrication* one of the two surfaces moves at a higher speed and the sliding motion brings the lubricating fluid between the contact surfaces, causing the friction to decrease. Finally, in the 4 regime of *total lubrication*, the speed is such that full lubrication is ensured, so that the two bodies are completely separated and move according to the dynamics typical of viscous friction, with constant slope as the speed increases [37].

For the modelling of dry friction, no reference was made to mathematical models of tribological derivation, but it was preferred to use numerical performance models representative of dynamic performance in the time domain. In this way, local effects and dynamics are neglected but a global representation of the phenomenon is provided.

An accurate numerical modelling of the phenomenon requires a description of the behaviour of the mechanical element subject to friction as a function of the loads. In particular, it is necessary that the model can represent the following four conditions [37]:

- initially stationary mechanical element that must remain stationary;
- initially stationary mechanical element that must move;
- mechanical element initially in motion that must remain so;
- mechanical element initially in motion that must stop.

In this way, confidence in the simulation results is also increased and the model is able to investigate critical and undesirable operating conditions.

For this reason, the *Borello - Dalla Vedova model* has been used, a Coulombian friction model which allows the four conditions to be met, and which is also effective in situations where the sign of the actuation speed is reversed. In this way, the simplifying assumptions, which in other models compromise compliance with the Coulombian model itself and affect the accuracy and performance of the system, are not necessary and it is possible to describe the operating conditions through a robust, versatile and user-manageable calculation algorithm.

The distinctive features of Borello's method are [37]:

- the sign of the friction torque varies with the direction of the speed;
- grip conditions are distinct from dynamic conditions;
- the possibility that the mechanical element initially in motion may stop is considered;
- the mechanical element is properly maintained under conditions of adhesion or motion;
- the possibility that the mechanical element initially stationary may restart is considered;
- considers possible end-of-stroke (inelastic impact).

As mentioned above, the model is based on the classical Coulombian formulation and describes the effects of dry friction as a function of speed and active torque.

The corresponding mathematical formulation evaluates in this case the friction torque $T_f$ as follows:

$$T_f = \begin{cases} T & \dot{\theta}_m = 0 \cap |T| \leq T_{sj} \\ sgn(T) \cdot T_{sj} & \dot{\theta}_m = 0 \cap |T| > T_{sj} \\ T_{dj} & \dot{\theta}_m \neq 0 \end{cases} \tag{3.1}$$

where $T_f$ is the friction torque output from the system model, $T_{sj}$ is the static friction torque under static conditions and $T_{dj}$ is the dynamic friction torque in and dynamic conditions. The representation of the model is shown in the figure 3.2.

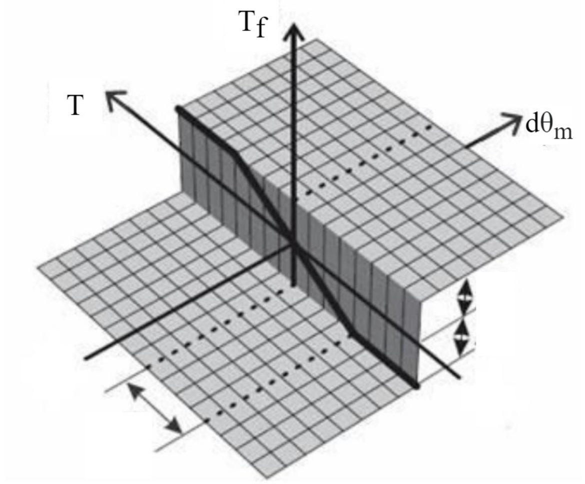


Figure 3.2: *Borello dry friction model representation*

Typically, in order to avoid instability phenomena, if a speed inversion occurs during the integration step, the mechanical system must be stopped, as described by the formula 3.2.

$$\dot{\theta}_m (t_i + 1) = 0 \qquad \text{if} \qquad \dot{\theta}_m (t_i + 1) \cdot \dot{\theta}_m (t_i) \leq 0 \tag{3.2}$$

However, this stop is not always correct, which is why in these circumstances the model described allows, as anticipated in the paragraph 2.1.6, that at the next integration step the unbalance of the overall forces acting on the system causes the system to be restarted by a reset command and the model returns to the adherence condition.
The figure 3.3 shows the Borello model implemented in the dynamic block of the system on Matlab-Simulink.

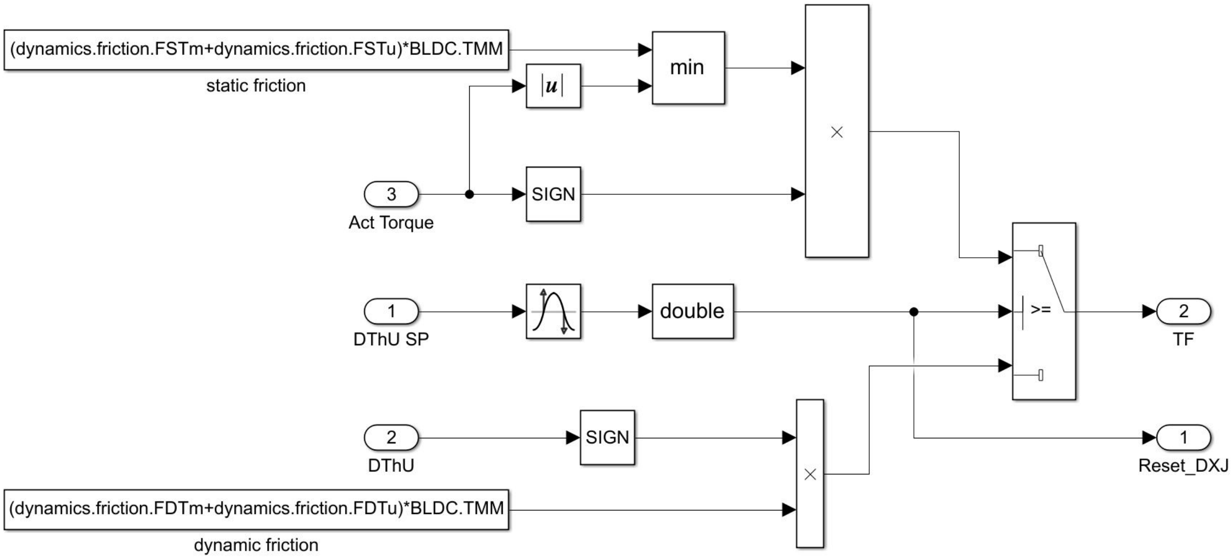

(dynamics.friction.FSTm+dynamics.friction.FSTu)*BLDC.TMM
static friction
|u|
min
×
3
Act Torque
SIGN
1
DThU SP
double
>=
2
TF
2
DThU
SIGN
×
1
Reset_DXJ
(dynamics.friction.FDTm+dynamics.friction.FDTu)*BLDC.TMM
dynamic friction

Figure 3.3: *Borello dry friction Simulink implementation*

Having defined the numerical model used, lets now look at the response of the system in the presence of the single dry friction fault. First a step command is considered, then a chirp command.
To obtain the curve in the figures below for the step control, the static and dynamic friction coefficients were varied from the nominal value (NF) to 3 times the nominal value.

In figures 3.4 and it is observed that as the friction increases, the position of the user tends more slowly to the commanded position because of the higher torque to overcome. This, for example, affects the readiness of the command.

The figure 3.5 show how the maximum speed attainable by the motor decreases as the friction force increases and the time constant diminishes, so the system becomes slower.

Finally, the equivalent current trend is shown in the figure 3.7. It can be seen that as friction increases, more power is required due to the higher torque required to maintain maximum rotation speed. As we have already seen, this results in a higher current demand. Certainly this condition is undesirable as an increase in current leads to greater dissipation due to the Joule effect which can lead to further damage to the material and circuits.

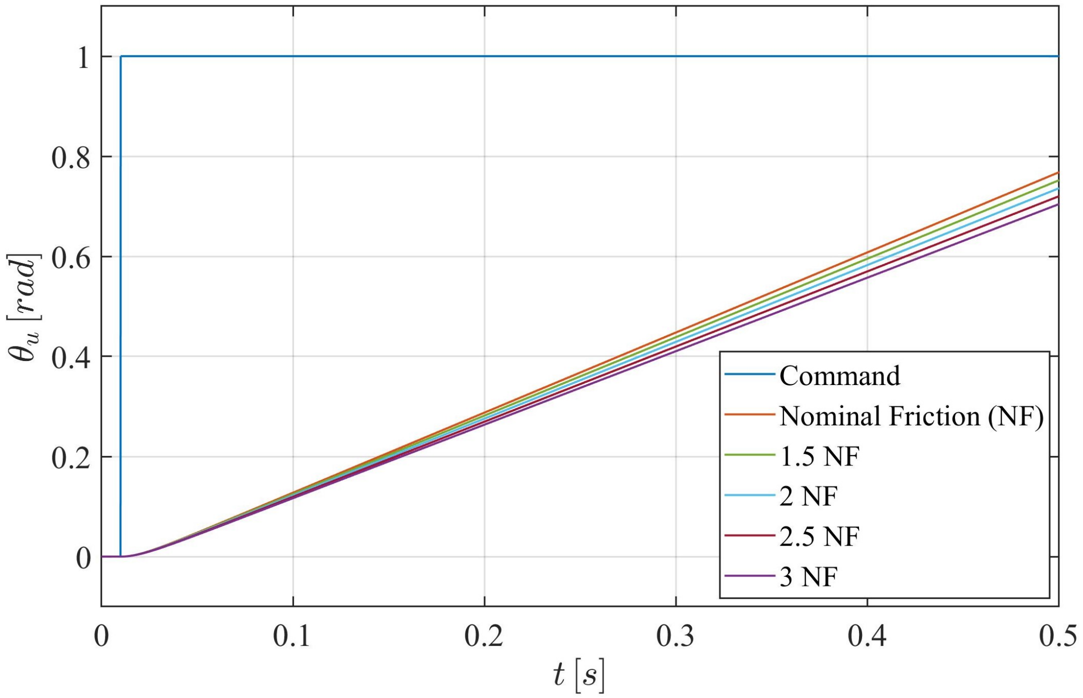


Figure 3.4: *User position for a step command - Dry Friction*

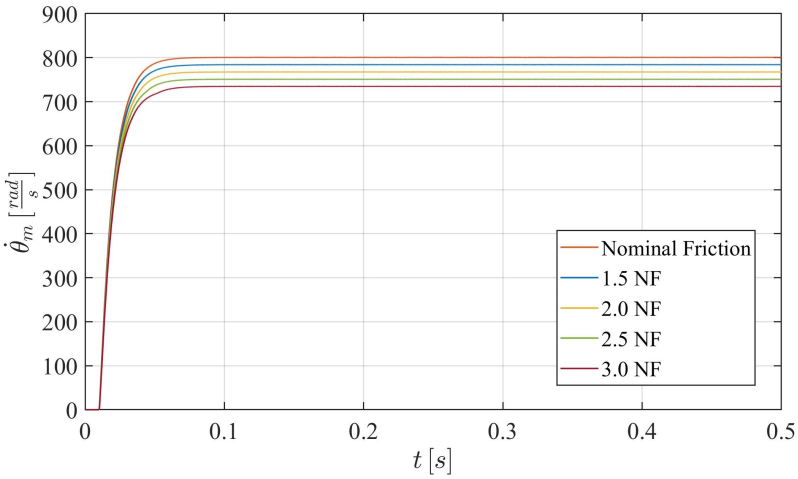


Figure 3.5: *Motor velocity for a step command - Dry Friction*

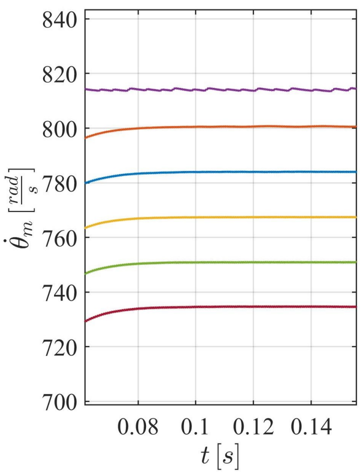


Figure 3.6: *Motor velocity for a step command - Zoom - Dry Friction*

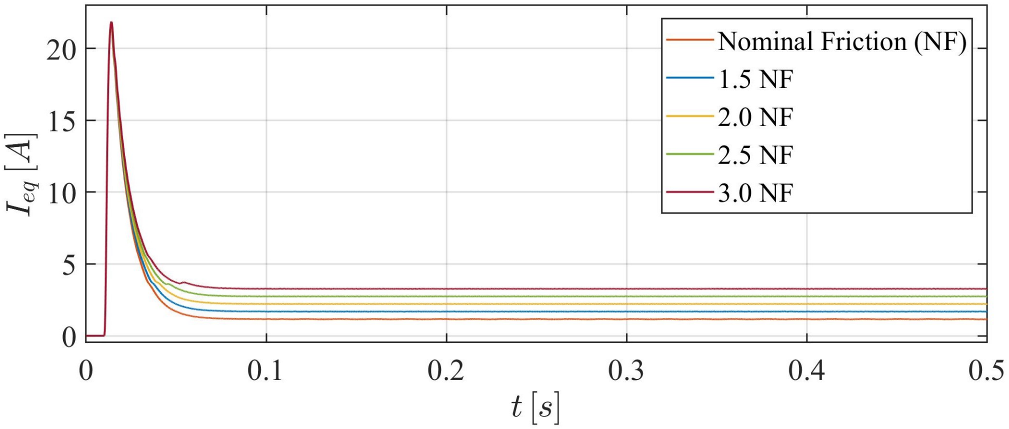


Figure 3.7: *Equivalent current for a step command - Dry Friction*

Now consider the system's response to the chirp command. Again, the friction failure was simulated by varying the friction coefficients between $NF$ and 3 NF.

The figures 3.8 and 3.9 shows that the position of the user is not significantly affected by the friction fault. In fact, the different friction conditions are superimposable, indicating that the control can always be well executed as the necessary current is still supplied. The impact of friction is slightly more pronounced at start-up and when reversing the control direction. In any case, this effect tends to diminish as the drive frequency increases.
The difference can be seen between the system response and the chirp command. This is due to the evaluation time of the controller, which does not allow instantaneous tracking; in addition, the dynamics of all the components and backlash present in the system must be taken into account.

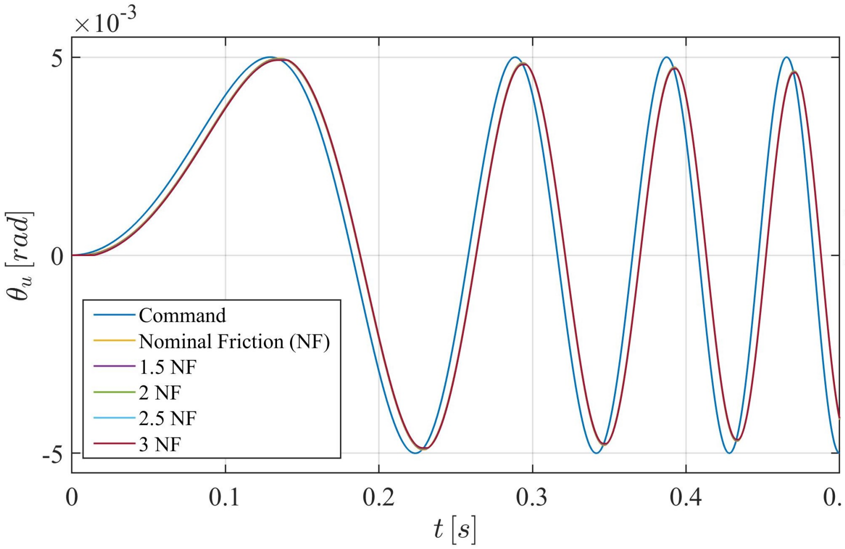


Figure 3.8: *User position for a chirp command - Dry Friction*

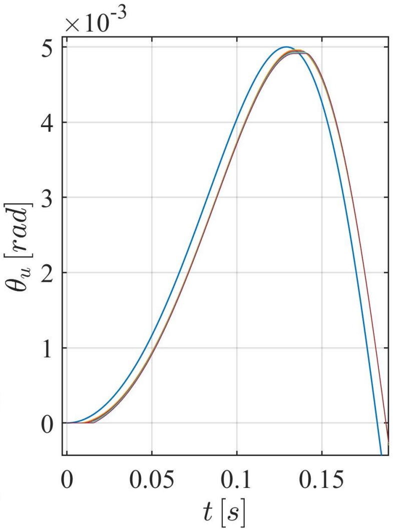


Figure 3.9: *User position for a chirp command - Zoom - Dry Friction*

The figures 3.10 and 3.11 show the system's response in terms of motor speed. It can be seen that the effect of friction is most influential when the motor speed is about 0. In fact, if the static friction coefficient is higher than the nominal value, a greater action will be required when changing direction in order to overcome the static friction torque. This translates into a longer adhesion time before motion restarts in the opposite direction before power is supplied to the system.

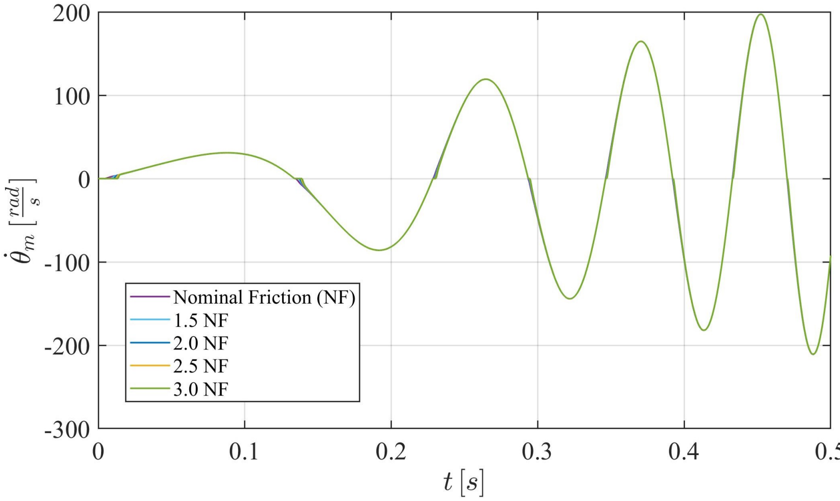


Figure 3.10: *User position for a chirp command - Dry Friction*

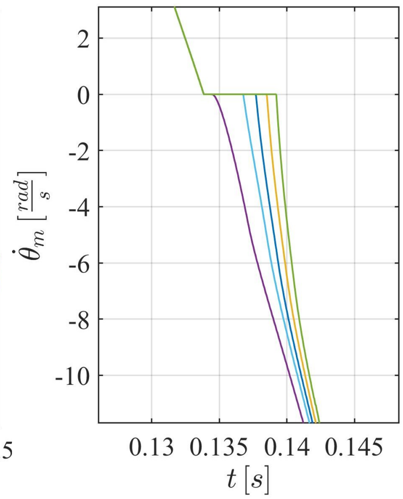


Figure 3.11: *User position for a chirp command - Zoom - Dry Friction*

Finally, the figure 3.12 shows the response of the system over time in terms of equivalent current as a function of the variation of the friction coefficients.

Confirming what has been said about the speed response, it can be observed that the current presents a peak at starting and at almost zero speed during direction reversal. This is related to the fact that static friction increases as the friction coefficients increase, so more torque is needed to counteract it. Since torque is directly proportional to current, as torque increases, current demand also increases. Once the adherence condition has been overcome, since the dynamic friction is taken into account, characterised by a lower friction coefficient than the static one, the torque and current requirements will also be lower. As the friction force increases, this effect is exacerbated.
It can also be observed that as the frequency of oscillation of the drive increases, the speed increases, so the effect of friction on the torque demand, and therefore on the current, is also less marked.

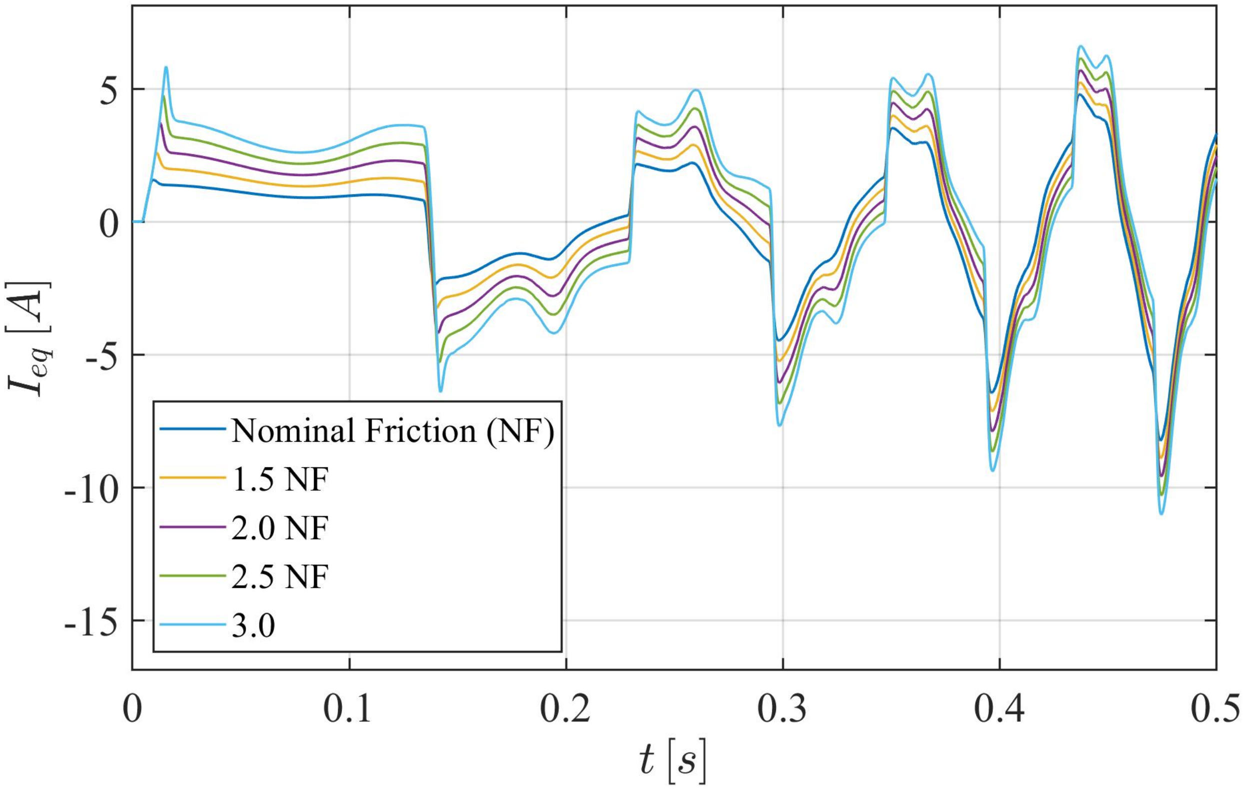


Figure 3.12: *Equivalent current for a chirp command - Dry Friction*

### 3.2.2 Backlash

Backlash is mechanical play that occurs between two coupled components in an assembly when a force is applied to one as a result of the movement of the other. It occurs, for example, in the joints that hold the components together, in the gears between the coupled toothed wheels, in the ballscrews between the ball and the raceway on which it runs or in the bearings.

Depending on the application, a certain amount of backlash is expected and accepted. In fact, a minimum clearance between the components must be ensured so that deflections under load and thermal expansion of the material can be guaranteed and a certain amount of lubrication is ensured. This space between the components is also linked to the geometric manufacturing tolerances which depend on the precision required for the components to perform at their best and the economic resources made available by the producer.

However, the continuous use of the components and the various stresses to which they are subjected have the consequence of facilitating component wear and thus increasing the gap and the play between the two components.
For this reason, the term backlash can be understood as both the error in the movement of the components or the gap between the components responsible for this error.

It is evident that the effect of an increase in backlash is undesirable, especially in precision position actuators such as the one under consideration; furthermore, it is considered that the effect is exacerbated in drives involving changes in the direction of motion. For this reason, when the fault is detected, the components should be replaced.
The effect of backlash is different depending on the speed at which the system is subjected. At higher speeds, i.e. at greater powers, mechanical vibrations increase and the components are subjected to unwanted shock loads which have a negative effect on the life and reliability of the transmission.
At lower frequencies, i.e. at reduced power, the effect of failures caused by fatigue of materials subjected to cyclically varying loads is predominant. In this case, backlash leads to inaccurate calculations due to small position errors introduced at each change of direction [38].

Backlash introduces nonlinearity into the model. In particular, it is defined by a dead band that defines the amount of backlash in the system. This is a simplification of the exact physical model; for example, all dynamic interactions between components, including internal damping of the shaft, are neglected. This is why the backlash block, which is placed in the two reference and monitor models in the module describing the system dynamics at the final ballscrew of the actuator, is considered to be an equivalent mechanical backlash that takes into account the plays that involve the mechanical transmission of the electromechanical actuator as a whole. In this way, as demonstrated in [39], the model is able to simulate the seizure and the wear that occurs when the balls slide in the ballscrew and the blockage of the bearings.

The effect of the fault is simulated in Matlab-Simulink through the percentage variation of the nominal value of the dead band corresponding to different fault entities.
The dead band value is calculated as

$$DB = \frac{BLK}{\tau} = \frac{10^{-5}}{\frac{1}{500}} = 5.0 \cdot 10^{-3} \tag{3.3}$$

Faults were then evaluated as 2, 5, 10, 50 and 100 times $BLK$ both for step and chirp command.

From the figures 3.13 and 3.14 it can be seen that for a step control, even for clearance values far in excess of the nominal conditions, the position of the user is not significantly changed. As we have said, in fact, the backlash is particularly evident for devices that carry out a reversal of the direction of rotation. A step control does not belong to this case this category and in fact the change in position is not very significant except when the control is started.

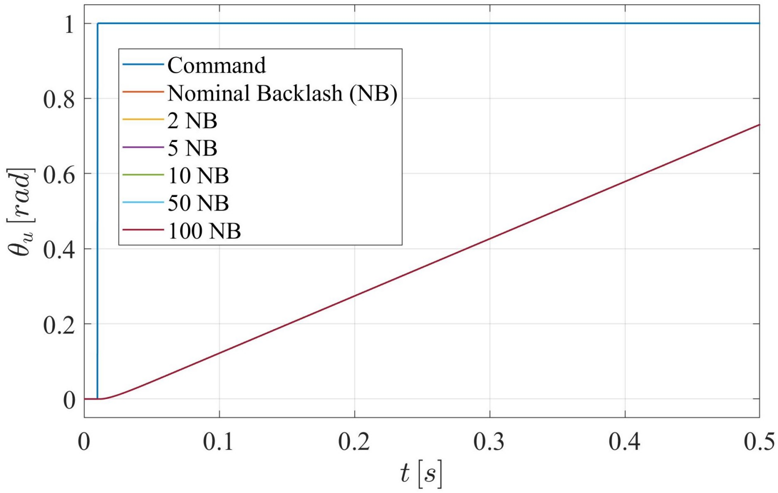


Figure 3.13: *User position for a step command - Backlash*

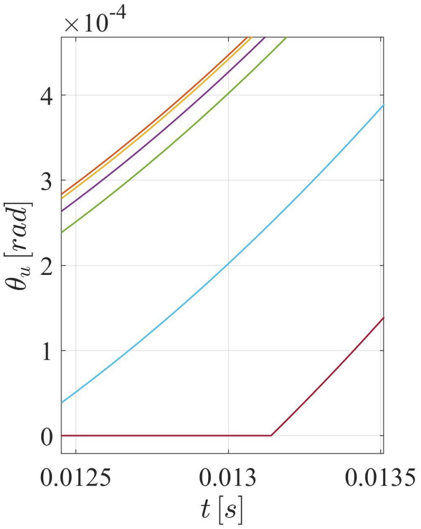


Figure 3.14: *User position for a step command - Zoom - Backlash*

In the figure 3.15, however, it can be seen that the backlash variation does not induce any change in the speed response of the motor. In fact, the fault only affects the position cycle and, consequently, the position error. For this reason, for any value of backlash, the curve is the same and corresponds to the nominal value.

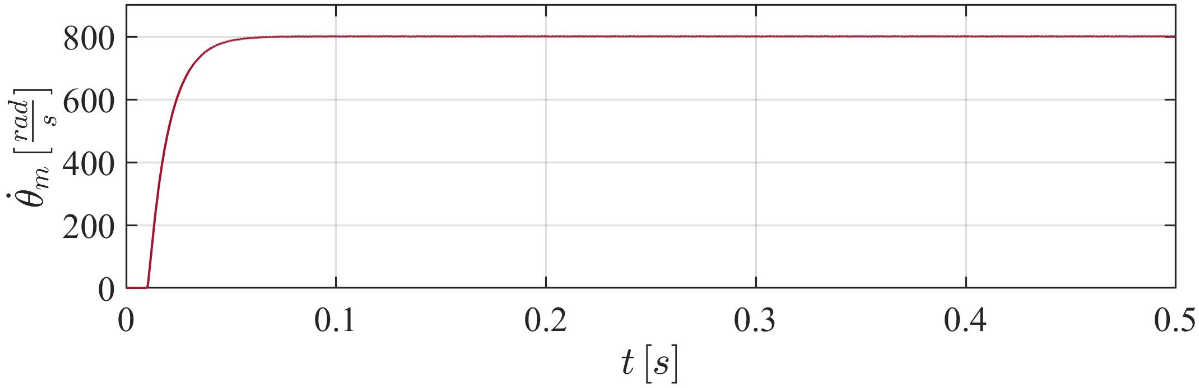


Figure 3.15: *Motor velocity for a step command - Backlash*

The same considerations are made for the current, whose trend is shown in the graph in the figure 3.16.

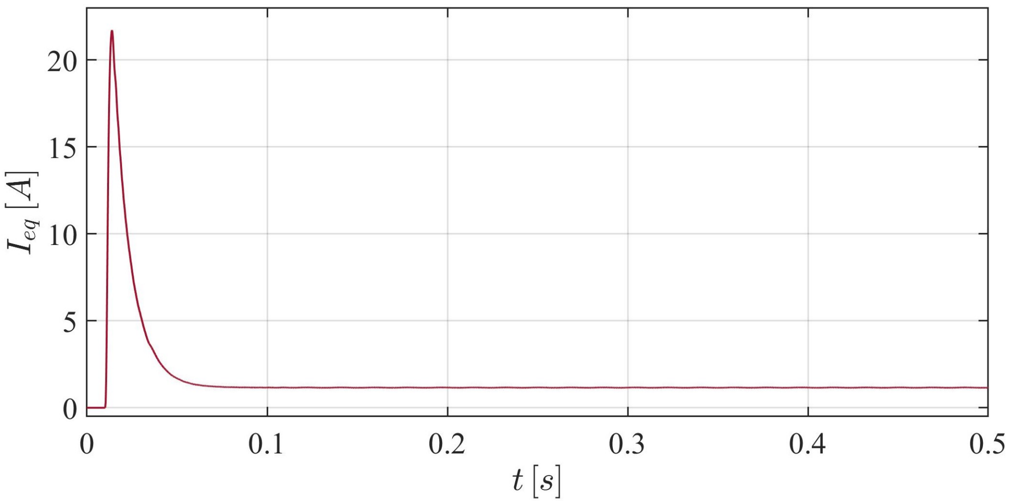
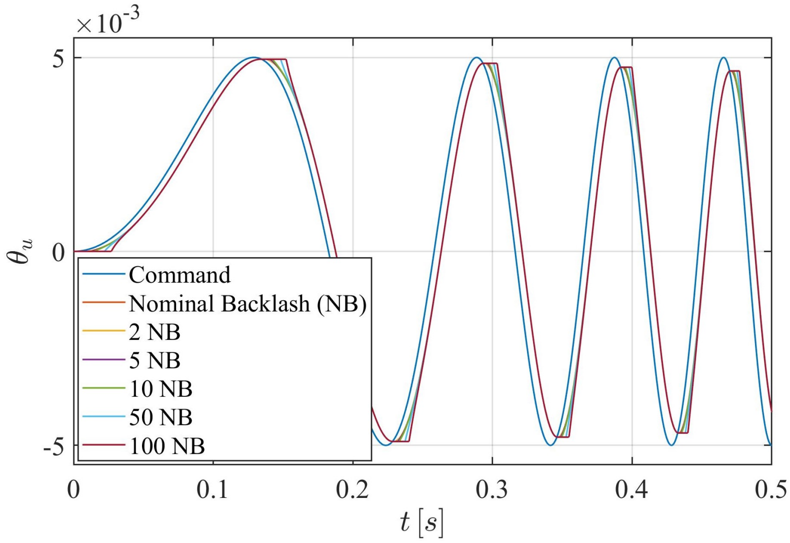


Figure 3.16: *Equivalent current for a step command - Backlash*

The results obtained for the chirp command are shown below.

Looking at the following images, it can be seen at first glance that the effect of backlash on the system's response to a chirp command is more distinct than in the previous case.

Particularly in the figures 3.17 and 3.18, it can be seen that with each change in the direction of motion, the delay induced by the fault is greater as the backlash increases. The effect is reduced as the frequency of the command increases, since the system is faster so the clearance is compensated more readily.



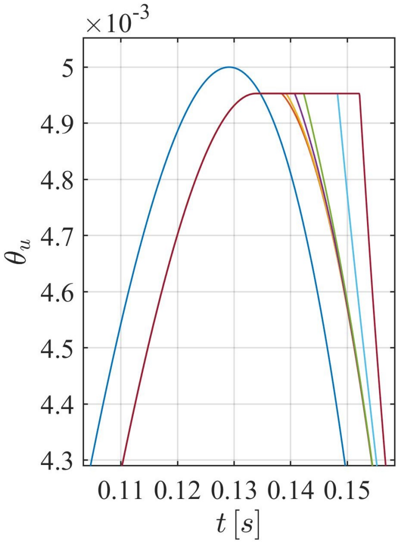


Figure 3.17: *User position for a chirp command - Backlash*

Figure 3.18: *User position for a chirp command - Zoom - Backlash*

Since speed is the time derivative of motor position, a position error will correspond to a speed error. This is why the graphs in the figures 3.19 and 3.20 show a speed peak at the change of direction.

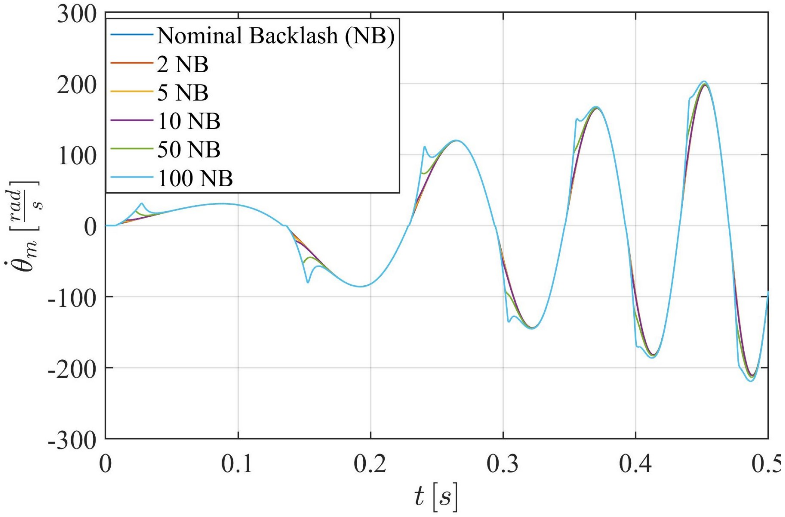


Figure 3.19: *Motor velocity for a chirp command - Backlash*

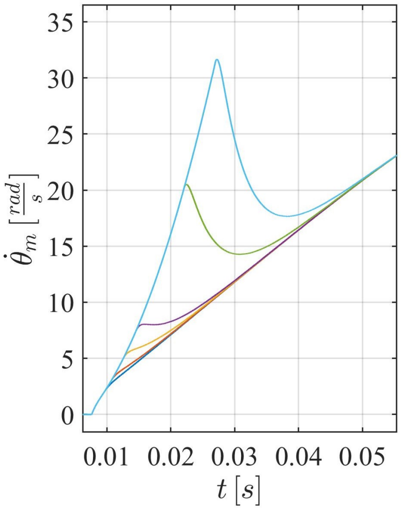


Figure 3.20: *Motor velocity for a chirp command - Zoom - Backlash*

For a similar reason to that discussed for friction in the paragraph 3.2.1, an increase in speed corresponds to an increase in the power required by the system, and therefore in the equivalent current. For this reason a current peak is always observed in the figures 3.21 and 3.22 at the same points, especially for very high backlash levels. In these cases, the system's speed is too great in relation to that required to move the system, so the current automatically tends to decrease in order to decelerate the system. As the frequency increases, the effect of backlash failure becomes increasingly apparent.

Current peaks are certainly not desired by the system, which may experience overheating that induces permanent damage in the components.

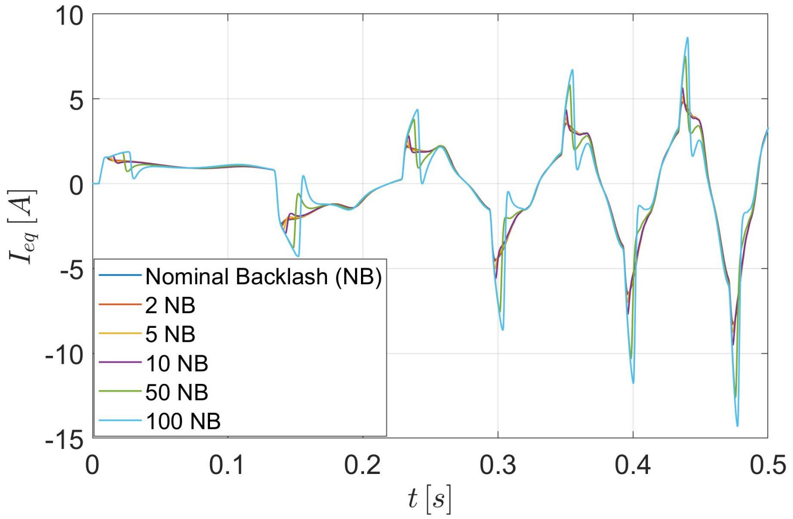


Figure 3.21: *Equivalent current for a chirp command - Backlash*

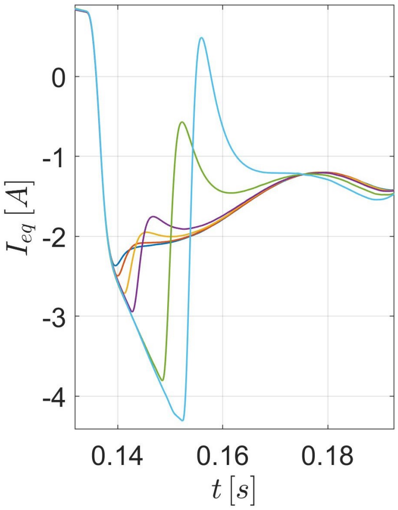


Figure 3.22: *Equivalent current for a chirp command - Zoom - Backlash*

### 3.2.3 Short Circuit

As we have already seen in the section 3.2.3, the electric motor under nominal operating conditions transforms electrical energy into mechanical one that is useful to the rotor shaft to drive the actuator. During operation, depending on the moment of inertia and the speed of rotation, a certain amount of dynamic energy is stored.

If a short circuit occurs, i.e. if there is contact between relatively low resistances or impedances in the circuit, the current tends to flow through the point of connection, as there is less resistance to its flow. However, the accumulated energy is still able to keep the rotor rotating and the kinetic energy is converted into electrical energy; this provides the short circuit itself with a fault current in addition to the supply current, which is not expected under the rated operating conditions. The overcurrent, known as *short-circuit current*, is very high in the short-circuited coils, so it causes localised heating and promotes rapid spread of the fault [40].

The main cause of the short circuit is the deterioration of the insulating materials of the conductors [41], which is amplified by:

- increased operating temperature;
- voltage fluctuations;
- increase in voltage due to pulse width modulated switching;
- partial electrostatic discharges;
- inadequate cooling;
- insufficient maintenance.

Generally, stator winding insulation failure starts with a short circuit between coils of the same phase, which is why it is called a *coil-coil short circuit.* In this case the motor can continue to run, but for a limited time depending on various conditions. It is important, however, that the fault is detected in advance because, as already mentioned, the fault tends to propagate easily and induce *phase-to-phase* or *phase-to-ground* short circuits. The latter two types of failure are instantaneous, the motor is no longer able to run and no corrective action can be taken.

For this reason, the coil-coil type of fault is considered in the discussion.

The short-circuit fault has been implemented differently in the two reference and monitor models. The reference model is considered first.

The fault can be assessed by evaluating the external magnetic field flux, which influences the motor counter-electromotive force coefficient $k_E$ (equal to the torque gain $G_M$) according to the formulation

$$k_E = G_M = \frac{\partial \phi}{\partial \theta_m} = NA\,\frac{\partial\left(\int_A \int B \cdot \hat{n}\, dS\right)}{\partial \theta_m} \tag{3.4}$$

where $N$ is is the winding number, $A$ the winding area and $B$ is the rotor magnetic flux density. This formula is a compact equation for the three formulations 2.11, 2.12 and 2.13 seen in the paragraph . By imposing a percentage $N_i$ of the winding affected by the fault for each phase $i$, the counter-electromotive force coefficient for each phase and for that specific fault state can be calculated as follows.

$$k_{E,i} = k_E \cdot N_i \tag{3.5}$$

For a coil-coil short circuit, the resistance and inductance values for that specific phase are respectively

$$R_i = \frac{R_s}{2 \cdot N_i} \tag{3.6}$$

$$L_i = \frac{L_s}{2 \cdot N_i^2} \tag{3.7}$$

where $R_s$ and $L_s$ are the resistance and inductance of the circuit under nominal operating conditions.
However, in the case of a phase-to-phase short circuit the resistance and inductance values are calculated as follows.

$$R_{i,j} = \frac{R_s}{2 \cdot (N_i + N_j)} \tag{3.8}$$

$$L_{i,j} = \frac{L_s}{2 \cdot \left(N_i^2 + N_j^2\right)} \tag{3.9}$$

Now, applying the circuit equation 2.15 and knowing that when the short circuit occurs the voltage is null, the value of the current in the circuit for each phase can be easily derived.

Note that under nominal conditions the percentage of windings not affected by short-circuit failure is 100% in each phase, so $N_A = N_B = N_C = 1$. It follows that the values of resistance and inductance are equal to

$$R_i = \frac{R_s}{2} \tag{3.10}$$

$$L_i = \frac{L_s}{2} \tag{3.11}$$

In the monitor model, the evaluation of resistance, inductance and counter-electromotive force coefficient was carried out in a different and very simplified manner. In fact, since the equivalent current will be considered, we must take into account the equivalent values of $N$

to assess the total equivalent percentage of windings affected by the failure.

$$N_{eq} = \frac{N_A + N_B + N_C}{3} \tag{3.12}$$

Finally, given the values of the counter-electromotive force coefficient $k_{cfem_{eq,NC}}$, torque gain $G_{M_{eq,NC}}$, resistance $R_{eq,NC}$ and inductance $L_{eq,NC}$ in nominal conditions, the respective equivalent values can be calculated. The following relationships are used for this purpose:

$$k_{cfem_{eq}} = k_{cfem_{eq,NC}} \cdot N_{eq} \tag{3.13}$$

$$R_{eq} = R_{eq,NC} \cdot N_{eq} \tag{3.14}$$

$$L_{eq} = L_{eq,NC} \cdot N_{eq}^2 \tag{3.15}$$

$$G_{M_{eq}} = G_{M_{eq,NC}} \cdot N_{eq} \tag{3.16}$$

As mentioned in the paragraph 2.2.2, the number of phases actually involved influences the values of counter-electromotive force and motor torque.
In order for the monitor model to be as faithful as possible to the reference one, a correction linked to the winding coils was introduced. This correspondence uses the knowledge of the angular position of the motor to determine the electrical characteristics of the motor, in particular the contribution of each individual phase to the short-circuit fault $N_i$ and the total contribution of the three phases $N_P$. The error between the two models was minimised by taking into account the following correction function:

$$f(\theta_m) = \frac{2}{3} N_P \left[ N_A \sin^2 \theta_m + N_B \sin^2 \left( \theta_m - \frac{2\pi}{3} \right) + N_C \sin^2 \left( \theta_m + \frac{2\pi}{3} \right) \right] \tag{3.17}$$

To recap, the effects that short-circuit correction has on the electric motor modelling block in the monitor model are:

- it changes the counter-electromotive force coefficient;
- induces a decrease in equivalent resistance;
- induces a decrease in torque gain.

The Matlab-Simulink implementation of the block in discussion is shown in the figure 3.23.

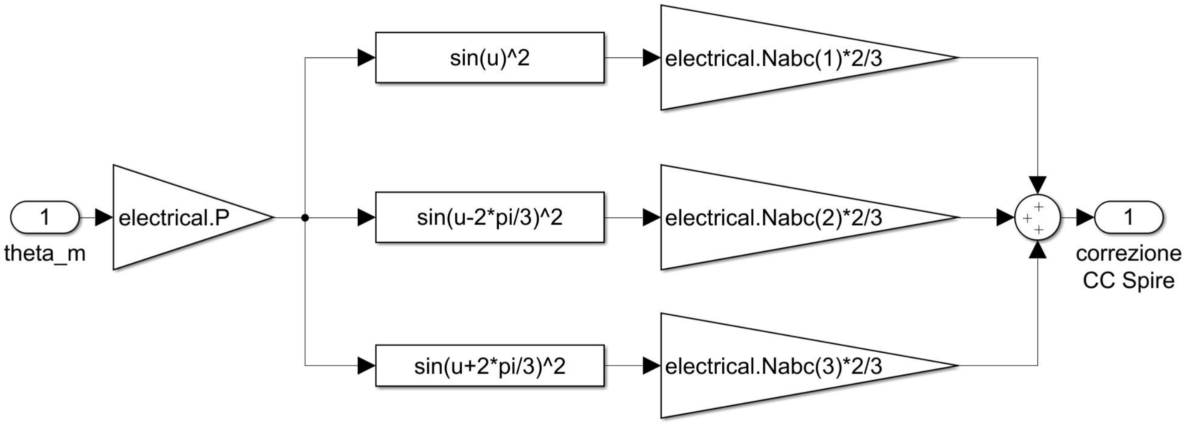


Figure 3.23: *Short-circuit correction block - Monitor Model*

The following are the results obtained from the simulation in terms of the dynamic response to a step command and a chirp command under conditions of malfunction due to a coil-coil short circuit. For this reason, in both cases the fault was implemented for a single phase, in particular for phase $C$. The failure rate was varied from 0% to 100%, i.e. $N_C$ was set equal to 1, 0.75, 0.50, 0.25 and 0. It should be noted that in the implementation of short-circuit at 100% a value of $N_C = 0$ was not set as this would have led to errors during the simulation. For this reason, the lowest possible value of $10^{-6}$ was imposed.

The dynamic response to the step command is analysed first.

The number of working windings in a phase affects the amount of magnetic flux developed by the defective phase itself. This is proportional to the electromotive force constant, which consequently decreases as the magnitude of the fault increases. In addition, the inductance, defined as the ratio between the magnetic field flux through the surface of the coil and the current intensity through it, also decreases. As we have already seen, in the presence of a short circuit the circuit resistance decreases and the current increases by compensating for the reduction in the torque constant due to the reduced effective number of windings.

As can be seen from the figures 3.24, 3.25, 3.26 and 3.27 as the number of turns involved in the fault increases, the regime speed increases, so the angular position reached by the user in the same time interval is greater. In particular, it can be observed that the system is slower during the transient period for larger fault entities, while at steady state it reaches higher speeds. It is also observed that the introduction of the coil-coil short-circuit fault introduces visible oscillations on the speed response.

In addition, there is a significant difference in terms of user position and motor speed in the transition from the nominal condition to the generic fault condition. In fact, the magnitude of the variations of the two quantities are progressively smaller as the failure rate increases from the condition of 25% short-circuited turns.

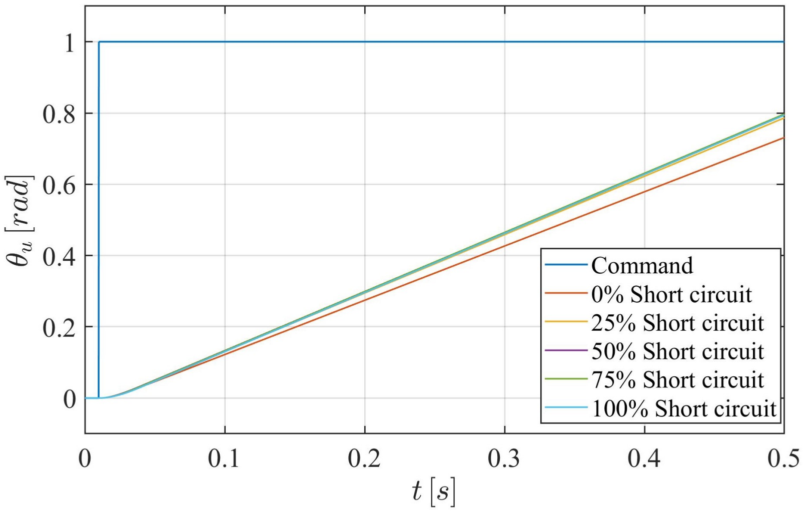


Figure 3.24: *User position for a step command - Short Circuit*

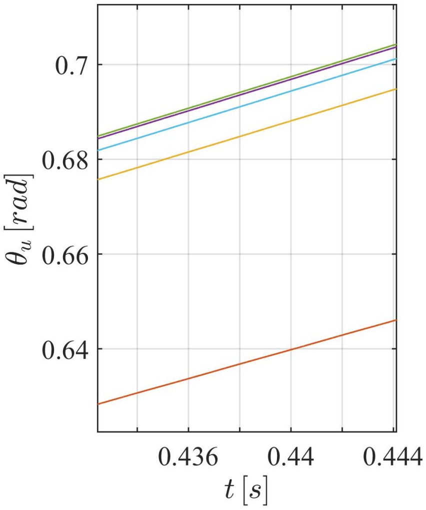


Figure 3.25: *User position for a step command - Zoom - Short Circuit*

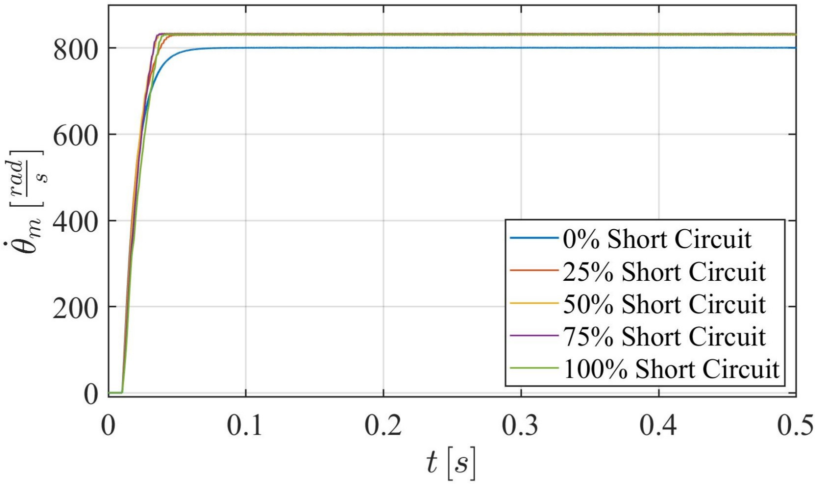


Figure 3.26: *Motor velocity for a step command - Short Circuit*

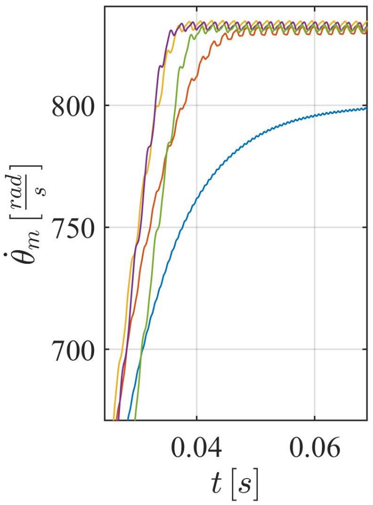


Figure 3.27: *Motor velocity for a step command - Zoom - Short Circuit*

Instead, the figures 3.28 and 3.29 show the evolution of the equivalent current as a function of the fault percentage. As the number of turns operating during the transient decreases, the current value tends to increase. Particularly during the transient, the oscillations introduced by the fault on the current pattern are more evident for an advanced failure state. When fully operational, however, the fluctuations tend to decrease.

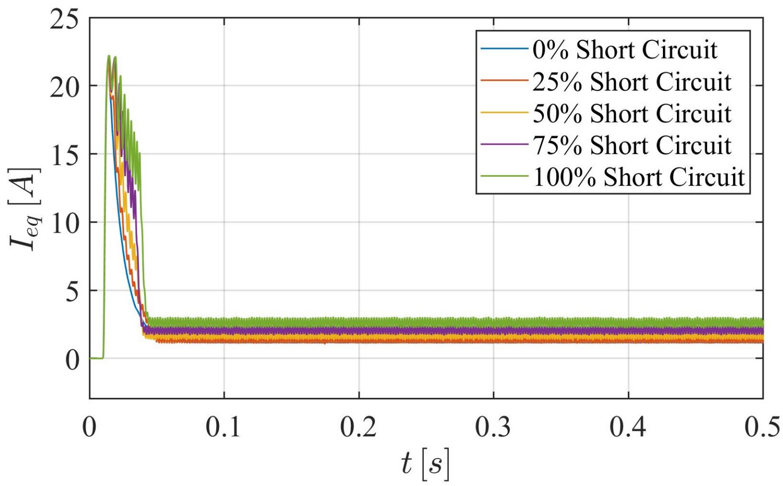


Figure 3.28: *Equivalent current for a step command - Short Circuit*

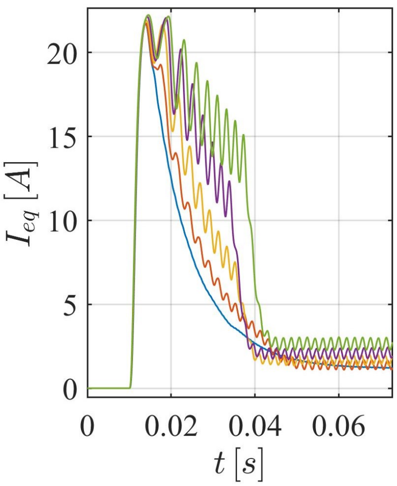


Figure 3.29: *Equivalent current for a step command - Zoom - Short Circuit*

The simulation results obtained for the system subjected to the chirp command are now reported.

In the figures 3.30, 3.31, 3.32 and 3.33 it can be seen that the position of the user and the speed of the motor are not particularly affected by the presence of the short circuit, except to a very small extent in the change of direction of the motor, as seen for the dry friction fault.

In addition, fluctuations around the nominal 0% fault condition are evident in the figure 3.32, related to the periodic activation of the phase $C$ subjected to the short circuit conditions.

This is also reflected in the figure 3.34 where it seen that the failure of a single phase has significant effects on the dynamic current response. In fact, the system is subjected to a strong current increase when phase $C$ is activated. When this is inactive, the current resumes its nominal trend, as seen in the figure 3.34. The increase in current is more evident the higher the frequency of the command.

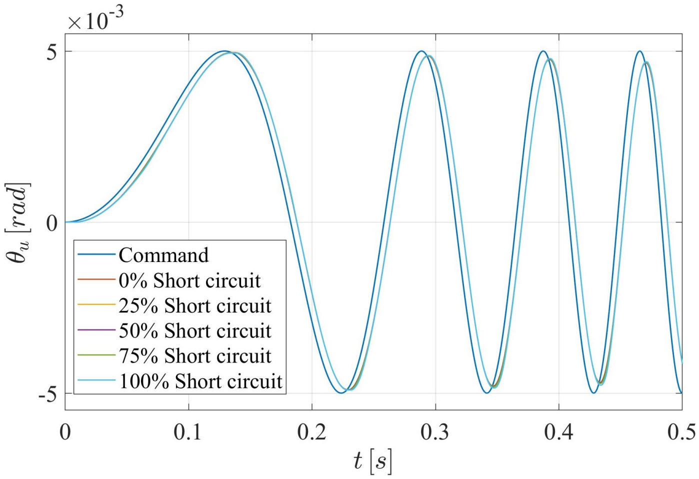


Figure 3.30: *User position for a chirp command - Short Circuit*

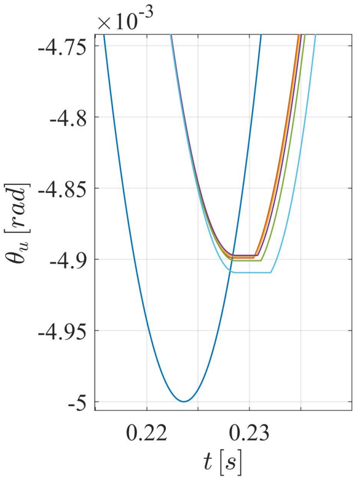


Figure 3.31: *User position for a chirp command - Zoom - Short Circuit*

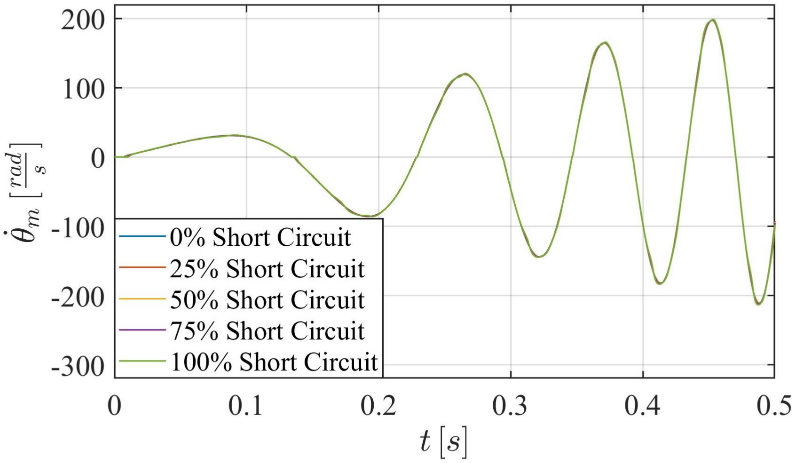


Figure 3.32: *Motor velocity for a chirp command - Short Circuit*

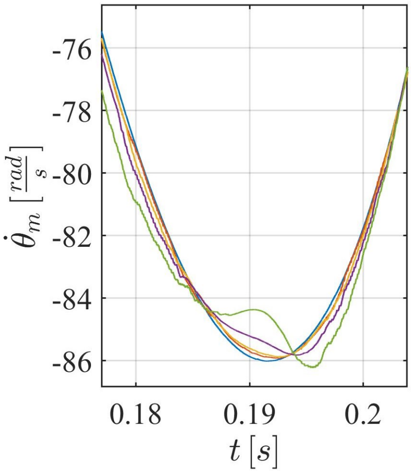


Figure 3.33: *Motor velocity for a chirp command - Zoom - Short Circuit*

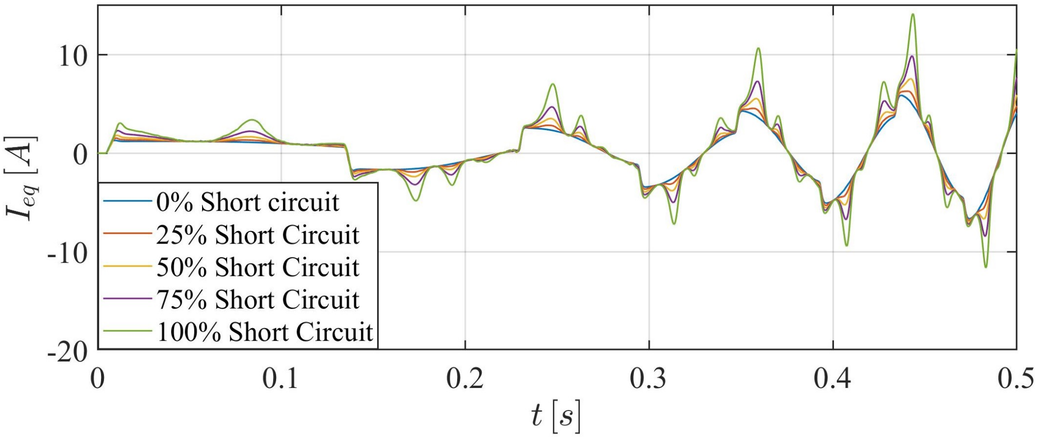


Figure 3.34: *Equivalent current for a chirp command - Short Circuit*

### 3.2.4 Eccentricity

Eccentricity, in general, is a mechanical failure that occurs when the rotor and stator of a motor rotate around different axes so that their mutual position is no longer concentric.

However, given the electrical nature of the motor, this type of failure has strong implications for the electromagnetic interactions in the system.
In fact, it should be remembered that a rotating electric machine has several sources of vibration [42]:

– response of the stator core to the attractive force developed between rotor and stator;

– response of the stator windings to the electromagnetic forces in the conductors;

- dynamic behaviour of the rotor;
- response of the shaft bearings to the vibration transmitted by the rotor.

These types of response are mutually dependent and it is inevitable that the occurrence of one will affect all the others.
In our case, the misalignment of the shaft due to tolerances or uncertainties during motor manufacture or the progressive wear of the rotor shaft bearings induces a certain value of eccentricity; this in turn causes a periodic force on the motor which vibrates the stator.

During motor operation, two phenomena are at play: the *Lorentz Force*, which is exerted on the current-carrying conductor immersed in the magnetic field, and the *Maxwell Force*, which is produced between the ferromagnetic structures of the rotor and stator as a result of the magnetic flux in the air gap. The latter, as discussed below, is strongly impacted by eccentricity [42].

In particular, Maxwell's forces act symmetrically and perpendicularly to the rotor and stator surfaces, so that under nominal conditions, i.e. when perfect concentricity occurs, the resultant is zero over the whole system.
A simplified representation of the rotor and stator sections in nominal conditions is shown in the figure 3.35.

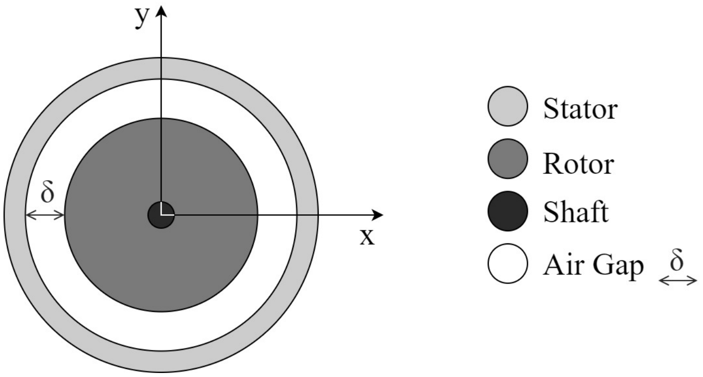


Figure 3.35: *Nominal condition - Eccentricity*

Under these conditions the flux at the air gap $\phi$ is directly proportional to the magnetomotor forces $M$ and inversely proportional to the reluctance $R$ of the magnetic circuit traversed by the flux, according to the simplified relation 3.18.

$$\phi = \frac{M}{R} \tag{3.18}$$

Reluctance is defined as

$$R = \frac{2\delta}{\mu_0 S} \tag{3.19}$$

where $\mu_0$ is the magnetic permeability in vacuum, $\delta$ the air gap thickness and $S$ the normal surface through which the flow passes.

It follows that the magnetomotive force is equal to

$$M = \phi R = BS\frac{2\delta}{\mu_0 S} = \frac{2\delta B}{\mu_0} \tag{3.20}$$

where $B$ is the flux density at the air gap.

The main harmonic component of the magnetomotive force has a sinusoidal distribution whose period depends on the number of pole pairs in the motor and whose amplitude is a function of the supply frequency. Consequently, the magnetic flux in the air gap also maintains the same dependence. However, under nominal conditions, integrating the magnetic field $B$ along the air gap gives a zero resultant, as mentioned above [42].

Under non-nominal conditions, two eccentricity conditions can occur, *static* and *dynamic* eccentricity, represented in figures 3.36 and 3.37 respectively.

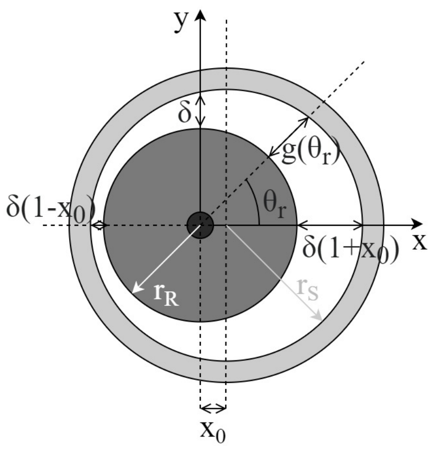


Figure 3.36: *Static eccentricity*

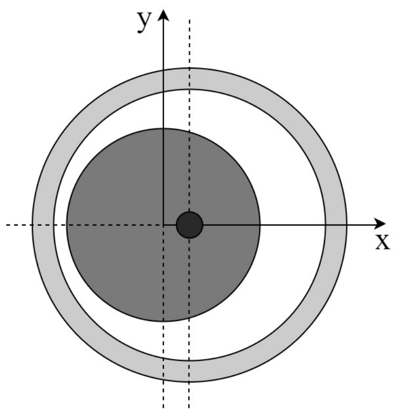


Figure 3.37: *Dynamic eccentricity*

In the case of static eccentricity, the rotor, solid with the reference system $xy$, rotates about its own axis, which coincides with the axis of the shaft, but does not correspond to the axis of stator symmetry with respect to which it is at distance $x_0$. The rotor in this case is symmetrical with respect to its own axis and there is no mechanical unbalance, so that the minimum air gap ($\delta(1 - x_0)$ in the figure 3.36) remains fixed in space and time.

If this time we integrate the magnetic field along the air gap, we observe a resultant different from 0 in the direction of the minimum air gap. Therefore, the presence of the static eccentricity induces an *Unbalanced Magnetic Pull* (UMP) on the rotor, which tries to move the geometrical centre of the rotor away from that of the stator in the direction of the minimum air gap with an intensity that varies depending on the angular position of the rotor $\theta_r$. This additional force, which is not foreseen under nominal conditions, leads to an increase in vibration up to excessive stresses on the motor and an acceleration of bearing wear. Such a failure may also develop into a real sliding of the rotor on the motor leading to insulation failures of the stator windings [42].

In the case of dynamic eccentricity, on the other hand, the rotor rotates around the stator axis but not around its own axis; this generates a mechanical imbalance by adding a centrifugal force which rotates at the speed of rotation of the motor. This type of fault has not been implemented as it cannot be assessed by simply analysing current and speed trends, but an evaluation of vibration harmonics at different frequencies is carried out.

Referring to the figure 3.36, we can evaluate the expression air gap $g$ as a function of rotor angle $\theta_r$. Consequently we can evaluate the expression of the magnetic flux $\phi$ flowing in the air gap.

For the two circumferences of the rotor and stator, we can write relations 3.21 and 3.22 respectively as a function of the coordinates $x$ and $y$.

$$x^2 + y^2 = r_r^2 \tag{3.21}$$

$$(x - x_0)^2 + y^2 = r_s^2 \tag{3.22}$$

Introducing polar coordinates, knowing that $\rho = r_r$, we can write

$$\begin{cases} x = \rho \cos(\theta_r) \\ y = \rho \sin(\theta_r) \end{cases} \tag{3.23}$$

By substituting relation 3.23 into relations 3.21 and 3.22, combining equations 3.21 and 3.22, we obtain the expression

$$\rho^2 - 2\rho x_0 \cos(\theta_r) + x_0^2 - r_S^2 = 0 \tag{3.24}$$

If we define $\delta = g_0 = r_s - r_r$ as the nominal air gap thickness, we can write the expression of the air gap as a function of the angular position of the rotor as

$$g(\theta_r) = x_0 \cos(\theta_r) + g_0 \implies g(\theta_r) = g_0 \left[1 + \zeta \cos(\theta_r)\right] \tag{3.25}$$

where $\zeta = \frac{x_0}{g_0}$.

Having obtained the expression of the air gap as a function of $\theta_r$, by substituting in the relations 3.20 and 3.18, we obtain the function of the air gap flux as a function of the angular position of the rotor.

As we have seen, the treatment of eccentricity is inseparable from the electrical treatment of the motor, so describing electrical degradation phenomena such as short circuits by neglecting the eccentricity of the rotor does not allow an accurate description of the problem.

However, in the reference model the coupling of electrical and mechanical behaviours has been implemented in Matlab-Simulink is well evident in the motor model, through the calculation and implementation of the counter-electromotive force constants, as anticipated in the paragraph 3.2.3. In this way, the magnetic interactions between the rotor and stator are accurately represented.

In the monitor model, as already seen for the short circuit, a correction block has also been implemented for eccentricity, shown in the figure 3.38.

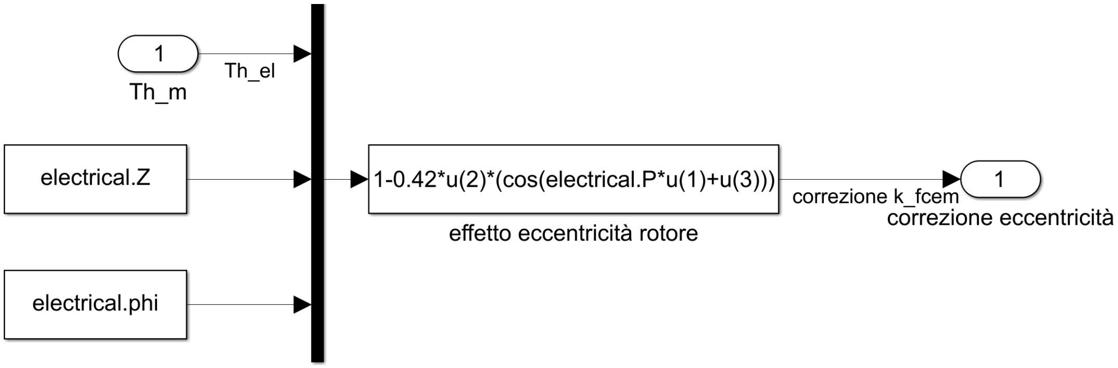


Figure 3.38: *Eccentricity correction block - Monitor Model*

As a first input parameter, it receives the value $\zeta$ representing the magnitude of the failure: it varies between 0 and 1, two values that correspond respectively to the perfect coincidence of the axes of the rotor and stator and to the absence of air gap between rotor and stator resulting in their mutual contact.
The second input parameter is the eccentricity phase $\phi$, which represents the angular position, with respect to the reference system, to which the minimum air gap corresponds, i.e. where the offset occurs. The value of $\phi$ also varies between 0 and 1. In fact, it represents the normalised value of the mechanical or electrical angle, according to the relations given below.

$$0 \leq \phi \leq 1 \iff 0 \leq \tilde{\phi}_m \leq (2\pi)_m \iff 0 \leq \tilde{\phi}_e \leq (4\pi)_e \tag{3.26}$$

where $\tilde{\phi}_m$ represents the mechanical angle and $\tilde{\phi}_e$ is the electricl angle.
Finally, the eccentricity correction block receives as input the rotor angle, which is converted into an electrical one.

These values are processed by the form function 3.27 obtained by trial and error procedure, so that the appropriate correction of the electrical parameters of the motor can take place as a function of the angular position.

$$\phi_E = 1 - 0.42 \cdot \zeta \left[\cos\left(\theta_e + \tilde{\phi}_e\right)\right] \tag{3.27}$$

The dynamic responses of the system in the presence of a variation of eccentricity to step and chirp commands are now considered.
The variation of the entity of failure is obtained by varying the coefficient $\zeta$ between the extremes 0 and 1.

The figures 3.22 and 3.40 show that as the eccentricity increases, the system is slower both in transient and steady state. It can be seen that the effect described due to the increase in eccentricity is opposite to that induced by the progression of the short circuit fault.

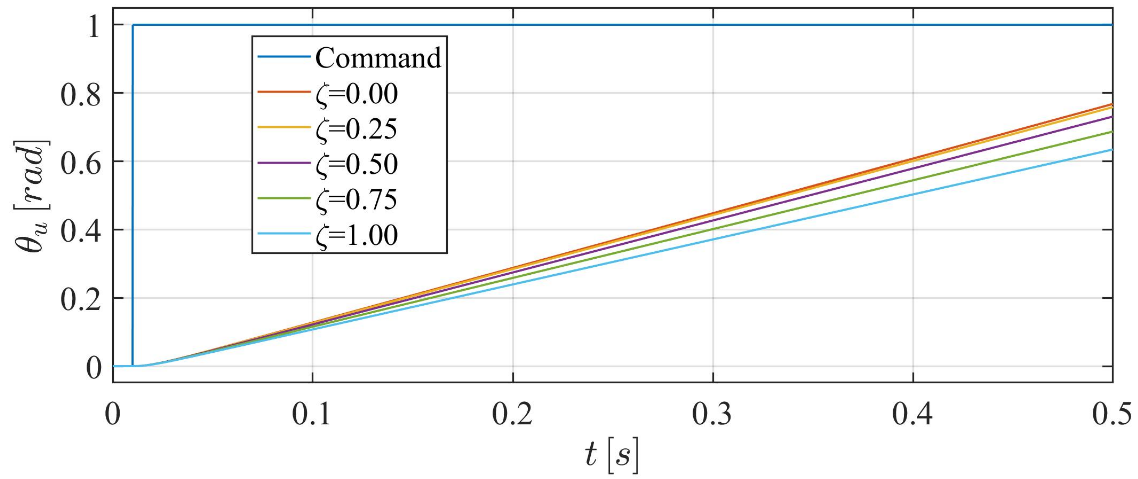


Figure 3.39: *User position for a step command - Eccentricity*

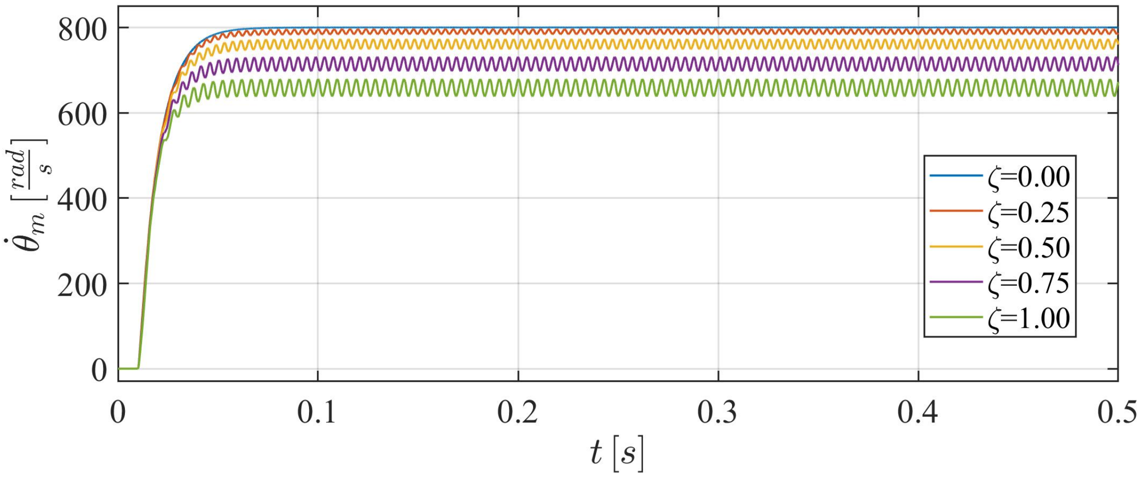


Figure 3.40: *Motor velocity for a step command - Eccentricity*

Again, the introduction of the fault introduces oscillations in the speed and current responses, as can be seen in the figures 3.40 and 3.41.

As we have seen, static eccentricity produces an increase in magnetic induction where the air gap is thinner. This produces an anisotropy in the flux path, so that in order to achieve the same value of rotor speed, it is necessary for the affected motor to have a higher current value than in the nominal case.

This can be appreciated in the figures 3.41 and 3.42, where it can be seen that an increase in the failure rate corresponds to a greater amplitude of the current ripple and a higher mean value.

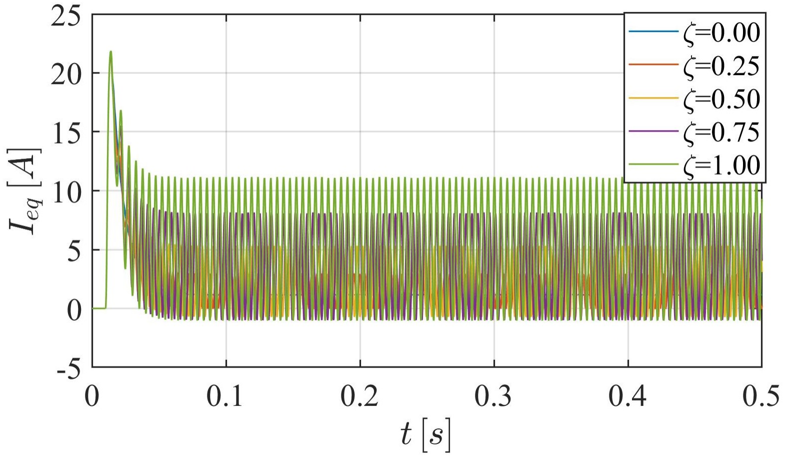


Figure 3.41: *Equivalent current for a step command - Eccentricity*

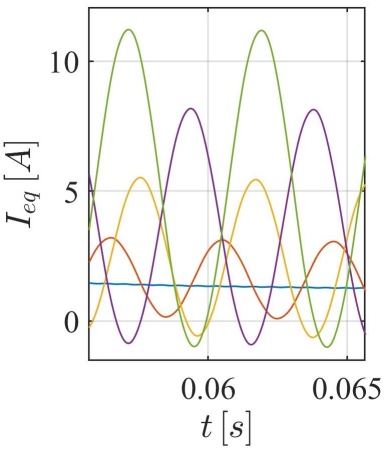


Figure 3.42: *Equivalent current for a step command - Zoom - Eccentricity*

The dynamic response of the system to a chirp command in the presence of an eccentricity fault is now considered.

As in the case of the short circuit, the position of the user, shown in the figure 3.43, is not greatly affected by the variation in eccentricity of the rotor. A slightly more significant deviation occurs when the control direction is reversed, as can be seen in the figure 3.44. Here it can be seen that even for the chirp command, the opposite trend is observed with respect to the case of the short circuit, i.e. as the magnitude of the fault increases, the response speed is lower.

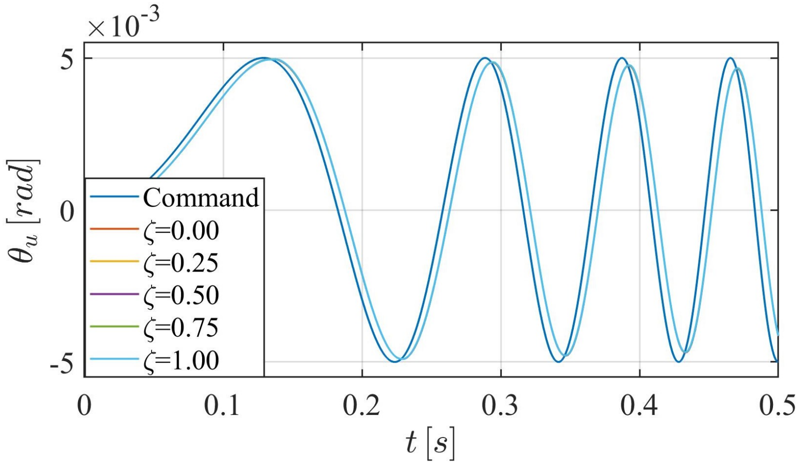


Figure 3.43: *User position for a chirp command - Eccentricity*

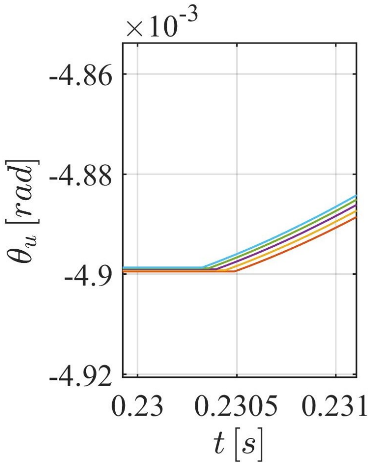


Figure 3.44: *User position for a step command - Zoom - Eccentricity*

The same observation can be made for speed response. In fact, the motor is driven by electronic commutation that provides a certain frequency to reach the rotor speed. For this reason, the controller, in the presence of unbalanced magnetic forces, is able to recognise position and speed errors and correct them to ensure that the nominal speed and position are maintained.

However, as can be seen in the figures 3.45 and 3.46, when the sign of the speed is reversed there is a slight delay in the response which tends to decrease as the speed increases. This is due to the fact that at high speed values the differences in vibration spectra between the non-faulty and faulted motor are less noticeable. This is due to two aspects: closed-loop control cancels out the effects caused by rotor eccentricity, especially at high speeds; moreover, at high speeds the iron between the stator teeth around the minimum thickness air gap is close to saturation and causes an apparent reduction in eccentricity.

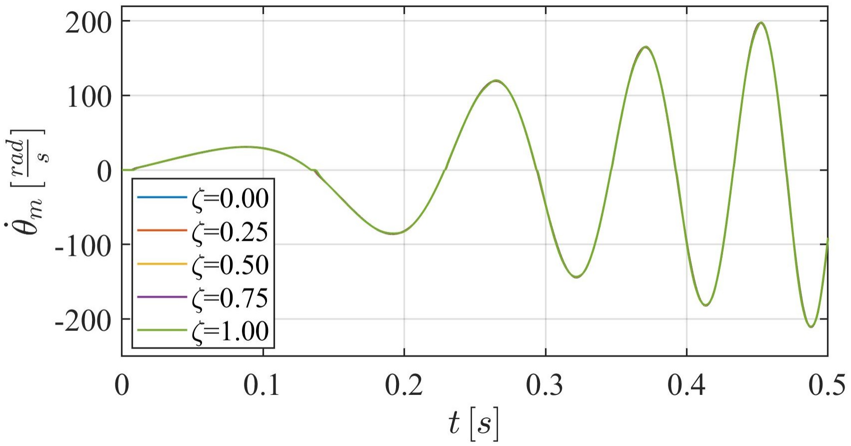


Figure 3.45: *Motor velocity for a chirp command - Eccentricity*

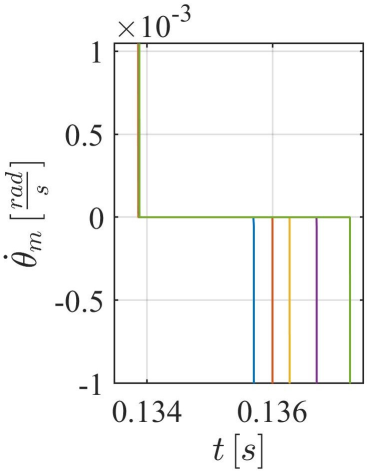


Figure 3.46: *Motor velocity for a step command - Zoom - Eccentricity*

Finally, the current response to a chirp command in the presence of an eccentricity fault is shown in figure 3.47.

As already seen for the step command, the current value increases as the fault magnitude increases. The effect is greater the higher the frequency of the command.

In this case it is evident that two effects overlap: one linked to the variation of the fault entity depending on the angular position of the motor which varies sinusoidally according to the variation of the air gap thickness; the other is linked to the activation/deactivation of the motor phases.

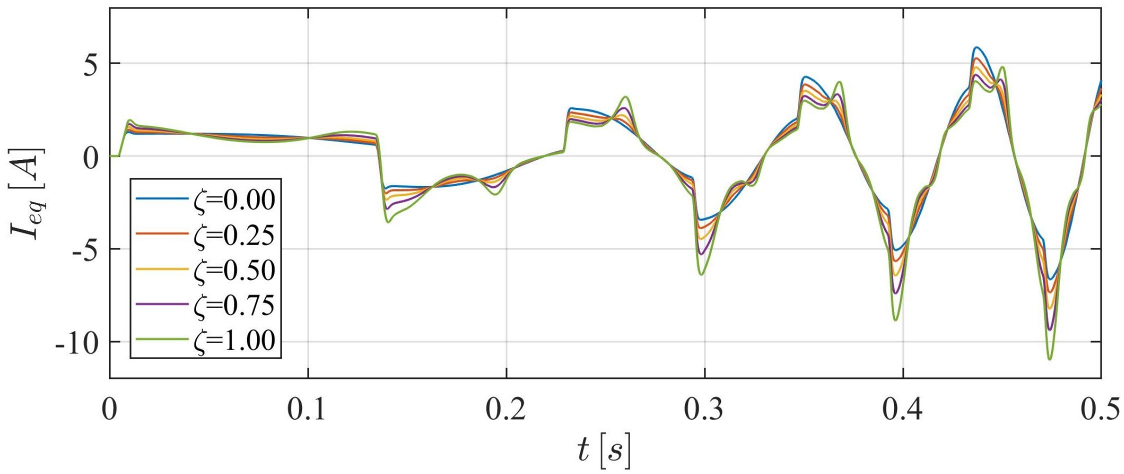


Figure 3.47: *Equivalent current for a chirp command - Eccentricity*

### 3.2.5 Proportional Gain

As seen in the introductory chapter, especially in aerospace, the role of electronics is becoming increasingly important, not only in the field of actuation but in various domains such as communication, on-board functions or component automation.

As a result, the probability of faults relating to such components is increasing, so it is necessary to implement a method that can detect these failures.
Most of these occur in transistors or capacitors and this can lead to a short circuit with the associated consequences. The other types of faults are also shown in the tables 3.3.

The problem with electric/electronics failures, however, is that they often occur without any alarm signal, so they would be unsuitable for prognostic methods. Generally, this is remedied by using redundancies so that they can replace the broken components without affecting the availability of the components.

These types of failure are implemented in Matlab-Simulink by varying the proportional gain used in the controller block. This parameter affects the dynamics of the system and, in particular, its response speed.

Under nominal conditions its value is $G_{prop} = 1 \cdot 10^5 \, \frac{1}{s}$. Since the proportional gain can vary the dynamics of the system by slowing down or speeding up its response to a specific command, the fault was implemented in Matlab-Simulink by varying it from $0.5 \cdot G_{prop}$ to $1.5 \cdot G_{prop}$.

The dynamic responses to a step command are shown below.
From the figures 3.48, 3.49 and 3.50 it can be seen that the gain variation proportional to a step command has no influence on the system dynamics.

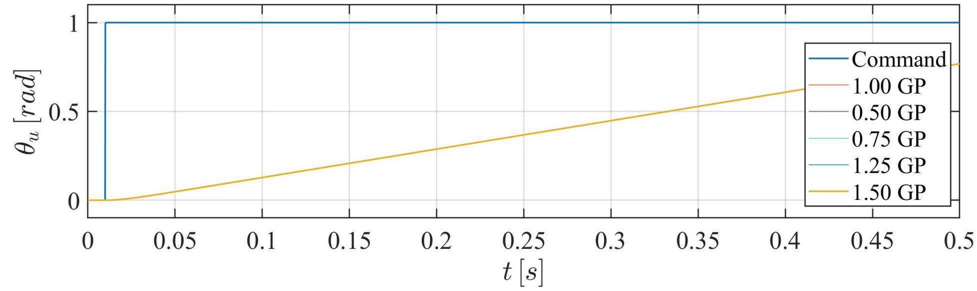


Figure 3.48: *User position for a step command - Proportional Gain*

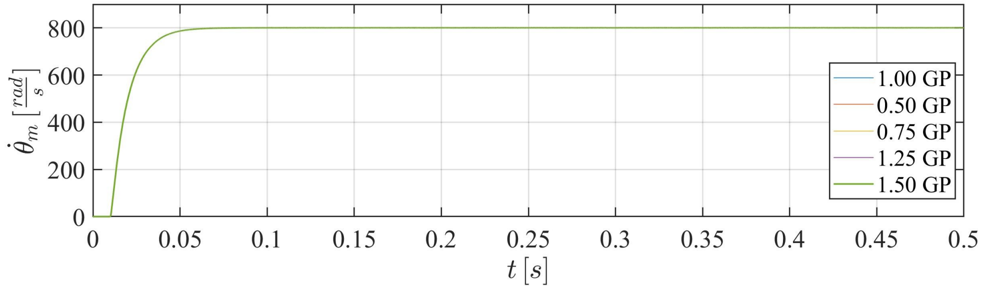


Figure 3.49: *Motor velocity for a step command - Proportional Gain*

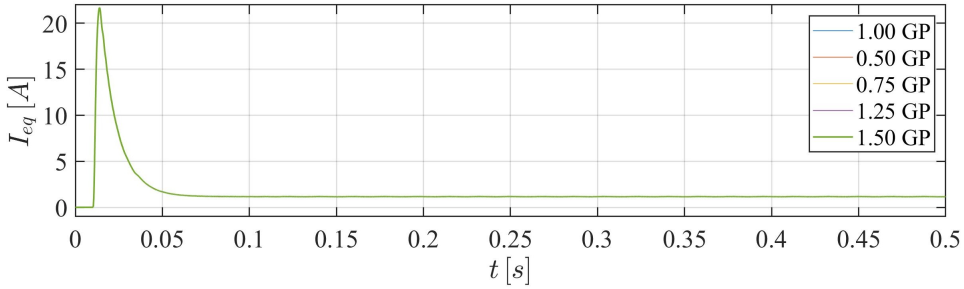


Figure 3.50: *Equivalent current for a step command - Proportional Gain*

The following shows the dynamic response of the system to a chirp command.

From the graphs in the figures 3.51, 3.52, 3.53, 3.54, 3.56 and 3.55 it can be seen that a decrease in proportional gain corresponds to a reduction in motor speed, which has consequences for motor readiness. In fact, a lower proportional gain induces the control law to delay the calculation of the current, which results in a delay in speed and position. As a result, the system is unable to reach maximum speed because when it reverses, the control has already imposed a decrease in current.

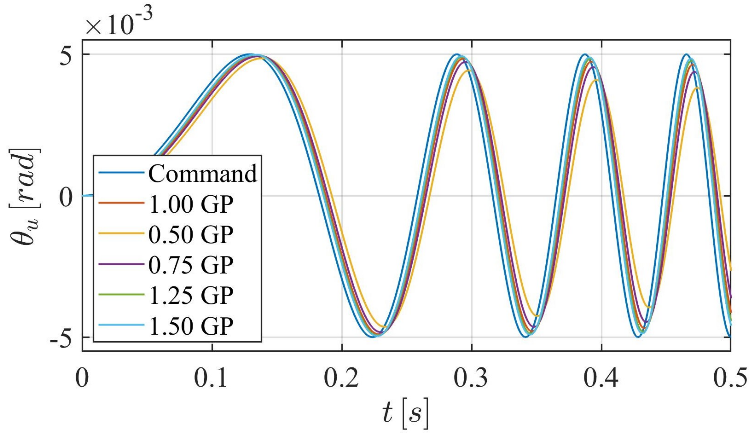


Figure 3.51: *User position for a chirp command - Proportional Gain*

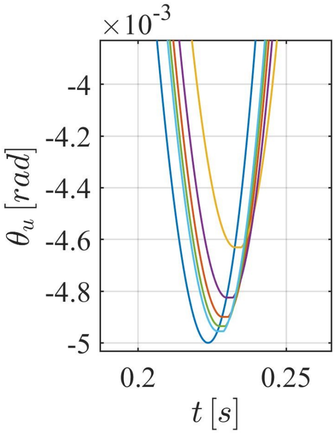


Figure 3.52: *User position for a chirp command - Zoom - Proportional Gain*

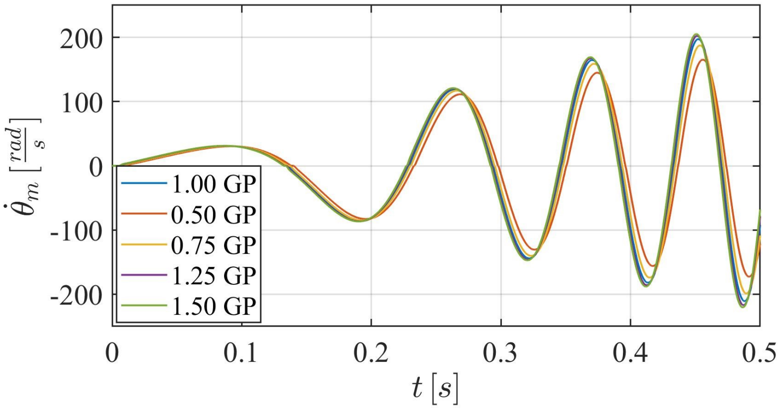


Figure 3.53: *Motor velocity for a chirp command - Proportional Gain*

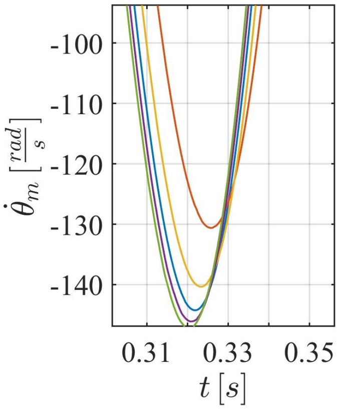


Figure 3.54: *Motor velocity for a chirp command - Zoom - Proportional Gain*

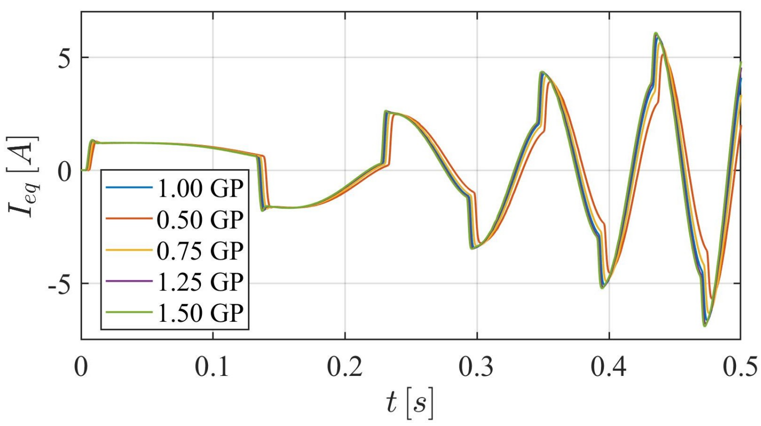


Figure 3.55: *Equivalent current for a chirp command - Proportional Gain*

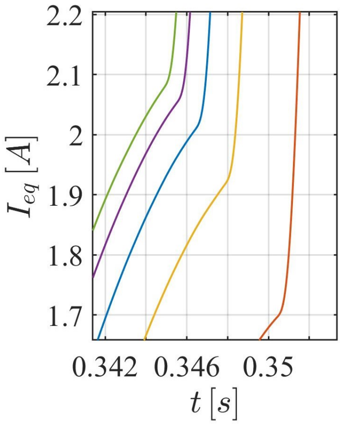


Figure 3.56: *Equivalent current for a chirp command -Zoom - Proportional Gain*

# Chapter 4

# Algorithms

# Bibliography

[1] B. Sarlioglu and C. T. Morris, “More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft,” *IEEE transactions on Transportation Electrification*, vol. 1, no. 1, pp. 54–64, 2015.

[2] R. Islam, “Evolution of aircraft flight control system and fly-by-light flight control system,” vol. 3, pp. 2250–2459, 12 2013.

[3] M. D. L. Dalla Vedova, “Servomeccanismi elettro-idraulici,” *Master degree course in Modelling, simulation and experimentation of aerospace systems - Exercise 7*.

[4] A. Garcia, I. Cusido, J. Rosero, J. Ortega, and L. Romeral, “Reliable electro-mechanical actuators in aircraft,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, no. 8, pp. 19–25, 2008.

[5] M. D. L. Dalla Vedova, “Servomeccanismi elettromeccanici,” *Master degree course in Modelling, simulation and experimentation of aerospace systems - Exercise 6*.

[6] R. D. Telford, S. J. Galloway, and G. M. Burt, “Evaluating the reliability amp; availability of more-electric aircraft power systems,” in *2012 47th International Universities Power Engineering Conference (UPEC)*, 2012, pp. 1–6.

[7] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[8] D. van Schrick, “Remarks on terminology in the field of supervision, fault detection and diagnosis,” *IFAC Proceedings Volumes*, vol. 30, no. 18, pp. 959–964, 1997, iFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Kingston upon Hull, UK, 26-28 August 1997.

[9] G. Swerdon, M. Watson, S. Bharadwaj, C. Byington, M. Smith, K. Goebel, and E. Balaban, “A systems engineering approach to electro-mechanical actuator diagnostic and prognostic development,” 09 2021.

[10] C. Byington, M. Watson, D. Edwards, and P. Stoelting, “A model-based approach to prognostics and health management for flight control actuators,” in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 6, 2004, pp. 3551–3562 Vol.6.

[11] M. Wahde, “Biologically inspired optimization methods,” 2008.

[12] M. Begotto, “Approccio evolutivo e analisi delle serie storiche,” B.S. thesis, Università Ca’Foscari Venezia, 2014.

[13] R. Tadei, F. Della Croce, and A. Grosso, *Fondamenti di ottimizzazione.* Società Editrice Esculapio, 2020.

[14] P. Serafini, “Metodi euristici,” in *Ricerca Operativa.* Springer, 2009, pp. 211–224.

[15] D. Floreano and C. Mattiussi, *Bio-inspired artificial intelligence: theories, methods, and technologies.* MIT press, 2008.

[16] A. Darwish, “Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications,” *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 231–246, 2018.

[17] X. Sun, C. Hu, G. Lei, Y. Guo, and J. Zhu, “State feedback control for a pm hub motor based on gray wolf optimization algorithm,” *IEEE Transactions on Power Electronics*, vol. 35, no. 1, pp. 1136–1146, 2020.

[18] V. Kumar and D. Kumar, “An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems,” *Advances in Engineering Software*, vol. 112, pp. 231–254, 2017.

[19] H. Cai and D. Hu, “On pmsm model fidelity and its implementation in simulation,” in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2018, pp. 1674–1681.

[20] P. C. Berri, M. D. Dalla Vedova, and P. Maggiore, “A simplified monitor model for ema prognostics,” in *MATEC Web of Conferences*, vol. 233. EDP Sciences, 2018, p. 00016.

[21] S. Re, “Development and comparison of prognostic methodologies applied to electromechanical servosystems (ema) for aerospace purposes,” 2018.

[22] MathWorks, “Matlab documentation,” www.mathworks.com/help/.

[23] G. Superti Furga, “Modellistica dei sistemi elettromeccanici: Trasformazione di park,” 2005.

[24] F. Bedeschi, “Controllo di una macchina brushless multi-fase con fasi statoriche aperte e compensatore della potenza reattiva,” 2019.

[25] O. Aksadi, “Study of innovatice model-based prognostic algorithms applied to aerospace electromechanical actuators,” 2021.

[26] L. Taponecco, “Appunti di meccatronica: motori elettrici.”

[27] P. Maggiore, “Azionamenti elettrici,” *Master degree course in Modelling, simulation and experimentation of aerospace systems.*

[28] D. Arzelà, "Modellazione, simulazione e controllo di un attuatore elettromeccanico con motore a riluttanza variabile per applicazione su uav," 2021.

[29] M. D. Dalla Vedova, A. Germanà, P. C. Berri, and P. Maggiore, "Model-based fault detection and identification for prognostics of electromechanical actuators using genetic algorithms," *Aerospace*, vol. 6, no. 9, p. 94, 2019.

[30] P. Berri, M. Dalla Vedova, P. Maggiore, and F. Viglione, "A simplified monitoring model for pmsm servoactuator prognostics," *MATEC Web of Conferences*, vol. 304, p. 04013, 01 2019.

[31] E. Balaban, P. Bansal, P. Stoelting, A. Saxena, K. F. Goebel, and S. Curran, "A diagnostic approach for electro-mechanical actuators in aerospace systems," in *2009 IEEE Aerospace conference.* IEEE, 2009, pp. 1–13.

[32] D. Belmonte, M. D. Vedova, C. Ferro, and P. Maggiore, "Electromechanical actuators affected by multiple failures: Prognostic method based on spectral analysis techniques," in *AIP Conference Proceedings*, vol. 1836, no. 1. AIP Publishing LLC, 2017, p. 020020.

[33] S. Corpino, "Rams," *Master degree course in Risk management, costs and integrated logistical support of aerospace sustems - Lesson 5.*

[34] M. Mazzoleni, Y. Maccarana, F. Previdi, G. Pispola, M. Nardi, F. Perni, and S. Toro, "Development of a reliable electro-mechanical actuator for primary control surfaces in small aircrafts," in *2017 IEEE international conference on advanced intelligent mechatronics (AIM).* IEEE, 2017, pp. 1142–1147.

[35] "Resistenze d'attrito," https://www.istitutopesenti.edu.it/dipartimenti/meccanica/Meccanica/ATTRITO.pdf.

[36] "Attrito radente (o di strisciamento)," http://web.inge.unige.it/DidRes/MApMac2/3_attrito_asciutto1112.pdf.

[37] P. Maggiore, "Modelli di attrito coulombiano," *Master degree course in Modelling, simulation and experimentation of aerospace systems.*

[38] J. Margielewicz, D. Gaska, and G. Litak, "Modelling of the gear backlash," *Nonlinear Dynamics*, vol. 97, no. 1, pp. 355–368, 2019.

[39] M. Dalla Vedova, D. Lauria, P. Maggiore, and L. Pace, "Linear electromechanical actuators affected by mechanical backlash: a fault identification method based on simulated annealing algorithm." *WSEAS transactions on systems*, pp. 268–276, 2015.

[40] F. di Ingegneria, "Correnti di corto circuito," *Università degli studi di Bergamo.*

[41] P. Arumugam, T. Hamiti, and C. Gerada, "Turn–turn short circuit fault management in permanent magnet machines," *IET electric power applications*, vol. 9, no. 9, pp. 634–641, 2015.

[42] L. Frosini, “Analisi delle vibrazioni per la diagnostica delle macchine rotanti,” *Dipartimento di Ingegneria Industriale e dell’Informazione Università di Pavia.*