

# **POLITECNICO DI TORINO**

Master's Degree in Mechatronic Engineering



Master's Degree Thesis

## **Study of innovative model-based prognostic algorithms applied to aerospace electromechanical actuators**

Supervisors:

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

Candidate:

Omayma AKSADI

APRIL 2021



# Acknowledgements

It gives me great pleasure to dedicate a small portion of my work to the people who have contributed, with their tireless support, to its realization.

First of all, a special thanks to my supervisors, Paolo Maggiore, Matteo D. L. Dalla Vedova and Pier Carlo Berri, for their immense patience, for their indispensable advice and for the knowledge transmitted throughout the drafting process.

I am infinitely grateful to my parents who have always supported me, supporting my every decision, right from the choice of my course of study. Not to be forgotten are my two little brothers, Soufiane and Zakaria, two important and fundamental pieces of my life. Without all of them, their immense love and sacrifices, I wouldn't have made it this far.

A heartfelt thanks to my two friends and colleagues, Suzy and Roberta, with whom I shared my entire university career with its ups and downs. It is thanks to them that I have overcome the most difficult moments. Without their advice, I would never have made it.

Last but not least, I dedicate this thesis to myself, to my sacrifices and my tenacity that have allowed me to go through the hardships to get here.

Alhamdulillah



# Abstract

The prior knowledge of incipient failures of primary flight command electromechanical actuators (EMAs) with prognostic algorithms can be very beneficial. Indeed, an early and proper detection and interpretation of the deterioration pattern can warn for the replacement of the servomechanism before the actual manifestation of the abnormal behavior. Such algorithms often exploit a model-based approach established on the direct comparison between the real (High Fidelity) system and the monitor (Low Fidelity) system to identify fault parameters through optimization processes, with the monitor model allowing to acquire accurate and precise results with a contained computational effort.

To this purpose, the thesis aims at presenting a prognostic technique consisting of a simplified monitor model capable of faithfully reproducing the dynamic response of the reference model in terms of position, speed and equivalent current, taking into account the presence of several mechanical and electrical faults: friction, backlash, coil short circuit, static rotor eccentricity, and proportional gain. Fault detection and isolation is performed by comparing the output signal of the reference system with the one obtained from the monitor model.

After that, the Genetic Algorithm is chosen as the optimization algorithm to match the two signals by iteratively changing the fault parameters to detect the global minimum of a quadratic error function. Once a suitable fit is obtained, the corresponding parameters are assumed to be acceptable. The reference models analyzed in this work have been previously developed in Matlab-Simulink by researchers of the ASTRA Group of the Department of Mechanical and Aerospace Engineering of the Politecnico di Torino.



# Table of contents

<b>List of Tables.....</b>	<b>I</b>
<b>List of Figures .....</b>	<b>II</b>
<b>1. Introduction .....</b>	<b>1</b>
1.1 Overview.....	1
1.2 Flight controls .....	3
1.2.1 Primary Flight Controls .....	4
1.2.2 Secondary Flight Controls .....	5
1.3 Actuation systems .....	7
1.3.1 Hydromechanical actuation .....	8
1.3.2 Electrohydraulic actuation .....	9
1.3.3 Electrohydrostatic actuation.....	10
1.3.4 Electromechanical actuators .....	12
1.4 Brushless Electrical Motors .....	15
1.4.1 BLDC motors.....	15
1.4.2 PMSM motors.....	18
<b>2. High Fidelity Model.....</b>	<b>21</b>
2.1 Model description .....	21
2.1.1 Com block .....	23
2.1.2 TR block.....	24
2.1.3 Control Electronics (PID) block .....	24
2.1.4 Resolver block .....	25
2.1.5 Inverter block.....	25
2.1.6 PMSM Electromagnetic Model block .....	29
2.1.7 Motor-Transmission Dynamical Model block.....	31
2.1.8 Signal acquisition block.....	32
2.2 Reference Model Output Response .....	33
<b>3. Low Fidelity Model.....</b>	<b>34</b>
3.1 Model description .....	34
3.1.1 Controller block .....	35
3.1.2 Electrical Model block.....	36
3.1.2.1 Winding SC Correction.....	38
3.1.2.2 Rotor Eccentricity Correction .....	39
3.1.3 Mechanical Model block.....	41
3.2 Monitor Model Output Response.....	43

<b>4. Fault analysis and implementation .....</b>	<b>44</b>
4.1 Introduction to faults.....	44
4.2 Friction.....	45
4.3 Backlash.....	46
4.4 Short circuit.....	48
4.5 Static rotor eccentricity .....	50
4.6 Proportional gain.....	52
<b>5. Optimization Process.....</b>	<b>54</b>
5.1 Optimization algorithms .....	54
5.2 Genetic algorithm.....	55
5.2 Optimization toolbox .....	57
5.4 Choice of the fitness function .....	60
5.5 Normalization of the fitness function arguments .....	63
5.6 Parallelization .....	64
<b>6. Model optimization.....</b>	<b>65</b>
6.1 Optimization parameters .....	65
6.2 Model calibration in normal conditions .....	66
6.3 Single fault isolation .....	70
6.3.1 Friction fault.....	71
6.3.2 Backlash fault.....	73
6.3.3 Short circuit fault .....	75
6.3.4 Rotor eccentricity fault .....	77
6.3.5 Proportional gain fault .....	80
6.4 Multiple fault isolation.....	82
<b>7 Conclusions .....</b>	<b>84</b>
<b>Bibliography.....</b>	<b>87</b>

# List of Tables

Table 6. 1 – Results of calibration at nominal conditions.....	69
Table 6. 2 – Low friction optimization results.....	71
Table 6. 3 – Medium friction optimization results .....	72
Table 6. 4 – High friction optimization results .....	72
Table 6. 5 – Low backlash optimization results .....	73
Table 6. 6 – Medium backlash optimization results .....	74
Table 6. 7 – High backlash optimization results.....	75
Table 6. 8 – Low short circuit optimization results .....	76
Table 6. 9 – Medium short circuit optimization results .....	76
Table 6. 10 – High short circuit optimization results.....	77
Table 6. 11 – Low rotor eccentricity optimization results .....	78
Table 6. 12 – Medium rotor eccentricity optimization results.....	78
Table 6. 13 – High rotor eccentricity optimization results .....	79
Table 6. 14 – Low proportional gain optimization results.....	80
Table 6. 15 – Medium proportional gain optimization results.....	81
Table 6. 16 – High proportional gain optimization results .....	81
Table 6. 17 – Random multiple fault parameters results .....	82

# List of Figures

Figure 1. 1 – Prognostics and Health Management taxonomy .....	2
Figure 1. 2 – Mechanical flight control system .....	3
Figure 1. 3 – Airplane controls, movements, axes of rotation and type of stability .....	4
Figure 1. 4 – Five common types of flaps .....	6
Figure 1. 5 – Leading edge high lift devices .....	6
Figure 1. 6 – Trim tabs.....	7
Figure 1. 7 – Simplified schematic of hydraulic circuit.....	9
Figure 1. 8 – Simplified schematic for an electrohydraulic actuator .....	9
Figure 1. 9 – Aircraft control surface system powered by an electrohydraulic servovalve.....	10
Figure 1. 10 - Simplified schematic for an electrohydrostatic actuator .....	11
Figure 1. 11 – Diagram of the main components of an electrohydrostatic actuator .....	11
Figure 1. 12 - Simplified schematic for an electromechanical actuator.....	12
Figure 1. 13 – Diagram of the main components of an electromechanical actuator .....	13
Figure 1. 14 – Structure of the BLDC motor .....	16
Figure 1. 15 – Trends of currents and of e.m.f E in BLDC motors .....	16
Figure 1. 16 – Cogging and skewing phenomena in BLDC motors .....	17
Figure 1. 17 – Block diagram of a BLDC motor drive.....	18
Figure 1. 18 – Structure of a PMSM motor .....	18
Figure 1. 19 – PMSM motor waveforms .....	19
Figure 1. 20 - Block diagram of a PMSM motor drive.....	20
Figure 2. 1 – Overview of the High Fidelity model.....	21
Figure 2. 2 – Inside the Sinusoidal EMA block .....	23
Figure 2. 3 – Com block .....	23
Figure 2. 4 – TR block .....	24
Figure 2. 5 – Control Electronics (PID) block .....	25
Figure 2. 6 – Resolver block.....	25
Figure 2. 7 – Inverter block.....	26
Figure 2. 8 – Evaluation of phase currents subsystem.....	26
Figure 2. 9 – inv Park subsystem .....	27
Figure 2. 10 – inv Clarke subsystem.....	27
Figure 2. 11 – Hysteresis PWM subsystem .....	27
Figure 2. 12 – 3-phase bridge subsystem.....	28
Figure 2. 13 – PMSM Electromagnetic Model block.....	29
Figure 2. 14 – Computation of back-EMF coefficients block .....	29
Figure 2. 15 – Computation of motor torque subsystem .....	30
Figure 2. 16 – Three phase RL model.....	30
Figure 2. 17 – Motor-Transmission Dynamical Model block .....	31
Figure 2. 18 – Borello Friction Model subsystem .....	31
Figure 2. 19 – Signal acquisition block .....	32
Figure 2. 20 – Output response of the reference model .....	33
Figure 3. 1 – Overview of the Low Fidelity model .....	34

Figure 3. 2 – Controller block.....	35
Figure 3. 3 – Electrical Model block .....	36
Figure 3. 4 – Winding SC Correction subsystem .....	38
Figure 3. 5 – Rotor Eccentricity Correction subsystem .....	39
Figure 3. 6 – Static eccentricity scheme .....	40
Figure 3. 7 – Mechanical Model block .....	41
Figure 3. 8 – Output response of the monitor model .....	43
Figure 4. 1 – Simulink implementation of the Borello friction block .....	46
Figure 4. 2 – The backlash fault implemented in the reference model in Simulink .....	47
Figure 4. 3 – The backlash fault implemented in the monitor model in Simulink .....	47
Figure 4. 4 – The short circuit fault implemented in the monitor model.....	50
Figure 4. 5 – Scheme of the air gap in the presence of rotor eccentricity .....	50
Figure 4. 6 – Implementation of the rotor eccentricity fault in the reference model .....	51
Figure 4. 7 – Implementation of the rotor eccentricity fault in the monitor model .....	52
Figure 4. 8 – Implementation of the proportional gain fault in the reference model.....	53
Figure 4. 9 – Implementation of the proportional gain fault in the monitor model.....	53
Figure 5. 1 – Population, chromosomes and genes of a genetic algorithm .....	56
Figure 5. 2 – Roulette wheel selection.....	56
Figure 5. 3 – The Optintool GUI for the genetic algorithm.....	57
Figure 5. 4 – Representation of the Total Least Squares method .....	61
Figure 6. 1 – Comparison of the output currents before calibration .....	66
Figure 6. 2 – Close-up of the comparison of the output currents before calibration .....	67
Figure 6. 3 – Comparison of the two output currents after calibration.....	69



# 1. Introduction

## 1.1 Overview

Prognostics is a field of study focused on predicting when a particular component loses its functionality and is no longer able to be fully effective or achieve the required performance. It is based on the analysis and prediction of all possible failure scenarios and of the ability to detect the early signs of aging and wear. When properly assembled and organized, such a database can be used effectively as an input to an appropriate failure propagation model. For what concerns other technological fields, the application of prognostics to aeronautics could result into a valuable asset on the maintenance aspect, as it could cut down both costs and inspection time. The purpose of what is known as Prognostics and Health Management (PHM) is the provision of real-time data regarding the ongoing state of the system and to deduce the Remaining Useful Life (RUL) before the occurrence of a fault which can lead to the inability of the component to perform its functionalities at the required degree. The benefit of implementing PHM strategies is clearly evident from the comparison with traditional concepts of monitoring and maintenance, based on overhaul or parts with limited lifespan.

Primary flight controls are a critical feature of the aircraft system and hence designed with a conservative safe-life approach, which requires replacing associated components after enduring a certain number of flight hours or operating cycles. When applying this approach, however, the actual condition of the components is not assessed, and maintenance is limited to the specific scheduled operation. Particularly, the safe life design criterion does not have the ability to assess initial defects, which could arise from the manufacturing process and could escalate into a sudden fault that compromises the safety of the aircraft. As a matter of fact, the gradual degradation of a component of the system, which initially does not present any unacceptable behavior, often leads to a condition in which the efficiency of such a component is weakened, and as a consequence the functionality of the actuator is

compromised. In addition, the safe-life criterion does not allow the detection of the cause and location of the malfunction, whereas an accurate identification of the specific failed subcomponent could be effective in reducing maintenance inefficiencies and costs, such as replacing the single subcomponent, instead of the entire system, might be sufficient to restore the system functionality.

By applying PHM strategies, failure may be managed more effectively, with the following benefits:

- Lower operating costs;
- Less maintenance interventions;
- Limited amount of necessary redundancies to be installed on board aircraft;
- Improvement of the aircraft safety and reliability;
- Simpler logistic, as maintenance can be scheduled properly with the instantaneous outcome of limited downtime and related costs and a more effective management of spare parts warehouses.

It should be noted that prognostic concepts, due to the variety of applications and the considerable impact they generate, have attracted great interest in the scientific and technological world and, especially in recent years, have been the subject of in-depth development and dissemination in the field of scientific literature. Very often these contributions, although extremely innovative and significant, are too theoretical or specific and tend to overlook a more comprehensive approach, a systematic vision, focusing on well-defined aspects of the problem considered.

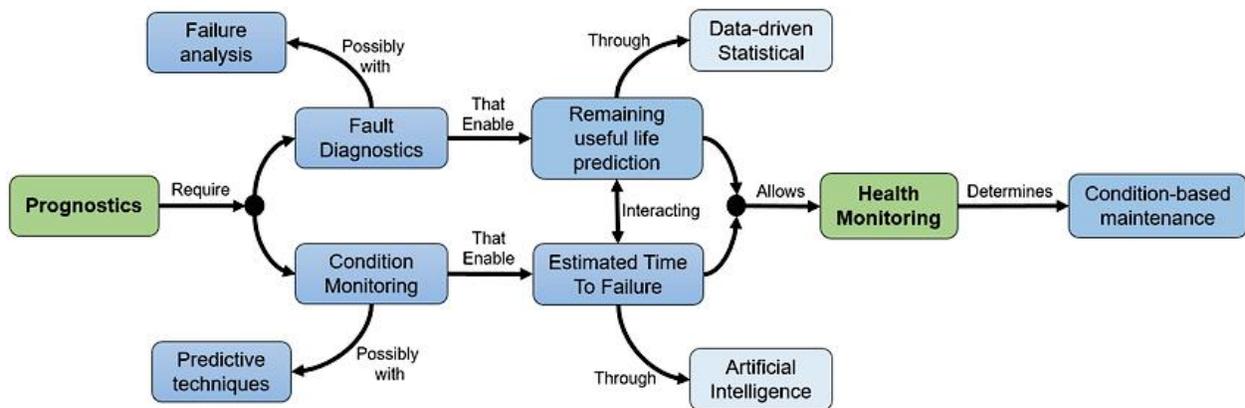


Figure 1.1 – Prognostics and Health Management taxonomy

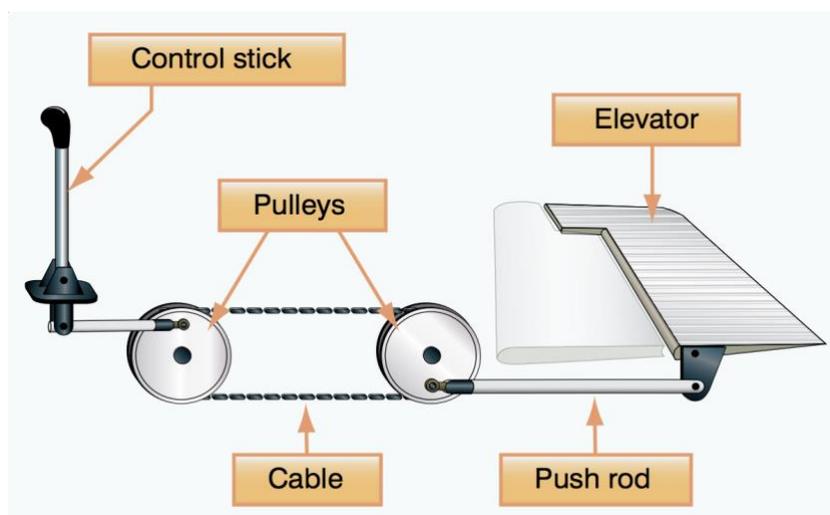
Figure 1.1 illustrates the development of prognostics, which depends mainly on fault diagnosis parameters that connect the failure modes and calculates an estimated

Remaining Useful Life (RUL) of the components. As a result, it is necessary to use the dependability analysis techniques, because dependability, in general, involves the identification of reliability, availability and maintainability of the system. Thus, after detecting the operating condition of the component, the collected data may be processed by statistical algorithms and/or models of AI to Estimated Time To Failure (ETTF), which is a prediction of the time period from the measurement of a current operating status to a time when the machine enters a functional fault condition.

Once PHM is implemented, health monitoring machines, by means of inspection routes, can aid decision making for planning and scheduling of Condition-based Maintenance (CBM). Thus, timely interventions for recovery or repair can be carried out, mitigating contingencies and ensuring manufacturing productivity.

## 1.2 Flight controls

Flight control systems are employed by pilots to control the flight forces as well as the direction and attitude of the aircraft. Their characteristics can vary considerably depending on the type of aircraft used. The most basic designs of flight control systems are mechanical and date back to the earliest airplanes. They work with a set of mechanical parts, such as rods, cables, pulleys and sometimes chains so forces can be transmitted to the control surfaces from the flight deck controls. Mechanical flight control systems are still used today in small aircraft where aerodynamic forces are not enormous.



*Figure 1. 2 – Mechanical flight control system*

Aircraft flight control systems include primary and secondary systems. The primary control system is constituted by ailerons, elevator (or stabilizer), and rudder which are necessary to control an aircraft safely during flight. Wing flaps, leading edge devices, spoilers and trim systems are part of the secondary control system which improves the performance characteristics of the aircraft and makes excessive control forces manageable by the pilot.

### 1.2.1 Primary Flight Controls

Aircraft control systems are carefully designed to provide suitable responses to control inputs while allowing a natural feel. At low speeds, the controls are generally soft and slow, and the aircraft responds gradually to control applications. At higher speeds, the controls become much more stable and the aircraft response is faster. Air circulation and pressure distribution over and around the airfoil change by means of the movement of one of the three main flight control surfaces: ailerons, elevator or stabilizer, or rudder. These changes affect the lift and drag produced and allow a pilot to manage the vehicle around its three axes of rotation.

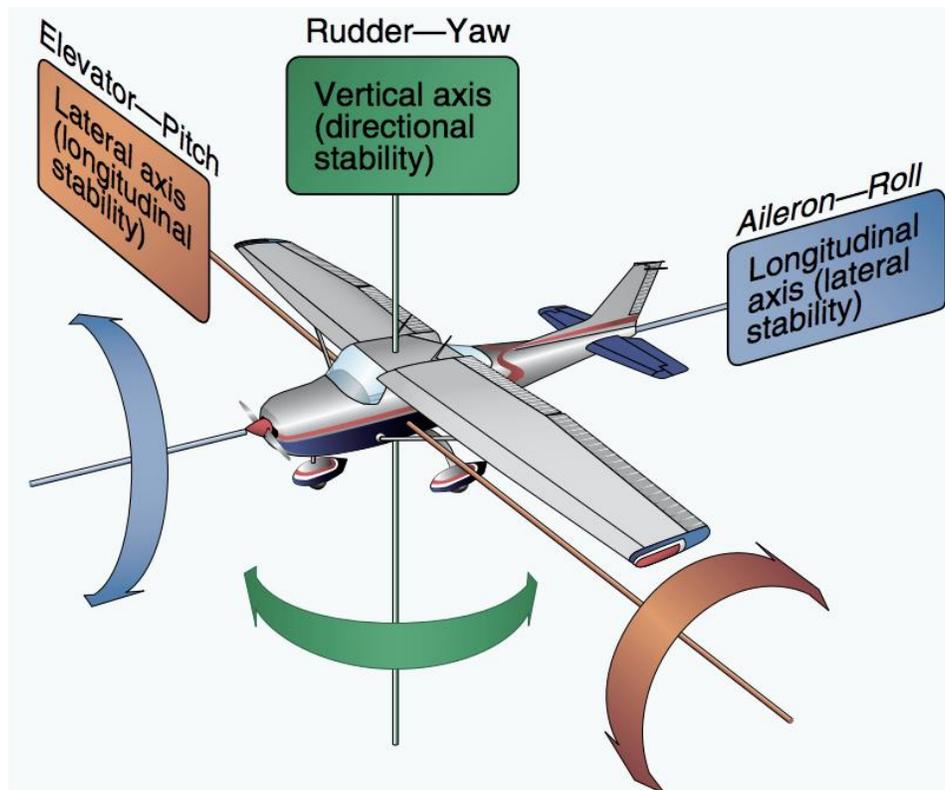


Figure 1. 3 – Airplane controls, movements, axes of rotation and type of stability

*Ailerons* control the roll movement about the longitudinal axis. They are attached to the outboard trailing edge of each wing and work opposite to each other. They are connected to a control wheel by means of cables, bell cranks, pulleys, and/or push-pull tubes. By moving such wheel to the right side, the right aileron deflects upward reducing the camber and resulting in lower lift on the right wing whereas the left aileron deflects downward increasing the camber and resulting in higher lift on the left wing. Consequently, the aircraft will roll to the right. The opposite occurs when moving the control wheel to the left.

The *elevator* controls the pitch movement about the lateral axis. Like in the case of ailerons on small aircraft, a series of mechanical links connect the elevator to the control column on the flight. The backward movement of such column deflects the trailing edge of the elevator surface up, reducing the camber of the elevator and creating a downward aerodynamic force greater than the normal tail-down force that exists in the straight flight. The overall effect causes the tail of the aircraft to descend and the nose to pitch up about the center of gravity (CG). The strength of the pitching moment can be established by the distance between the CG and the horizontal tail surface, as well as by the aerodynamic efficiency of the horizontal tail surface. The opposite effect is obtained by moving the control column forward.

The *rudder* controls the yaw movement of the aircraft about its vertical axis. It is a movable surface hinged to the vertical stabilizer and it is controlled by the left and right rudder pedals. By stepping on the left pedal, the rudder moves left altering the airflow around the vertical stabilizer and creating a sideward lift that moves the tail to the right and yaws the nose of the airplane to the left. Rudder effectiveness increases with speed.

## 1.2.2 Secondary Flight Controls

Secondary flight controls are intended to change aerodynamic force coefficient to adapt them to different flight conditions. They may consist of wing flaps, leading edge devices, spoilers, and trim systems.

*Flaps* are the most common high-lift devices used on aircraft. They are attached to the trailing edge of the wing and they increase both lift and induced drag for any value of the angle of attack. They provide a compromise between high cruising speed and low landing speed thanks to their feature that allows them to extend when needed and retract into the wing structure when not needed. There are five common types of flaps: plain, split, slotted, Fowler flaps and slotted Fowler flaps.

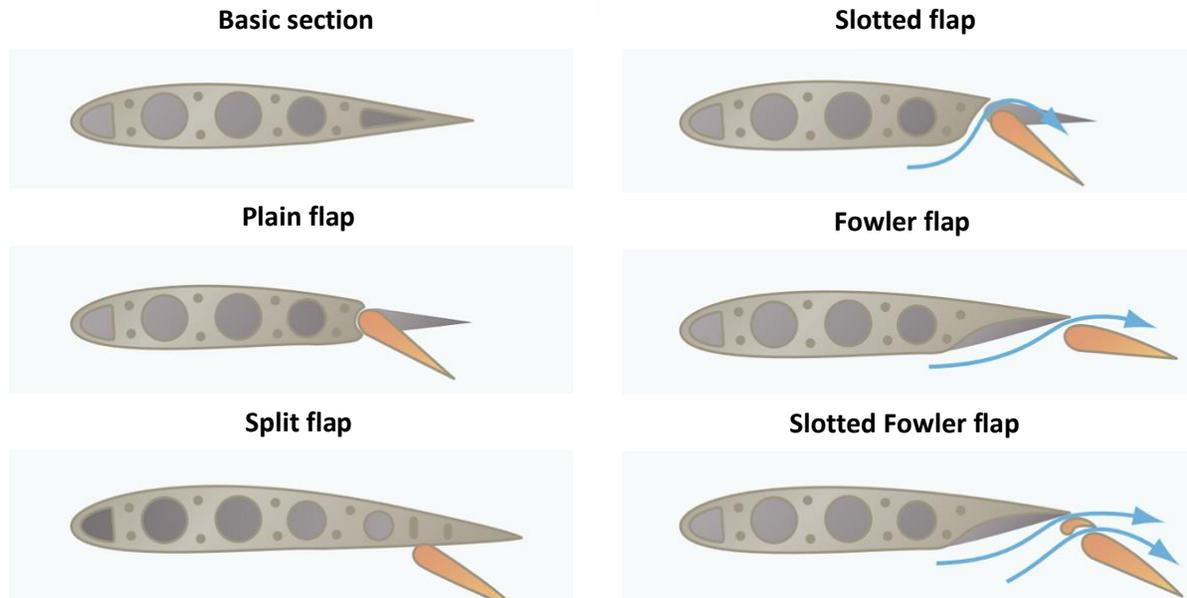


Figure 1. 4 – Five common types of flaps

High-lift devices can be applied to the *leading edge* of the airfoil, the part which first meets the oncoming air. The most common types are fixed slots, movable slots, leading edge flaps and cuffs.

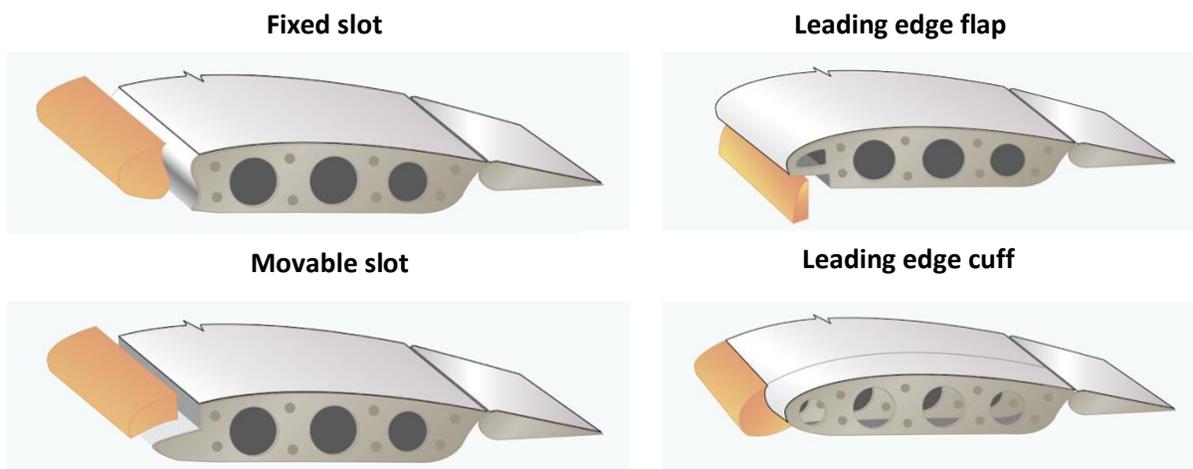
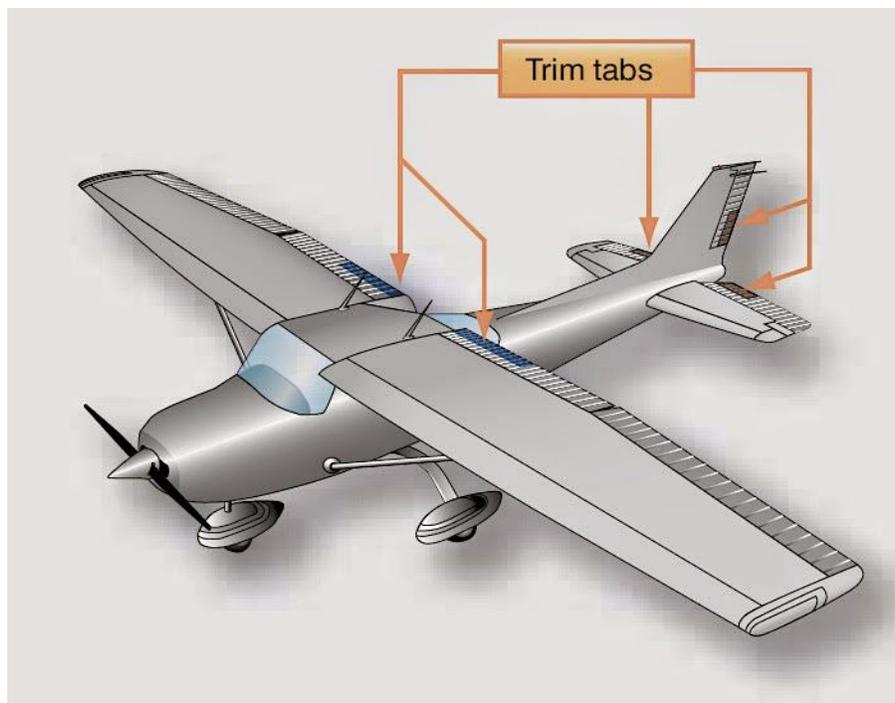


Figure 1. 5 – Leading edge high lift devices

*Spoilers* can be found on some fixed-wing aircraft and they are deployed from the wings to spoil the smooth airflow, with a consequent reduction of lift and increase of drag. They are mostly used on gliders to control the descent rate for landings,

whereas on other planes they are usually employed to control roll leading to the elimination of adverse yaw. Using spoilers on both wings simultaneously allows the aircraft to descend without an increase in terms of speed. They can be used to help reduce roll after landing as well.

*Trim systems* are employed to help the pilot maintain a constant pressure on flight controls, and often include flight deck controls and small hinged devices attached to the trailing edge of one or more of the primary flight control surfaces. They aerodynamically assist the movement and position of the flight control surface to which they are attached, being designed to minimize a pilot's workload throughout the flight. Common types of trim systems include trim tabs, balance tabs, antiservo tabs, ground adjustable tabs, and an adjustable stabilizer.



*Figure 1. 6 – Trim tabs*

### **1.3 Actuation systems**

Aircraft actuators perform several functions including adjusting flight control surfaces. Over the past decades, the source control signals together with power for actuators onboard aircraft has evolved, with actuation technology gradually advancing toward hydraulically and electrically driven solutions moving from

manual sources like cables and rods. Such transition started with hydromechanical systems in which the operation of control valves in the hydraulic circuits could operate by means of the movement of a control column or lever transmitted mechanically, filling and emptying cylinders to produce actuator movement.

Fly-by-wire systems then replaced mechanical linkages with electrical cables. In such systems, a flight computer interprets the pilot's control column movements and sends electrical signals to actuator control electronics, which instruct the operation of either hydraulic control valves to move hydraulic actuators or electric motors to put in motion electromechanical actuators. Fly-by-wire systems allowed aircraft manufacturers to integrate more electrically powered actuators in aircraft systems such as electrohydraulic actuators and electromechanical actuators, which are powered by electricity produced by engine-driven generators delivered over power-by-wire systems. The reason why the aerospace industry wants to convert mechanical, pneumatic and hydraulic systems to electric systems is driven by a desire to optimize aircraft performance, reduce maintenance and operating costs, increase fuel efficiency and reduce emissions.

### **1.3.1 Hydromechanical actuation**

The complexity and weight of flight control systems greatly increase with the size and performance of the aircraft, but the use of hydromechanical actuators helps overcome these limitations. With hydraulic flight control systems, the size and performance limitations of the aircraft are due to economic reasons and not the pilot's muscular strength. In the early days, only partially assisted systems were used, so the pilot still had the feeling of aerodynamic loads on the control surfaces

A hydromechanical flight control system is mainly composed of:

- A mechanical circuit, which connects the cockpit controls with the hydraulic circuits; it is composed of rods, cables, pulleys, and sometimes chains.
- A hydraulic circuit, with hydraulic pumps, tanks, filters, pipes, valves and hydraulic actuators. The actuators are powered by the hydraulic pressure generated by the pumps in the hydraulic circuit and convert the hydraulic pressure into movements of the control surfaces. Servo valves control the movement of the actuators.

The pilot's action on a control in the cabin causes the mechanical circuit to open the corresponding servovalve in the hydraulic circuit. The hydraulic circuit activates the actuators that move the control surfaces. When the actuator comes into operation, the servo valve is closed by a feedback link which blocks the movement of the control surface to the desired position.

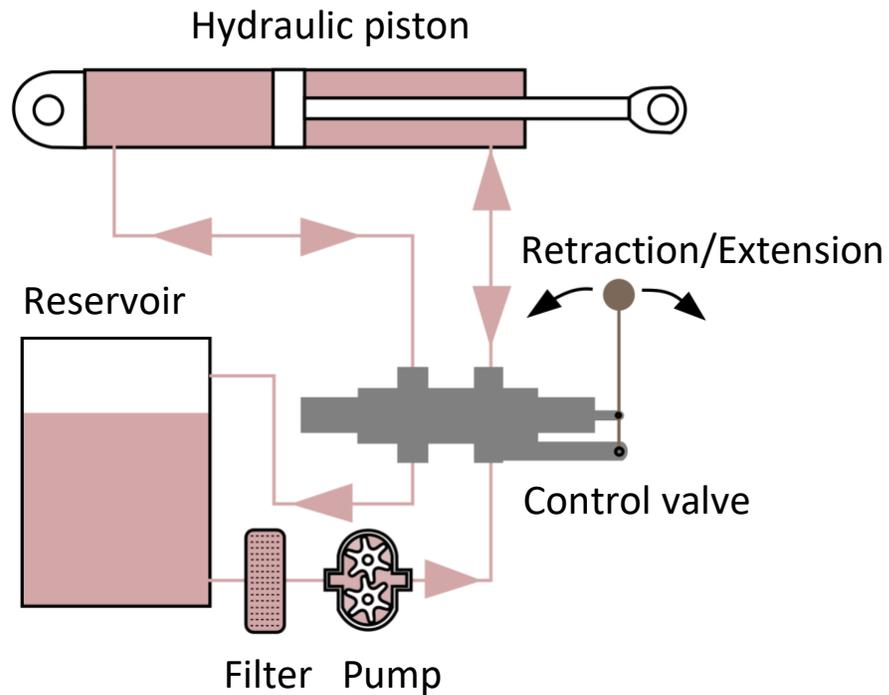


Figure 1. 7 – Simplified schematic of hydraulic circuit

### 1.3.2 Electrohydraulic actuation

Conventional electrohydraulic actuator systems need a central hydraulic power supply with hydraulic lines connected to each actuator. The amount of hydraulic fluid supplied to the actuator from the main hydraulic fluid supply can be varied by a servovalve which is controlled by electric command signals.

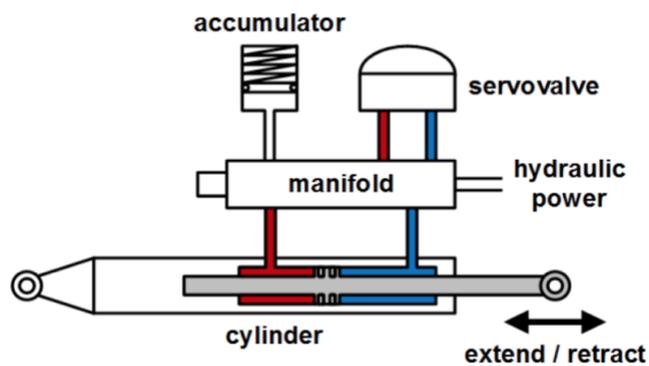
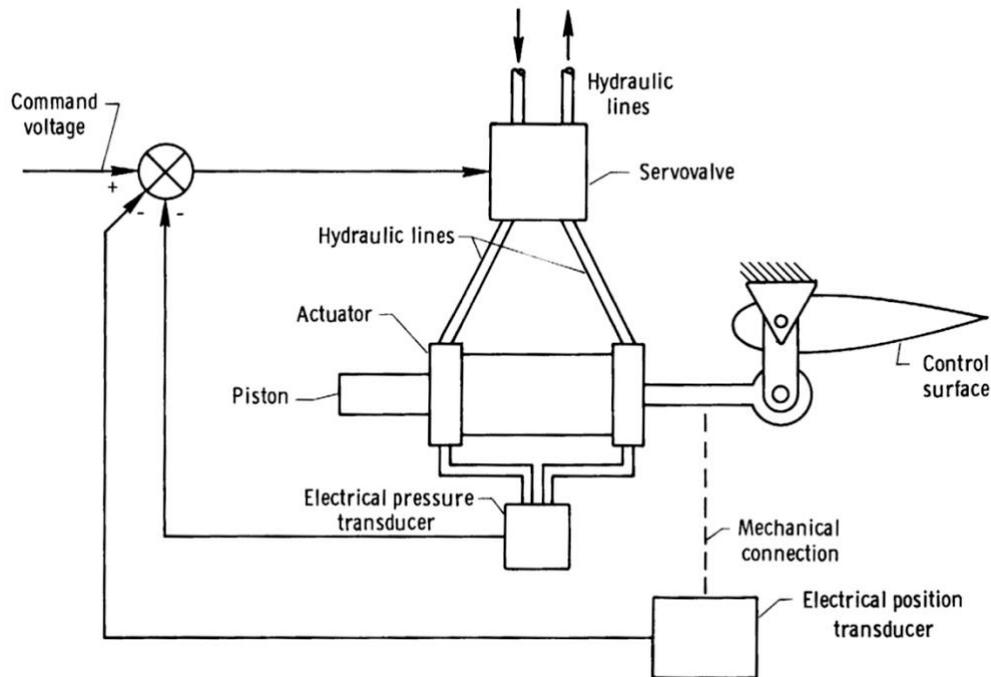


Figure 1. 8 – Simplified schematic for an electrohydraulic actuator



*Figure 1. 9 – Aircraft control surface system powered by an electrohydraulic servovalve*

This kind of electrohydraulic servo-actuator system is able of producing very high forces without backlash, but it necessitates of a centralized hydraulic network that should be maintained at constant pressure (3000÷5000 psi) by hydraulic pumps relentlessly draining energy from the engines. The continuous consumption of energy leads to the heating of the hydraulic fluid, which then required a cooling system to maintain an acceptable temperature of the hydraulic fluid.

The central hydraulic network also needs a system of pipes to deliver pressurized hydraulic fluid to actuators distributed throughout the aircraft, adding extra weight and occupying space. The large hydraulic network increases the risk of leaks and requires a large volume of hydraulic fluid.

Traditional electrohydraulic actuators have remarkable power density (kW/kg) equipment wise but low power density when considering the power distribution network.

### 1.3.3 Electrohydrostatic actuation

Electrohydrostatic actuators are self-contained hydraulic units that do not a central hydraulic power source and the associated hydraulic lines. They take advantage of

the high power density at the equipment level and eliminate the inefficiencies related to a central hydraulic power distribution network.

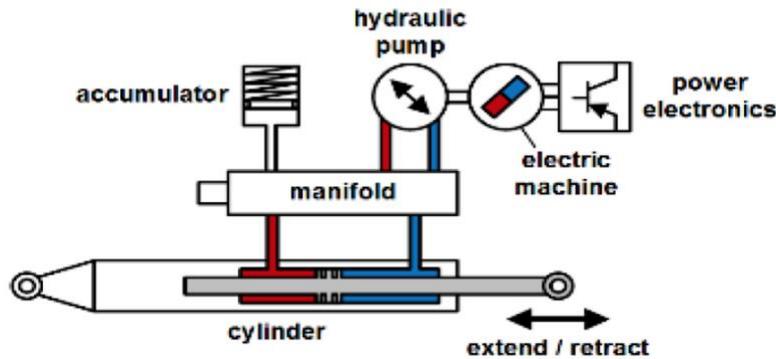


Figure 1. 10 - Simplified schematic for an electrohydrostatic actuator

Electrohydrostatic actuation systems convert electrical energy locally at the actuation location to hydraulic energy and then to mechanical energy. The standalone unit consists of an electric motor that drives a hydraulic pump to pressurize the fluid for a hydraulic actuator. In order to control the output of an electrohydrostatic actuator, the output flow of the pump must be controlled. This can be achieved with a variable speed electric motor that drives a fixed displacement hydraulic pump or a fixed speed electric motor that drives a variable displacement hydraulic pump.

The electronic control signals are sent to the electrohydrostatic actuator by means of electric cables. They control the speed of the electric motor to provide rotational power to the hydraulic pump which creates pressurized hydraulic fluid to move the hydraulic cylinder locally at the actuator.

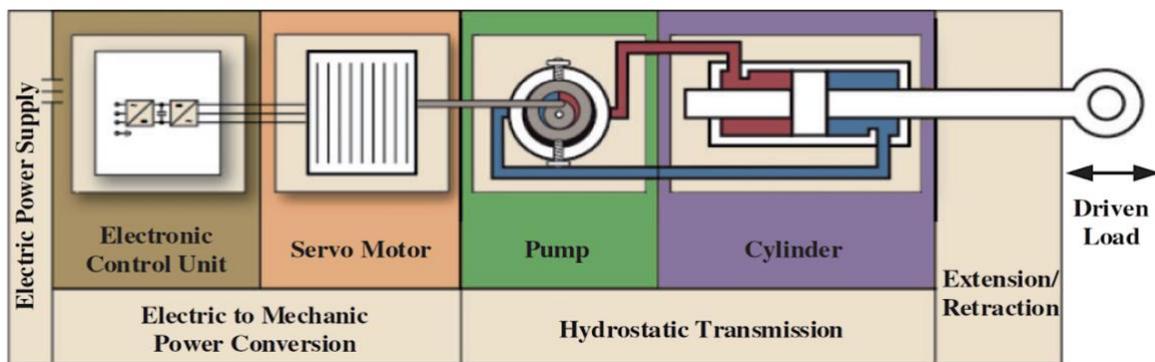


Figure 1. 11 – Diagram of the main components of an electrohydrostatic actuator

Electrohydrostatic actuators work with greater energy efficiency compared to conventional electrohydraulic actuators. Rather than continuously venting the power from the motors to keep a large hydraulic network at constant pressure, electrohydrostatic actuators only consume electricity when they are moving the load. Lower energy consumption creates less heat in the hydraulic fluid, so no cooling system is required. With fewer faults compared to a central hydraulic network with extensive pipelines, the leakage potential as well as the maintenance effort are reduced. Moreover, a higher power density enables them to generate higher forces in a more compact housing. They also have no backlash problems and allow precise positioning with no errors caused by gaps between mechanical components. In addition, electrohydrostatic actuators do not suffer from the risk of jamming caused by interference between gear teeth or screw threads in electromechanical actuators. A disadvantage of electrohydrostatic actuators is that they require hydraulic fluid to operate. Although the fluid volume is greatly reduced compared to conventional electro-hydraulic actuation systems, the presence of the fluid invalidates the potential for 100% leakage-free operation.

### 1.3.4 Electromechanical actuators

A major advantage of electromechanical actuators (EMAs) over traditional electrohydraulic actuators is the elimination of hydraulic fluid. The lack of this liquid and the pipes required for it leads to increased safety, lower weight, more space, higher energy efficiency and lower environmental pollution. Easier maintenance is facilitated by the lack of leaks and of liquid conditioning tasks such as filling, loading, flushing and filtering.

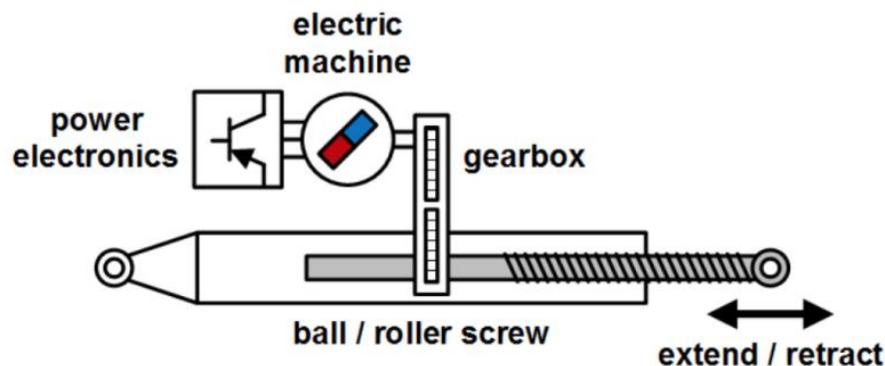


Figure 1. 12 - Simplified schematic for an electromechanical actuator

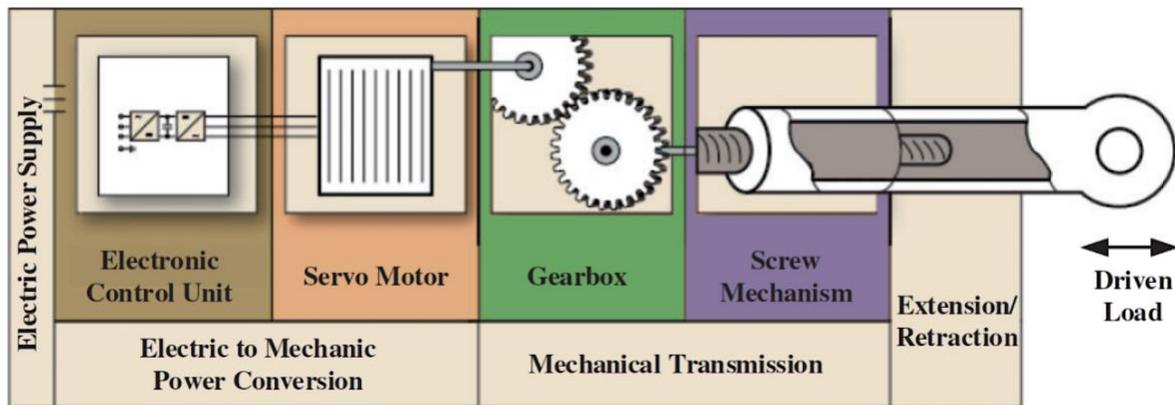


Figure 1. 13 – Diagram of the main components of an electromechanical actuator

The functions and characteristics of the main components are described as follows:

- *Electronic Control Unit*: it works with a voltage of 28V from a power supply unit. The power control determines the motor current by voltage pulse width modulation to the motor in response to a position or torque command signal. The EMA position and load sensor sends the position / speed and load information to the ECU for position feedback and current limitation. For the flight status when the external load is high, additional temperature sensors are usually installed on the controller housing near the connections to record the rise in outside temperature and to alert high temperatures.
- *Servo Motor*: it has variable speed, high reliability, high power density, and acceptable heat dissipation. Several types of motors may be suitable, and three common options are the permanent magnet synchronous motor (PMSM), the BLDC motor, and the switched reluctance motor (SR). The choice of motors usually depends on the power supply on board. For example, the PMSM may be a suitable type when an AC power supply is directly driving a servo motor. The motor control is used to control the speed and direction through the electrical switch of the windings via power electronic devices.
- *Gearbox*: its main purpose is to convert the high speed and low torque of the servomotor into low speed and high torque of a screw mechanism. Harmonic reducers, cycloidal reducers, or planetary gears are an effective option because of their compact structure, easy accessibility without backlash, and high efficiency. It should be noted that when planetary gears are used, a large part of the actuator mass can be devoted to the servo motor due to its low gear ratio compared to the high reduction ratios of harmonic and cycloidal gear reducers.

- *Screw Mechanism*: either a ball screw mechanism or a planetary roller screw mechanism is used to convert the rotary motion into linear motion with a required force. By connecting the servo motor and the external load, the torque and speed are matched to one another by the transition mechanism. The overall mass of the actuator tends to decrease as the gear ratio increases by increasing the gear ratio of the reducer and decreasing the pitch of the screw mechanism. To maximize stiffness while minimizing weight, a hollow screw shaft is selected to accommodate a linear variable differential transformer and measure the linear position of the screw rod to close the loop.

Electromechanical actuators appear to be an optimal solution for more electric and fully electric aircraft as they completely eliminate the need for hydraulic fluid. Current electromechanical actuator technology, however, has limitations in applications that require high output forces. In such cases, hydraulic actuators have a power density advantage that can generate large forces in a small space under harsh conditions without the need for a liquid cooling system for the engine.

The drawbacks of electromechanical actuators include backlash, jamming, and thermal management problems. Backlash can occur due to gaps between intermeshing gear teeth or screw threads and can lead to positional inaccuracies. Backlash can increase as repeated cycles of wear lead to surface deterioration. Jamming is a risk for electromechanical actuators due to possible failures where screw components interfere or become stuck, preventing the actuator from moving. This can have the following causes:

- Mechanical wear of the gear and screw assembly including fatigue from external loads that cause high contact stresses on the raceway;
- Reduced lubricant viscosity and thickness due to high temperatures;
- Catastrophic failure of components.

Thermal management is also a problem for electromechanical actuators in high load applications. In electromechanical actuators, heat is generated in the electric motor due to the electrical resistance in the copper stator windings and the iron stator core, as well as the friction in the gearbox and screw mechanism. Heat dissipation in conventional hydraulic systems is more easily accomplished through hydraulic fluid circulation and heat exchange in the main tank. Electromechanical actuators, on the other hand, have to deal with heat dissipation locally. Possible solutions include heat sinks, heat pipes, liquid cooling, and phase change materials.

Despite the challenges, electromechanical actuator technology continues to advance as the industry seeks improvements in reliability, thermal efficiency, and case sizes.

## 1.4 Brushless Electrical Motors

Brushless motors are ideally derived from DC motors with the purpose of eliminating the commutator and therefore the brushes. There are two types of brushless motor:

- Trapezoidal or *Brushless Direct Current (BLDC)*;
- Sinusoidal or *Permanent Magnet Synchronous Motor (PMSM)*;

Since brushless DC motors have no brushes that can worn out with continuous use, they represent a major leap forward in modern technology. Brushless motors are significantly more efficient and less prone to mechanical wear than their brushed counterparts.

They have several other benefits such as:

- Higher torque to weight ratio;
- Higher efficiency due to the increased torque per watt of power input;
- Greater reliability and reduced maintenance requirements;
- Lower operational and mechanical noise;
- Longer lifespan;
- No ionizing sparks from the commutator;
- Minimal electromagnetic interference.

### 1.4.1 BLDC motors

In a brushed DC motor, the rotor winding (induced) is powered from the outside through the brushes and the commutator blades. The rotation of the rotor determines the relative movement of the brushes on the commutator and the consequent feeding of the appropriate coils. The commutator together with the brushes acts as an automatic power switch on the rotor windings.

In brushless motors, the commutator is replaced by an electronic controlled switch which makes it possible to exchange the positions between rotor and stator, ie the field is now generated by permanent magnets placed on the rotor, while the windings are placed on the stator. An evolution has led to the creation of this machine with three stator windings placed at  $120^\circ$ .

The trapezoidal brushless has such a magnetization of the rotor that, once set in rotation, it produces an induced e.m.f on the stator windings with trapezoidal characteristics. This will therefore be the profile of the e.m.f  $E$  during the operation of the machine. The three stator windings, placed at  $120^\circ$  from each other, will have  $E$  with their respective time delays.

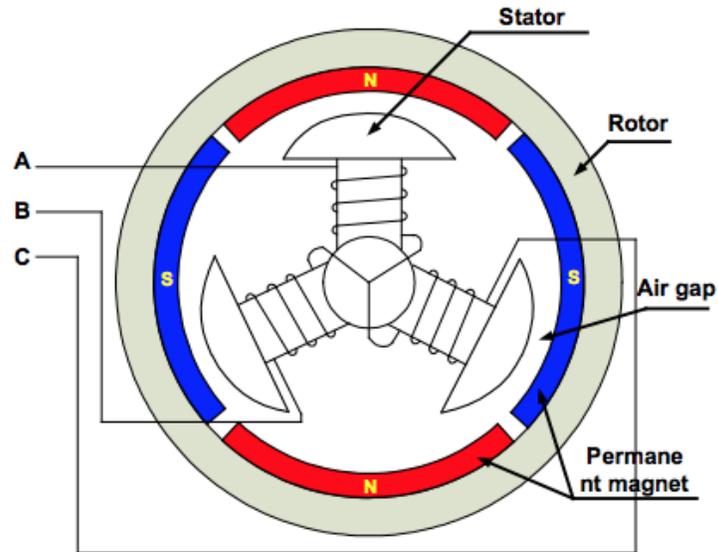


Figure 1. 14 – Structure of the BLDC motor

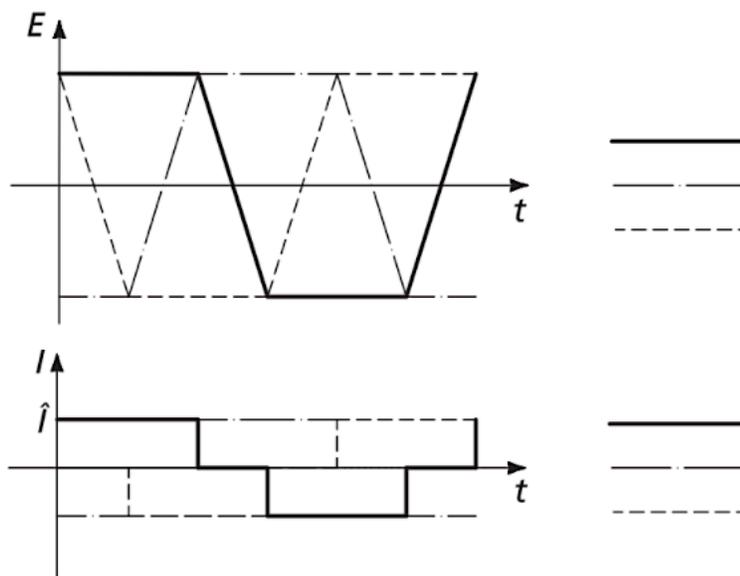


Figure 1. 15 – Trends of currents and of e.m.f  $E$  in BLDC motors

By powering the motor and appropriately piloting the currents  $I$  so that they are as in *Figure 15*, it can be seen that at every instant the product  $E \cdot I$ , equivalent to the mechanical power and therefore linked to the torque, is due to two windings at a time and is constant. Therefore:

$$P = 2 \cdot E \cdot I \tag{1.1}$$

In order to operate, the motor needs a control circuit that takes into account the angular position of the shaft, through some sensors (Hall effect), which must be correctly positioned in phase on the stator.

For this motor, the torque can be expressed as:

$$T = I \cdot \left( \frac{d}{dt} \Phi_m \cdot \frac{1}{\omega} \right) = I \cdot \left( E \cdot \frac{1}{\omega} \right) \quad (1.2)$$

where  $I$  is the matrix of the stator currents,  $\Phi_m$  is the magnetic flux of the rotor and  $\omega$  is the angular velocity.

If you manually rotate the shaft of a brushless DC motor, you will hear some clicks. These are signs of a negative cogging effect. The rotor tends to position itself in such a way as to create magnetic paths with minimum reluctance, such as the sections of the stator magnetic circuit between two consecutive slots. This gives rise to a torque discontinuity called torque ripple which can have amplitudes of up to 30%. In order to reduce the problem, the motor is built with a certain rotor skewing or with a slight longitudinal screwing of the magnetic circuit or in the arrangement of the magnets.

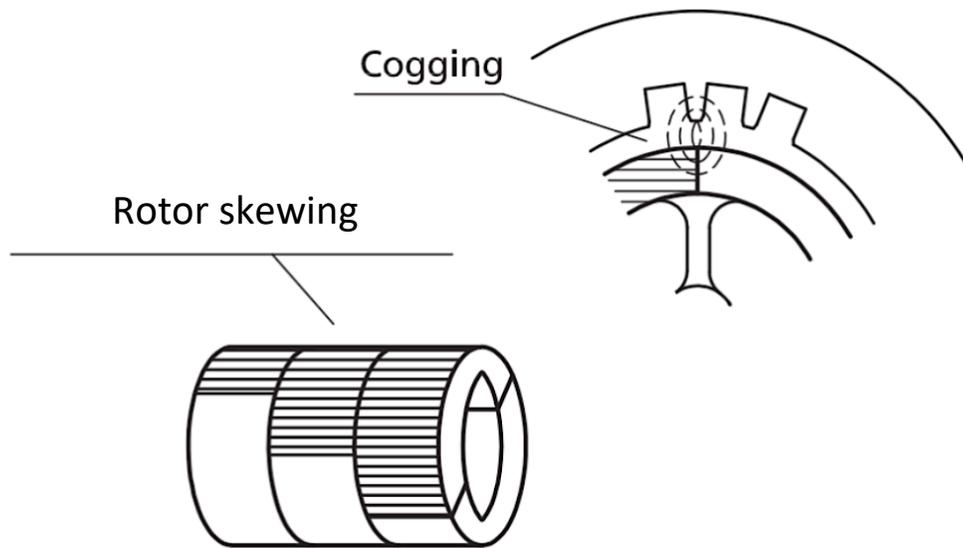


Figure 1. 16 – Cogging and skewing phenomena in BLDC motors

The figure below shows the block diagram of a drive for brushless DC motors.

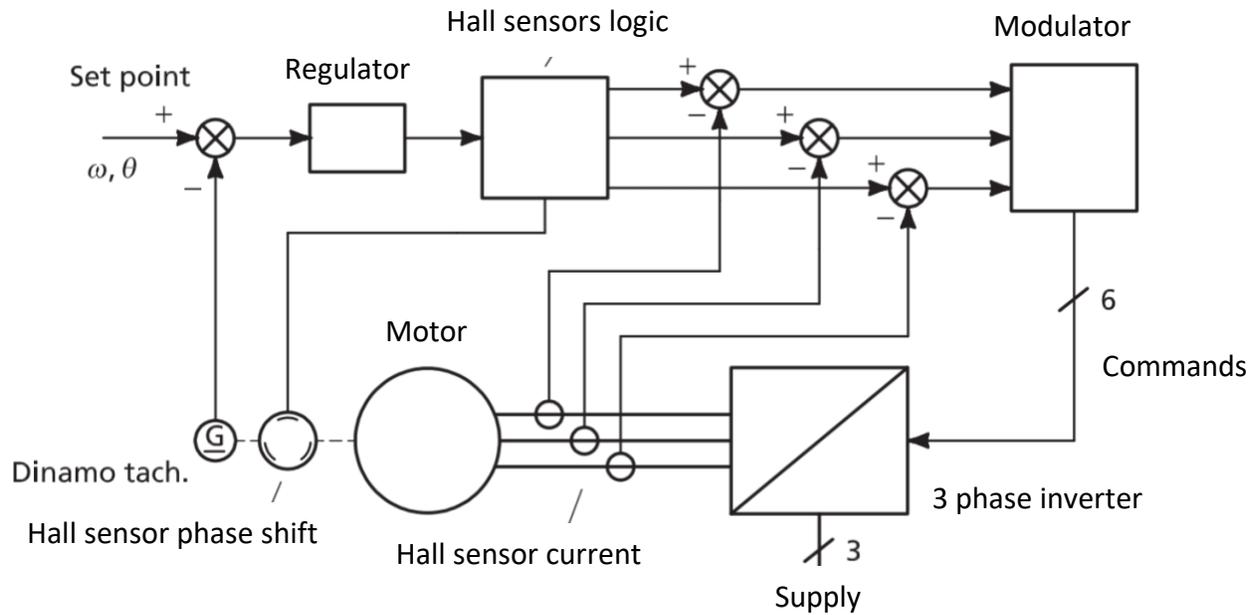


Figure 1. 17 – Block diagram of a BLDC motor drive

### 1.4.2 PMSM motors

The brushless AC motor has some similarities to the BLDC motor but is driven by a sinusoidal signal for less torque ripple. It is essentially an evolution of the DC one.

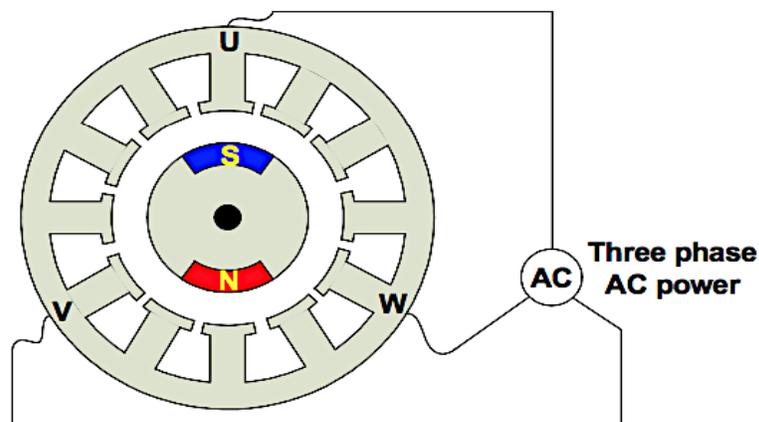


Figure 1. 18 – Structure of a PMSM motor

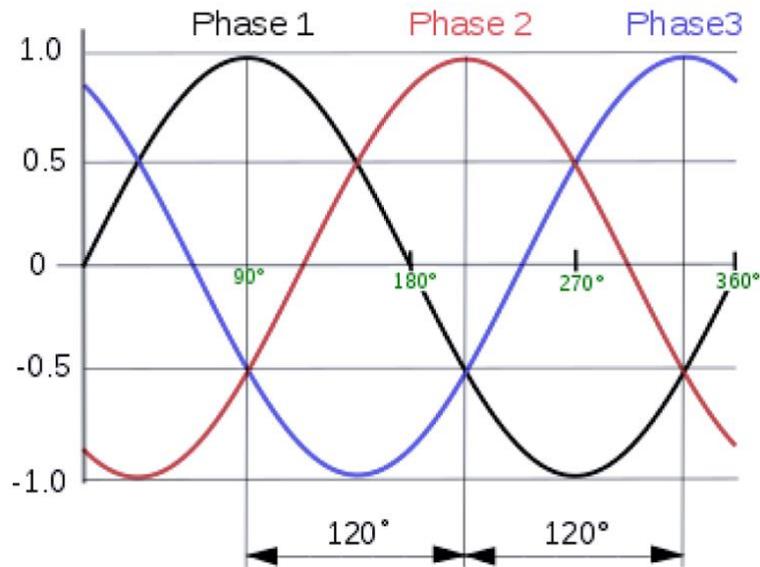


Figure 1.19 – PMSM motor waveforms

The torque  $T_m$  is still determined by the product of the e.m.f.  $E$  and from the armature current  $I_m$ ,

$$T_m = \frac{E \cdot I_m}{\omega} \quad (1.3)$$

In order to obtain a constant torque, it is necessary that the e.m.f. also has a sinusoidal trend. This is achieved by suitably magnetizing the rotor.

The torque is linked to the currents, but since these are sinusoidal, it is difficult to control by the drive. It is therefore necessary to have a model that highlights a controllable component of the current that influences the torque itself: if, in particular, it is possible to decompose  $I_m$  into the component that generates torque and into the component that generates flux, it leads back to the well-known situation of the DC motor. Two transformations are needed:

- Three-phase-two-phase transformation of the reference for electrical quantities;
- Transformation of the reference from fixed to rotating.

These transformations lead to refer the quantities to two orthogonal rotating axes: the d axis aligned with the flow and the q axis placed in quadrature (oriented machine).

It can be shown that the expression of the mechanical torque is:

$$T_m = \varphi_m \cdot i_q \quad (1.4)$$

Thus, with the appropriate orientation of the machine, the mechanical torque control is carried out by verifying the  $i_q$  component of the absorbed current. However, it should be noted that  $i_q$  will not be directly read by a sensor, but will be calculated in real time through the I absorbed by the phases and taking into account the exact rotor position. The brushless AC therefore needs at least two current transducers to read the absorbed current, a rotor position transducer and a very fast calculation system, such as a DSP (Digital Signal Processor), to obtain the  $i_q$  and to consequently control the supply.

Obviously, having the need to vary the three three-phase voltages that drive the motor as desired, it is also necessary to integrate an inverter into the system.

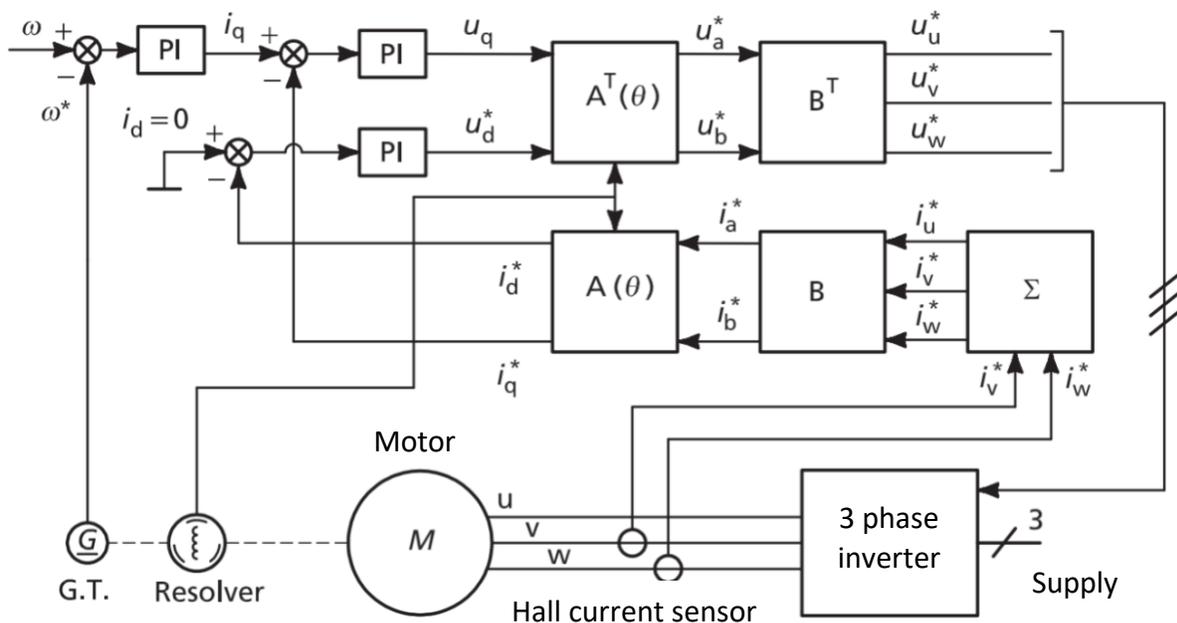


Figure 1. 20 - Block diagram of a PMSM motor drive

## 2. High Fidelity Model

### 2.1 Model description

It represents the numerical reference model able to simulate the behavior of a real servomechanism in order to carry out an early identification of the symptoms that are considered to be the failure precursors of EMA degradations. A suitable simulation test bench has been developed in MATLAB/Simulink.

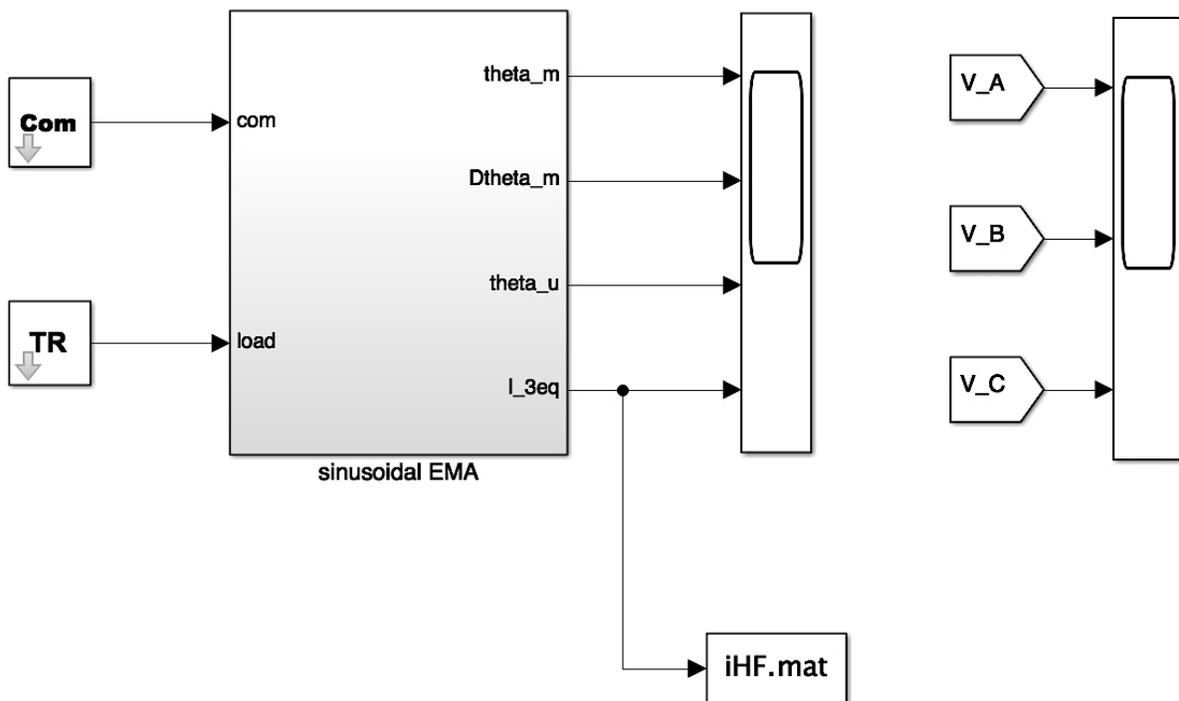


Figure 2. 1 – Overview of the High Fidelity model

The detailed model simulates a flap control with a response time between the primary and the secondary flight control systems. *Table 2.1* reports the simulation parameters that have been considered.

Parameter	Symbol	Value	UOM
Controller proportional gain	controller.Gprop	$10^5$	$\frac{1}{s}$
Proportional gain of the PID controller	controller.PID.GAP	0.05	$\frac{Nms}{rad}$
Integrative gain of the PID controller	controller.PID.GAI	0	$\frac{Nm}{rad}$
Derivative gain of the PID controller	controller.PID.GAD	0	$\frac{Nms^2}{rad}$
Maximum current	controller.I_Max	22.5	A
Maximum power supply voltage	inverter.Hbridge.Vdc	48	V
Current hysteresis bandwidth	inverter.PWM.hb	0.5	A
Maximum motor torque	PMSM.TMM	1.689	Nm
Back-EMF constant	PMSM. $k_e$	$\frac{0.0752}{2}$	$\frac{Nm}{A}$
Phase-to-phase resistance	PMSM.Rs	2.13	$\Omega$
Phase-to-phase inductance	PMSM.Ls	$7.2 \cdot 10^{-4}$	H
Number of pole-pairs per phase	PMSM.P	2	-
Nominal rotor static eccentricity	PMSM.zeta	0.5	-
Nominal rotor static eccentricity phase	PMSM.phi	1	-
Inertial Torque of the motor	dynamics.JM	$1.3 \cdot 10^{-5}$	$kg \cdot m^2$
Viscous damping coefficient of the motor	dynamics.CM	$\frac{30}{\pi} \cdot 10^{-6}$	$\frac{Nms}{rad}$
Inertial Torque of the user	dynamics.JU	$1.2 \cdot 10^{-5}$	$kg \cdot m^2$
Viscous damping coefficient of the user	dynamics.CU	$4.5 \cdot 10^{-7}$	$\frac{Nms}{rad}$
Static friction torque of the motor	dynamics.friction.FSTm	$0.06 \cdot PMSM.TMM$	Nm
Dynamic friction torque of the motor	dynamics.friction.FDTm	$dynamics.friction.FSTm/2$	Nm
Static friction torque of the user	dynamics.friction.FSTu	$0.04 \cdot PMSM.TMM$	Nm
Dynamic friction torque of the user	dynamics.friction.FDTu	$dynamics.friction.FSTu/2$	Nm
Nominal backlash	dynamics.BLK	$10^5$	rad

*Table 2. 1 – Main simulation parameters*

The *sinusoidal EMA block* includes the subsystems shown in *Figure 2.2*.

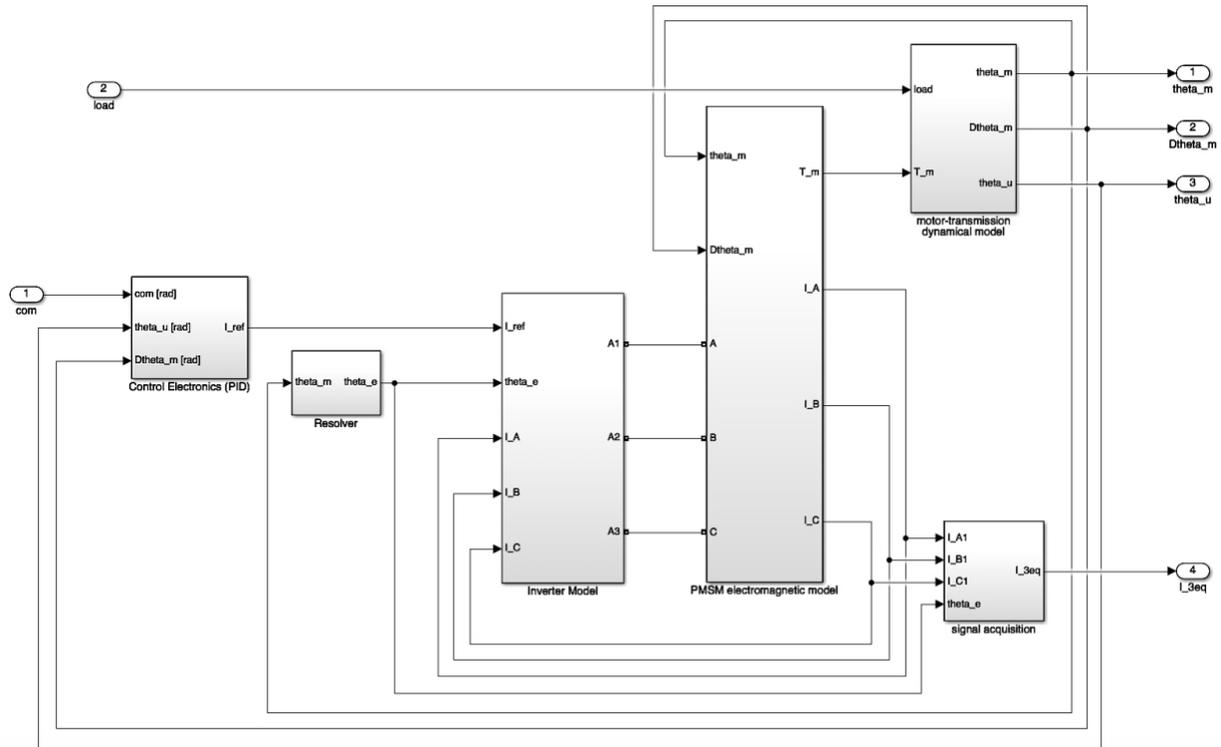


Figure 2. 2 – Inside the Sinusoidal EMA block

### 2.1.1 Com block

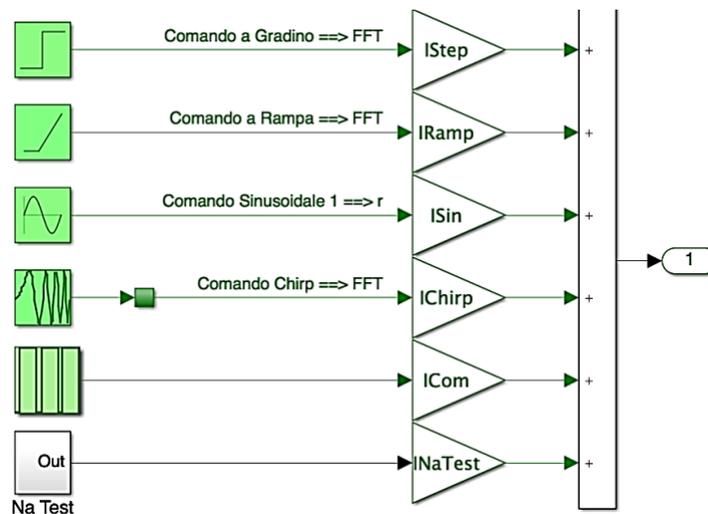


Figure 2. 3 – Com block

It allows to choose from several input commands such as step, ramp, sine wave, chirp or user defined time history. The selected command is sent to the actuator, which gives as output an angle that is then sent to the *Sinusoidal EMA* block. For this work, a chirp command has been chosen.

## 2.1.2 TR block

It is the torque of the external shaft and it represents the load needed by the *Motor Transmission Dynamical Model* block to produce the necessary data. It simulates the aerodynamic torques acting on the moving surfaces controlled by the actuator.

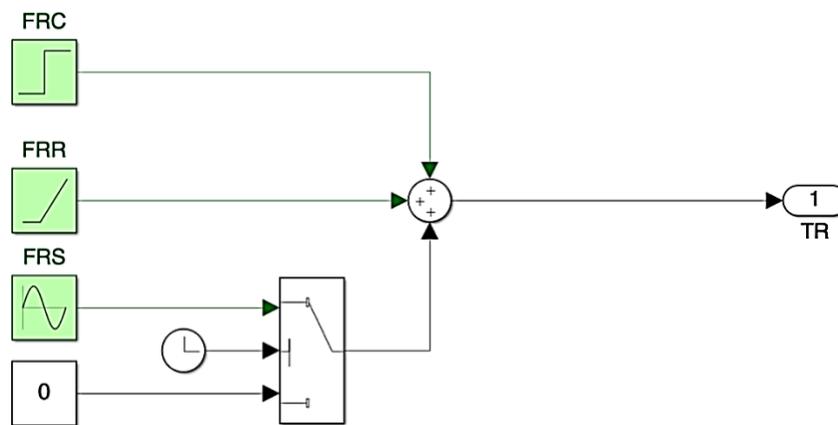


Figure 2. 4 – TR block

## 2.1.3 Control Electronics (PID) block

This block allows to control the overall EMA system. Given the inputs **com** (the position command),  $\theta_u$  (the feedback user position) and  $\dot{\theta}_m$  (the motor angular speed), it provides the reference current  $I_{ref}$  as output.

The position error  $Err_{pos}$ , given by comparing the position command **com** with the actual user position  $\theta_u$ , is suitably transformed into a speed signal by multiplying it by the *Controller Proportional Gain*  $G_{prop}$ , and it is limited by the saturation block between -8000 rpm and +8000 rpm. The resulting speed value  $\omega_{ref}$  is then compared to  $\dot{\theta}_m$ , producing the error sent to the *PID controller*. Here, the input signal is transformed into a reference torque, which is divided by the torque gain in order to obtain the reference current that is saturated at the maximum value of 22,5A

so as to avoid any failure condition of the motor. Before the computation of the real value of the reference current, a *white-noise disturbance* block is added generating random numbers that are suitable for continuous or hybrid systems. The acquired values are multiplied by  $10^{-6}$  to make the two signals comparable. The noise gain *Knoise* is then set to 0 since it would possibly amplify white noise affecting the feedback signals.

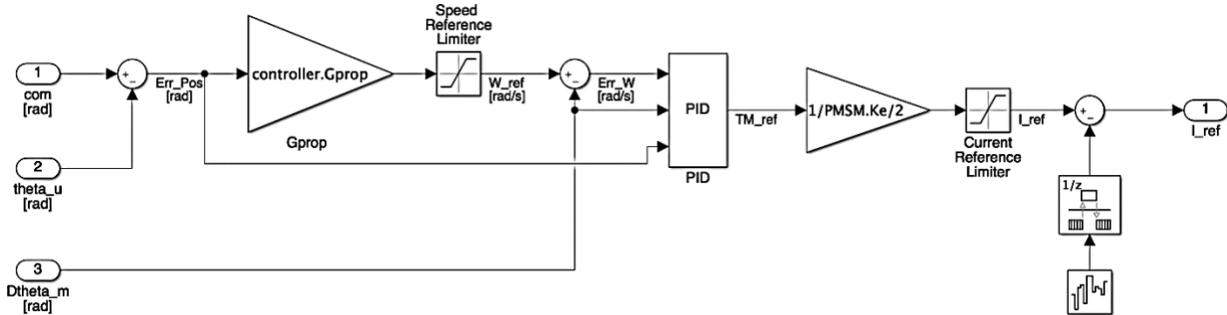


Figure 2. 5 – Control Electronics (PID) block

### 2.1.4 Resolver block

It converts the motor mechanical angular position  $\theta_m$  to the electrical one  $\theta_e$ , through the following expression:

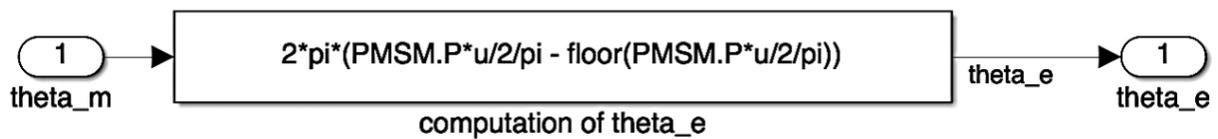


Figure 2. 6 – Resolver block

### 2.1.5 Inverter block

It takes in the reference current  $I_{ref}$ , the motor electrical position  $\theta_e$ , and the three-phased currents  $I_A$ ,  $I_B$  and  $I_C$  as feedback from the output of the *PMSM*

*electromagnetic model* to which it is physically connected through A, B, C connections.

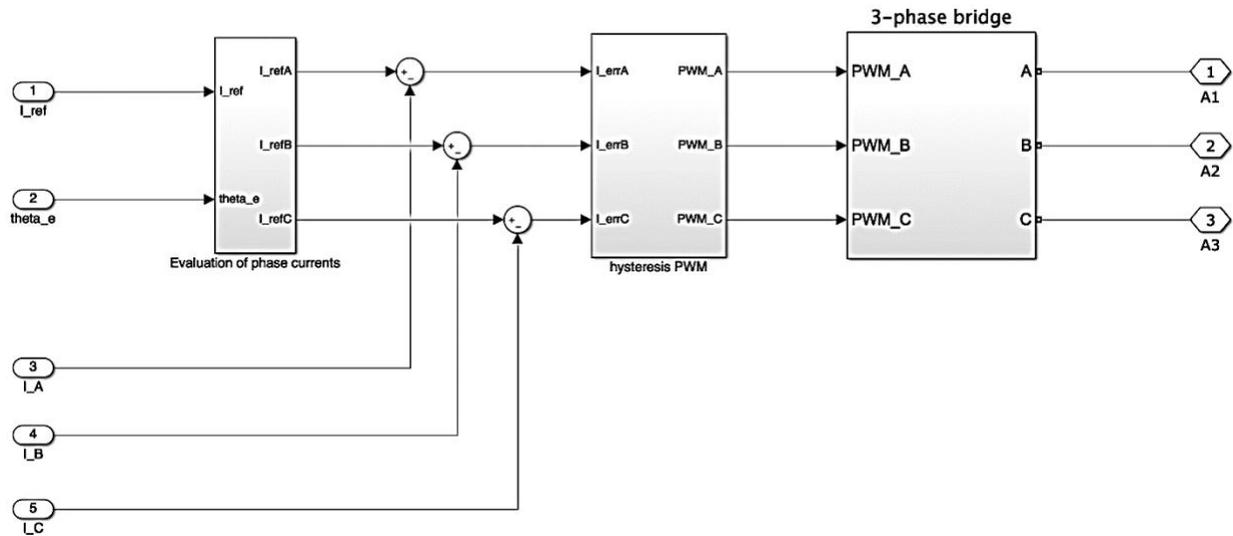


Figure 2. 7 – Inverter block

The *Evaluation of phase currents* subsystem transforms the single-phased reference current  $I_{ref}$  into the three phases  $I_{refA}$ ,  $I_{refB}$  and  $I_{refC}$  through the Clarke-Park transformation, as showed below.

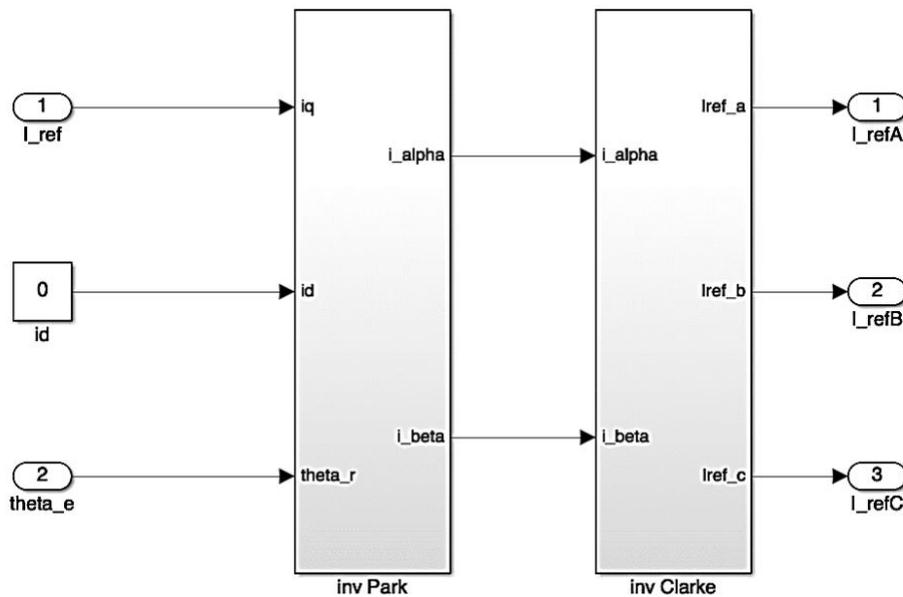


Figure 2. 8 – Evaluation of phase currents subsystem

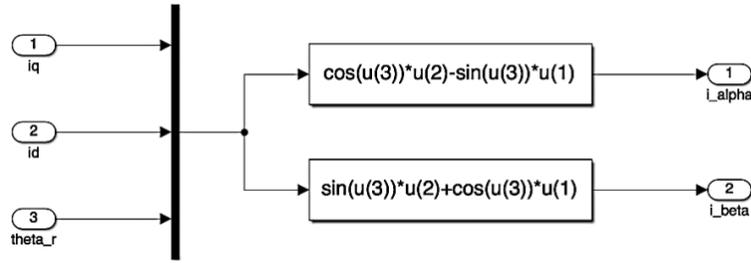


Figure 2. 9 – inv Park subsystem

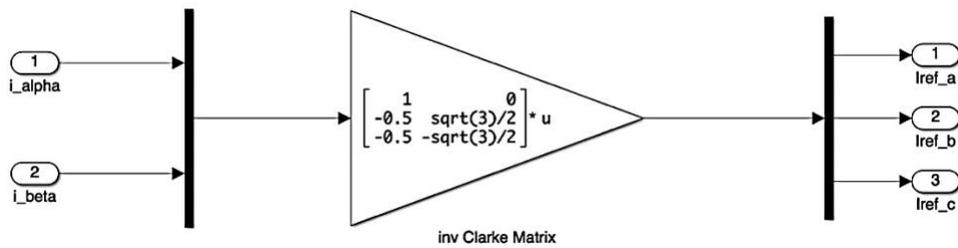


Figure 2. 10 – inv Clarke subsystem

Then, the feedback currents  $I_A$ ,  $I_B$  and  $I_C$  are subtracted from the acquired three-phased currents producing current errors  $I_{errA}$ ,  $I_{errB}$  and  $I_{errC}$ . These will be the inputs of the *Hysteresis PWM* subsystem containing the pulse width modulation process, whose products feed the *3-phase Bridge* box containing the three-phase bridge electrical model.

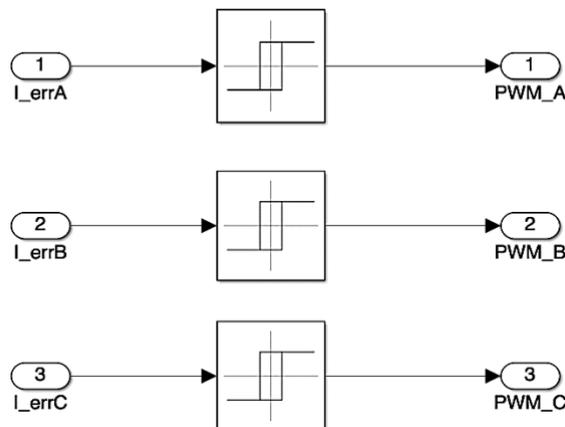


Figure 2. 11 – Hysteresis PWM subsystem

The *three-phase bridge* block simulates the H-bridge static power converter. It employs a *Universal Bridge* Simulink block from the Simscape library, and it connects it to a DC voltage source corresponding to the inverter supply voltage  $V_{DC}$ . Each PWM Boolean signal and its negation activate one of the six MOSFET transistors of the three phase H-bridge. Being the two PWM signals complementary, when the transistor connecting one phase to the supply voltage is on, the corresponding one connecting the same phase to ground is off and vice versa. In this way, it avoids any short-circuit that may occur between supply and ground inside the bridge. The three output voltages from the H-bridge are then fed to the *Three Phase RL Model* of the *PMSM Electromagnetic Model* block.

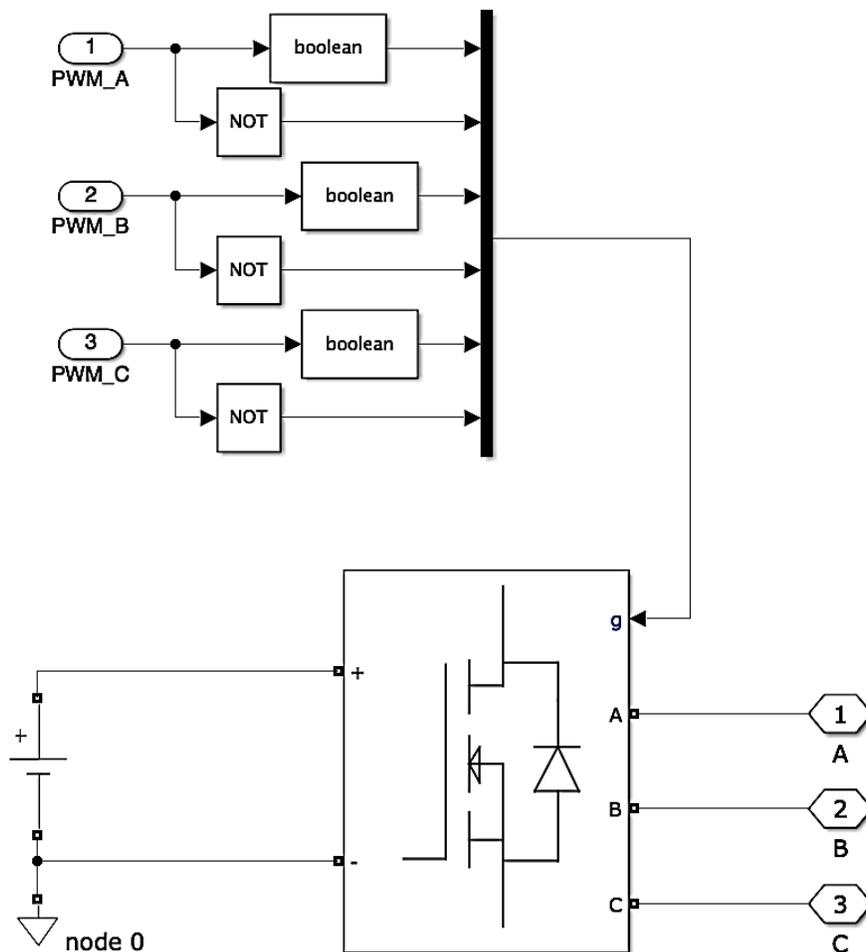


Figure 2. 12 – 3-phase bridge subsystem

## 2.1.6 PMSM Electromagnetic Model block

This block receives the motor angular position  $\theta_m$  and speed  $\dot{\theta}_m$  as inputs, computes the back EMF coefficients and, through an ohmic-inductive model, produces the motor torque and phase currents.

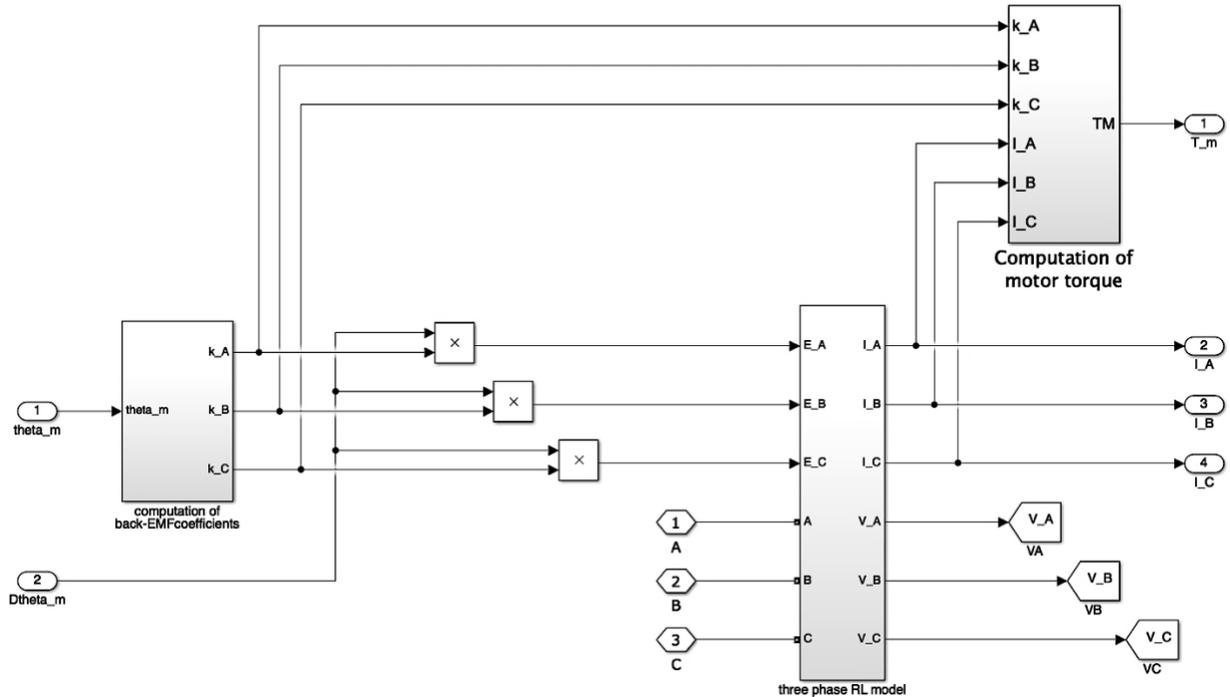


Figure 2. 13 – PMSM Electromagnetic Model block

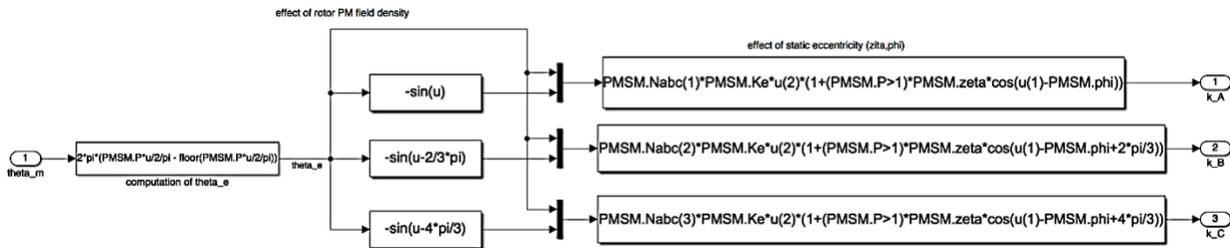


Figure 2. 14 – Computation of back-EMF coefficients block

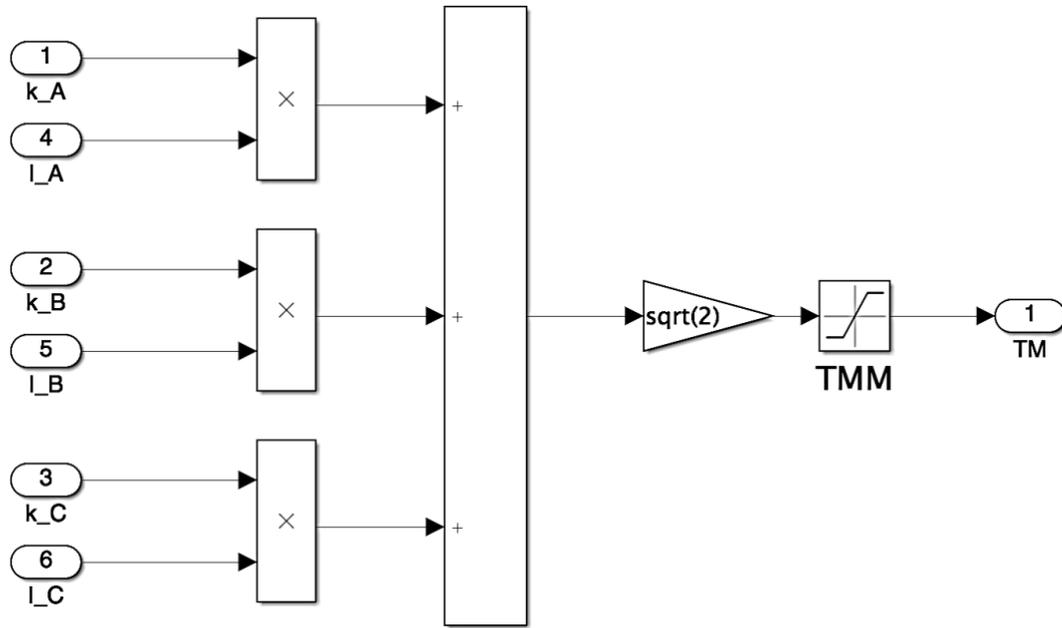


Figure 2. 15 – Computation of motor torque subsystem

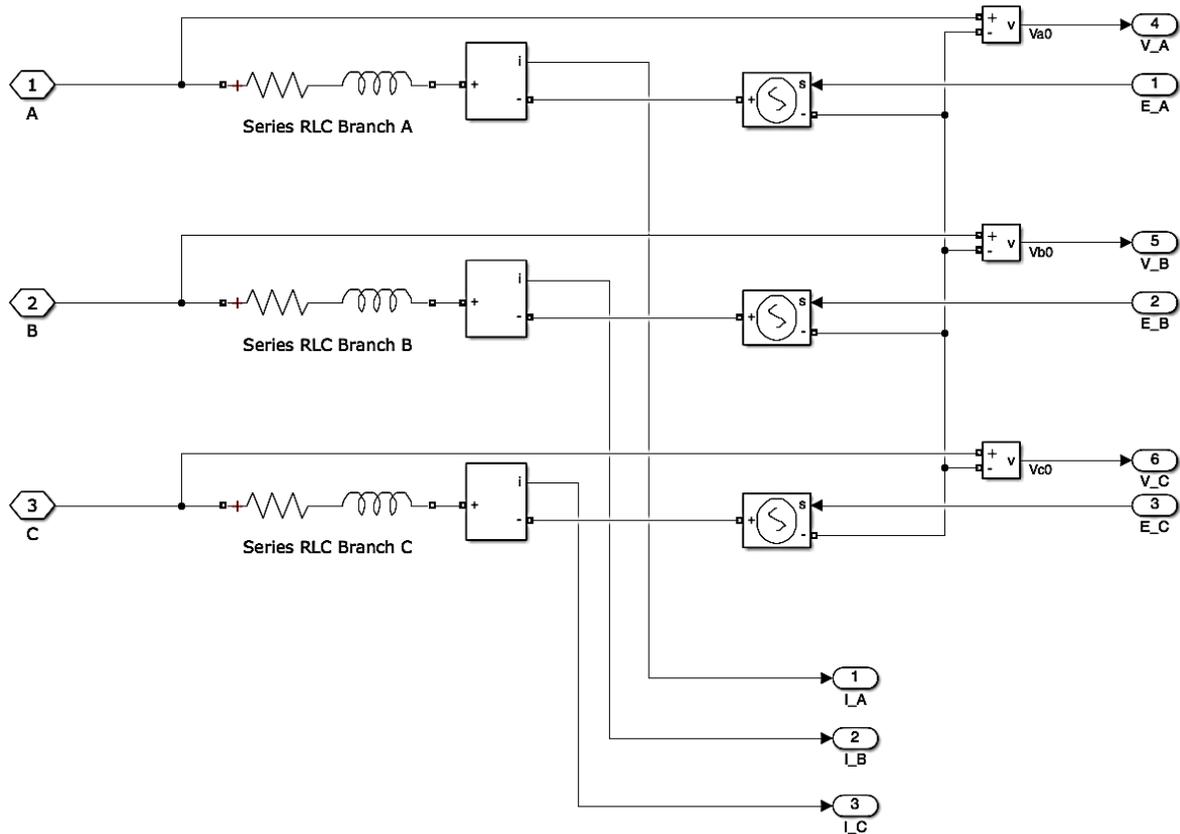


Figure 2. 16 – Three phase RL model

## 2.1.7 Motor-Transmission Dynamical Model block

It contains the mechanical model of the motor-reducer group, including the Borello's friction model. It is able to provide the motor and the user angular speeds and positions, starting from the motor torque and the external load.

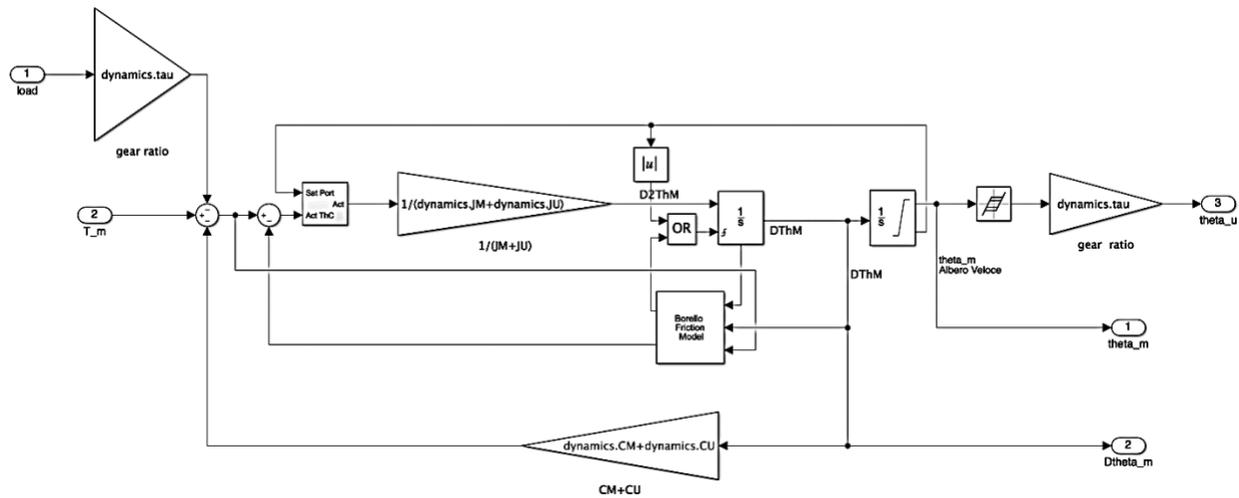


Figure 2. 17 – Motor-Transmission Dynamical Model block

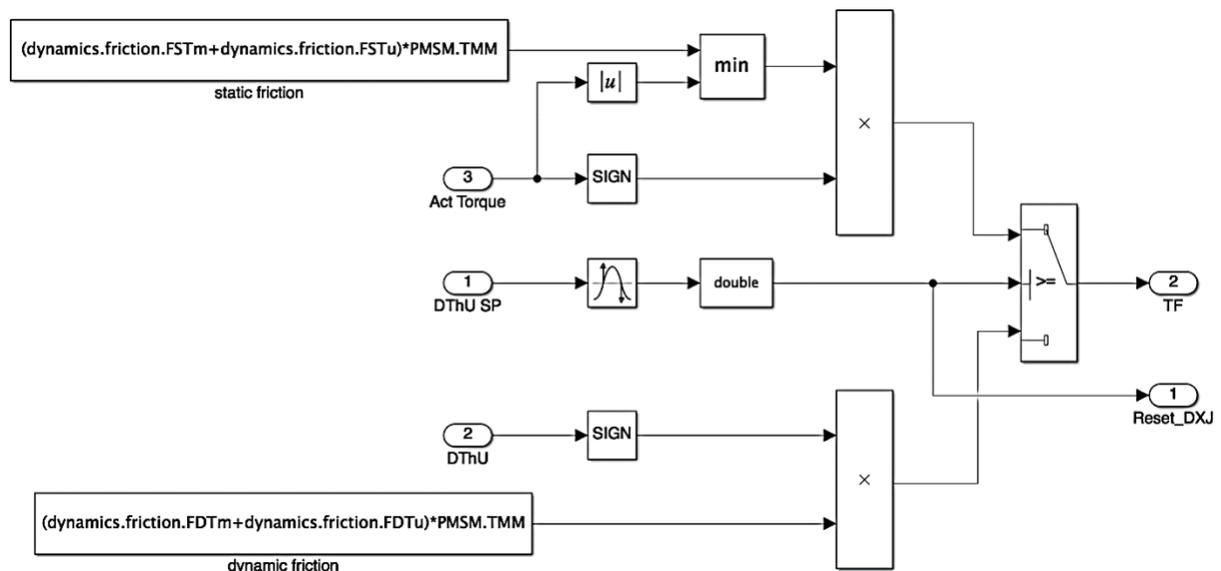


Figure 2. 18 – Borello Friction Model subsystem

## 2.1.8 Signal acquisition block

It computes a single-phasic current “proportional” to the three-phasic one.

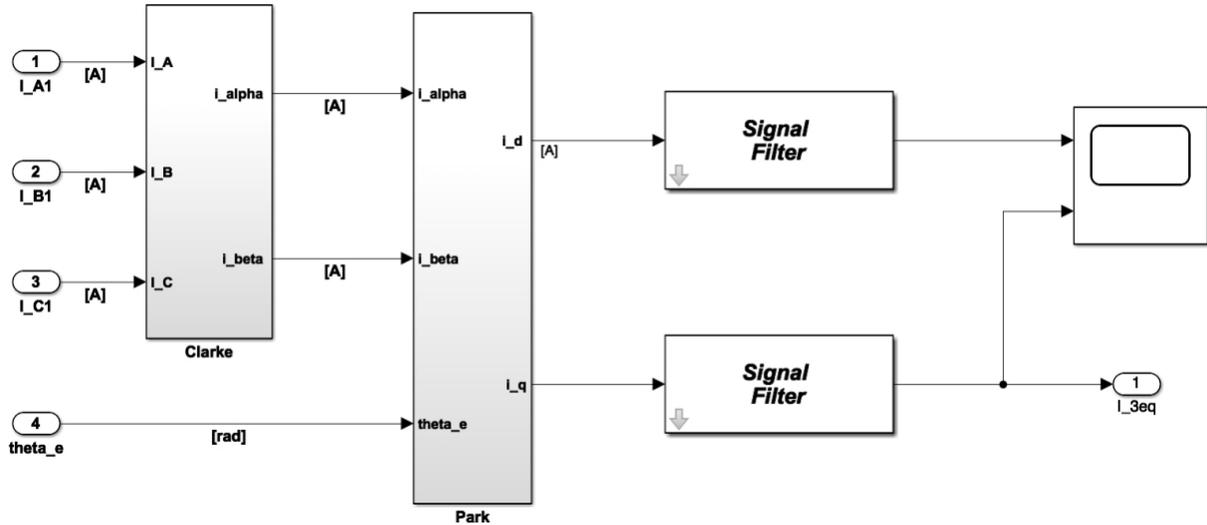


Figure 2. 19 – Signal acquisition block

The single-phase currents  $I_{A1}$ ,  $I_{B1}$ , and  $I_{C1}$ , which are the contributions of the single coils coming from the PMSM electromagnetic model, go through the Clarke-Park transformation boxes to produce the direct and quadrature currents,  $I_d$  and  $I_q$ . A filtering stage is needed to get rid of higher frequencies in the signal caused by PWM and undesired noise.

Being the output parameter of the monitor model single phasic, the output of the real model should be a single-phase equivalent current in order to be comparable to it. This current is a function of the direct and quadrature currents:

$$I_{3eq} = I_q \cdot \hat{q} + I_d \cdot \hat{d} \quad (2.1)$$

$$I_q = \left[ -I_A + \frac{1}{2}(I_B + I_C) \right] \sin\theta_e + \frac{\sqrt{3}}{2}(I_B - I_C) \cos\theta_e \quad (2.2)$$

To ensure a proper permutation sequence, the resultant coils supply current should always be perpendicular (in terms of electric angle) to the rotor magnetic field.

As a consequence,  $I_d$  is always going to be null, thus causing:

$$I_d = 0 \quad (2.3)$$

$$I_{3eq} = I_q \cdot \hat{q} \quad (2.4)$$

The direct current  $I_{3eq}$  will then be compared to the one resulting from the monitor model.

## 2.2 Reference Model Output Response

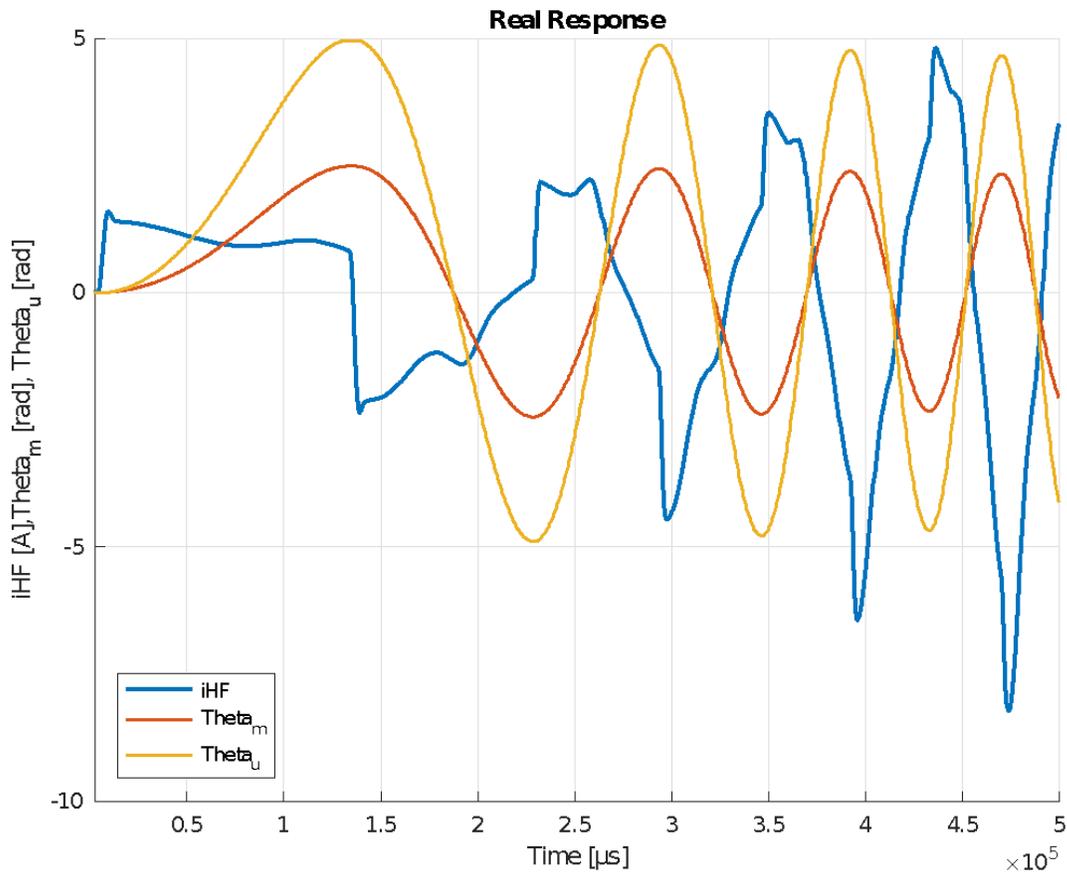


Figure 2. 20 – Output response of the reference model

### 3. Low Fidelity Model

#### 3.1 Model description

The monitor model is a simplification of the real model. The three-phase current conversion has been skipped and with it the Clarke-Park inverse transformation, hysteresis PWM generation and the three-phase electrical model. The electromagnetic model, on the other hand, does not require the computation of the EMF parameters and it is replaced by a single-phase ohmic-inductive model which the *Electrical Model* block.

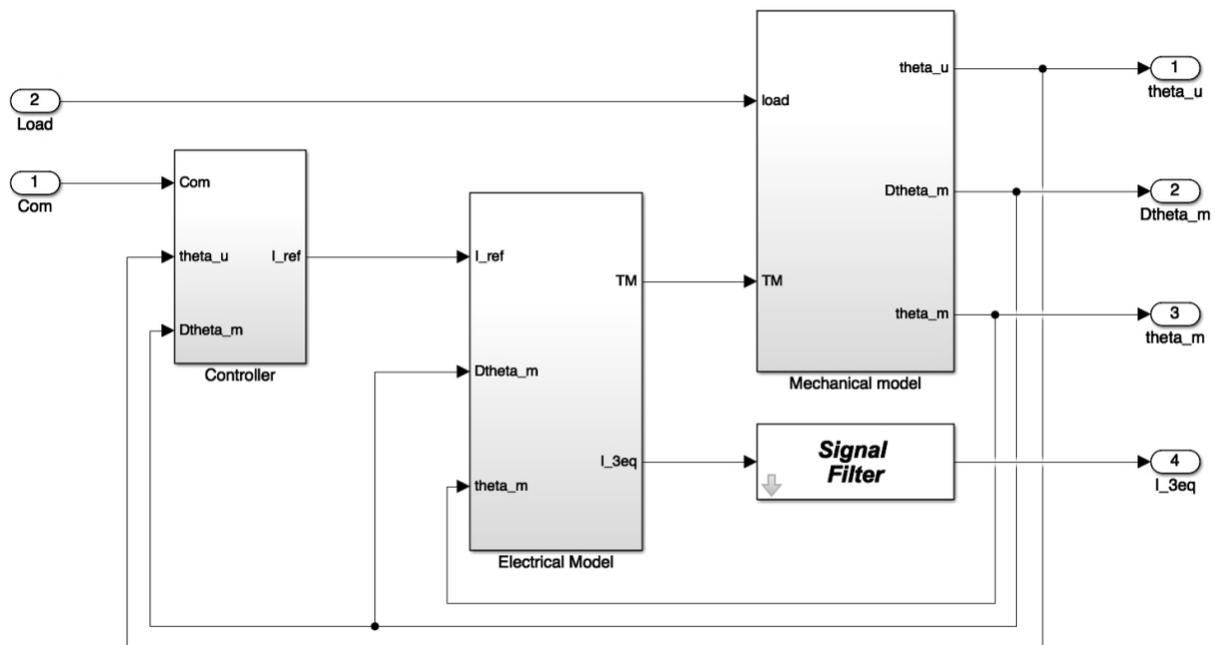


Figure 3. 1 – Overview of the Low Fidelity model

**Com** is the position command in radians elaborated by the *Control Electronics (PID)* block, together with “low speed” shaft angular position ( $\theta_u$ , due to a reduction stage downstream of the motor) and motor speed loops ( $\dot{\theta}_m$ , related to “high speed” shaft), which produces a reference current  $I_{ref}$  as output.

### 3.1.1 Controller block

The control electronics block computes the position error ( $Err_{pos} = Com - \theta_u$ ) and multiplies it by the controller proportional gain, set to:

$$GAPm1 = 10^{-5} \left[ \frac{1}{s} \right] \quad (3.1)$$

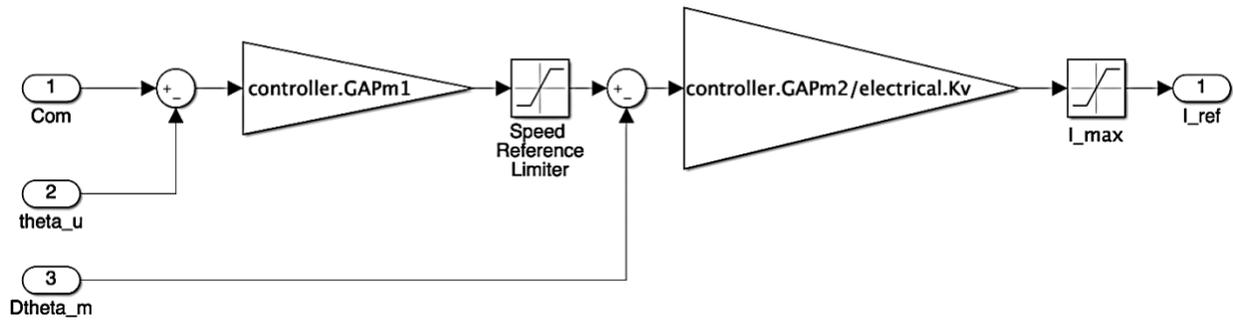


Figure 3. 2 – Controller block

Thus, the actuation speed is obtained, followed by its saturation set to:

$$\omega_{reflim} = \pm 8000 \text{ [RPM]} \cdot \frac{2\pi}{60} = \pm 800 \cdot \frac{\pi}{3} \left[ \frac{rad}{s} \right] \quad (3.2)$$

Next is the speed error calculation ( $Err_w = W_{ref} - \dot{\theta}_m$ ), which is then multiplied by the PID proportional gain. The reference current  $I_{ref}$  is obtained, and its admitted boundary values are set by the  $I_{max}$  box:

$$I_{reflim} = \pm I_{maxHF} = \pm 22.5 \text{ [A]} \quad (3.3)$$

where  $I_{maxHF} = \pm 22.5 \text{ [A]}$  is the maximum allowed current for the high fidelity model.

### 3.1.2 Electrical Model block

Once  $I_{ref}$  is obtained from the controller, the signal enters the *Electrical model* system, containing the following:

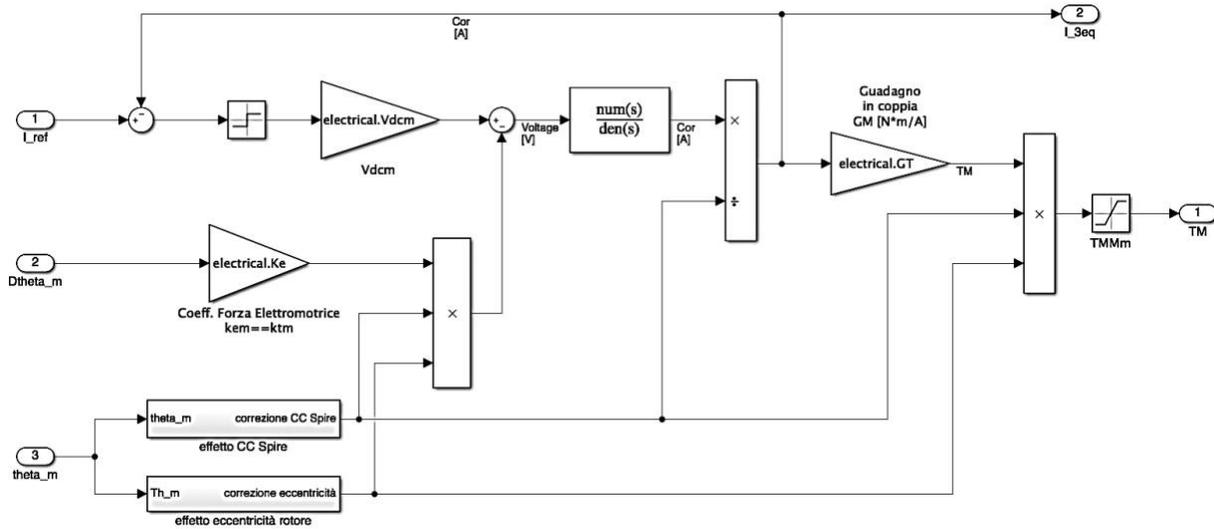


Figure 3. 3 – Electrical Model block

The  $I_{ref}$ , subtracted of the Cor loop, produces  $I_{err}$ :

$$I_{err} = I_{ref} - Cor [A] \quad (3.4)$$

Which enters the Current Sign block, deducting its sign in order to assign it to the Supply Voltage:

$$V_{dcm} = \pm 48 [V] \quad (3.5)$$

Another subtraction stage deprives  $V_{dcm}$  of the series of multiplications which involve:

- The *Back ElectroMotive Force* contribution, given by the motor angular speed  $\dot{\theta}_m$  multiplied by the *Back EMF Coefficient*, equal to:

$$K_e = \frac{0.0752}{3} \left[ \frac{V \cdot s}{rad} \right] \quad (3.6)$$

Thus having:

$$V [V] = Dtheta\_m \left[ \frac{rad}{s} \right] \cdot K_e \left[ \frac{V \cdot s}{rad} \right] \quad (3.7)$$

- The *Winding SC Correction* and *Rotor Eccentricity Correction*, which take  $\theta_m$  as input and contain a form function able to model, respectively, the effect of partial windings short circuit and mechanical failures due to rotor deformation or misalignment. In both cases, a non-dimensional multiplicative coefficient is generated and is going to affect  $V$  (Back EMF contribution), producing  $V_f [V]$ ,  $I_{out}$ , giving  $I [A]$  and  $TM$ , having  $TM_f [Nm]$ .

The mechanical correction has no direct effects on the ohmic-inductive motor model. Thus,  $I_{out}$  is given by the motor phase-to-phase resistance  $R_m$  which, being proportional to the number of working coils, needs to be multiplied by the SC correction which has a precise contribution of the inductance  $L_m$  multiplying it by SC correction too. A better solution has been considered using the mean working coils number  $N_{tot}/3$ . This way, the motor transfer function affected by faults, is going to be:

$$\frac{1}{\Phi_{SC} \cdot R_m \left[ \left( \frac{L_m}{R_m} \cdot \frac{N_{tot}}{3} \right) s + 1 \right]} \quad (3.8)$$

For nominal condition we have the following data:

$$\left\{ \begin{array}{l} R_m = 2.130 [\Omega] \\ L_m = 720 [\mu H] \\ N_A = N_B = N_C = 1 \\ N_{tot} = N_A + N_B + N_C = 3 \end{array} \right. \quad \begin{array}{l} (3.9) \\ (3.10) \\ (3.11) \\ (3.12) \end{array}$$

$\Phi_{SC}$  is the SC correction function value and  $N_{A,B,C}$  is the percentage of windings A, B, C working coils with  $0 \leq N_{A,B,C} \leq 1$ , where 1 means completely functional, while 0 means totally damaged.

The  $I_{out}$  goes both to the output 2, as  $I$ , and through the Gain Torque:

$$GT = 0.0752 \left[ \frac{Nm}{A} \right] \quad (3.13)$$

obtained by the calibration process.

It then goes through the *Torque Saturation* box given by the eventual  $I_{err}$  saturation and equal to:

$$TM_{lim} = \pm 1.689 [Nm] \quad (3.14)$$

TM is the output number 1.

After that, one has:

$$\begin{cases} V_f = \Phi_{SC} \cdot \Phi_E \cdot V [\Omega] & (3.15) \\ TM_f = \Phi_{SC} \cdot \Phi_E \cdot TM [Nm] & (3.16) \end{cases}$$

$\Phi_E$  being the eccentricity correction function parameter.

It can be observed from the overall picture that the torque output needs to be modified as well by the form function: that's because the gain torque, which is basically the same as BEMF coefficient, is affected by the number of working coils and by the air-gap depth, whose contributions are not included into the motor model transfer function modification, as might be thought.

### 3.1.2.1 Winding SC Correction

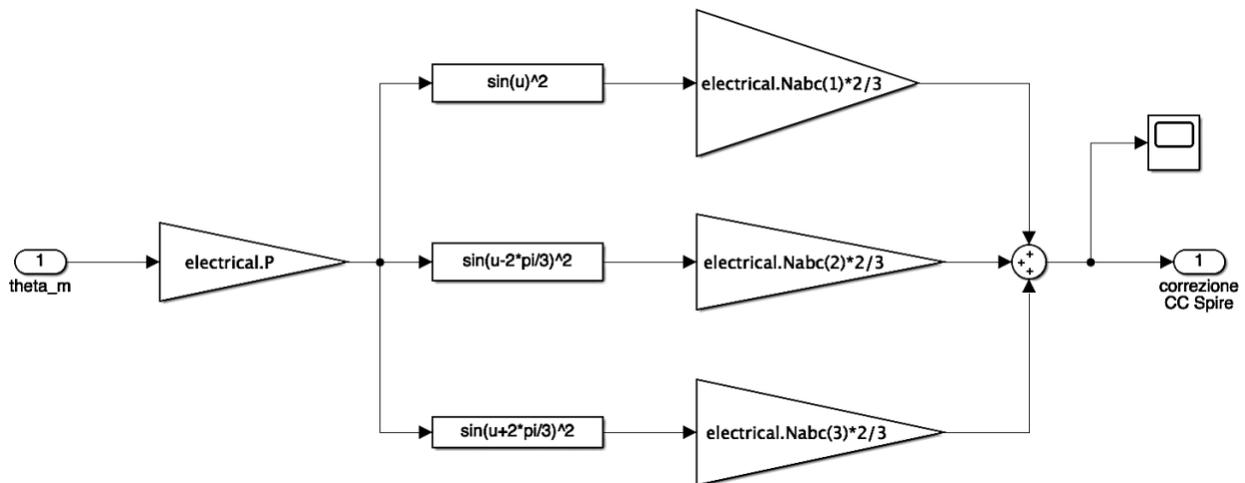


Figure 3. 4 – Winding SC Correction subsystem

The *Winding SC correction* subsystem provides as an output the correction coefficient which is proportional to the percentage of working coils  $[N_A, N_B, N_C]$  and to the motor angular position.

It starts by converting the mechanical angular position into an electrical one by multiplying it by the number of pole pairs per phase, whereas the second section of the block contains the actual form function which models the possible short circuits within the single phase as follows:

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

Where  $N_P$  is the short circuit total contribution gain and  $N_{A,B,C}$  is the short circuit single contribution gain, and they are both needed to minimize the error adjusting between the high fidelity and the monitor model.

### 3.1.2.2 Rotor Eccentricity Correction

A similar procedure has been adopted for the *Rotor Eccentricity Correction* subsystem.

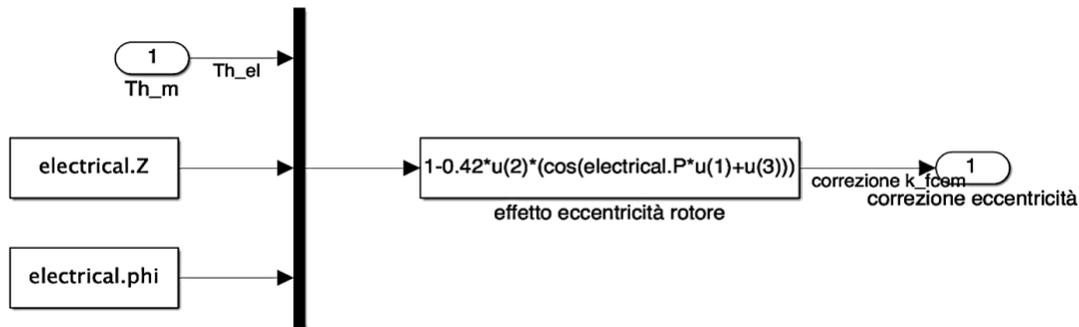


Figure 3. 5 – Rotor Eccentricity Correction subsystem

The computation of the rotor eccentricity generates a non-dimensional correction coefficient which has an effect on the Back emf and the output torque, without affecting the output current thanks to its mechanical nature. Specifically, the aim of this subsystem is to represent the static misalignment caused by a possible deformation of the rotor axis, by accepting as input a value in the range:

$$0 \leq Z \leq 1 \rightarrow 0 \leq \frac{\Delta}{Z_0} \leq 1 \quad (3.18)$$

where  $\Delta$  is the distance between the rotational axis in normal conditions and the deformed one and  $Z_0$  is the nominal air gap depth.

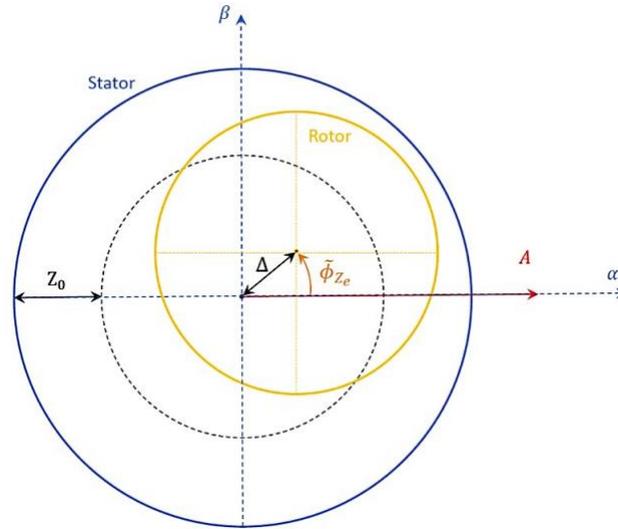


Figure 3. 6 – Static eccentricity scheme

Therefore, static eccentricity is present when the rotational axis coincides with the deformed rotor axis so that the closest and the furthest points to the stator are in the same angular position.

The value 0 represents the case with no axis deformation whereas 1 is the condition which can cause creeping between stator and rotor.

The second input is the phase eccentricity which indicates the maximum deflection angular position with respect to the reference system:

$$0 \leq \phi_Z \leq 1 \rightarrow 0 \leq \widetilde{\phi}_{Z_m} \leq (2\pi)_m \rightarrow 0 \leq \widetilde{\phi}_{Z_e} \leq (4\pi)_e \quad (3.19)$$

$$\phi_Z = \frac{\widetilde{\phi}_{Z_m}}{(2\pi)_m} = \frac{\widetilde{\phi}_{Z_e}}{(4\pi)_e} \quad (3.20)$$

Where  $m$  denotes the mechanical angle,  $e$  the electrical angle and  $\phi_Z$  the normalized value of  $\widetilde{\phi}_Z$  with respect to the chosen angle.

The third input is theta m which is then converted in  $\theta_e$ :

$$\theta_e = N_p \cdot \theta_m = 2 \cdot \theta_m \quad (3.21)$$

The total effect of the rotor eccentricity is then given by the following equation:

$$\Phi_E = 1 - 0.42 \cdot Z [\cos(\theta_e + \widetilde{\phi}_{Z_e})] \quad (3.22)$$

where No is the number of pair poles and the value 0.42 is the starting value that has been chosen since it works well for the PMSM motor.

### 3.1.3 Mechanical Model block

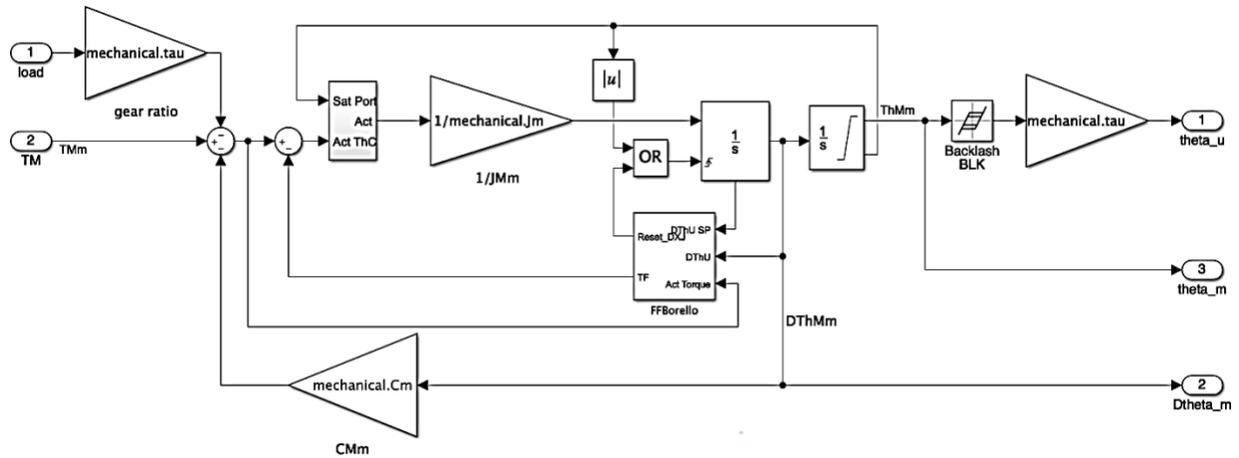


Figure 3. 7 – Mechanical Model block

The inputs are given by **Load** and motor torque **TM** coming from the Electrical Model. The first one is multiplied by the Gear Ratio:

$$\tau = \frac{1}{500} \quad (3.23)$$

**TM**, on the other hand, is subtracted of the load contribution and of viscous friction factor, given by:

$$(C_m + C_u) \cdot \dot{\theta}_m = C_{Mm} \cdot \dot{\theta}_m \quad (3.24)$$

Where the dumping component of the system is equal to:

$$C_{Mm} = C_m + C_u = 5.172^{-5} \left[ \frac{Nm \cdot s}{rad} \right] \quad (3.25)$$

With  $C_m$  and  $C_u$  being, respectively, the motor and user contribution. This way one has:

$$TM [Nm] - \tau \cdot Load [Nm] - C_{Mm} \cdot \dot{\theta}_m \left[ \frac{Nm \cdot s}{rad} \right] \cdot \left[ \frac{rad}{T} \right] = T [Nm] \quad (3.26)$$

The next step is the active torque calculation **Act\_ThC** producing **Act** as output:

$$Act\_ThC = T - T_f [Nm] \quad (3.27)$$

With  $T_f$  being the Coulomb friction contribution to the motor torque, computed by Borello Friction Model. The FFFBorello block chooses between static and dynamic friction coefficients by evaluating whether the motor angular speed changed sign or not, through a detection point. When this happens, it sends a reset signal to the first integrator, in order to set the speed output to zero. Then, the friction model receives as inputs two consecutive motor speed steps ( $\dot{\theta}_{Mm}$  **(-1)** and  $\dot{\theta}_{Mm}$ ) and the torque **T**, which is needed to understand whether the conditions are complementary or opposite. The **Act ThC** enters a switch box which verifies whether mechanical end-stops conditions occurred or not and states which output we are going to have, respectively, between zero and the input active torque. Hence, this box needs the saturation port Saturation signal, coming from the integrator to  $\theta_m$ , which is going to produce the following values:

$$\begin{cases} +1 & \text{Upper mechanical end - stop} \\ 0 & \text{Not limited} \\ -1 & \text{Lower mechanical end - stop} \end{cases} \quad (3.28)$$

Hence, if **Act\_ThC** has the same sign as Saturation, then the output is going to be null, while in all the other cases, it states **Act = Act\_ThC**. Notice that, if Saturation is other than zero, a second kind of reset signal is sent to the first integrator, allowing it to produce a null speed value for the next integration step. The **Act** encounters the gain:

$$\frac{1}{J_m + J_u} = \frac{1}{J_{tot}} = \frac{1}{2.5^{-5}} \left[ \frac{rad}{kg \cdot m^2} \right] \quad (3.29)$$

With  $J_{tot}$  being the system total moment of inertia, given by the sum of the motor ( $J_m$ ) and user ( $J_u$ ) ones. In such a way, we obtain the motor angular acceleration  $\ddot{\theta}_m$ , applying the fundamental law of rotational dynamics, given by:

$$T = J \cdot \ddot{\theta}_m \Rightarrow \ddot{\theta}_m = \frac{T}{J} \quad (3.30)$$

The following step is the first integration, providing the angular speed  $\dot{\theta}_m$  and whose box was a reset signal input port and a state port about which has already been discussed. The second integration box is limited, as could be understood from what was said above, and these limits represent the mechanical end-stops in terms of motor angular position; its output is, indeed, motor mechanical angle  $\theta_m$ . As last “non-linearity” we have a Backlash box, coarsely modeling the reducer mechanical clearance.

### 3.2 Monitor Model Output Response

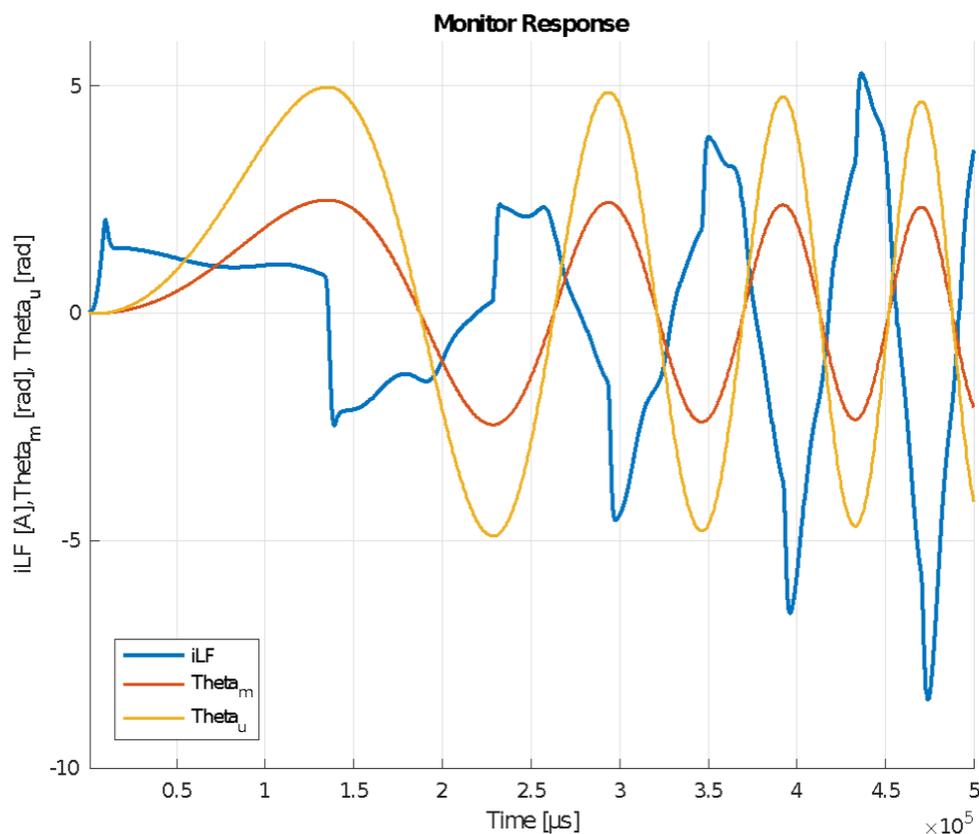


Figure 3. 8 – Output response of the monitor model

## 4. Fault analysis and implementation

### 4.1 Introduction to faults

As stated before, electromechanical actuators have been employed in aerospace applications only recently, making the accumulated flight hours or installations on board insufficient data to use to get reliable statistics on recurring failures. However, it is possible to distinguish between four main categories of failures:

- *Mechanical or structural*, affecting gear reducer and transmission and they are mainly due to high loads, manufacturing defects and problems related to lubrication;
- *PMSM motor failures*, linked to the vibrations and stress applied to the rotor bearing because of the high rotational speed of the motor. Overheating as well can produce damages as temperature is hard to control in electrical motors compared to hydraulic ones;
- *Electronics failures*, due to overheating and overcurrents that can lead to short circuit and vibrations;
- *Sensor failures*, mainly bias, scaling or drift faults. They can influence the dynamic response of the system as feedback signals can be measured incorrectly.

Only the first two categories will be analyzed in this work, in particular the effects of mechanical failures resulting from progressive wear, which manifests itself in an increase of backlash and friction, as well as two common PMSM motor failures, the coil-short circuits and the bearing gear generating rotor static eccentricity, together with a drift of the proportional gain of the controller. Very often, the detection as well as the assessment of mechanical failure due to friction and/or backlash is carried out directly by performing an analysis on specific properties of the dynamic response of the overall actuation system, in terms of position, speed or acceleration. On the

other hand, when taking into account motor progressive failures such as coil short circuit or rotor static eccentricity, the characteristics of the mechanical transmission, present as inertia, dry and viscous frictions, backlashes, etc., may conceal or in some cases reduce the effects that such failures can cause resulting in the inaccuracy, and in serious cases the inefficiency, of any prognostic attempt. In these situations, a more accurate result and better understanding of the incipient failure and its progression may be achieved by analyzing the electrical harmonics (e.g. phase currents). Electronic and sensor failures are no less relevant, although their failure precursors are often hard to identify and analyze as they usually occur very quickly, if not instantaneously.

## 4.2 Friction

Dry friction can be defined as the resistance to relative movements of two solid surfaces in contact and all machines consume frictional energy because their parts slide against each other. Therefore, it occurs when two surfaces are in relative motion and, as its coefficient increases due to wear, the reaction torque grows as well leading to the need of greater torques from the motor to operate on the control surfaces. Such an increase of the dry friction causes the breakdown of the entire system in addition to a reduction in the servomechanism accuracy and occasionally the generation of unexpected behaviors, such as stick-slip or limit cycles, in the system dynamic response. It can cause other types of instabilities due to the decrease in the frictional force with increasing sliding speed, by material expansion due to the generation of heat during friction, or by pure dynamic effects of sliding of two elastic materials. Friction is implemented in both the reference and the monitor model in the *Borello* block as a linear Coulomb friction. The numerical method has been employed in the time domain and its mathematical model can be modeled as follows:

$$F_f = \begin{cases} F_{act} & \dot{x} = 0 \cap |F_{act}| \leq F_{sj} \\ F_{dj} \cdot \text{sign}(F_{act}) & \dot{x} = 0 \cap |F_{act}| > F_{sj} \\ F_{dj} \cdot \text{sign}(\dot{x}) & \dot{x} \neq 0 \end{cases} \quad (4.1)$$

where  $F_f$  is the computed friction force,  $F_{act}$  is the active force applied to the system,  $F_{sj}$  is the friction force in stick condition and  $F_{dj}$  the friction force in dynamic conditions.

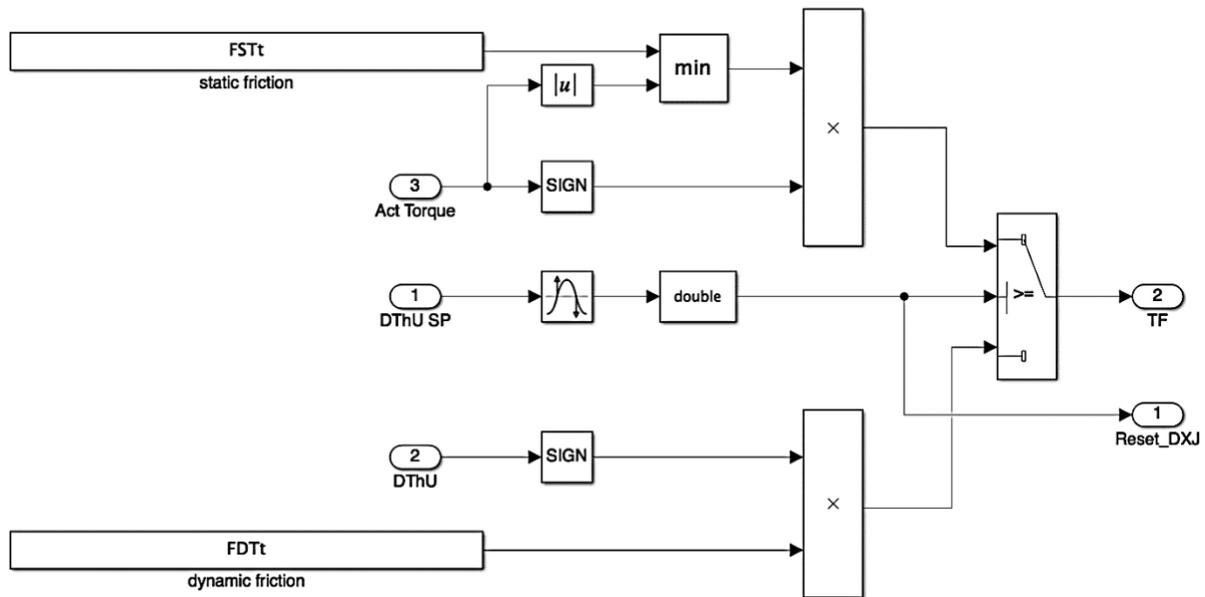


Figure 4. 1 – Simulink implementation of the Borello friction block

### 4.3 Backlash

Another outcome of mechanical wear is backlash, which can be generated in the moving components of the electromechanical actuators such as gears, hinges, bearings and especially screw actuators. Consequently, a major power consumption takes place resulting into jamming or early on breakdown of the motor in case the issue is not solved promptly.

Backlash is the mechanical play between two moving components and in the case of electromechanical actuators it is represented by the gap between the surface of the motor and user shafts. Backlash may have many causes and one of the main factors is poor lubrication. It may be minimized by design, but there must be a certain amount to allow lubrication between moving parts. If there is no lubrication or it is present in contained quantities, the rubbing of the parts creates undesirable clearances that add extra play. Another cause of backlash is poor gear manufacturing, especially if the gear teeth are cut too deep.

Too much backlash in a system can lead to several many failures. A simple failure can cause the destruction or the loss of manufactured parts, which is a waste of material and time. This is because the machine is no longer as precise as it used to be due to the loss of position when moving in either direction. This loss leads to the

machine moving and falling below its target position and achieving undesirable results during execution.

Backlash is implemented in both the reference and the monitor model in Simulink by selecting the Backlash block from the Discontinuities library and it is positioned downstream the motor position sensors. As a consequence, only the outer position loop will be affected by this fault and in the meanwhile the electrical control law will remain unchanged as the rotor angular position sensors reports the correct measured value.

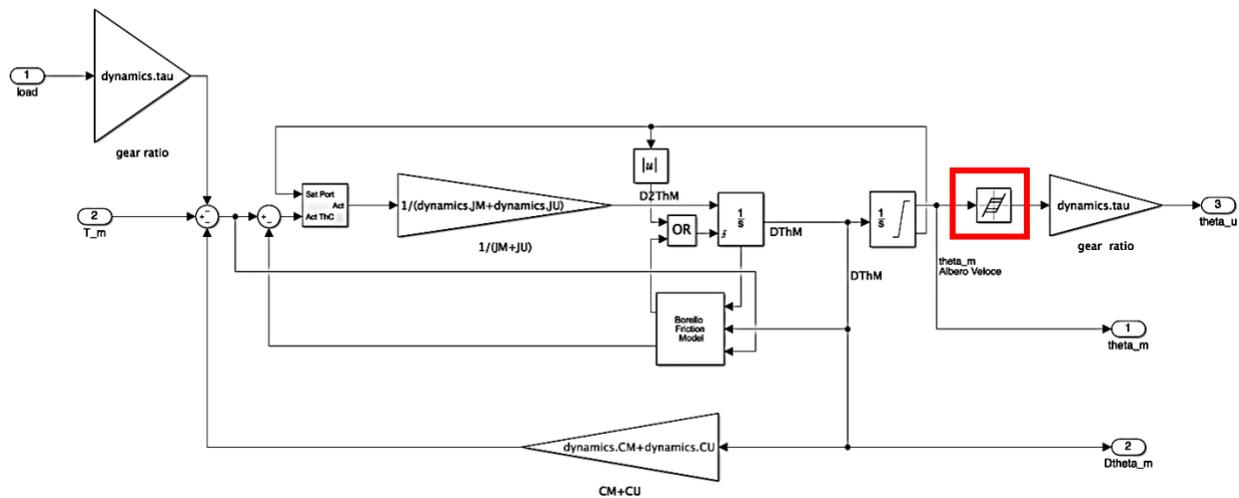


Figure 4. 2 – The backlash fault implemented in the reference model in Simulink

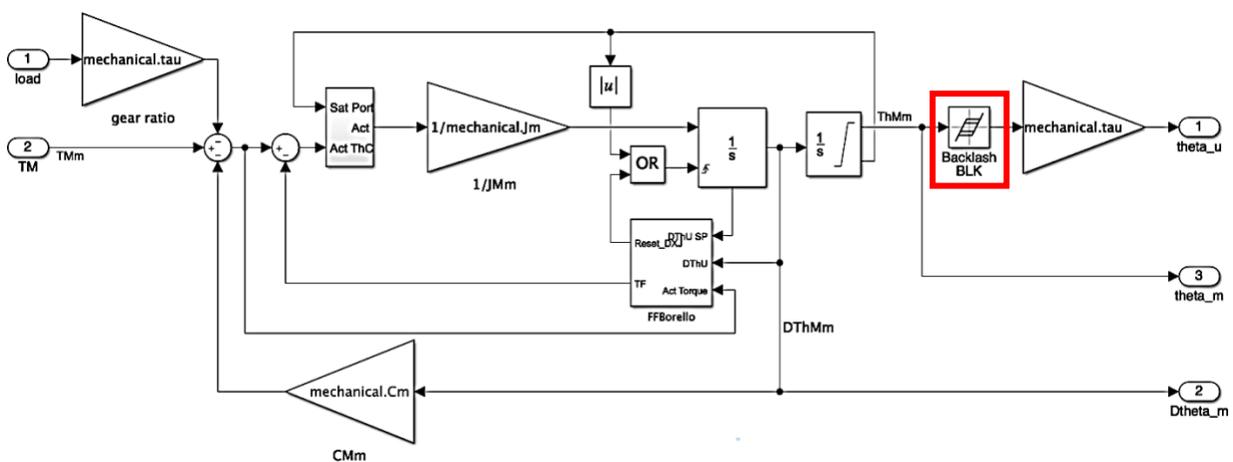


Figure 4. 3 – The backlash fault implemented in the monitor model in Simulink

## 4.4 Short circuit

Moving on to the electrical failures, it is possible to have either progressive coil short circuits or rotor static eccentricity generated by bearing wear. Short circuits often begin between a pair of coils belonging to the same phase. In short-circuited coils, the voltage doesn't vary while the resistance is reduced, resulting in an increase of the currents as well as the generation of a localized temperature in the conductor, which promotes the propagation of the failure to adjacent coils. This situation can cause serious damages if not detected promptly.

Short circuit can manifest itself in different modes:

- Between two windings that are part of the same phase (coil-coil);
- Between two windings that are part of different phases (phase-phase);
- Between a winding and the iron stator core (phase-ground).

In a complete system breakdown scenario, the short circuit fault starts as a coil-coil type; then, the damage caused by the located heating produced will propagate to a phase-phase or a phase-ground short circuit. Accordingly, prognostic analysis mainly focuses on the early identification of a coil-coil short circuit.

In the reference model, the short circuit fault is implemented in the highly detailed modeling of the three phases of the motor stator. The partial coil-coil short circuit can lead to a decrease of the phase resistance as well as the inductance and it can cause a reduction of the counter electromotive and torque gains, which can be initially approximated as follows:

$$K_e = G_M = \frac{\partial \phi}{\partial \theta_m} = NA \frac{\partial \left( \int_A \int B \cdot \bar{n} dS \right)}{\partial \theta_m} \quad (4.2)$$

where  $N$  is the number of windings,  $A$  the winding area and  $B$  the magnetic flux density of the rotor.

Thus, the percentage of short circuit windings of the  $i$ -th phase ( $N_i$ ) can affect the following computations:

$$K_{ei} = K_e \cdot N_i \quad (4.3)$$

$$R_{ij} = \frac{R_s}{2 \cdot (N_i + N_j)} \quad (4.4)$$

$$L_{ij} = \frac{L_s}{2 \cdot (N_i^2 + N_j^2)} \quad (4.5)$$

$$R_i = \frac{R_s}{2 \cdot N_i} \quad (4.6)$$

$$L_i = \frac{L_s}{2 \cdot N_i^2} \quad (4.7)$$

where  $K_{ei}$  is the counter electromotive coefficient used to compute the counter electromotive force,  $R_s$  and  $L_s$  are respectively the phase-phase resistance and inductance of the motor without faults,  $R_{ij}$  and  $L_{ij}$  are respectively the phase-phase resistance and inductance of the faulty motor, and  $R_i$  and  $L_i$  the coil-coil resistance and inductance of the motor when taking into consideration faults.

In nominal conditions,

$$\begin{cases} N_A = N_B = N_C = 1 & (4.8) \\ R_i = \frac{R_s}{2} & (4.9) \\ L_i = \frac{L_s}{2} & (4.10) \end{cases}$$

In the monitor model, the short circuit fault is implemented differently since the model simulates the behavior of a single-phase equivalent motor whereas in the case of the reference model the three currents are assessed for a single-phase equivalent actuator. Therefore, the percentage of short circuit can be evaluated as:

$$N_{equiv} = \frac{N_A + N_B + N_C}{3} \quad (4.11)$$

Consequently, the electrical parameters are computed as:

$$R_{equiv} = N_{equiv} \cdot R_{equivNC} \quad (4.12)$$

$$L_{equiv} = N_{equiv}^2 \cdot L_{equivNC} \quad (4.13)$$

$$k_{fcem} = N_{equiv} \cdot k_{fcemNC} \quad (4.14)$$

$$G_{Mequiv} = N_{equiv} \cdot G_{MequivNC} \quad (4.15)$$

where ‘NC’ stands for normal conditions.

The rest of the procedure has already been discussed in the paragraph 3.1.2.1.

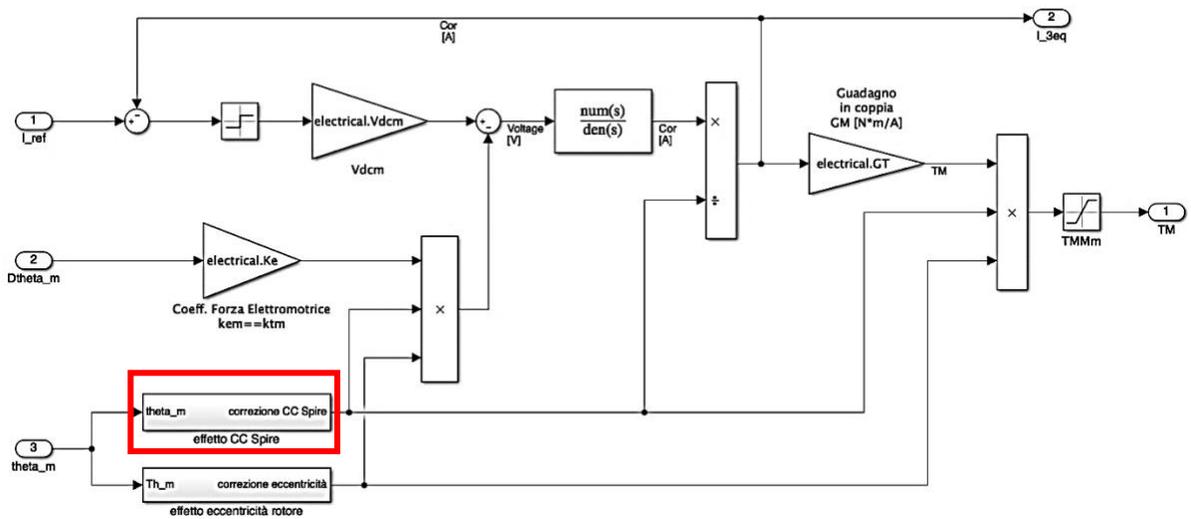


Figure 4. 4 – The short circuit fault implemented in the monitor model

## 4.5 Static rotor eccentricity

The rotor static eccentricity  $\zeta$ , instead, consists in the misalignment between the rotor rotation axis and the stator axis symmetry. This misalignment is mainly due to tolerances and imperfections that occur during the construction of the motor or to a gradual increase in rotor shaft bearing wear. When this error occurs, the motor with more than one polar pair will generate a periodically variable magnetic flux as the air gap changes during its  $360^\circ$  rotation.

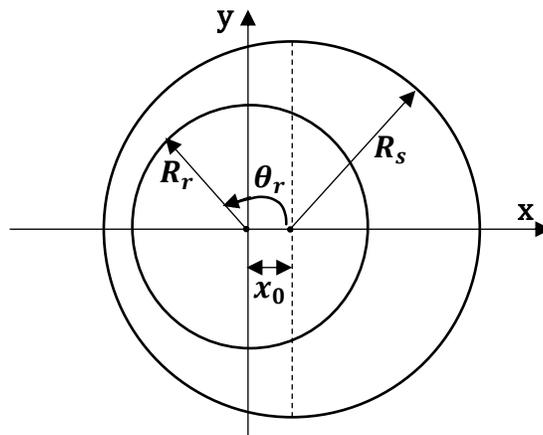


Figure 4. 5 – Scheme of the air gap in the presence of rotor eccentricity

With static eccentricity, the air gap changes during the rotor rotation depending on the rotor position  $\theta_r$ :

$$g'(\theta_r) = g_0 + x_0 \cos(\theta_r)$$

where  $g_0 = R_s - R_r$  is the clearance between the stator and rotor with no misalignment and  $x_0 \cos(\theta_r)$  the variation of the air gap.

Regarding motor performances, the torque provided is less than under nominal conditions, while spectral analysis shows the presence of subharmonics, which increase with higher eccentricities. The static eccentricity of the rotor and the partial short-circuit effects of the stator coil have been modeled using a simplified numerical algorithm. Since both failures change the magnetic coupling between the stator and rotor, they can actually be modeled by changing the values and angle modulations of the back-EMF coefficients:

$$K_{eA} = K_{eA} \cdot C_{eA} \cdot (1 + \zeta \cos(\theta_r))$$

where  $\zeta$  is the rotor static eccentricity. The constants  $K_{eA}$ ,  $K_{eB}$  and  $K_{eC}$  are employed to compute the corresponding counter-electromotive forces and to assess the mechanical couples,  $C_{eA}$ ,  $C_{eB}$  and  $C_{eC}$ , generated by the three motor phases.

In the reference model, the rotor eccentricity is implemented by computing the coefficients of the counter electromotive force as a function of the rotor eccentricity and angular position in the *PMSM electromagnetic model* block. In this way, it enables the exact reproduction of the system behavior with the help of an analytical representation of magnetic interactions between rotor and stator.

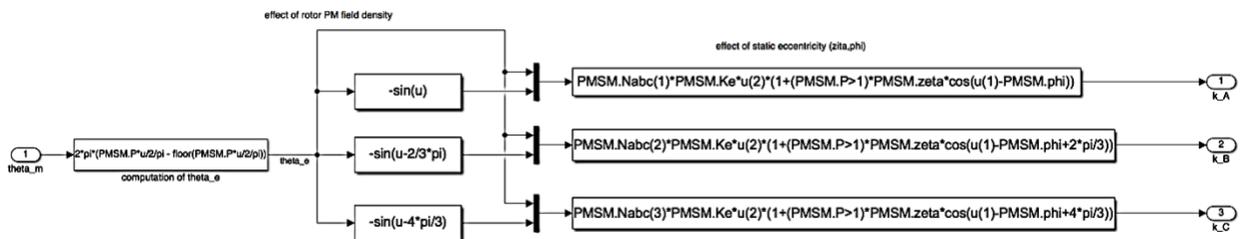


Figure 4. 6 – Implementation of the rotor eccentricity fault in the reference model

The implementation of the rotor eccentricity in the monitor model has already been discussed in the paragraph 3.1.2.2.

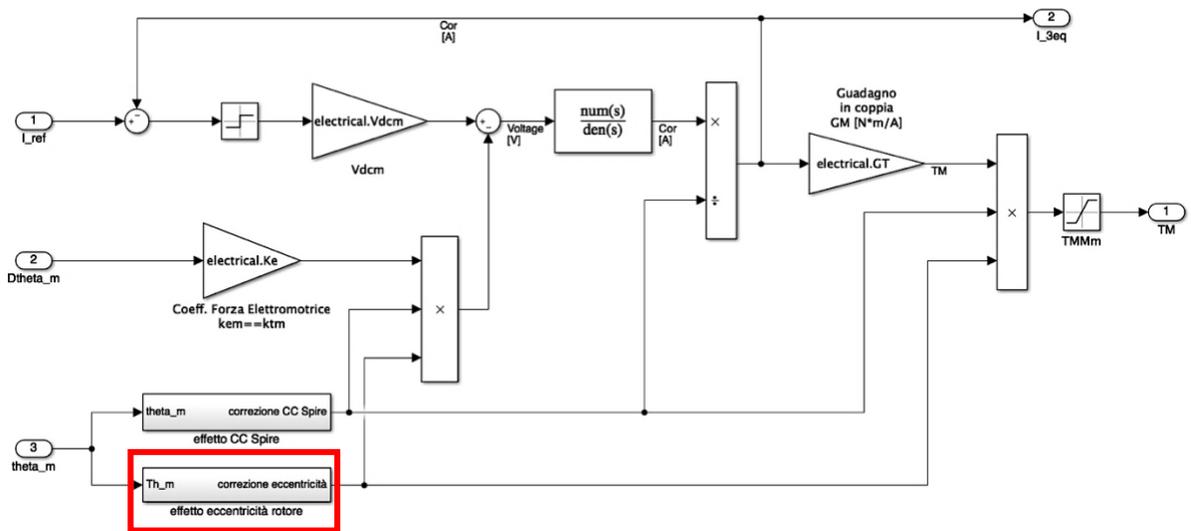


Figure 4. 7 – Implementation of the rotor eccentricity fault in the monitor model

## 4.6 Proportional gain

The last fault to be considered is proportional gain  $G_{prop}$  which is part of the *Control Electronics* block. Due to the abrupt nature of most electronic faults, a prognostic analysis cannot be applied as it is not possible to evaluate their expected RUL. In order to prevent a catastrophic system failure, the only solution is to employ redundancies so the damaged part can be easily identified, isolated and substituted. The proportional gain can affect the actuator dynamic response and by varying its value it is possible to see the change of the dynamic response of the models.

The implementation of this fault in both the reference and the monitor model is relatively easy as it only requires the modification of the input parameter: it is multiplied by a value in the range 0.5-1.5 so the nominal value equal to  $10^5 [s^{-1}]$  can assume a value between  $0.5 \cdot 10^5 [s^{-1}]$  and  $1.5 \cdot 10^5 [s^{-1}]$ .



## 5. Optimization Process

### 5.1 Optimization algorithms

Optimization algorithms are numerical methods which are carried out iteratively by comparing different solutions until a satisfactory solution is found. They can be divided in two main branches:

- *Deterministic algorithms*, which employ certain rules to move from one solution to another and they rely on precise mathematical models;
- *Stochastic algorithms*, which are in nature with probabilistic transition rules and use a random search of the best solution.

Although a probabilistic method can provide lower computational cost and effort when the number of variables considered is high, deterministic algorithms are still better since they provide the same output given a particular input, so their convergence relies greatly on the initial condition and this is important for searching local minima.

In this work, optimization algorithms may be employed to have the monitor model generate parameters as close as possible to the one produced by the reference model. A predetermined fault is implemented in the reference model before it is launched. Thereby, it is possible with the simulation to receive all signals of position, speed, torque and current in a faulty state as an output. After that, the simulation of the monitor model is executed: it is necessary to approximate the parameters of the reference as closely as possible in order to detect the introduced fault. To attain this goal, the monitor model must run many times with different parameters and select those that come closest to the reference model: these variations of the main features of the monitor model in each simulation are made by suitable optimization algorithms. The Genetic Algorithm (GA) has been chosen as the optimization algorithm for this work.

## 5.2 Genetic algorithm

A genetic algorithm is a search heuristic inspired by the theory of natural evolution of Charles Darwin. This algorithm mirrors the process of natural selection, in which the most suitable individuals are chosen for reproduction in order to produce offspring of the next generation.

The process of natural selection begins with choosing the best individuals from a population by means of an objective function. Such individuals produce offspring that inherit the characteristics of the parents and are added to the next generation. When parents are in better fitness, their offspring will be better than them and will have a better chance of survival. The process keeps repeating itself and in the end a generation with the best people is acquired.

Genetic algorithms can be used for a search problem by considering a number of solutions and selecting the best out of them. The genetic algorithm consists of five phases:

- *Initial population*: the process starts with a group of individuals composing a population which is fixed. Each one of them is a solution to the problem they want to solve. One individual is characterized by the **genes**, which consist of a number of parameters (variables). These genes are put together in a string to form a **chromosome**, which is the solution.
- *Fitness function*: it establishes how fit an individual is in terms of ability to compete with others. It gives each individual a **fitness score** that will be the factor determining the probability an individual will be selected for reproduction.
- *Selection*: here the most suitable individuals are picked, and they are allowed to pass their genes on to the next generation. The **parents** are selected based on their fitness score. When new generations are formed, the least fit individuals die and make room for new offspring.

Many selection processes employ a “roulette wheel” mechanism to choose in a probabilistic manner the individuals based on some measures of their performance. A real-valued interval, *sum*, is computed either as the sum of the expected selection probabilities of the individual or the sum of its fitness score over all other individuals of the population. Individuals are then mapped one to one in connected intervals in the range  $[0, sum]$ . The size of each individual interval corresponds to the fitness score of the associated individual. An example applied to five individuals is shown in *Figure 5.2*.

- *Crossover*: it is the most important phase in a generic algorithm. A **crossover point** is randomly selected from the genes of each pair of parents to be mated;

the genes from the parents are exchanged until the crossover point is reached, creating new offspring which will added to the population.

- *Mutation*: there is a low chance that some of the genes of the newly formed offspring may be mutated. This implies that some of the bits in the bit sequence may be flipped. Mutation occurs in order to retain diversity within the population and avoid having premature convergence.

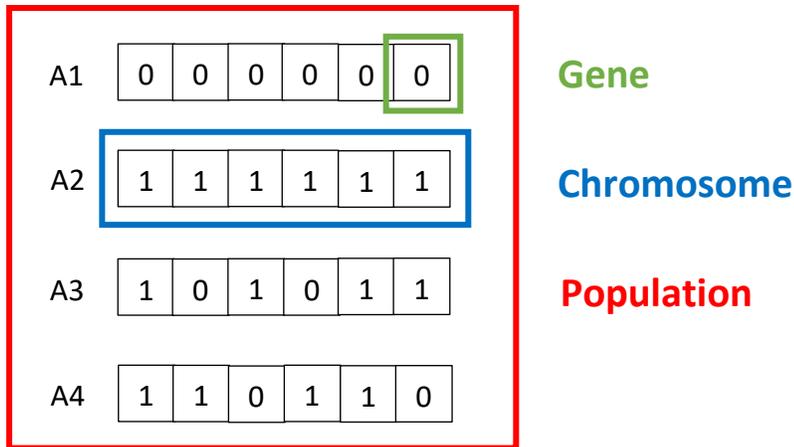


Figure 5. 1 – Population, chromosomes and genes of a genetic algorithm

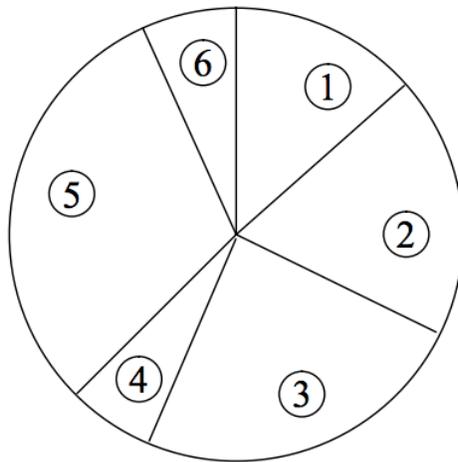


Figure 5. 2 – Roulette wheel selection

The sequence of phases is repeated to generate individuals in each new generation who are better than the previous one.

The algorithm ends when the population has converged as no offspring that differs from the previous generation has been created. It is then that a set of solution are said to have been provided to the problem.

## 5.2 Optimization toolbox

The MATLAB Optimizer consists of a graphical user interface that can be used to solve an optimization problem without having to write a script to invoke the Toolbox functions. In fact, it offers a user-friendly, fast and uncomplicated way of setting the desired optimization algorithm.

The tool can be started either by typing 'optimtool' on the command line or from the MATLAB Apps tab. Consequently, the user interface will appear. With the tool, the user can then select a solver from the Optimization Toolbox, which ranges from gradient-based algorithms to non-deterministic methods such as simulated annealing, genetic algorithms or pattern searches. The user interface depends on the solver selection, based on the options available for the selected algorithm. Only the genetic algorithm has been considered for this work by setting the solver to “ga”.

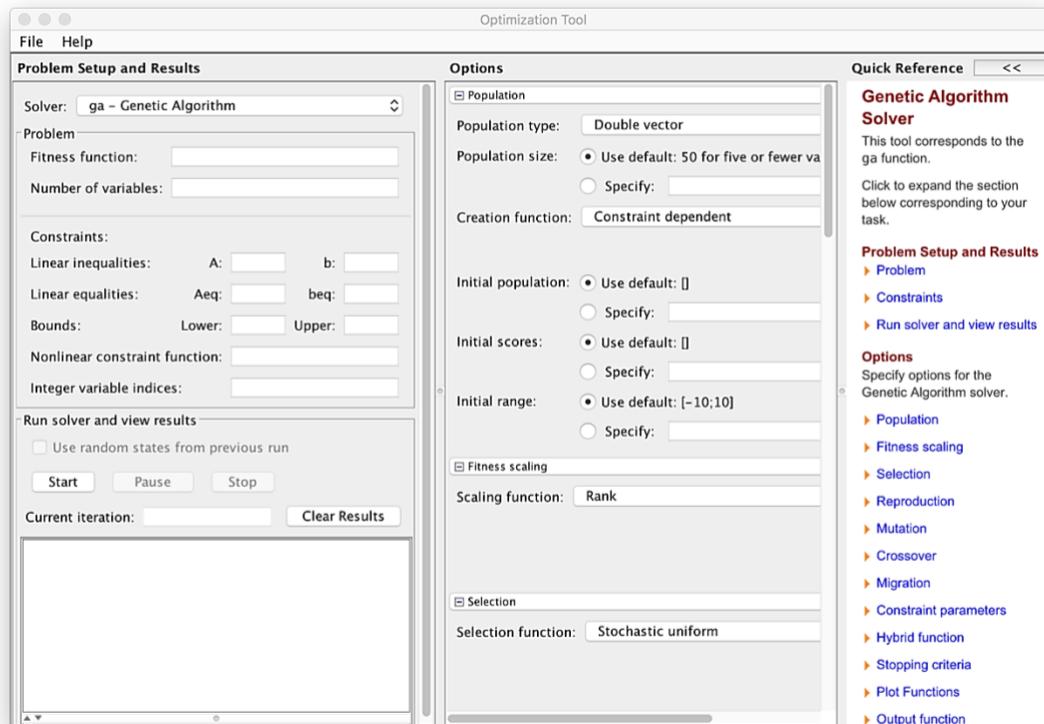


Figure 5. 3 – The Optimtool GUI for the genetic algorithm

The main setup options are located on the left side of the GUI. Firstly, there is the *Problem Setup and Results* section, which defines the problem and starts optimization. The *Problem* subsection contains the *Fitness Function* and *Number of Variables* fields, which are both mandatory: the objective function must be inserted as a function handle in the form '@objfun', where 'objfun.m' is the matlab file that contains the fitness function that must receive a vector as input and return a scalar. The dimension of the input vector must be specified in the *Number of Variables* field. In the *Constraints* subsection, *linear equalities* constraints can be specified in the form  $[A]\{x\} = \{b\}$ , *linear inequalities* in the form  $[A]\{x\} \leq \{b\}$ , *bounds* constraints, or *nonlinear constraints* defined by a user-defined function. At least one bound constraint is required to start optimization.

In the subsection *Run Solver and View Results* one can start, stop and pause the optimization. Details about the optimization are displayed in the *Status and Results* area, while the resolution to the problem is displayed in the *Final point* area.

In the middle area of the Optimtool window, *Options* for optimization can be set which are useful for improving the robustness of the algorithm, the computational efficiency as well as the convergence. The subsection *Population* enables the selection of the population type between *double vector*, *bit string* or *custom*. *Population size* indicates the number of individuals who make up each generation. The *Creation function* is used to generate people of the original population: it can be *Constraint dependent* (i.e. *uniform* if there are no restrictions and otherwise *feasible population*) or *custom*. It is also possible to set the *initial population*, *initial scores*, or *initial range* by specifying each individual, rating, or the upper and lower limits for the first generation, respectively.

*Fitness scaling* allows a scaling function to be defined, i.e., an algorithm for converting raw fitness scores returned by the fitness function into values fit for the selection function. The *Rank* option scales the ratings of each individual depending on their position in the sorted ratings: the fittest person receives rank 1, the second strongest rank 2, and so on, which eliminates the effect of scattering the raw scores. *Proportional scaling* sets everyone's expectations in proportion to their score: this is the simplest option, but can have a weakness if the raw scores are not in a favorable range. The *Top* option picks the best  $n$  individuals for reproduction with the same probability, where  $n$  is the number given in the *Quantity* field, while the others have no reproduction probability. *Shift linear* scales the raw values in such a way that the expectation of the fitter individual is equal to the constant specified as the *Maximum survival rate*, multiplied by the average score, whereas *Custom* allows the user to define its own scaling function in the form of '@Scalefcn'.

The *selection function* selects parents for the next generation based on their scaled fitness values. By specifying the *stochastic uniform*, the parents are selected at random, with the probability being proportional to their expectations.

The *Remainder* deterministically assigns the parents from the integer part of each individual's scaled value and then uses the roulette selection for the remaining fraction. *Uniform* randomly selects parents from an even distribution using the expectations and number of parents: the user can use it to test the genetic algorithm. *Roulette* simulates a roulette wheel, with the area of each segment proportional to its expectation. The algorithm then uses a random number to select one of the sections with a probability that corresponds to its area. *Tournament* randomly selects a subgroup of individuals whose size is reported as the *Tournament size* and then picks the best person from the subgroup as the parent. Finally, with *Custom* the user can specify a selection function as "@SelectFcn".

The *Reproduction* options establish how the algorithm generates individuals for the next generation. The *Elite count* indicates the number of individuals who survived from the previous generation, while the *Crossover fraction* establishes the next generation fraction produced by crossover.

The *Mutation* function carries out small random variations to the individuals in the population that offer genetic diversity and allow the genetic algorithm to search a wider space. This parameter can be selected under *Constraint dependent*, which is *Gaussian* if there are no constraints or *Adaptive feasible* otherwise. *Uniform*, where a fraction of an individual's vector entry is replaced with a probability equal to the mutation rate with a random number uniformly selected from the range for that entry or *Custom* that allows the user to write a mutation function that is called '@MutateFcn'.

The *Crossover function* brings two parents together to form one child. *Scattered* crossover randomly selects each gene from one of the parents and assigns it to the child. *Intermediate* Crossover generates a child as a weighted average between the parents with random weights. *Heuristic* crossover randomly creates a child in the parent row who is near the best match parent away from the worst match parent. *Arithmetic* crossover creates a child as a random arithmetic average of the parents, while the *Custom* option enables a custom function to be entered as usual.

If more than one subpopulation is accounted for in the population subsection by entering a vector with a dimension greater than one as the population size, it is feasible to implement the *migration* option which simulates the movement of people between subpopulations. The *Direction* indicates whether people can migrate *Forward* or in *Both* directions. The *Fraction* establishes the number of individuals who migrate, expressed as a fraction of the smallest population, while the *Interval* controls how many generations cross between migrations.

In the *Constraint Parameters* subsection, one can configure the *initial penalty* and *penalty factor* used by the nonlinear constraint solver algorithm.

The *Hybrid function* option enables further minimization after completing the genetic algorithm. The options available are *fminsearch*, *patternearch*, *fminunc*, and *fmincon*.

The *Stopping criteria* settings establish why the algorithm stops. It is possible to limit the maximum number of generations, the maximum computation time, the maximum number of stable generations (over which changes in the fitness function are less than the functional tolerance) or the tolerance for the maximum nonlinear constraint violation .

From the *Plot Functions* subsection, various aspects of the genetic algorithm can be plotted during its execution. Each quantity is shown on a separate axis in the same figure. The choices available are the best fitness value for each generation, entries of the best individual, the average distance between individuals at each iteration, the expected number of children compared to the raw values, the genealogy number of people (only available at the end of optimization) maximum, minimum, and mean fitness, score of individuals of each generation, parents selected for reproduction, stopping criteria, maximum nonlinear constraint violation, or a custom function specified as '@PlotFcn'.

In addition, it is possible to specify an *output function* that the genetic algorithm calls every generation and the amount of diagnostic information that is displayed in the command window.

As a final note, the *User Function Evaluation* option establishes how fitness and constraint functions are rated. With *in serial* selection, the functions are assessed separately for each individual; The *vectorized* evaluation calculates the fitness and constraints for all individuals in a population in a function call, but the functions must be set in such a way that they accept vectorized input. Finally, the *in parallel* option allows the fitness and limitations of a group of processors to be assessed, which can significantly speed up optimization when assessing a computationally intensive function on a multicore processor. To enable this feature, the worker cluster must be started with the 'matlabpool' command before the solver is started. This will open the specified number of MATLAB processes, which will allow the processor resources to be used to the full.

## 5.4 Choice of the fitness function

The objective function is used in order to provide an evaluation of how individuals performed in the problem. In the case of a minimization problem, the most suitable individuals have the lowest numerical value of the associated objective function. It is also known as the fitness function: it expresses how much the monitor model is comparable to the high fidelity one in a satisfactory manner. If the algorithm

executes the monitor model with a certain fault vector  $k$  at every sample time, the single-phase current is stored in the workspace of Matlab in a suitable variable. This value is then compared with the reference equivalent current, which is assessed as described at the beginning of *Chapter 2* in order to compute the error between the two signals. The comparison has been made as a first approximation using the *Least Squares* method:

$$err = \sum_t (I_{HF}(t_0) - I_{LF}(t_0))^2 \quad (5.1)$$

where  $I_{HF}$  is the reference model output current,  $I_{LF}$  is the monitor model output current and  $t_0$  is a generic time instant of the simulation time. The two current trends are compared in the equation 5.1, whereby the minimum of the signal described is determined. This means that the error must stay as close to zero as possible. On closer inspection, the equation 5.1 turned out to be not quite accurate as it resulted in very small differences between the behavior of the reference and monitor models not being detected, a situation which could lead to a phase shift in the rotor angular position measured between the two models. This inaccuracy is notably exacerbated with abrupt changes in the commanded current.

A solution to this problem could be implemented by employing the *Total Least Squares* method: it takes into account not only the dependent variable, but also the independent variable. It evaluates the accuracy of the fit in different ways from the *Least squares* method as it minimizes the sum of the squared vertical distances from the acquired data to the fit curve.

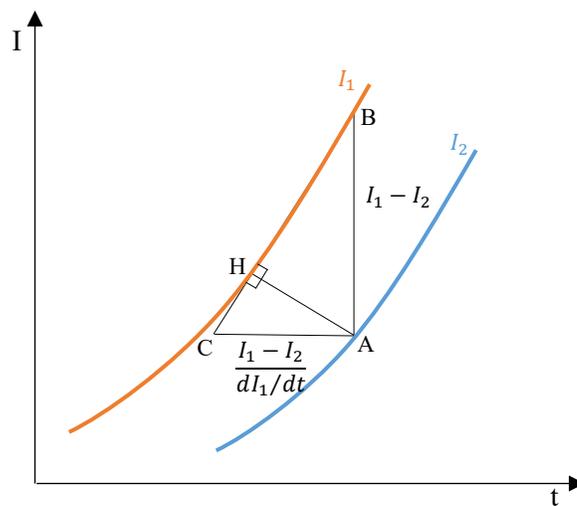


Figure 5. 4 – Representation of the Total Least Squares method

In our case, the acquired data is the output of the reference model, and the approximation is performed by the optimization algorithm applied to the monitor model. When using the Total Least Squares method, the computation of the error is only possible when assessing the normal distance between a data trend and a fit curve as shown in *Figure 5.4*.

Considering  $I_1 = I_{HF}$  and  $I_2 = I_{LF}$  and assuming that the distance  $\overline{BC}$  is small enough to match the segment with the curve, the length of the segment  $\overline{AC}$  would be:

$$\overline{AC} = \frac{I_1 - I_2}{dI_1/dt} \quad (5.2)$$

Being  $\overline{AH}$  the height right-angled triangle, one can say that for every time instant:

$$\overline{AH} = \frac{\overline{AB} \cdot \overline{AC}}{\overline{BC}} = \frac{(I_1 - I_2) \cdot \frac{I_1 - I_2}{dI_1/dt}}{\sqrt{(I_1 - I_2)^2 + \left(\frac{I_1 - I_2}{dI_1/dt}\right)^2}} = \frac{(I_1 - I_2)}{\sqrt{\frac{dI_1^2}{dt} + 1}} \quad (5.3)$$

Thus, the error with the Total Least Squares method can be computed as:

$$err = \sum_t \frac{(I_1(t_0) - I_2(t_0))^2}{\sqrt{\frac{dI_1(t_0)^2}{dt} + 1}} = \sum_t \frac{(I_{HF}(t_0) - I_{LF}(t_0))^2}{\sqrt{\frac{dI_{HF}(t_0)^2}{dt} + 1}} \quad (5.4)$$

The error is finally multiplied by the sample time to avoid its dependence on the compiling time.

A major concern resides on the unit of measure when defining the error related to the use of Pythagoras theorem to evaluate the sum of the two values with different units of measure. As a matter of fact, the nominator of equation 5.2 is a distance between values with the same unit of measurement, whereas the derivative is introduced at the denominator. In order to solve this issue, some solutions can be applied by either replacing the normal distance with the horizontal and vertical distance remainders or normalizing the variables with the precision measurement analysis. Both solutions have some drawbacks: the first one undervalues the error when the derivative of the curve is too small, whereas with the second it is not easy to acquire the precision measurement as the error must be equal to zero from time to time in nominal conditions. It has been decided to replace the derivative at the

denominator with its *Root Mean Square* value in normal conditions so the current can attain a unitary average derivative.

## 5.5 Normalization of the fitness function arguments

Fault parameters are introduced to both the reference and monitor models in the form of an eight-elements normalized vector  $k$ . These elements represent the arguments of the fitness function described in the previous paragraph. Because genetic algorithms are found to provide a faster convergence with normalized parameters, each argument of the fitness function has been made to vary between 0 and 1 by performing a linear interpolation based on the minimum and maximum values for each fault:

- $k(1) \in [0,1]$  refers to the normalized friction fault:  $k(1) = 0$  in normal conditions, whereas  $k(1) = 1$  when the value is three times the one in normal conditions.
- $k(2) \in [0,1]$  refers to the normalized backlash fault:  $k(2) = 0$  in normal conditions, whereas  $k(2) = 1$  when the value is one hundred times the one in nominal conditions.
- $k(3), k(4), k(5) \in [0,1]$  represent respectively the normalized short circuit of phases A, B and C:  $k(3) = 0$  when phase A is fully functional, whereas  $k(3) = 1$  when there's a complete short circuit for the same phase. In order to avoid any divergence through the simulation, whenever two of the three parameters reach simultaneously the maximum value (1), they are immediately set to 0.99 because otherwise two fully short-circuited phases will make the monitor model current diverge to infinite. This condition may lead to the total breakdown of the motor.
- $k(6), k(7) \in [0,1]$  represent the eccentricity fault in terms of amplitude and phase.  $k(6)$  is the rotor eccentricity amplitude,  $k(6) = 0$  when the rotor eccentricity is null whereas  $k(6) = 1$  when it is equal to 1.  $k(7)$  is the rotor eccentricity phase, which is the direction of the minimum air gap,  $k(7) = 0$  when the phase is equal to  $-\pi$  whereas  $k(7) = 1$  when it is equal to  $\pi$ . Due to the fact that the eccentricity phase can assume any value between  $-\pi$  and  $\pi$  when the magnitude is null, it can to be suitably managed during the assessment of the function error.
- $k(8) \in [0,1]$  is the proportional gain fault.  $k(8) = 0$  when only the 50% of the nominal value is considered and  $k(8) = 1$  when the percentage is increased to 150%. Thus, in nominal conditions  $k(8) = 0.5$ .

Once normalized, the fault vector is:

$$k = [0, 0, 0, 0, 0, 0, 0.5, 0.5] \quad (5.5)$$

During the execution of the optimization algorithm, the fault parameters of the reference model are varied depending on whether single fault or multiple fault optimization is chosen. By comparing the current trends of the two models, the algorithm generates suitable values of the fault parameters in the monitor model.

## 5.6 Parallelization

Parallelization is the choice of whether the computational effort should be divided between different cores or not. Not parallelizing means that the simulation is executed in serial mode so that the algorithm only calls up the fitness function for one individual at a time. The use of the parallel calculation means that the algorithm calls simultaneously several individuals, based on the number of core processors, to assess their suitability with the goal. A MacBook Pro has been employed to carry out the simulation, so two core processors were available for the execution in parallel mode.

## 6. Model optimization

### 6.1 Optimization parameters

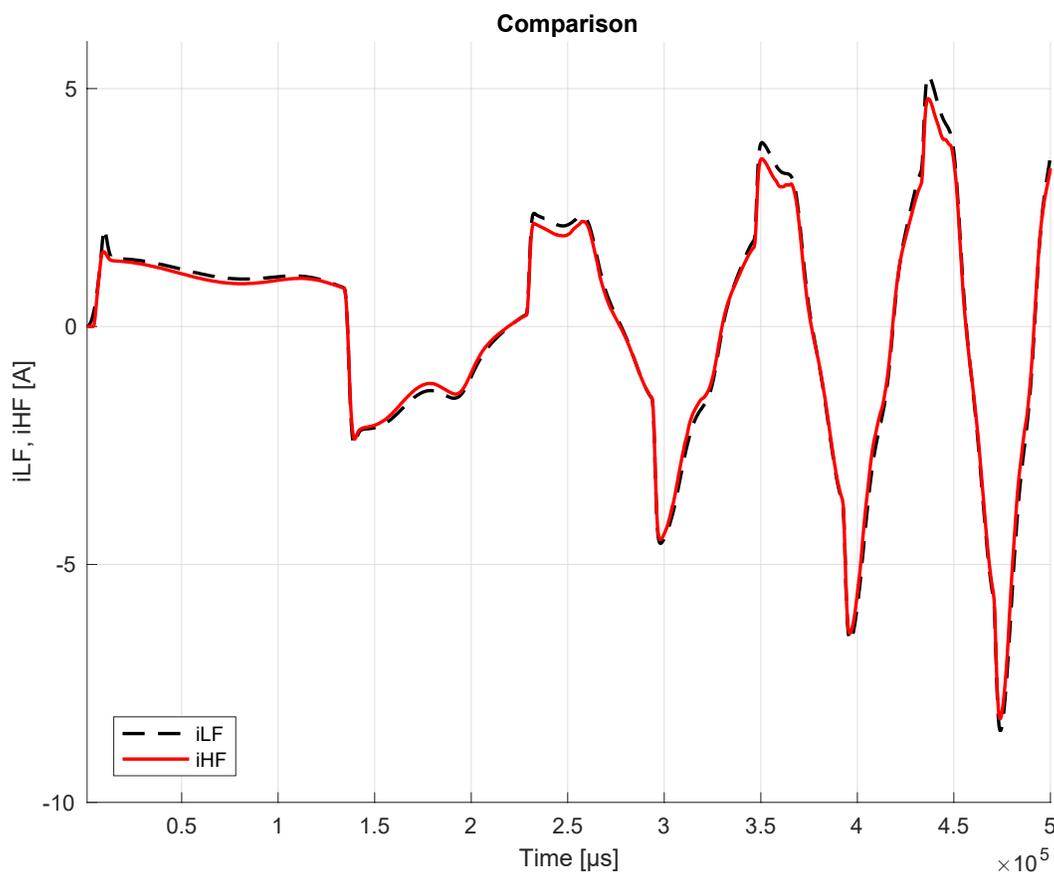
In order to acquire reliable and precise data from the simulation, the right optimization parameters have to be chosen. In particular, the main parameters taken into account are:

- *Population size*: it represents the number of individuals of each generation. Increasing this parameter can provide better accuracy and reliable data, but the computation time is no longer acceptable. After several calibration attempts, a population size of 20 has been chosen.
- *Number of generations*: it represents the maximum number iterations. In each generation, the best individuals are selected as a temporary solution to the problem: the algorithm stops when one of these solutions is found to be the best out of all. Increasing the number of generations has the same effect as in the case of the population size: it increases the computational time by a lot even though it gives better solutions. Given that the number of variables (faults) considered is 8 and the number of generations is obtained by multiplying it by 100, it should be equal to 800. However, after several calibration attempts, it has been found that a number of generations equal to 200 can give acceptable results with lower computational time.
- *Function tolerance*: it represents the average relative variation in the best score of the fitness function. By setting this number, the algorithm halts the iteration in case the relative variation between the best value of one generation and the one of the next generation is lower than the set value. The value set by default is  $10^{-6}$  and it has been changed to  $10^{-9}$  for this simulation.

- *Parallelization*: as already explained in *paragraph 5.6*, two core processors could be employed reducing significantly the completion time of the optimization process.

The user can vary any of these parameters taking into account that the genetic algorithm will halt as soon as one of these conditions is met.

## 6.2 Model calibration in normal conditions



*Figure 6. 1 – Comparison of the output currents before calibration*

In both signals, the amplitude rises as the command frequency increases, even though the magnitude of the latter does not vary during the whole actuation. This is

due to the inertial forces which increase alongside the frequency and when faster direction inversions are needed, the system produces larger acceleration with

consequent larger speeds and torque which are required when imposing the new direction. The current is directly proportional to the motor torque, as a consequence it increases and decreases at the same rate. At the initial stage of the response, the signal line is mainly horizontal, and it becomes steeper as the actuation speed increases together with the position error, until they come vertical. This new trend represents the condition in which the static friction has to be overcome: once the direction is changed, the system reacts with its own dynamics and the actuator has to stop by inverting the actual position speed. Consequently, as the static friction torque is greater than the dynamic one, a sudden increase of current is required in order to initialize again the system. This explains the jumps in the current trend.

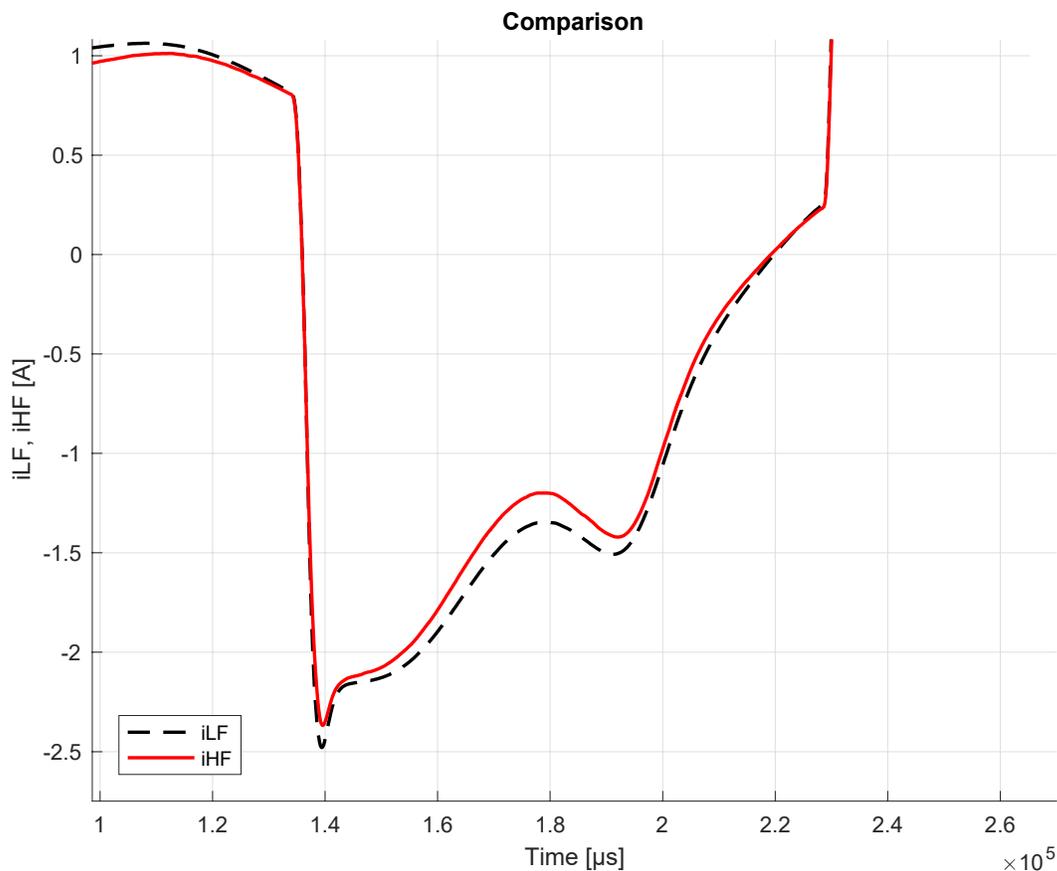


Figure 6. 2 – Close-up of the comparison of the output currents before calibration

At the first stage of actuation, the signal has initially a linear behavior and then one can observe an overshoot typical of second order systems, which not the case for this study.

When considering the models in nominal conditions, one can observe how the current trends are comparable, they differ for a small offset in terms of magnitude that can be regarded as an ‘error’ which can be compensated through the Genetic Algorithm included in Matlab Optimization Toolbox. This error is due to a slightly lower stator-rotor electromagnetic coupling in the monitor model and this results in a higher equivalent current when the same operating conditions in terms of torque and speed are considered. Another consequence of this configuration is that the maximum zero-load actuating speed of the monitor model results moderately higher and leads to the accumulation of a position offset when working near saturation. Hence, a different response can be observed from position and velocity time histories as the faster model will reach first the commanded position. When a fault in the electrical model is taken into account, the angular position phase displacement gives current ripples signal, hindering the correct fault identification.

In order to minimize the position error in nominal condition, the BEMF coefficient  $k_e$  as well as the motor gain torque  $G_T$  of the monitor model have to be calibrated by applying the genetic algorithm to find the values for which the minimum error is obtained. After that, the objective function that the genetic algorithm will try to minimize is chosen as the Mean Squared Error (MSE) computed as follows:

$$MSE = \sum_{i=1} (I_{HF} - I_{LF})^2 [A^2] \quad (6.1)$$

where  $I_{HF}$  and  $I_{LF}$  are, respectively, the high fidelity and the low fidelity current components at each integration step, whereas  $n$  is the sampling number. Before the calibration, the MSE has a value equal to:

$$MSE_{init} = 0,1292 [A^2] \quad (6.2)$$

With the acquired values of  $k_e$  and  $G_T$ , the following result is obtained:

$$MSE_{cal} = 0,0644 [A^2] \quad (6.3)$$

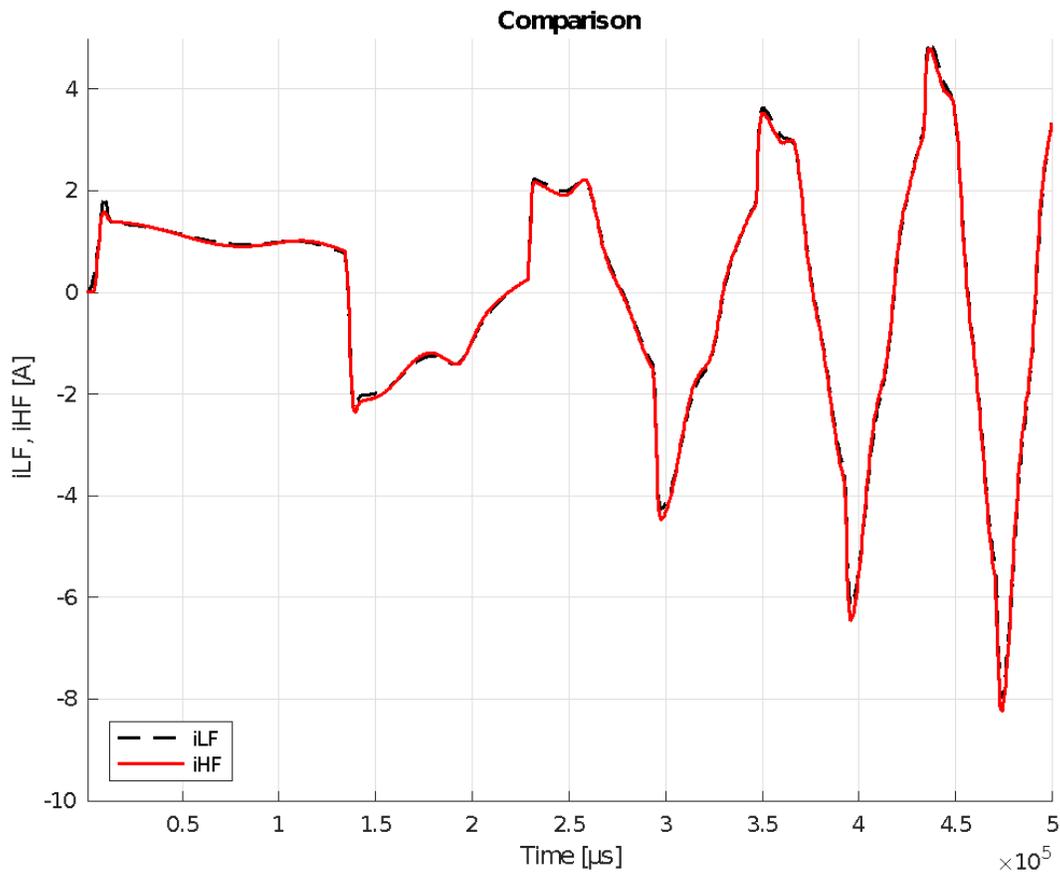


Figure 6. 3 – Comparison of the two output currents after calibration

	KE	GT	MSE
NOMINAL CONDITIONS	0.0752	0.0752	0.1292
AFTER CALIBRATION	0.085897	0.080072	0.0644

Table 6. 1 – Results of calibration at nominal conditions

### 6.3 Single fault isolation

As previously stated, the reference current is produced by feeding a fault vector  $k$  into the reference model. The fault detection is then performed by the genetic algorithm which, through the monitor model, attempts to approximate as closely as possible the values of each fault coefficient corresponding to the equivalent current trend. The acquired results are shown in the tables of the following paragraphs for each fault. The percentage of error is computed by using the following relation:

$$\%err = 100 \cdot \sqrt{\sum_{i=1}^6 (k_i - \bar{k}_i)^2 + \bar{k}_6 \cdot (k_7 - \bar{k}_7)^2 + (k_8 - \bar{k}_8)^2} \quad (6.4)$$

where  $k_i$  is the value of the  $i$ -th fault parameter of the monitor model and  $\bar{k}_i$  is the corresponding  $i$ -th fault parameter of the reference model.

The relation 6.4 is exactly the same as a mean square error with a minor dissimilarity in the definition of the eccentricity phase error  $k_7$ , which can assume any value when the eccentricity coefficient  $\zeta$  is null.

Three distinct objective functions are examined for each fault: the low fault detection (with  $\bar{k}_i \leq 0.25$ ), medium fault detection (with  $0.25 < \bar{k}_i < 0.7$ ) and high fault detection (with  $\bar{k}_i \geq 0.7$ ).

The optimization has been performed not only for the single fault application, but also for the multiple fault application.

In order to emphasize the stochastic behavior of the genetic algorithm, ten optimizations have been carried out for each case, with slightly different results being achieved each time. The execution time has an average value of 12'15", reaching a maximum of 17'42". The computation time is mainly affected by the number of iterations that are executed to achieve convergence.

### 6.3.1 Friction fault

Friction fault has been assessed by implementing a low ( $k_1 = 0.25$ ), medium ( $k_1 = 0.5$ ) and high ( $k_1 = 0.75$ ) level of damage to analyze the behavior of the system in each of these scenarios. For each case, ten optimizations have been executed and the results have been reported in the tables below.

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0.25</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.5</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.2390	0.0028	0.0013	0.0007	0.0039	0.0029	0.7492	0.4758	
	1.1%	0.28%	0.13%	0.07%	0.39%	0.29%	24.92%	2.42%	2.49%
<b>2</b>	0.2471	0.0004	0.0060	0.0022	0	0.0047	0.9652	0.4859	
	0.29%	0.04%	0.6%	0.22%	0%	0.47%	46.52%	0.41%	1.62%
<b>3</b>	0.2399	0.0013	0	0.0051	0.0049	0.0032	0.6273	0.4836	
	1.01%	0.13%	0%	0.51%	0.49%	0.32%	12.73%	1.64%	1.82%
<b>4</b>	0.2463	0.0017	0.0053	0.0048	0.0023	0.0005	0.0027	0.5105	
	0.37%	0.17%	0.53%	0.48%	0.23%	0.05%	49.73%	1.05%	1.30%
<b>5</b>	0.2506	0	0.0051	0.0015	0	0.0009	0.3653	0.4863	
	0.06%	0%	0.51%	0.15%	0%	0.09%	13.47%	1.37%	1.47%
<b>6</b>	0.2447	0.0034	0	0	0.0045	0.0039	0.4678	0.5154	
	0.53%	0.34%	0%	0%	0.45%	0.39%	3.22%	1.54%	1.69%
<b>7</b>	0.2471	0.0019	0.0038	0.0052	0.0019	0.0074	0.6330	0.4778	
	0.29%	0.19%	0.38%	0.52%	0.19%	0.74%	13.30%	2.22%	2.44%
<b>8</b>	0.2501	0.0023	0.0015	0.0082	0.0008	0.0071	0.0762	0.4756	
	0.01%	0.23%	0.15%	0.82%	0.08%	0.71%	42.38%	2.44%	2.69%
<b>9</b>	0.2478	0.0001	0	0.0097	0.0013	0.0057	0.5049	0.4893	
	0.22%	0.01%	0%	0.97%	0.13%	0.57%	0.49%	1.07%	1.56%
<b>10</b>	0.2422	0.0009	0.0024	0.0014	0.0037	0	0.9955	0.4723	
	0.78%	0.09%	0.24%	0.14%	0.37%	0%	49.55%	2.77%	2.81%

Table 6. 2 – Low friction optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0.5</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.5</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.4938	0.0102	0.0008	0.0127	0.0079	0.0107	0.7192	0.4858	
	0.62%	1.02%	0.08%	1.27%	0.79%	1.07%	21.92%	1.42%	2.54%
<b>2</b>	0.5012	0.0068	0.0038	0	0.0114	0.0139	0.7530	0.5024	
	0.12%	0.68%	0.38%	0%	1.14%	1.39%	25.30%	0.24%	1.97%
<b>3</b>	0.4841	0.0125	0	0.0084	0	0.0082	0.4084	0.4872	
	1.59%	1.25%	0%	0.84%	0%	0.82%	9.16%	1.28%	2.14%
<b>4</b>	0.4748	0.0183	0.0014	0.0140	0.0036	0	0.6334	0.4775	
	2.52%	1.83%	0.14%	1.4%	0.36%	0%	13.34%	2.25%	3.24%

<b>5</b>	0.4948	0.0179	0.0009	0	0.0006	0.0064	0.3463	0.4933	
	0.52%	1.79%	0.09%	0%	0.06%	0.64%	15.37%	0.67%	2.02%
<b>6</b>	0.4809	0.0033	0	0	0.0032	0.0008	0.1049	0.4768	
	1.91%	0.33%	0%	0%	0.32%	0.08%	39.51%	2.32%	2.37%
<b>7</b>	0.4957	0.0098	0.0034	0.0093	0.0095	0.0021	0.8051	0.4931	
	0.43%	0.98%	0.34%	0.93%	0.95%	0.21%	30.51%	0.69%	1.83%
<b>8</b>	0.4714	0.0046	0	0.0089	0.0014	0.0069	0.9035	0.4873	
	2.86%	0.46%	0%	0.89%	0.14%	0.69%	40.35%	1.27%	1.76%
<b>9</b>	0.4882	0.0075	0.0116	0.0036	0	0.0162	0.0057	0.5036	
	1.18%	0.75%	1.16%	0.36%	0%	1.62%	49.43%	0.36%	2.19%
<b>10</b>	0.5042	0.0221	0.0022	0	0.0093	0	0.5948	0.4904	
	0.42%	2.21%	0.22%	0%	0.93%	0%	9.48%	0.96%	2.59%

Table 6. 3 – Medium friction optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0.75</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.5</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.7382	0.0005	0	0.0012	0.0005	0.0097	0.7259	0.4835	
	1.18%	0.05%	0%	0.12%	0.05%	0.97%	22.59%	1.65%	1.92%
<b>2</b>	0.7408	0.0057	0.0003	0	0.0307	0.0109	0.0095	0.4924	
	0.92%	0.57%	0.03%	0%	3.07%	1.09%	49.05%	0.76%	3.39%
<b>3</b>	0.7390	0	0.0046	0.0191	0.0032	0	0.6939	0.4081	
	1.1%	0%	0.46%	1.91%	0.32%	0%	19.39%	9.19%	9.40%
<b>4</b>	0.7449	0.0136	0.0064	0	0	0.0012	0.6655	0.3992	
	0.51%	1.36%	0.64%	0%	0%	0.12%	16.55%	10.08%	10.19%
<b>5</b>	0.7389	0.0026	0	0.0129	0	0	0.2059	0.4048	
	1.11%	0.26%	0%	1.29%	0%	0%	29.41%	9.52%	9.61%
<b>6</b>	0.7401	0.0003	0.0002	0.0021	0.0055	0.0040	0.5862	0.4731	
	0.99%	0.03%	0.02%	0.21%	0.55%	0.40%	8.62%	2.69%	2.78%
<b>7</b>	0.7461	0	0.0028	0	0	0.0007	0.3140	0.4160	
	0.39%	0%	0.28%	0%	0%	0.07%	18.60%	8.4%	8.40%
<b>8</b>	0.7368	0.0039	0.0071	0.0073	0.0016	0.0071	0.1747	0.4769	
	1.32%	0.39%	0.71%	0.73%	0.16%	0.71%	32.53%	2.31%	2.66%
<b>9</b>	0.7493	0.0014	0	0.0048	0.0032	0	0.6461	0.4235	
	0.07%	0.14%	0%	0.48%	0.32%	0%	14.61%	7.65%	7.67%
<b>10</b>	0.7482	0.0031	0.0001	0	0.0081	0.0037	0.0070	0.3844	
	0.18%	0.31%	0.01%	0%	0.81%	0.37%	49.30%	11.56%	11.60%

Table 6. 4 – High friction optimization results

It can be observed from the tables how the average error is higher for higher values of the friction compared to the low friction optimization results. This is due to the nature of optimization processes as they work better with values that are closer to the nominal ones in non-faulty conditions.

### 6.3.2 Backlash fault

Backlash fault has been assessed by implementing a low ( $k_2 = 0.2$ ), medium ( $k_2 = 0.6$ ) and high ( $k_2 = 0.8$ ) level of damage to analyze the behavior of the system in each of these scenarios. For each case, ten optimizations have been executed and the results have been reported in the tables below.

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0.2	0	0	0	0	0.5	0.5	%err
1	0.0029	0.1889	0.0013	0	0.0002	0.0018	0.2094	0.4846	
	0.29%	1.11%	0.13%	0%	0.02%	0.18%	29.06%	1.54%	1.91%
2	0.0003	0.1941	0.0011	0.0009	0.0014	0	0.1044	0.4735	
	0.03%	0.59%	0.11%	0.09%	0.14%	0%	39.56%	2.65%	2.72%
3	0.0032	0.1892	0	0.0016	0.0038	0.0009	0.6330	0.5083	
	0.32%	1.08%	0%	0.16%	0.38%	0.09%	13.3%	0.83%	1.43%
4	0.0017	0.1907	0.0005	0	0.0017	0.0012	0.2241	0.4749	
	0.17%	0.93%	0.05%	0%	0.17%	0.12%	27.59%	2.51%	2.69%
5	0.0013	0.1871	0.0021	0.0031	0	0	0.8656	0.4930	
	0.13%	1.29%	0.21%	0.31%	0%	0%	36.56%	0.7%	1.51%
6	0.0001	0.1893	0	0.0019	0.0001	0.0029	0.9728	0.4895	
	0.01%	1.07%	0%	0.19%	0.01%	0.29%	47.28%	1.05%	1.54%
7	0	0.1909	0	0.0004	0.0001	0.0016	0.5742	0.4775	
	0%	0.91%	0%	0.04%	0.01%	0.16%	7.42%	2.25%	2.43%
8	0	0.1958	0.0003	0	0.0015	0	0.3967	0.4909	
	0%	0.42%	0.03%	0%	0.15%	0%	10.33%	0.91%	1.21%
9	0.0011	0.1981	0.0012	0.0004	0	0.0009	0.2803	0.4821	
	0.11%	0.19%	0.12%	0.04%	0%	0.09%	21.97%	1.79%	1.81%
10	0.0038	0.1869	0	0.0011	0.0023	0.0002	0.6350	0.4870	
	0.38%	1.31%	0%	0.11%	0.23%	0.02%	13.5%	1.3%	1.86%

Table 6. 5 – Low backlash optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0.6	0	0	0	0	0.5	0.5	%err
1	0.0003	0.5983	0.0009	0	0.0005	0	0.3449	0.4732	
	0.03%	0.17%	0.09%	0%	0.05%	0%	15.51%	2.68%	2.69%
2	0.0011	0.5785	0.0016	0.0002	0.0007	0	0.5950	0.4974	
	0.11%	2.15%	0.16%	0.02%	0.07%	0%	9.50%	0.26%	2.17%
3	0	0.6030	0	0	0	0.0027	0.7535	0.4750	
	0%	0.3%	0%	0%	0%	0.27%	25.35%	2.5%	2.53%
4	0	0.6039	0.0002	0.0013	0.0004	0.0019	0.3979	0.4966	
	0%	0.39%	0.02%	0.13%	0.04%	0.19%	10.21%	0.34%	0.57%
5	0.0004	0.574	0.0007	0	0.0011	0.0009	0.0717	0.5045	
	0.04%	2.6%	0.07%	0%	0.11%	0.09%	42.83%	0.45%	2.64%
6	0.0018	0.6042	0	0.0021	0.0011	0.0014	0.6136	0.4795	
	0.18%	0.42%	0%	0.21%	0.11%	0.14%	11.36%	2.05%	2.11%
7	0.0003	0.5870	0.0012	0.0025	0.0009	0.0017	0.2658	0.4834	
	0.03%	1.43%	0.12%	0.25%	0.09%	0.17%	23.42%	1.66%	2.22%
8	0	0.6045	0.0012	0.0009	0	0	0.2963	0.4781	
	0%	0.45%	0.12%	0.09%	0%	0%	20.37%	2.19%	2.24%
9	0.0021	0.5784	0.0008	0	0	0.0022	0.4485	0.4738	
	0.21%	2.16%	0.08%	0%	0%	0.22%	5.15%	2.62%	3.40%
10	0.0016	0.5849	0.0003	0	0.0002	0.0016	0.9619	0.4893	
	0.16%	1.51%	0.03%	0%	0.02%	0.16%	46.19%	1.07%	1.86%

Table 6. 6 – Medium backlash optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0.8	0	0	0	0	0.5	0.5	%err
1	0.0003	0.7959	0.0008	0	0.0010	0.0023	0.6434	0.4858	
	0.03%	0.41%	0.08%	0%	0.10%	0.23%	14.34%	1.42%	1.50%
2	0.0015	0.7839	0.0004	0.0015	0.0035	0	0.1066	0.4915	
	0.15%	1.61%	0.04%	0.15%	0.35%	0%	39.34%	0.85%	1.86%
3	0.0017	0.7929	0	0.0019	0.0021	0.0034	0.5855	0.4760	
	0.17%	0.71%	0%	0.19%	0.21%	0.34%	8.55%	2.4%	2.54%
4	0.0017	0.8092	0.0001	0.0124	0	0.0182	0.2041	0.4947	
	0.17%	0.92%	0.01%	1.24%	0%	1.82%	29.59%	0.53%	2.44%
5	0.0008	0.7847	0.0011	0.0001	0.0004	0.0018	0.5879	0.5075	
	0.08%	1.53%	0.011%	0.01%	0.04%	0.18%	8.79%	0.75%	1.72%
6	0.0003	0.7965	0.0028	0.0004	0	0.0124	0.1839	0.5032	
	0.03%	0.35%	0.28%	0.04%	0%	1.24%	31.61%	0.32%	1.36%
7	0.0008	0.7837	0	0.0021	0.0031	0	0.3023	0.4961	
	0.08%	1.63%	0%	0.21%	0.31%	0%	19.77%	0.39%	1.72%
8	0.0005	0.7873	0.0002	0.0211	0.0018	0.0009	0.0654	0.4841	
	0.05%	1.27%	0.02%	2.11%	0.18%	0.09%	43.46%	1.59%	2.94%

<b>9</b>	0.0018	0.7934	0.0009	0	0.0014	0.0209	0.5541	0.4825	
	0.18%	0.66%	0.09%	0%	0.14%	2.09%	5.41%	1.75%	2.81%
<b>10</b>	0.0006	0.7920	0.0012	0.0025	0.0014	0	0.9476	0.5090	
	0.06%	0.8%	0.12%	0.25%	0.14%	0%	44.76%	0.9%	1.24%

Table 6. 7 – High backlash optimization results

Backlash detection is more straightforward compared to the other faults and the results acquired through the optimization process confirm that. Indeed, the average error as well as the compiling time for each level of damage are relatively low providing higher accuracy with lower computational effort.

### 6.3.3 Short circuit fault

Short circuit fault parameters  $k_3$ ,  $k_4$  and  $k_5$  behave in a similar fashion, so only the case in which phase A is in short circuit has been considered. It has been assessed by implementing a low ( $k_3 = 0.2$ ), medium ( $k_3 = 0.4$ ) and high ( $k_3 = 0.7$ ) level of damage to analyze the behavior of the system in each of these scenarios. For each case, ten optimizations have been executed and the results have been reported in the tables below.

Ref.	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	%err
	0	0	0.2	0	0	0	0.5	0.5	
<b>1</b>	0.0002	0.0128	0.1935	0	0.0028	0.0031	0.7852	0.4926	
	0.02%	1.28%	0.65%	0%	0.28%	0.31%	28.52%	0.74%	1.67%
<b>2</b>	0	0.0029	0.1883	0.0021	0.0011	0	0.6630	0.5468	
	0%	0.29%	1.17%	0.21%	0.11%	0%	16.30%	4.68%	4.84%
<b>3</b>	0.0019	0.0071	0.1297	0	0.0231	0	0.2479	0.5247	
	0.19%	0.71%	7.03%	0%	2.31%	0%	25.21%	2.47%	7.83%
<b>4</b>	0.0020	0	0.1908	0.0004	0.0029	0.0327	0.4127	0.4918	
	0.20%	0%	0.92%	0.04%	0.29%	3.27%	8.73%	0.82%	3.51%
<b>5</b>	0.0009	0	0.2004	0.0223	0	0	0.0173	0.5350	
	0.09%	0%	0.04	2.23%	0%	0%	48.27%	4.5%	5.02%
<b>6</b>	0.0024	0.0155	0.1228	0.0019	0.0001	0.0063	0.5331	0.5061	
	0.24%	1.55%	7.72%	0.19%	0.01%	0.63%	3.31%	0.61%	8.01%
<b>7</b>	0.0039	0.0082	0.1835	0	0.0010	0.0025	0.3485	0.4944	
	0.39%	0.82%	1.65%	0%	0.10%	0.25%	15.15%	0.56%	1.94%
<b>8</b>	0.0006	0.0047	0.1367	0	0.0026	0	0.2345	0.5169	
	0.06%	0.47%	6.33%	0%	0.26%	0%	26.55%	1.69%	6.57%

<b>9</b>	0	0.0203	0.1890	0.0108	0	0.0003	0.6864	0.5387	
	0%	2.03%	1.1%	1.08%	0%	0.03%	18.64%	3.87%	4.63%
<b>10</b>	0.0001	0.0025	0.1913	0.0014	0.0039	0.0008	0.8239	0.5326	
	0.01%	0.25%	0.87%	0.14%	0.39%	0.08%	32.39%	3.26%	3.41%

Table 6. 8 – Low short circuit optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0</b>	<b>0</b>	<b>0.4</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.5</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.0047	0.0054	0.3650	0.0095	0.0126	0.0038	0.7858	0.5102	
	0.47%	0.54%	3.5%	0.95%	1.26%	0.38%	28.58%	1.02%	4.03%
<b>2</b>	0.0028	0.0007	0.3868	0.0203	0.0027	0.0154	0.4847	0.5070	
	0.28%	0.07%	1.32%	2.03%	0.27%	1.54%	1.53%	0.7%	2.97%
<b>3</b>	0.0011	0.0056	0.3747	0.0079	0.0150	0.0237	0.2589	0.5149	
	0.11%	0.56%	2.53%	0.79%	1.50%	2.37%	24.11%	1.49%	4.17%
<b>4</b>	0.0005	0.0042	0.3224	0.0176	0.0163	0.0073	0.0652	0.5229	
	0.05%	0.42%	7.76%	1.76%	1.63%	0.73%	43.48%	2.29%	8.48%
<b>5</b>	0.0027	0.0002	0.3789	0.0227	0.0202	0.0024	0.0398	0.5023	
	0.27%	0.02%	2.11%	2.27%	2.02%	0.24%	46.02%	0.23%	3.71%
<b>6</b>	0.0042	0.0008	0.3713	0.0403	0.0084	0.0192	0.7571	0.5053	
	0.42%	0.08%	2.87%	4.03%	0.84%	1.92%	25.71%	0.53%	5.40%
<b>7</b>	0.0004	0.0027	0.3490	0.0116	0.0059	0.0005	0.2637	0.5527	
	0.04%	0.27%	5.1%	1.16%	0.59%	0.05%	23.63%	2.27%	5.47%
<b>8</b>	0.0017	0.0047	0.3744	0.0008	0.0172	0.0048	0.4337	0.5176	
	0.17%	0.47%	2.56%	0.08%	1.72%	0.48%	6.63%	1.76%	3.61%
<b>9</b>	0.0017	0.0015	0.3705	0.0074	0.0038	0.0304	0.8831	0.5041	
	0.17%	0.15%	2.95%	0.74%	0.38%	3.04%	38.31%	0.41%	4.34%
<b>10</b>	0.0009	0.0023	0.3527	0.0091	0.0067	0.0056	0.2264	0.4951	
	0.09%	0.23%	4.73%	0.91%	0.67%	0.56%	27.36%	0.49%	4.93%

Table 6. 9 – Medium short circuit optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0</b>	<b>0</b>	<b>0.7</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.5</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.0053	0.0048	0.6364	0.0241	0.0082	0.0307	0.4080	0.5316	
	0.53%	0.48%	6.36%	2.41%	0.82%	3.07%	9.20%	3.16%	8.16%
<b>2</b>	0.0038	0.0024	0.6545	0.0093	0.0127	0.0069	0.8464	0.5239	
	0.38%	0.24%	4.55%	0.93%	1.27%	0.69%	34.64%	2.39%	5.42%
<b>3</b>	0.0021	0.0041	0.6593	0.0089	0.0420	0.0314	0.3244	0.5087	
	0.21%	0.41%	4.07%	0.89%	4.20%	3.14%	17.56%	0.87%	6.77%

<b>4</b>	0.0027	0.0018	0.6624	0.0103	0.0079	0.0083	0.7617	0.5248	
	0.27%	0.18%	3.76%	1.03%	0.79%	0.83%	26.17%	2.48%	4.76%
<b>5</b>	0.0045	0.0005	0.6473	0.0218	0.0062	0.0148	0.0211	0.5313	
	0.45%	0.05%	5.27%	2.18%	0.62%	1.48%	47.89%	3.13%	6.70%
<b>6</b>	0.0009	0.0026	0.6534	0.0327	0.0227	0.0008	0.7087	0.5159	
	0.09%	0.26%	4.66%	3.27%	2.27%	0.08%	20.87%	1.59%	6.34%
<b>7</b>	0.0005	0.0069	0.6622	0.0060	0.0053	0.0005	0.5542	0.5381	
	0.05%	0.69%	3.78%	0.60%	0.53%	0.05%	5.42%	3.81%	5.47%
<b>8</b>	0.0039	0.0032	0.6491	0.0027	0.0089	0.0041	0.9108	0.5010	
	0.39%	0.32%	5.09%	0.27%	0.89%	0.41%	41.08%	0.1%	5.20%
<b>9</b>	0.0024	0.0011	0.6337	0.0092	0.0006	0.0162	0.0719	0.5173	
	0.24%	0.11%	6.63%	0.92%	0.06%	1.62%	42.81%	1.73%	7.10%
<b>10</b>	0.0013	0.0038	0.6672	0.0278	0.0198	0.0025	0.6919	0.5381	
	0.13%	0.38%	3.28%	2.78%	1.98%	0.25%	19.19%	3.81%	6.09%

Table 6. 10 – High short circuit optimization results

It can be observed from the tables how the average error increases as the short circuit parameter value approaches 1. However, the accuracy provided is still satisfactory compared to other faults.

### 6.3.4 Rotor eccentricity fault

Rotor eccentricity fault has been assessed by implementing a low ( $k_6 = 0.2$ ), medium ( $k_6 = 0.4$ ) and high ( $k_6 = 0.7$ ) level of damage to analyze the behavior of the system in each of these scenarios. For each case, ten optimizations have been executed and the results have been reported in the tables below.

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0	0	0	0	0.2	0.5	0.5	%err
<b>1</b>	0.0005	0	0.0003	0.0032	0.0016	0.2068	0.5062	0.4993	
	0.05%	0%	0.03%	0.32%	0.16%	0.68%	0.62%	0.07%	0.77%
<b>2</b>	0.0016	0.0032	0	0.0010	0.0035	0.2073	0.4996	0.4945	
	0.16%	0.32%	0%	0.10%	0.35%	0.73%	0.04%	0.55%	1.03%
<b>3</b>	0.0002	0.0017	0.0021	0.0005	0	0.2007	0.5027	0.5043	
	0.02%	0.17%	0.21%	0.05%	0%	0.07%	0.27%	0.43%	0.52%
<b>4</b>	0	0.0010	0.0004	0.0005	0.0001	0.2061	0.5004	0.4917	
	0%	0.10%	0.04%	0.05%	0.01%	0.61%	0.04%	0.83%	1.04%

<b>5</b>	0	0.0026	0.0004	0	0.0021	0.2075	0.5022	0.4933	
	0%	0.26%	0.04%	0%	0.21%	0.75%	0.22%	0.67%	1.06%
<b>6</b>	0.0012	0.0005	0.0018	0.0027	0.0025	0.2106	0.5015	0.5036	
	0.12%	0.05%	0.18%	0.27%	0.25%	1.06%	0.15%	0.36%	1.19%
<b>7</b>	0.0007	0.0081	0.0029	0.0009	0.0003	0	0.4991	0.4998	
	0.07%	0.81%	0.29%	0.09%	0.03%	20%	0.09%	0.02%	20.02%
<b>8</b>	0.0027	0.0017	0	0.0015	0.0018	0.2075	0.5039	0.5005	
	0.27%	0.17%	0%	0.15%	0.18%	0.75%	0.39%	0.05%	0.81%
<b>9</b>	0.0015	0	0.0004	0.0003	0	0.2055	0.5009	0.4937	
	0.15%	0%	0.04%	0.03%	0%	0.55%	0.09%	0.63%	0.84%
<b>10</b>	0.0008	0.0002	0.0018	0.0011	0.0006	0.2124	0.5066	0.5011	
	0.08%	0.02%	0.18%	0.11%	0.06%	1.24%	0.66%	0.11%	1.26%

Table 6. 11 – Low rotor eccentricity optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.4</b>	<b>0.5</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.0028	0.0003	0.0018	0	0.0023	0.4096	0.4974	0.4753	
	0.28%	0.03%	0.18%	0%	0.23%	0.96%	0.26%	2.47%	2.67%
<b>2</b>	0.0013	0	0.0023	0.0011	0	0.3993	0.5170	0.4706	
	0.13%	0%	0.23%	0.11%	0%	0.07%	1.70%	2.94%	2.95%
<b>3</b>	0.0001	0.0026	0	0.0007	0.0005	0.4084	0.4941	0.5099	
	0.01%	0.26%	0%	0.07%	0.05%	0.84%	0.59%	0.99%	1.33%
<b>4</b>	0	0.0013	0.0009	0	0.0005	0.3839	0.5089	0.4798	
	0%	0.13%	0.09%	0%	0.05%	1.61%	0.89%	2.02%	2.59%
<b>5</b>	0	0.0002	0.0004	0.0022	0	0.3679	0.4945	0.5132	
	0%	0.02%	0.04%	0.22%	0%	3.21%	0.55%	1.32%	3.48%
<b>6</b>	0.0030	0.0002	0	0.0006	0	0.3942	0.4854	0.4777	
	0.30%	0.02%	0%	0.06%	0%	0.58%	1.46%	2.23%	2.31%
<b>7</b>	0.0012	0.0017	0	0	0.0001	0.3896	0.4897	0.4730	
	0.12%	0.17%	0%	0%	0.01%	1.04%	1.03%	2.7%	2.90%
<b>8</b>	0.0035	0.0009	0.0012	0	0.0010	0.3768	0.5095	0.5186	
	0.35%	0.09%	0.12%	0%	0.10%	2.32%	0.95%	1.86%	2.98%
<b>9</b>	0.0008	0	0.0007	0.0010	0.0002	0.3890	0.4965	0.5169	
	0.08%	0%	0.07%	0.10%	0.02%	1.1%	0.35%	1.69%	2.02%
<b>10</b>	0	0.0003	0.0020	0.0107	0	0.3945	0.4896	0.4979	
	0%	0.03%	0.20%	1.07%	0%	0.55%	1.04%	0.21%	1.24%

Table 6. 12 – Medium rotor eccentricity optimization results

<b>Ref.</b>	$k_1$ <b>0</b>	$k_2$ <b>0</b>	$k_3$ <b>0</b>	$k_4$ <b>0</b>	$k_5$ <b>0</b>	$k_6$ <b>0.7</b>	$k_7$ <b>0.5</b>	$k_8$ <b>0.5</b>	<b>%err</b>
<b>1</b>	0.0030	0	0.0001	0.0004	0.0016	0.7072	0.5024	0.4744	
	0.30%	0%	0.01%	0.04%	0.16%	0.72%	0.24%	2.56%	2.66%
<b>2</b>	0.0004	0	0.0022	0	0	0.7062	0.4764	0.5159	
	0.04%	0%	0.22%	0%	0%	0.62%	2.36%	1.59%	1.72%
<b>3</b>	0.0004	0.0002	0.0009	0.0017	0.0004	0.6961	0.4867	0.4722	
	0.04%	0.02%	0.09%	0.17%	0.04%	0.39%	1.33%	2.78%	2.81%
<b>4</b>	0	0.0002	0.0009	0	0.0002	0.6908	0.4718	0.4877	
	0%	0.02%	0.09%	0%	0.02%	0.92%	2.82%	1.23%	1.54%
<b>5</b>	0.0012	0.0027	0.0018	0.0026	0.0013	0.7055	0.4879	0.4781	
	0.12%	0.27%	0.18%	0.26%	0.13%	0.55%	1.21%	2.19%	2.30%
<b>6</b>	0.0029	0.0015	0	0.0028	0.0022	0.7068	0.5182	0.4906	
	0.29%	0.15%	0%	0.28%	0.22%	0.68%	1.82%	0.94%	1.22%
<b>7</b>	0	0.0017	0.0027	0.0003	0.0006	0.6983	0.5071	0.5180	
	0%	0.17%	0.27%	0.03%	0.06%	0.17%	0.71%	1.8%	1.84%
<b>8</b>	0	0.0021	0.0010	0.0016	0	0.7021	0.4716	0.4871	
	0%	0.21%	0.10%	0.16%	0%	0.21%	2.84%	1.29%	1.34%
<b>9</b>	0.0008	0	0.00019	0.0009	0.0001	0.6923	0.4757	0.5118	
	0.08%	0%	0.19%	0.09%	0.01%	0.77%	2.43%	1.18%	1.42%
<b>10</b>	0.0017	0	0.0003	0	0.0012	0.6951	0.4858	0.5101	
	0.17%	0%	0.03%	0%	0.12%	0.49%	1.42%	1.01%	1.13%

*Table 6. 13 – High rotor eccentricity optimization results*

The average error in all three cases is relatively small and an acceptable accuracy is acquired. It is noteworthy to point out how the rotor eccentricity phase is easier to detect in this case and it's more accurate compared to when the rotor eccentricity amplitude value was very small: this is because the optimization process is unable to understand which is the correct rotor phase as all the considered rotor phase values produce small effects, so it gives as output random values that may be or not comparable with the reference value. Thus, a further calibration of the genetic algorithm parameters is needed to solve this issue.

### 6.3.5 Proportional gain fault

Proportional gain fault has been assessed by implementing a low ( $k_8 = 0.2$ ), medium ( $k_8 = 0.4$ ) and high ( $k_8 = 0.7$ ) level of damage to analyze the behavior of the system in each of these scenarios. For each case, ten optimizations have been executed and the results have been reported in the tables below.

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0	0	0	0	0	0.5	0.2	%err
1	0.0016	0.0034	0.0013	0.0036	0.0037	0.0169	0.5541	0.2086	
	0.16%	0.34%	0.13%	0.36%	0.37%	1.69%	5.41%	0.86%	2%
2	0.0007	0.0012	0	0.0024	0	0.0086	0.1132	0.2069	
	0.07%	0.12%	0%	0.24%	0%	0.86%	38.68%	0.69%	1.13%
3	0.0029	0.0010	0.0027	0.0003	0.0013	0.0052	0.7416	0.2130	
	0.29%	0.10%	0.27%	0.03%	0.13%	0.52%	24.16%	1.3%	1.44%
4	0.0018	0	0.0018	0.0011	0.0045	0.0205	0.9781	0.2048	
	0.18%	0%	0.18%	0.11%	0.45%	2.05%	47.81%	0.48%	2.16%
5	0.0030	0.0023	0.0006	0.0028	0.0004	0.0170	0.8665	0.2024	
	0.30%	0.23%	0.06%	0.28%	0.04%	1.70%	36.65%	0.24%	1.76%
6	0.0004	0.0017	0.0012	0.0007	0.0026	0.0069	0.1577	0.2149	
	0.04%	0.17%	0.12%	0.07%	0.26%	0.69%	34.23%	1.49%	1.68%
7	0	0.0001	0.0025	0.0012	0.0009	0.0181	0.7016	0.2074	
	0%	0.01%	0.25%	0.12%	0.09%	1.81%	20.16%	0.74%	1.98%
8	0.0006	0.0016	0.0031	0.0010	0.0021	0.0089	0.2155	0.2181	
	0.06%	0.16%	0.31%	0.10%	0.21%	0.89%	28.45%	1.81%	2.06%
9	0.0013	0.0005	0.0017	0	0.0047	0.0244	0.4521	0.2134	
	0.13%	0.05%	0.17%	0%	0.47%	2.44%	4.79%	1.34%	2.83%
10	0.0003	0	0.0002	0.0008	0.0019	0.0094	0.7168	0.2087	
	0.03%	0%	0.02%	0.08%	0.19%	0.94%	21.68%	0.87%	1.30%

Table 6. 14 – Low proportional gain optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0	0	0	0	0	0.5	0.4	%err
1	0.0008	0.0014	0.0007	0.0005	0.0025	0	0.6589	0.4095	
	0.08%	0.14%	0.07%	0.05%	0.25%	0%	15.89%	0.95%	1%
2	0.0011	0.0027	0.0003	0.0029	0.0009	0.0010	0.3236	0.4183	
	0.11%	0.27%	0.03%	0.29%	0.09%	0.10%	17.64%	1.83%	1.88%
3	0	0.0004	0	0.0017	0.0005	0.0010	0.8106	0.4143	
	0%	0.04%	0%	0.17%	0.05%	0.10%	31.06%	1.43%	1.44%

<b>4</b>	0.0009	0.0004	0.0003	0.0031	0	0.0016	0.9166	0.4084	
	0.09%	0.04%	0.03%	0.31%	0%	0.16%	41.66%	0.84%	0.92%
<b>5</b>	0.0009	0	0	0.0006	0	0	0.2071	0.4162	
	0.09%	0%	0%	0.06%	0%	0%	29.29%	1.62%	1.62%
<b>6</b>	0	0.0010	0.0014	0.0004	0.0001	0	0.6650	0.4057	
	0%	0.10%	0.14%	0.04%	0.01%	0%	16.50%	0.57%	0.6%
<b>7</b>	0.0022	0	0.0012	0.0011	0	0	0.6402	0.4113	
	0.22%	0%	0.12%	0.11%	0%	0%	14.02%	1.13%	1.16%
<b>8</b>	0.0002	0.0001	0.0008	0.0002	0	0.0028	0.2612	0.4170	
	0.02%	0.01%	0.08%	0.02%	0%	0.28%	23.88%	1.7%	1.73%
<b>9</b>	0.0010	0.0012	0.0017	0	0.0023	0.0035	0.1713	0.4037	
	0.10%	0.12%	0.17%	0%	0.23%	0.35%	32.87%	0.37%	0.6%
<b>10</b>	0	0	0.0011	0	0.0015	0.0018	0.5924	0.4093	
	0%	0%	0.11%	0%	0.15%	0.18%	9.24%	0.93%	0.97%

Table 6. 15 – Medium proportional gain optimization results

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
Ref.	0	0	0	0	0	0	0.5	0.7	%err
<b>1</b>	0.0005	0.0002	0.0009	0.0004	0.0011	0.0029	0.3461	0.6903	
	0.05%	0.02%	0.09%	0.04%	0.11%	0.29%	15.39%	0.97%	1.02%
<b>2</b>	0.0003	0	0.0002	0.0012	0.0009	0.0038	0.3648	0.7009	
	0.03%	0%	0.02%	0.12%	0.09%	0.38%	13.52%	0.09%	0.42%
<b>3</b>	0.0007	0.0004	0	0	0.0002	0.0074	0.2677	0.6938	
	0.07%	0.04%	0%	0%	0.02%	0.74%	23.23%	0.62%	0.97%
<b>4</b>	0	0.0010	0.0002	0	0.0002	0.0043	0.9420	0.6992	
	0%	0.10%	0.02%	0%	0.02%	0.43%	44.20%	0.08%	0.45%
<b>5</b>	0.0011	0.0001	0	0.0001	0.0008	0.0086	0.1862	0.7043	
	0.11%	0.01%	0%	0.01%	0.08%	0.86%	31.38%	0.43%	0.97%
<b>6</b>	0.0004	0	0.0009	0.0006	0	0.0012	0.4268	0.7130	
	0.04%	0%	0.09%	0.06%	0%	0.12%	7.32%	1.3%	1.31%
<b>7</b>	0.0004	0	0.0012	0.0006	0.0013	0.0035	0.7485	0.7115	
	0.04%	0%	0.12%	0.06%	0.13%	0.35%	24.85%	1.15%	1.22%
<b>8</b>	0	0.0002	0.0006	0	0.0027	0.0019	0.3179	0.7053	
	0%	0.02%	0.06%	0%	0.27%	0.19%	18.21%	0.53%	0.63%
<b>9</b>	0.0003	0.0008	0.0015	0.0004	0.0009	0.0028	0.8210	0.6901	
	0.03%	0.08%	0.15%	0.04%	0.09%	0.28%	32.10%	0.99%	1.05%
<b>10</b>	0.0001	0	0.0007	0.0010	0	0.0051	0.6537	0.6959	
	0.01%	0%	0.07%	0.10%	0%	0.51%	15.37%	0.41%	0.67%

Table 6. 16 – High proportional gain optimization results

By comparing the results acquired from the proportional gain fault isolation to the other faults, it can be deduced that it is the most detectable out of all. Indeed, the average error for all levels of damage is very low reaching a maximum of 2.83%, providing a very high accuracy.

## 6.4 Multiple fault isolation

In a real scenario, faults do not occur one at a time but there may be situations in which multiple faults are present. In order to test the performance and the accuracy of the genetic algorithm, a multiple fault optimization is executed. The reference values can be introduced either to liking or randomly. The latter method has been chosen and the results acquired are shown in the table below.

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$	$k_8$	
<b>Ref.</b>	<b>0.2383</b>	<b>0.5003</b>	<b><math>5.32 \cdot 10^{-7}</math></b>	<b>0.4503</b>	<b>0.0404</b>	<b>0.0028</b>	<b>0.2785</b>	<b>0.5</b>	<b>%err</b>
<b>1</b>	0.2051	0.4727	$9.44 \cdot 10^{-4}$	0.4414	0.0445	0.0041	0.4334	0.4950	
	3.32%	2.76%	0.09%	0.89%	0.41%	0.13%	15.49%	0.5%	4.46%
<b>2</b>	0.2137	0.4680	0.0111	0.4449	0.0033	0.0006	0.6516	0.4949	
	2.46%	3.23%	1.11%	0.54%	3.71%	0.22%	37.31%	0.51%	5.66%
<b>3</b>	0.2192	0.4858	$5.59 \cdot 10^{-5}$	0.4421	0.0026	0.0067	0.1195	0.5096	
	1.91%	1.45%	0.01%	0.82%	3.78%	0.39%	15.90%	0.96%	4.67%
<b>4</b>	0.2132	0.4732	0.0015	0.4466	0.0221	0.0010	0.0874	0.4962	
	2.51%	2.71%	0.15%	0.37%	1.83%	0.18%	19.11%	0.38%	4.16%
<b>5</b>	0.2137	0.4813	$2 \cdot 10^{-4}$	0.4504	0.0021	0.0067	0.1402	0.4970	
	2.46%	1.9%	0.02%	0.01%	3.83%	0.39%	13.83%	0.3%	4.96%
<b>6</b>	0.2143	0.4771	0.0012	0.4545	0.0031	0.0052	0.2438	0.4999	
	2.4%	2.32%	0.12%	0.42%	3.73%	0.24%	3.47%	0.01%	5.03%
<b>7</b>	0.2011	0.4936	$7.9 \cdot 10^{-4}$	0.5019	0.0178	0.0066	0.5405	0.5350	
	3.72%	0.67%	0.08%	5.16%	2.26%	0.38%	26.20%	3.5%	7.64%
<b>8</b>	0.1959	0.4807	$7.52 \cdot 10^{-4}$	0.4776	0.0373	0.0017	0.1992	0.4995	
	4.24%	1.96%	0.08%	2.73%	0.31%	0.11%	7.93%	0.05%	5.42%
<b>9</b>	0.2123	0.4780	$2.59 \cdot 10^{-4}$	0.4660	0.0087	0.0009	0.6436	0.4978	
	2.6%	2.23%	0.03%	1.57%	3.17%	0.19%	36.51%	0.22%	4.93%
<b>10</b>	0.2142	0.4784	$3.5 \cdot 10^{-5}$	0.4590	0.0018	0.0015	0.9062	0.4996	
	2.41%	2.19%	0%	0.87%	3.86%	0.13%	62.77%	0.04%	5.13%

Table 6. 17 – Random multiple fault parameters results

The algorithm has chosen mostly small fault parameters, showing a good accuracy and error values that are on a par with the ones obtained from the single fault isolation. It is worth noticing how backlash and short circuit faults contribute the most to the total error: that is due to the fact that they are more difficult to detect appropriately as they have a lower effect on the equivalent output current.

Overall, the genetic algorithm has demonstrated to be a valuable optimization method for both the single and the multiple fault isolation, providing satisfactory results with quite high accuracy.

## 7 Conclusions

The prior knowledge of incipient failures of primary flight command electromechanical actuators (EMAs) with prognostic algorithms constitutes the main topic of this work. Such algorithms often exploit a model-based approach established on the direct comparison between the real (High Fidelity) system and the monitor (Low Fidelity) system to identify fault parameters through optimization processes, with the monitor model allowing to acquire accurate and precise results with a contained computational effort. Indeed, employing an algorithm capable of detecting the most common errors at an early stage would enable a more effective maintenance program to be activated, as each component could only be replaced when needed and almost every corrective action planned in advance. To this end, the application of the Genetic Algorithm for fault detection in an electromechanical actuator has been performed attaining rather promising results.

First of all, the opening chapter deals with all the theoretical concepts concerning prognostics and the description of how flight controls, actuation systems as well as brushless electrical motors work. After that, the reference and the monitor models have been deeply analyzed and their output in terms of position, speed and current trends has been displayed. Then, a fault analysis has been performed to introduce the main mechanical and electrical faults that can occur in the system studied: friction, backlash, coil short circuit, static rotor eccentricity, and proportional gain. Fault detection and isolation is performed by comparing the output signal of the reference system with the one obtained from the monitor model. Finally, after the description of the Genetic Algorithm and the definition of the fitness function and its arguments, the optimization has been performed and its results have been disclosed.

Generally, genetic algorithms are intrinsically non-deterministic and, as such, it is unreasonable to expect a perfect success rate. Certainly, running the optimization taking into account the same reference data produces slightly different results each time and sometimes the impact on the overall error can be quite significant. However, executing the optimization ten times with the same inputs mostly results in a proper fault isolation and the outcome can be considered satisfactory, albeit this work leaves room for further improvements to attain a much higher accuracy.

One way of enhancing the efficiency of the system is by speeding up the optimization process. One solution has been already proposed in *paragraph 5.6* which is parallelization: by using a multi-core architecture, it is possible to reduce the compiling time up to 25% by assigning the job to multiple core processors instead of having one do all the optimization and, hence, making the process last for an unreasonable amount of time. Another solution could be to introduce some changes to the monitor model in terms of connection lines that may be simplified in some other form.

Another problem relies on the correct estimation of the rotor eccentricity phase as stated in *paragraph 6.3.4*: when the rotor eccentricity amplitude value is very small, the optimization process is unable to understand which is the correct rotor phase as all the considered rotor phase values produce small effects, so it gives as output random values that may be or not comparable with the reference value. Thus, a further development of the models has to be done in order to solve this issue.

Lastly, different optimization algorithms could certainly be introduced and examined to find out which is the best compromise to investigate the issue: truly, there are many rapidly growing algorithms such as Grey Wolf Optimization, Particle Swarm Optimization and Different Evolution that may achieve more reliable results. These algorithms may also be tested in a real situation, possibly using the data of a real sensor located on a test bench.



## Bibliography

- [1] Pier Carlo Berri, *Genetic Algorithms for Prognostics of Electromechanical Actuators*, M.D. Thesis, A.Y. 2016
- [2] Stefano Re, *Development and comparison of prognostic methodologies applied to Electromechanical Servosystems (EMA) for aerospace purposes*, M.D. Thesis, A.Y. 2018
- [3] Francesco Viglione, *Development and validation of a monitoring model for EMA fault analysis*, M.D. Thesis, A.Y. 2019
- [4] Manuela Battipede, Matteo D. L. Dalla Vedova, Paolo Maggiore and Simone Romeo, *Model based analysis of precursors of electromechanical servomechanisms using an artificial neural network*, AIAA Modeling and Simulation Technologies Conference, Kissimmee, Florida, 2015
- [5] Paolo Maggiore, Matteo D. L. Dalla Vedova, Lorenzo Pace and Alessio Desando, *Definition of parametric methods for fault analysis applied to an electromechanical servomechanism affected by multiple failures*, Second European Conference of the Prognostics and Health Management Society, Nantes, France, 2014
- [6] Matteo D. L. Dalla Vedova, Pier Carlo Berri and Paolo Maggiore, *A Smart Electromechanical Actuator Monitor for New Model-Based Prognostic Algorithms*, International Journal of Mechanics and Control, 2016
- [7] Matteo D. L. Dalla Vedova, Paolo Maggiore, Lorenzo Pace and Alessio Desando, *Evaluation of the Correlation Coefficient as a Prognostic Indicator for Electromechanical Servomechanism Failures*, International Journal of Prognostic and Health Management, 2015

- [8] Matteo D. L. Dalla Vedova, Paolo Maggiore and Lorenzo Pace, *Parametric Methods for fault analysis applied to a servomechanism affected by multiple failures*, Recent Advances in Applied Mathematics, Modelling and Simulation, 2014
- [9] Dario Belmonte, Matteo D. L. Dalla Vedova and Paolo Maggiore, *New Prognostic Method Based on Spectral Analysis Techniques Dealing with Motor Static Eccentricity for Aerospace Electromechanical Actuators*, World Scientific and Engineering Academy and Society, 2015
- [10] Federal Aviation Administration, *Pilot's Handbook of Aeronautical Knowledge*, FAA-H-8083-25B, 2016
- [11] Eric Olson, *Aircraft actuation technologies: how do electrohydraulic, electrohydrostatic and electromechanical actuators work?*, Globalspec, 2019
- [12] Guan Qiao, Geng Liu, Zhenghong Shi, Yawen Wang, Shangjun Ma and Teik C Lim, *A review of electromechanical actuators for More/All Electric aircraft systems*, Journal of Mechanical Engineering Science, 2017
- [13] Aldo Boglietti, *Dispense di macchine elettriche: Motori Brushless*, 2018
- [14] Vijini Mallawaarachchi, *Introduction of Genetic Algorithms*, Towards Data Science, 2017
- [15] Andrew Chipperfield, Peter Fleming, Hartmut Pohlheim and Carlos Fonseca, *Genetic Algorithm Toolbox User's Guide*, Department of Automatic Control and Systems Engineering of the University of Sheffield, 1994
- [16] MathWorks, *MATLAB Documentation*, [www.mathworks.com/help/](http://www.mathworks.com/help/)