


Doctoral Dissertation
Doctoral Program in Aerospace Engineering (33th Cycle)

Design and development of algorithms and technologies applied to prognostics of aerospace systems

Pier Carlo Berri

* * * * * *

Supervisors
Prof. Paolo Maggiore, Supervisor
Ing. Matteo D.L. Dalla Vedova, Co-Supervisor
Dr. Ing. Laura Mainini, Co-Supervisor

Doctoral Examination Committee:
Prof. Piero Baraldi, Referee, Politecnico di Milano
Prof. Elvio Bonisoli, Referee, Politecnico di Torino
Prof. Daniel Milanese, Referee, University of Parma
Prof. Francesco Tornabene, Università del Salento
Prof. Kimon P. Valavanis, Referee, University of Denver.

Politecnico di Torino
March 23, 2021

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial - NoDerivative Works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

…….....

Pier Carlo Berri

Turin, March 23, 2021

Summary

Prognostics and Health Management (PHM) is an emerging field that aims to determine the Remaining Useful Life (RUL) of systems, in order to plan in advance the required maintenance interventions. Currently, components are replaced at the end of their design life, which is the result of a tradeoff between maintenance costs and reliability and availability requirements. With a PHM approach, it will be possible to schedule replacements accounting for the actual condition of the system, without decommissioning operable equipment or flying with worn components. In addition, prognostics performed in real-time may allow adapting the operational envelope of a vehicle adaptively, in order to increase the system life without jeopardizing the success of the mission.

Most approaches to failure prognosis available in literature require a significant computational burden, not suitable for real-time computations, and are characterized by a large uncertainty associated to the RUL prediction. This in part is due to the inherent unpredictability of the propagation rate of damages, which is influenced by several variables that cannot be controlled nor measured; another source of uncertainty lies in the errors associated with the fault detection processes.

This study addresses these limitations and provides a comprehensive computational framework for fast and reliable RUL prediction. Physics-based models of the system dynamics are combined with supervised and unsupervised machine learning to obtain surrogate representations of the equipment and allow for real-time evaluations. The method is tested on the RUL prediction task of an electromechanical actuator for aircraft flight controls. This is a challenging and representative case study as flight controls are complex subsystems of a vehicle that involve the interaction between a number of heterogeneous disciplines, such as mechanics, electronics, fluid dynamics and control theory. Multiple fault modes can affect an actuator at the same time and influence each other, making the fault detection and RUL prediction tasks difficult. Highly detailed physics-based

simulations are employed as a simulated test bench for the PHM algorithms. An experimental validation of the numerical models is provided by a physical electromechanical actuator test rig.

Additionally, innovative sensor technology is discussed as a promising candidate to collect some of the required input data for the prognostic process. Specifically, precise measurements of in-flight aerodynamic loads on the flight control actuators are required for on-board prognostics as they influence significantly the response of the flight control system. Often this information is not available as it cannot be measured reliably and conveniently with traditional technologies. Optical sensors are considered for the task as they permit to achieve high frequency, accurate measurements with a good spatial resolution and a minimally invasive installation.

Acknowledgment

I wish to thank prof. Paolo Maggiore, Ing. Matteo D.L. Dalla Vedova and Dr. Laura Mainini for their invaluable support to this research. Rocco Antonio Cataldi, Carlo Murè, Aldo Negri and Matteo Malara of Siemens S.p.A, who helped us with the integration of the test bench for the experimental validation of the dynamical models employed in this work. All the master's and bachelor's thesis students, without whom much of the results presented here would not have been achieved. Last, but not least, I would like to thank Sonia and all my family, who encouraged and supported me throughout my PhD career.

Contents

List of Tables

List of Figures

Nomenclature

Problem statement and methodology

b	bias of neurons
G	Gappy matrix
$\boldsymbol{k}$	vector of fault parameters (i.e. fault combination or health condition)
$\boldsymbol{k}_{NC}$	vector of fault parameters in nominal conditions (i.e. without any faults)
K	matrix of fault combinations $\boldsymbol{k}$
l	index of the SOM neuron closest to an input
n_e	number of components of $\boldsymbol{y}$
n_k	number of components of $\boldsymbol{k}$
n_m	number of retained POD modes
n_s	number of samples in the importance sampling
n_ω	number of SOM neurons
t	time coordinate
T	training set
$\boldsymbol{w}$	weight vector
$\boldsymbol{x}$	reference coordinate (e.g. time, position, speed etc.) of a measured signal $y(x,k)$
$\hat{\boldsymbol{x}}$	compression mask (i.e. list of x coordinates to store and process online)
$\boldsymbol{y}(x,k)$	measured signal from the system
Y	matrix of measured signals $\boldsymbol{y}$
$\boldsymbol{\alpha}$	coefficients of the POD expansion
$\boldsymbol{\beta}$	gradient of SVM hyperplane
γ	weight parameter of the dynamic estimator filter for RUL estimation
λ_i	POD eigenvalues
$\boldsymbol{\nu}_i$	POD modes
$\phi_a(k)$	function for health condition assessment (binary valued failure threshold)
$\hat{\phi}_a(k)$	surrogate function for health condition assessment
Φ	matrix of values of ϕ_a
σ	cumulative sum of the POD eigenvalues
$\boldsymbol{\tau}$	training point for SOM

EMA models

b_h	hysteresis controller band
BLK	backlash amplitude
C_m	damping of the actuator's rotating assembly
G_P	proportional gain of PID controller
G_I	integrative gain of PID controller
G_D	derivative gain of PID controller
g	air gap width
i_j	actual current in the j-th phase
i_d	direct current of PMSM
i_q	quadrature current of PMSM
I_{ref}	reference current value (i.e. torque/current setpoint)
J_m	inertia of the actuator's rotating assembly
k_j	back-EMF coefficient of the j-th phase
L_j	inductance of the j-th phase
P	number of pole-pairs
R_j	resistance of the j-th phase
T_m	motor torque
T_l	load torque
V_j	voltage applied to the j-th phase
τ_D	time constant of PID derivative filter
θ_m	mechanical position of the motor
θ_e	electrical position of the motor
θ_u	user position
ζ	normalized rotor eccentricity
ω	angular speed

Estimation of actuator load

C_t	temperature coefficient
J_x	cross-section moment of inertia along the x axis
J_z	cross-section moment of inertia along the z axis
J_{xz}	cross-section $x - y$ product of inertia
$L(y)$	distribution of lift
$M(y)$	distribution of pitching moment
M_x	bending moment

M_t	torsion moment
n	index of refraction
p_e	photoelastic coefficient
T_z	shear force
ΔT	temperature variation
V	airspeed
y	spanwise coordinate
$\alpha(y)$	local angle of attack
α_0	global angle of attack
α_c	contribution of control surface deflection to angle of attack
α_g	wing twist contribution to angle of attack
α_i	induced angle of attack
α_f	thermal expansion coefficient
Γ	circulation
ε	strain
Λ_g	axial pitch of Fiber Bragg Grating
λ_B	peak wavelength of Fiber Bragg Grating
ρ	air density
ζ_f	thermo-optic coefficient
σ	normal stress
τ	shear stress

Validation of physics-based models

A_{in}	amplitude of system input (for dynamical response)
A_{out}	amplitude of system output (for dynamical response)
F_{lc}	force measured by loadcell
G_{db}	gain of transfer function
i	gear ratio
r	radial distance
T_m	motor torque
T_u	user torque
T_f	preload friction torque
z	number of teeth
ϕ_{in}	phase of system input (for dynamical response)
ϕ_{out}	phase of system output (for dynamical response)
η_D	direct efficiency (or opposing load efficiency)

η_I	inverse efficiency (or aiding load efficiency)
ω_m	motor speed
ω_u	user speed

Validation of Machine Learning models

A	accuracy of SVM classifier
err_y	error on signal reconstruction
err_α	error on POD coefficients
err_k	error on FDI
f_f	false positives
f_t	true positives
h_f	false negatives
h_t	true negatives
k_s	SVM kernel scale
FP	false negative ratio
MD	false positive ratio

Chapter 1

Introduction

The complexity of aerospace systems is increasing steadily, as next generation platforms are developed and enter the market. As a result, the design and operation of the vehicle relies on the integration of a multitude of heterogeneous components interacting with each other. Each piece of equipment within the system is characterized by its own failure modes, and its malfunction influences the operation of the system as a whole, in ways that may be difficult to predict. This makes the troubleshooting tasks difficult and time expensive, and ultimately yields to worsen the reliability and availability of the aerospace platform.

Traditionally, the approach to system safety and reliability relies on scheduling the maintenance intervention *a priori*: the periodical maintenance intervals are determined statically during the design phase, analysing the failure rate of the involved equipment [1, 2, 3]. The uncertainty associated to this approach is large, as two identical components subject to the same mission profile may fail at very different times. Then, the maintenance intervals are determined as a tradeoff between the risk of flying a damaged component and that of replacing equipment that is still operable. To mitigate the risk associated to failure of safety-critical equipment, redundancies are employed at component or system level [4, 5]; as a drawback, basic reliability and weight are worsened.

Advances in Prognostics and Health Management (PHM) disciplines permit to monitor continuously the actual health condition of components and estimate their Remaining Useful Life (RUL) [6, 7, 8]. This information is leveraged by newer methods for product life-cycle management like Condition Based Maintenance (CBM) [9, 10, 11] and Integrated Vehicle health Management (IVHM) [12, 13, 14] to plan the maintenance interventions in an optimal way, to increase the mission reliability of the vehicle, and to reduce costs related to the operation of the fleet. Then, a significant research effort is focused on giving to next generation system the capability to diagnose their damages and faults in advance and predicting the RUL during operations, autonomously. This skill permits the replacement of components only when actually necessary, avoiding additional downtime on

systems that are still operable, and allows to reconfigure the mission profile of the vehicle dynamically in order to ensure a longer and safer system life [15, 16].

Most PHM strategies available in literature rely either on a model-based approach (i.e. employing physics-based models of the monitored systems) or on a data-driven one. Tinga and Loendersloot [17] provide a review of model-based strategies for prognostics. In [18] a model of spacecraft dynamics is compared against the measured attitude of a satellite to identify faults in the Attitude Determination and Control System (ADCS). A similar approach is leveraged by Henry et al. in [19]. Battipede et al. [20] discuss a model-based diagnostic strategy that analyses the output of an actuator through Fast Fourier Transform (FFT). A structured residual between a digital twin and the system response is employed in [21] to determine faults of industrial compressors. Shi et al. [22] employ particle filtering to perform failure prognosis on electrohydrostatic actuators. In [23], Huang et al. present a review of data-driven prognostic strategies. Bektas et al. [24] propose a neural network framework for similarity-based prognostics. An Extreme Learning Machine is leveraged in [25] for condition monitoring of wind power equipment. In [26] a neural network is employed for fault detection of an aircraft electromechanical actuator. In [27], RUL prediction of milling machine cutting tools is performed employing autoregressive integrated moving average (ARIMA). A similar case study is addressed in [28] employing Deep Learning. Usually, model-based strategies need long computational times and significant hardware resources, so they are not suitable for real-time execution. On the contrary, data-driven approaches require large training datasets, which are impractical to collect from experimental or field data. For example, critical but uncommon failure modes tend to be underrepresented in field data, but they must be accounted for by prognostic algorithm.

This study discusses a computational framework for a nearly real-time prediction of the Remaining Useful Life of dynamical assemblies, initially proposed by Berri et al. in [29, 30]. The input consists in the measurements taken from sensors installed on the system, either for feedback or for diagnostic purposes; these data can be of various nature and depend on the specific characteristics of the monitored equipment. Examples are the currents and voltages applied to an electrical circuit, speed and position of an actuator or pressure, torque produced by a motor, flow rate and temperature of hydraulic fluid. The procedure is characterized by the combination of an optimal signal compression method with reduced order modelling and machine learning algorithms. This way a computationally efficient map that permits to associate the sensor measurements to a prediction of the system's RUL is built offline. As a result, the storage and processing power necessary for the real-time RUL prediction is reduced dramatically, even accounting for the time and hardware constraints usually associated to on-board computations. The proposed approach involves surrogate models of the system that are trained offline: these models are used online in order to reduce the computational burden, which is required for the diagnosis of the actual system health status and for the failure prognosis. In addition, the dimensionality of the problem is reduced by learning online an informative compression mask, which

allows to store and process online only a limited set of informative signal components, without compromising accuracy.

In this study, the actuators for aircraft Flight Control Systems (FCSs) are taken into account for the application of the proposed methodology. FCSs are among the most critical systems of an aircraft, since their failure may result, in extreme scenarios, in the impossibility to control the vehicle, with obvious safety concerns. For this reason, monitoring the health status of the FCSs is very important because it can produce substantial progresses in terms of operating costs, aircraft performance, mission reliability and even ease requirements on system redundancies.

The problem of FCS prognostics is intrinsically challenging: different disciplines interact in describing the model of FCS equipment; to guarantee the operation of the system, electrical/electronics, mechanical, aerodynamic, structural and hydraulic subsystems need to work concurrently. The dimensionality of the FDI problem is a result of the large amount of the possible failure modes. Different faults may hide each other, or may have similar effects to that they are easily misidentified. Additionally, specific environmental or operating conditions can trigger false positive fault detections. Then, the specific test case of flight control actuators is an interesting one for the demonstration of the proposed computational framework.

1.1 Problem formulation

Usually the PHM process includes three tasks, namely signal measurement and feature extraction, Fault Detection and Identification (FDI), and prediction of Remaining Useful Life (RUL), as shown in Figure 1.

In the signal measurement and feature extraction task, the output signals of the system are measured with a uniform acquisition frequency; they can be stored as time-series, or statistical features can be extracted such as moving averages, variance and skewness of the data. The monitored quantities shall be sensitive to the possible presence of incipient faults, and only marginally affected by changes

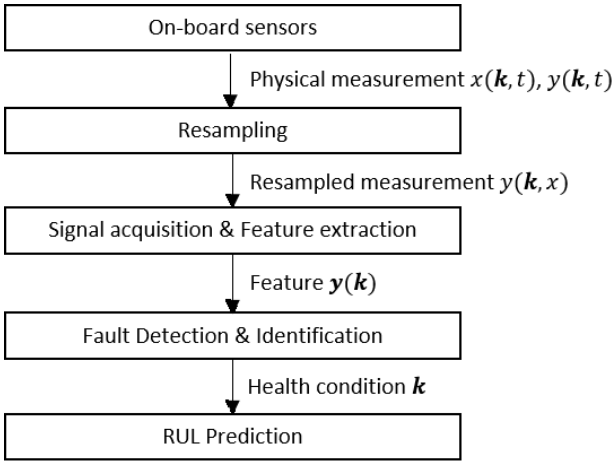

Figure 1.1: Common PHM flow.

in the operating conditions of the system. In the FDI task, the output of the previous phase is processed and analysed in search of the early signs of damage to determine the actual health status of the components. Eventually, in the RUL prediction task, the current condition of the system is employed as a starting point for the failure prognosis, i.e. to inform an estimate of the RUL of the system.

Traditionally, only the signal measurement and feature extraction task is performed online, as the computational burden required by FDI and RUL prediction is usually too high. The signal acquisition consists in acquiring a set of signals $y(\boldsymbol{k}, t)$, sampled with a constant frequency dependent on the capability of the hardware and on the characteristics of the information that shall be captured. The signals can be then resampled as a function of other parameters x of the system in order to obtain and informative map $y(\boldsymbol{k}, x)$, which can be stored in vector form $\boldsymbol{y}(\boldsymbol{k})$. This vector can be high dimensional and cumbersome to store and process; for this reason, the FDI and RUL prediction tasks are usually performed offline. In this work, this issue is addressed by finding a compressed representation of $\boldsymbol{y}(\boldsymbol{k})$ to reduce the dimensionality of the FDI task.

The informative map $\boldsymbol{y}(\boldsymbol{k})$ is processed in the FDI to obtain an estimate of the current health status of the system. Several FDI approaches are model-based: they rely on physics-based models of the monitored equipment and compare their output to that measured from the physical equipment. For example, Freeman et al. [31] compute the residual between a model of aircraft dynamics and the actual response of the vehicle; this residual is analysed by a statistical algorithm to detect anomalies. In [32], a similar strategy is employed to detect flight control actuator faults on a small UAS; this strategy relies on the analysis of the fault effect at vehicle level; then, small, incipient damages are hardly identifiable. Meng et al. [33] propose to use a Kalman filter for model-based FDI of wind turbines. In these approaches, the FDI task is formalised as an optimisation problem whose solution is the fault condition $\boldsymbol{k}$ that minimizes the discrepancies between the output $\boldsymbol{y}$ measured from the physical equipment and that of a model sensitive to the health condition $\boldsymbol{y}_m(\boldsymbol{k})$:

$$\boldsymbol{k} = \underset{\boldsymbol{k}}{\operatorname{argmin}}\, err_y(\boldsymbol{y} - \boldsymbol{y}_m(\boldsymbol{k})) \tag{1.1}$$

where, in general, the error $err_y\big(\boldsymbol{y} - \boldsymbol{y}_m(\boldsymbol{k})\big)$ is a monotonically increasing function of $\|\boldsymbol{y} - \boldsymbol{y}_m(\boldsymbol{k})\|$. The specific norm $\|\cdot\|$ to be employed depends on the peculiar characteristic of each individual application. Under this assumption, err_y has a global minimum where $\boldsymbol{y}_m$ is the best possible approximation of $\boldsymbol{y}$. In this condition, the vector $\boldsymbol{k}$ can be assumed as a good representation of the actual health of the system. The If a purely physics-based approach is used, the computation of the model $\boldsymbol{y}_m(\boldsymbol{k})$ can be expensive; the use of an optimization algorithm to solve the problem of Equation (1.1) requires to evaluate this model iteratively until convergence, and yields a computational effort that is not suitable for real-time, on-board FDI. In addition, the choice of an optimal error function may be challenging. On the contrary, data-driven algorithms allow for faster online computations, but require to learn surrogate models offline from large datasets that are often not available: their collection may require several thousands of hours of system

operation in nominal and faulty conditions, and can be obtained only after the system has entered service [34]. The output of the FDI task is an estimate of the health condition $\boldsymbol{k}$ to be used for the prediction of Remaining Useful Life.

The RUL of a system can be defined as the residual time before a failure occur; the term failure means the inability of the system to meet its functional or performance requirements [35, 36]. Formally, we can define a function $\phi_a(\boldsymbol{k})$ for the assessment of the system health [37]; ϕ_a is a binary classifier that compute the residual performance of the system under the effect of the faults $\boldsymbol{k}$ and compares it to the applicable requirements. If the requirements are still met, a "healthy" label is assigned to $\boldsymbol{k}$, otherwise a "faulty" label. Then, the RUL prediction problem can be written as:

$$\begin{aligned} &RUL = \min(t) \\ &\text{s.t.} \quad \phi_a\big(k(t)\big) = \text{"faulty"} \end{aligned} \tag{1.2}$$

assuming that the current time is $t = 0$. In the traditional approach to system lifecycle management, the life of a piece of equipment is computed in the design cycle, as a combination of the failure rates of individual components. This kind of estimate has a very large uncertainty margin, as two identical parts may age differently and fail at different times. On the contrary, PHM methods rely on a RUL prediction performed during the operation of the system, accounting for its actual operating condition and health status. In order to obtain a RUL estimate, most approaches to failure prognosis either extrapolate the observed fault propagation rate [38] or evaluate a model of damage growth until the health condition $\boldsymbol{k}$ reaches a failure threshold. In [39], a semi-markov model is combined with the Maximum Likelihood Estimation (MLE) method to infer a damage propagation model. Jacazio et al. [40] propose the use of Particle Filtering to predict the RUL. Li et al. [41] combine Particle Filtering with Canonical Variate Analysis (CVA) and Exponentially Weighted Moving Average (EWMA) for failure prognosis of industrial rotating machinery. Usually, the main limitations that characterize PHM strategies available in literature are related to the sensitivity to structured uncertainties (i.e. uncertainties associated to the parameters of the system model) and unstructured uncertainties (i.e. uncertainties associated to phenomena neglected by the system modes) affecting both the FDI and the model of damage propagation. In addition, the failure threshold is often set independently on each failure mode: however, when multiple faults interact in a complex system, the failure can happen before any individual fault mode reaches a critical value. Eventually, the computational time required for failure prognosis is usually not suitable for real-time computations.

This study attempts to address these limitations by proposing a comprehensive PHM framework to move the whole FDI and RUL prediction process on-board the vehicle. This is made possible by the use of surrogate models of the system response to faults, which are trained offline and evaluated online to speed up the computations and meet real-time constraints.

Chapter 2

Proposed methodology

The proposed methodology addresses the three phases of the general Prognostics and Health Management (PHM) problem described in Section 1.1. A combination of model-based and machine learning techniques is employed to reduce the computational burden associated with each phase of the process and speed up the estimation of the Remaining Useful Life (RUL). The purpose is to achieve a nearly real-time evaluation of the system RUL: such information can be employed to inform maintenance planning and possibly a dynamic reconfiguration of the mission, accounting for the residual capabilities of the equipment.

The block diagram of Figure 2.1 depicts the high-level structure of the proposed PHM strategy, highlighting the subdivision of each phase in an offline training and an online evaluation; the information flow between the different blocks is shown as well.

At first, information about the system behavior, both in nominal conditions and in presence of faults, is collected in a training dataset. The source of training data can be either an experimental campaign, a physics-based simulation of the equipment (calibrated and validated with respect to the response of the actual physical system), historical records of field data, or a combination of the three. Generally speaking, high-fidelity data collected from actual hardware is to be preferred, but often the amount of information required for training successfully the machine learning tools is not available; then, high fidelity data may be integrated with synthetic data computed with a simulation model. Specifically, for the proposed methodology, the following information is required:

- A set of fault combinations, sampled in the space of the considered fault parameters. The number of fault combinations, as well as the particular sampling criterion, are problem dependent. For the application discussed in this manuscript, a particular importance sampling technique is employed. A detailed description is provided and discussed in Section 2.1.
- A set of informative maps of the system behavior, associated to each of the aforementioned fault combinations. These maps shall be measurable with the available sensors installed on the system, and shall be highly dependent on the health condition of components; at the same time, dependency on operating and environmental conditions shall be reduced as possible.
- A set of Boolean values associated to each sampled fault combination, determining whether or not that specific health condition is still compliant with the requirements of the system. This information can be collected by evaluating an assessment function, as proposed in [37].

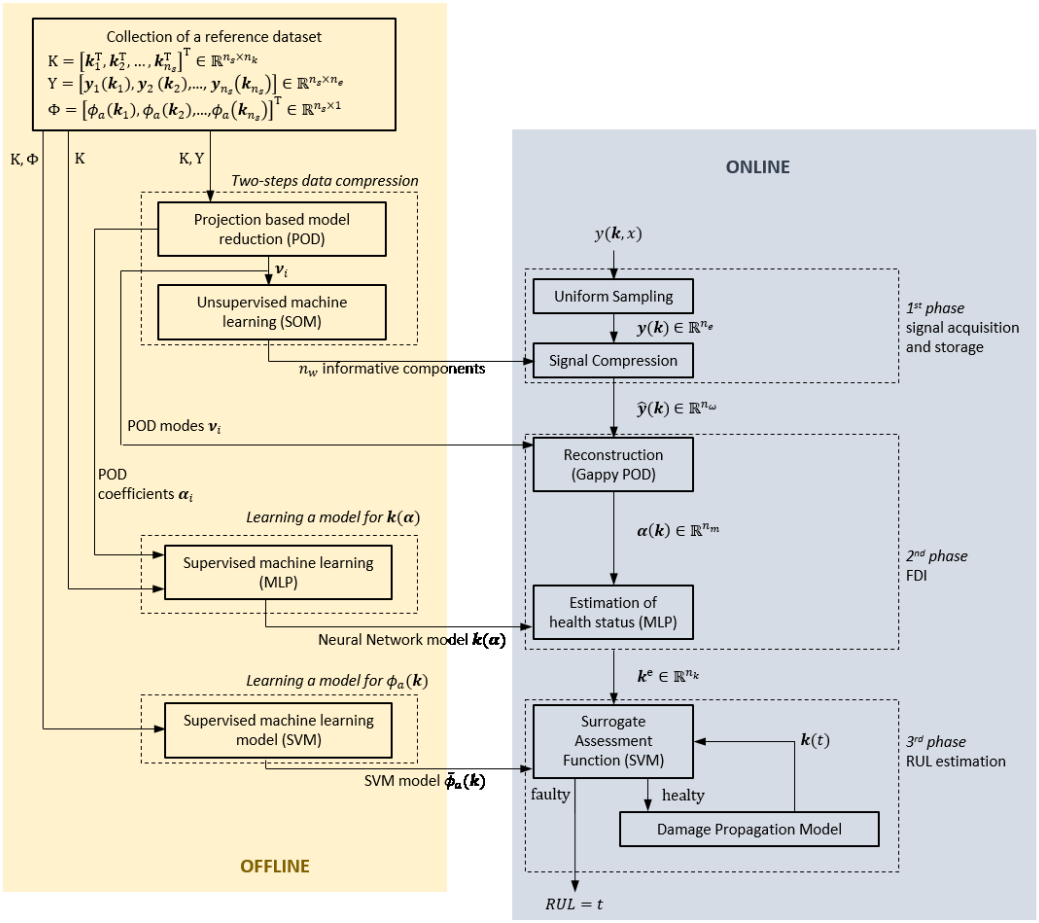

Figure 2.1: Block diagram of the proposed PHM strategy, highlighting the offline/online arrangement of each phase.

After the training dataset is collected, a compression map is determined offline to reduce the amount of data to be measured, stored and processed in real-time. A two-step signal compression, initially proposed by [42] and successfully demonstrated for structural health monitoring problems, is leveraged. The first step of compression employs Proper Orthogonal Decomposition (POD) to determine the informative principal components of the system behavior maps from the training set. Then, a Self-Organizing Map (SOM) yields a nonlinear projection of the principal components to a set of optimal locations for real-time measurement. Online, only those locations are considered for measurement of the system behavior map, thus reducing the required processing for FDI and RUL estimation. Most of the information of the complete system behavior map is reconstructed in real time via Gappy POD: as demonstrated in [29, 30], this results in a significantly improved computational time with respect to measuring the whole map.

After reconstruction via Gappy POD, the retained information from the system behavior map is encoded in a set of coefficients associated to the principal components of the measurements. A Multi-Layer Perceptron (MLP) is trained offline to associate the POD coefficients of the training set to the corresponding fault combination. Online, the POD coefficients estimated from the real-time measurements are fed to the trained MLP in evaluation: the output of the neural network constitutes an estimate for the actual health condition of the system.

The estimated health condition is employed as a starting point to determine the RUL of the system. To do so, a model for the propagation of faults is integrated with respect to time. In order to reduce the sensitivity of the RUL estimate to the uncertainty in FDI, all the last known health conditions are accounted for. A filter based on a dynamical estimator, similar to a Kalman Filter, is employed to combine the information from FDI with the known model of damage propagation, and to extrapolate the time evolution of the health condition until a failure happens [43]. The stopping criterion for the extrapolation is the assessment function for the system [37]; however, the computational cost is excessive and not suitable for real-time computation. Then, a Support Vector Machine (SVM) is trained offline as a surrogate assessment function to speed up computations online.

Several approaches available in literature, based e.g. on AutoAssociative Kernel Regression [44] or Principal Component Analysis [45] build fault detection models considering only nominal condition data. The proposed method builds the fault detection models considering data both from nominal condition systems and from faulty systems. This approach has the advantage of providing information not only about the magnitude of the deviation from a nominal state, but also about the specific fault modes detected in the equipment: this may ease the failure prognosis process, as well as the troubleshooting tasks needed to correct faulty subsystems. As a downside, a more demanding, but still feasible, data collection is required.

In this manuscript, Section 2.1 describes the particular importance sampling criterion employed for the training set; Section 2.2 details the signal acquisition and two-step compression; the FDI process employing MLPs is discussed in Section 2.3; eventually, Section 2.4 describes the RUL estimation algorithm.

2.1 Acquisition of a training dataset via particular importance sampling

The machine learning tools employed for the three phases of the PHM process require data-driven knowledge about the behavior of the equipment in nominal and off-nominal conditions. This information is collected with the acquisition of a training dataset, employed offline to train the surrogate models used through the whole PHM strategy.

Specifically, three sets of data are collected, arranged into three matrices.

- A matrix of fault combinations $\mathrm{K} = \left[\boldsymbol{k}_1^\top, \dots, \boldsymbol{k}_{n_s}^\top\right]^\top$. Each row $\boldsymbol{k}_i$ of this matrix is a n_k-dimensional vector encoding in its components the health condition of the system or, in other words, a set of parameters related to the state of the considered progressive damages. The components of k_i may be physical quantities related to the level of wear of components of the system: for example, the backlash or friction coefficient between two mechanical components, the resistance of an electric connector, the thermal conductivity of the heat sink of a circuit board. Since small, incipient faults are considered for prognostic interest, the drift of a fault combination from the nominal condition does not result immediately in a failure: the system

may still be able to meet its performance requirements for some time before failing.

- A matrix of informative behavior maps of the system $\mathrm{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_{n_s}]$. Each column vector $\boldsymbol{y}_i = \boldsymbol{y}(\boldsymbol{k}_1) = [y_1^i, \dots, y_{n_e}^i]$ encodes the behavior of the system associated to the i-th fault combination $\boldsymbol{k}_i$. The vectors $\boldsymbol{y}_i$ are given combinations of measurable operating parameters from the system, expressed as a function of another parameter $\boldsymbol{x}$: for example, $\boldsymbol{x}$ can be a time coordinate, the output position or speed of an actuator, or the external load on a control surface. The particular behavior maps chosen for an application are strongly problem dependent. The requirements for the chosen quantities are that (1) they can be derived from data available from sensors installed on the equipment and (2) they are highly sensitive to the health condition of the system, and marginally influenced by operating and environmental conditions.
- A matrix of values of a function for health condition assessment $\Phi = [\phi_{a,1}, \dots, \phi_{a,n_s}]$. The assessment function $\phi_{a,i} = \phi_a(\boldsymbol{k}_i)$ associates to the fault combination $\boldsymbol{k}$ a Boolean value expressing whether or not the considered system, affected by that fault combination, is still compliant to its requirements. Traditionally, common approaches to PHM employ a simple threshold for the individual fault parameters; however, this method is not able to consider accurately the combined effect of multiple fault modes, so a model-based approach is preferred.

The matrix K is sampled in the space of fault combinations. The sampling method used for a specific application is problem dependent. For prognostic applications, the specific interest is to capture the behavior of the system in presence of the early precursors of incipient faults. Indeed, the faults shall be identified in advance of the actual failure, early enough to plan corrective actions. Then, the training dataset shall allocate a relatively high number of samples for small faults, i.e. for fault vectors $\boldsymbol{k}$ near to the nominal condition $\boldsymbol{k}_{NC}$. For this reason, a particular importance sampling is adopted in this work.

Being the space of fault combinations relatively high dimensional, if a sampling with uniform probability distribution was to be employed, either a prohibitively large number of samples would be required, or the resultant density of samples would be very low. For example, let the fault combination be $\boldsymbol{k}$ an eight-dimensional vector (i.e. $n_k = 8$), its components normalized between 0 and 1, and let the nominal condition $\boldsymbol{k}_{NC}$ coincides with the origin of a Cartesian reference frame. The acceptable sampling space is an eight-dimensional hypercube with unit side; its (hyper)volume is $1^8 = 1$. For comparison, the volume of the eight-dimensional hypercube with half unit side is $0.5^8 \approx 3.9 \cdot 10^{-3}$, or that of the locus of points with positive coordinates whose distance from the origin is less than 1 is:

$$0.5^8 \cdot \pi \approx 0.012 \tag{2.1}$$

that is, only 1.2% of the total fault combinations have a Euclidean distance from the origin less than 1. Therefore, a uniform sampling distribution is not a viable option to achieve a good density near the nominal condition. To obtain a better

distribution for the samples, and assign more weight to health conditions close to the nominal one, we employ a standard Latin Hypercube sampling [46, 47, 48, 49] on an auxiliary set of variables, whose space is mapped to the space of the fault parameters via a nonlinear projection. At first, the auxiliary matrix J, n_s-by-n_k dimensional, is built as a standard Latin Hypercube sample. The elements of the fault matrix K are then computed as a nonlinear projection of J; at this purpose, in this work we employ the function:

$$\mathrm{K}_{ij} = \left(\mathrm{J}_{ij}\right)^{n_k} \tag{2.2}$$

It is possible to prove that a uniform distribution of the L_∞ norm of K is obtained this way. Specifically, let us consider the rows $\mathrm{J}_{i,:}$ contained in the matrix J, such that:

$$L_\infty\left(\mathrm{J}_{i,:}\right) < l \tag{2.3}$$

where $0 < l < 1$. These rows encode the coordinates of the points contained in a hypercube with side length l, in the space of the auxiliary variables. Being J a Latin Hypercube sample, its rows have a uniform the distribution; the number of points that satisfy the condition of Equation (2.3) is approximately:

$$n_s l^{n_k} \tag{2.4}$$

where n_s is the total number of points (i.e. rows of the matrix). These points are mapped to the space of the fault combinations $\boldsymbol{k}$ that are contained in the hypercube with side length l^{n_k}. Then, the corresponding rows of K are subject to the condition:

$$L_\infty\left(\mathrm{K}_{i,:}\right) < l^{n_k} \tag{2.5}$$

that is, for any given positive scalar a, the number of points such that $L_\infty\left(\mathrm{K}_{i,:}\right) < a$ is proportional to a, thus achieving a uniform distribution in the L_∞ of the sampling points, that is, an increasing density in the individual coordinates towards the origin. A graphical interpretation of the behavior of this sampling technique is shown in Figure 2.2, where a standard Latin Hypercube is compared to the proposed importance sampling method.

After the matrix K is determined, Y and Φ are obtained using physics-based models of the system, and evaluating those models for each fault combination $\boldsymbol{k}_i$.

2.2 Signal Acquisition, Feature extraction and Compression

The system behavior maps $\boldsymbol{y}(\boldsymbol{x})$ contained in the training dataset are usually quite high-dimensional. Considering for example the application presented in this work, if employing the back-EMF coefficient of the motor as the map $\boldsymbol{y}$, expressed as a function of the rotor angle $\boldsymbol{x}$, to identify electrical faults, the voltages, currents, rotor angle and speed shall be measured and acquired with a high frequency (at least one order of magnitude above the commutation frequency of the motor, so typically in in excess of 10kHz).

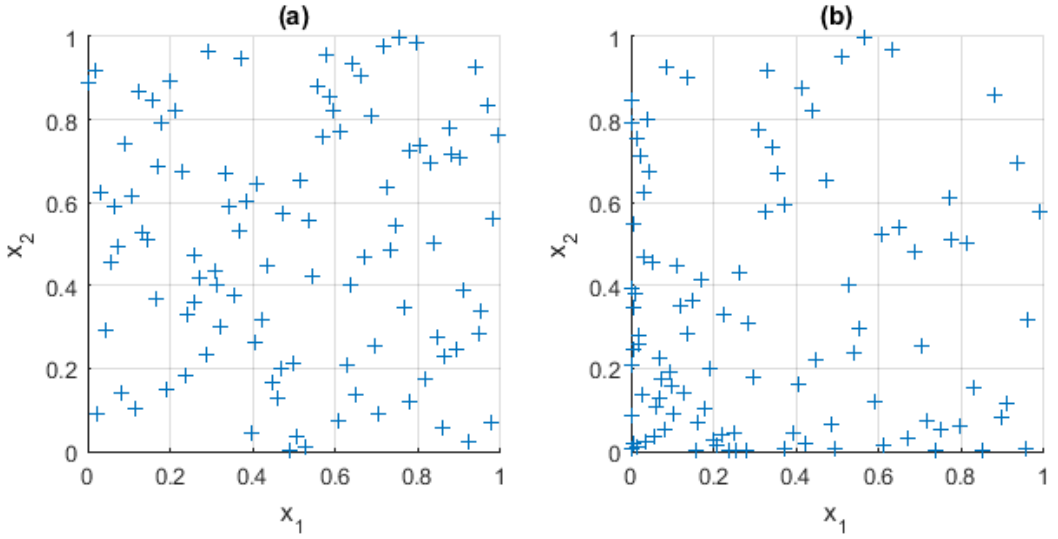

Figure 2.2: Behavior of the proposed importance sampling in two dimensions. (a) standard Latin Hypercube sampling; (b) importance sampling. The proposed importance sampling increases the density of data points near the origin.

This results in large amounts of data to store and process in real-time: assuming to allocate 16 bits for each signal, this approach generates a data stream in excess of 1Mbps. The system behavior map reconstructed from this data is, in turn, high-dimensional, and encoded in an n_e-dimensional vector $\boldsymbol{y}$. To process this data stream in real-time without some kind of compression would require substantial on-board hardware resources, that are prohibitive for health monitoring purposes.

Hence, an efficient method to reduce the amount of data is required. In this work, a two steps signal compression is employed, combining Proper Orthogonal Decomposition (POD) with Self-Organizing Maps (SOMs) to determine offline an optimal set of $n_w \ll n_e$ informative locations in which the system behavior maps are measured and processed. The method was initially developed by Mainini and Willcox [42, 50], and verified for structural health monitoring applications. Berri et al. demonstrated the application to actuator diagnostics in [29, 51].

Specifically, offline a set of informative locations (also referred to as a compression mask) is determined for measurement and processing of the system behavior map. Online, this compression mask is employed to speed up real-time computations; most of the information of the whole system map is recovered via Gappy POD.

2.2.1 Offline

In the offline step of the compression, the aim is to determine an efficient compression mask for the system output $\boldsymbol{y}(\boldsymbol{x})$, in the form of a set of informative locations in $\boldsymbol{x}$ where the observation $\boldsymbol{y}$ is measured, stored and processed. The method developed in [51] is employed. The compression strategy includes two steps: (1) at first, Proper Orthogonal Decomposition (POD) identifies a linear projection of Y to the space of its principal modes; (2) then, a Self-Organizing Map performs a nonlinear projection of the dominant POD modes to the space of the

weight vectors of the map. As a result, this procedure allows embedding physics-based knowledge in the signal processing, and obtaining an optimal compression mask to retain the significant information about the system. This dramatically reduces the amount of data to be processed in real-time.

The following Paragraphs describe respectively the first step of compression employing POD, and the SOM leveraged to obtain the optimal compression mask.

2.2.2 First step of compression: Proper Orthogonal Decomposition (POD)

This phase of the PHM process aims at computing a reduced representation of the system behavior map, identifying its most informative components. The input of this step is the matrix of informative behavior maps of the system $\mathrm{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_{n_s}]$, obtained in Section 2.1. The output is represented by the POD modes, that is, the principal directions of variation of the system behavior $\boldsymbol{y}$.

Proper Orthogonal Decomposition (POD) [52, 53] is a numerical procedure closely related to Principal Component Analysis (PCA) [45] and commonly employed to find compressed representation and reduced models, by identifying underlying structures of possibly correlated data. While PCA is often employed in statistics for finitely dimensional data, POD is commonly used in several fields of engineering, such as: fluid-dynamics, structural mechanics, and signal processing. Its formulation, based on the Karhunen-Loeve expansion, guarantees that the compressed representation of the data is optimal in the least squares sense and retains the structure of the underlying physical phenomena [54]. For example, in [55] POD is employed to obtain surrogates of nonlinear dynamical systems for model-based control. Walton et al. [56] combine POD and radial basis functions to obtain reduced order models of unsteady fluid flows. Willcox and Peraire [57] employ Proper Orthogonal Decomposition to perform a balanced reduction of high-order linear systems. In [58], POD is used to obtain a dynamical characterization and order reduction of linear and nonlinear dynamical systems.

POD is applied to the matrix Y with the method of Snapshots [54]. The observed system outputs $\boldsymbol{y}$ (snapshots) are arranged into the columns of $\mathrm{Y} \in \mathbb{R}^{n_e \times n_s}$, and Singular Value Decomposition (SVD) is leveraged to find two orthogonal matrices $\mathrm{U} \in \mathbb{R}^{n_e \times n_e}$ and $\mathrm{V} \in \mathbb{R}^{n_s \times n_s}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{n_e \times n_s}$ such that:

$$\mathrm{Y} = \mathrm{U}\Sigma\mathrm{V}^{\top} \tag{2.6}$$

An important property of SVD states that the first n_s columns of $\mathrm{U} = [\boldsymbol{v}_1 \dots \boldsymbol{v}_{n_s}]$ constitute an orthonormal basis for the columns of Y. As a result, a generic observation of the system $\boldsymbol{y}_i$ can be written as the linear superposition of the modes $\boldsymbol{v}$, weighted by the coefficients α:

$$\boldsymbol{y}_i = \boldsymbol{y}_0 + \sum_{j=1}^{n_s} \alpha_{i,j}\, \boldsymbol{v}_j \tag{2.7}$$

where $\boldsymbol{y}_0$ is a baseline observation, e.g. the average of all the individual observations. The POD modes $\boldsymbol{v}_j$ are optimal in a least squares sense, and are

ordered according to the associated eigenvalues λ_j. In a geometrical interpretation of POD, the observations $\boldsymbol{y}$ are represented as data points in the n_e-dimensional space; the POD modes are the principal directions of variation of the observation, i.e. the principal axes of an n_e-dimensional ellipsoid fitted over the distribution of the data points. The length of each axis is the eigenvalue associated to that mode, and provides a measure of the dispersion of the observations in that direction. The modes associated to the largest eigenvalues then explain most of the variability of the data; as a result, a compressed representation of the observations y can be achieved by truncating the POD expansion of Equation (2.7) to the first n_m modes:

$$\boldsymbol{y}_i \approx \boldsymbol{y}_0 + \sum_{j=1}^{n_m} \alpha_{i,j}\, \boldsymbol{v}_j \tag{2.8}$$

with $n_m \ll n_s$. The fraction of information of the original observation set retained by this representation is related to the cumulative sum σ of the eigenvalues associated to the retained modes:

$$\sigma = \frac{\sum_{j=1}^{n_m} \lambda_j}{\sum_{j=1}^{n_s} \lambda_j} \tag{2.9}$$

If the eigenvalues decay fast, i.e. if σ is already close to 100% for a small n_m, the compression provided by Equation (2.8) retains most of the original information within a limited set of parameters (i.e. the POD coefficients $\alpha_{i,j}$). Indeed, the set of modes is fixed for a given system, while the information associated to an individual observation is stored in the POD coefficients.

The output of this step includes the POD modes and coefficients computed for the matrix Y, to be employed by the following phases of the PHM process.

2.2.3 Second step of compression: Self-Organizing Map (SOM)

The first step of compression through POD allows to find a compressed representation of the observations to reduce the amount of data to process for FDI. A second step of compression leverages unsupervised machine learning to determine an optimal compression mask to reduce the amount of data to be measured and stored online. The input of this phase are the first n_m POD modes computed in Section 2.2.1.1.

Self-Organizing Maps (SOMs) were originally introduced by Kohonen [59, 60]; they are a class of single-layer neural networks that exploit unsupervised learning to identify clusters of self-similar data. Kaski [61] discusses the use of SOMs to find structure in large multidimensional datasets, with applications in engineering, statistics and data mining. In [62], SOMs are employed to extract interpretable patterns from satellite imagery. Svensson et al. [63] leverage SOMs to diagnose cooling system faults in a fleet of vehicles, by obtaining low dimensional representations of sensor measurements. Kohonen maps are employed with agglomerative hierarchical algorithms to detect failures of induction motors in [64].

A similar approach is tested for diagnostics of industrial automation equipment in [65].

Each of the n_w neurons of the SOM has two representations. One lies in the space of the weight vectors $\boldsymbol{w}$ (that is equivalent to the space of the input): this representation is updated iteratively during learning. The second representation is in the usually bi-dimensional topological space of the network, and is fixed.

The SOM exploits a form of unsupervised competitive learning. At first, the weight vectors are initialized to random values; during training, the examples $\boldsymbol{\tau}$ are fed one by one to the network. For each example $\boldsymbol{\tau}_i$, a winner neuron l is chosen as the one that minimizes the distance between its weight vector $\boldsymbol{w}_l$ and the current training point $\boldsymbol{\tau}_i$:

$$l = \arg\min_j(\|\tau_i - w_j\|) \tag{2.10}$$

For most applications, including the one considered in this work, $\|\cdot\|$ denotes an L_2 norm, that is the Euclidean distance between the weight vector and the training example. The weight vector of the winner neuron l and its neighbors are then updated according to the following equation:

$$w_j^{i+1} = w_j^i + \theta(j,l)\alpha(i)(\tau_i - w_j) \tag{2.11}$$

where $\theta(j,l)$ is a neighborhood function, $\alpha(i)$ is a monotonically decreasing learning coefficient, i denotes the iteration of the learning process, and j is the particular neuron being updated. The neighborhood function is typically a decreasing function of the distance from the winner neuron l, defined in the space of the topological representations of neurons. As a result, at each iteration, the weight vectors of the winner neuron and its neighbors are moved closer to the training point. This process is repeated for each input of the training set for several epochs. During each epoch, all the training points are submitted to the network, each time in a different, randomized order. The training is complete when one of the stopping criteria is met: for example, when a performance parameter reaches its goal value, or when the maximum number of epochs is reached.

For the application discussed in this work, the training set T is composed by the first n_m modes of the POD $\boldsymbol{v}_i$ and the associated coordinate $\boldsymbol{x}$**:**

$$\mathrm{T} = [\boldsymbol{x}, \boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_m}] \tag{2.12}$$

Each training point is a row of T, including a coordinate x_i and the corresponding values of the POD modes. A property of SOMs implies that, after training, the weight vectors represent a nonlinear projection of the high-dimensional training data to the lower dimensional space of the neurons [66]. A consequence is that the weight vectors, defined in the space of the input, encode representative vectors for clusters of self-similar points. Then, an efficient compression mask can be obtained from the components of the weight vectors associated to the coordinate $\boldsymbol{x}$: the system behavior maps $\boldsymbol{y}(\boldsymbol{x})$ will be measured and processed online only in correspondence of those n_w informative values $\hat{\boldsymbol{x}}$ of the coordinate $\boldsymbol{x}$, to obtain a compressed representation $\hat{\boldsymbol{y}}$.

2.2.4 Online

The compression mask $\hat{\boldsymbol{x}}$ computed offline is employed to reduce the computational effort needed in real-time. During the operation of the monitored system, the sensors installed onboard capture a constant stream of data, whose frequency depends on the primary application of the individual sensor: indeed, data used for health monitoring purposes is preferably measured by sensors installed for a primary application different from prognostics. This way, virtually no weight or complexity is added to the onboard system, and its performances and reliability are not decreased. Data measured onboard is processed to obtain the system behavior map $\boldsymbol{y}$. The reconstruction of the complete map $\boldsymbol{y}(\boldsymbol{x})$ would require substantial processing power, since the acquisition frequency of the sensors may be in the kilohertz range or above. Then, the compression mask allows to process only the n_w points whose coordinate $\hat{\boldsymbol{x}}$ have been determined offline.

Those points $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ constitute an incomplete measurement of the behavior map of the system. The efficient placement of the measurement points through the two-step compression guarantees that a large fraction of the information contained in the complete map is retained. The map could be fed directly to a regression neural network, as in [67]; however, random errors on the measurements could result in a biased fault detection. In this work, Gappy Proper Orthogonal Decomposition (Gappy POD) is employed to reconstruct an estimate of the POD coefficients associated to the incomplete map measured online.

Gappy POD [68, 69, 70] is a procedure derived from POD, commonly employed to reconstruct data from sparse measurements, leveraging physics-based knowledge of the structures of data through the use of the POD modes. Saini et al. [71] discuss the use of Gappy POD for data recovery from the noisy particle image velocimetry measurements in combustors of gas turbines. Willcox [72] employs a similar technique to reconstruct unsteady fluid flows from incomplete measurements. Bui Thanh et al. [73] employ gappy proper orthogonal decomposition for efficient inverse airfoil design. In [74], stabilized reducer order models of nonlinear eddy currents are obtained with Gappy POD.

For the implementation studied in this work, the goal is to reconstruct the POD coefficients associated to the incomplete measurement, as opposed to the complete observation. According to Equation (2.8), the quantity $\boldsymbol{y}$ measured online can be approximated by a linear superposition of the first n_m POD modes. The objective of Gappy POD is to find a set of coefficients $\boldsymbol{\alpha}$ that minimize the mean squared error between the original signal and the reconstructed one, limited to the n_w known components $\hat{\boldsymbol{y}}$. This is done by solving the linear system:

$$\mathrm{G}\boldsymbol{\alpha} = \boldsymbol{f} \tag{2.13}$$

where $\mathrm{G} = \hat{v}^{\mathrm{T}}\hat{v}$ is the Gappy Matrix, and $\hat{v} = [\hat{\boldsymbol{v}}_1, \dots, \hat{\boldsymbol{v}}_{n_m}]$ is a n_w-by-n_m dimensional matrix whose columns are the first n_m POD modes, considered only in their n_w informative elements corresponding to the coordinates $\hat{\boldsymbol{x}}$ of the compression mask. The vector $\boldsymbol{f}$ is obtained by projecting the compressed signal $\hat{\boldsymbol{y}}$ along the compressed POD modes $\hat{v}$:

$$\boldsymbol{f} = \hat{v}^{\top}\hat{\boldsymbol{y}} \tag{2.14}$$

In the traditional application of Gappy POD, after solving the system of Equation (2.13), an approximation of the complete measurement y is reconstructed as the linear superposition of the first n_m POD modes, weighted by the coefficients $\boldsymbol{\alpha}$ as per Equation (2.8). The specific implementation of the proposed methodology requires directly the use of the POD coefficients as a compressed representation of the high-dimensional observation $\boldsymbol{y}$ for the Fault Detection and Identification phase.

2.3 Fault Detection and Identification (FDI)

This step of the proposed PHM procedure receives as an input a compressed representation of the system behavior map $\boldsymbol{y}$ in the form of the POD coefficients α, to return in output the estimated health condition $\boldsymbol{k}$ of the equipment.

Common methods for Fault Detection and Identification (FDI) are either model-based or data-driven. The former category relies on physics-based digital twins of the monitored system: the measurements from the monitored equipment are compared to the output of the digital twin, and a parameter identification algorithm is employed to match the response of the two systems and estimate the actual health condition. This procedure can be highly accurate [75], but requires a computational effort that is usually not suitable for real-time execution. The latter category relies on machine learning tools to associate a measured output to a given fault condition. In this case, computational time is fast, but accuracy is not suitable to determine small, incipient faults [76], especially if the response of the system is influenced by several unpredictable variables.

The proposed methodology for FDI is essentially data-driven; however, thanks to the efficient data compression, and with a suitable choice of the monitored parameters, it permits to combine good accuracy and fast computational time in evaluation. The methodology employs supervised machine learning to associate the estimated POD coefficients $\boldsymbol{\alpha}$ to the health condition of the system $\boldsymbol{k}$. Specifically, a feedforward neural network is trained offline (with the fault conditions of the dataset K and the associated POD coefficients computed offline as per Section 2.2.1) and evaluated online on the POD coefficients estimated by Gappy POD (Section 2.2.2).

2.3.1 Offline

A neural network is trained offline to associate an estimate of the fault condition $\boldsymbol{k}$ to a set of POD coefficients $\boldsymbol{\alpha}$. The use of a machine learning approach instead of a model-based one allows to keep the computational time in evaluation low and compatible with real-time constraints. In this application, a standard implementation of a Multi-Layer Perceptron (MLP) with a single hidden layer demonstrated to be suitable for the task. Depending on the specific problem, more complex machine learning paradigms may be required.

Multi-Layer Perceptrons [77, 78] are a class of supervised learning, feedforward neural networks; their property of being universal approximators

makes them suitable for solving regression and classification problems. In [79] MLPs are employed to estimate fuel consumption of road vehicles accounting for variations of the operating condition. Li et al. [80] compare MLP networks and deterministic regression methods for the estimation of wind turbine power curves. In [81] MLPs and Support Vector Machines (SVMs) are combined to detect early faults of gearboxes from vibration measurements.

The architecture of an MLP features neurons arranged in one or more hidden layers, followed by an output layer. MLPs are fully connected networks, meaning that each neuron receives as an input the outputs of all the neurons of the previous layer, and its output is fed to all neurons of the next layer.

The implementation adopted in this work has a single hidden layer with n_h sigmoid activation function, and an output layer with linear saturated activation function. The number of neurons in the output layer is determined by the problem, and is equal to the dimension of the output: then, for this work there are n_k output neurons, being the output of the MLP an estimate of the fault vector. The training set is composed by the fault conditions of the training set K as the targets, and the associated POD coefficients $\boldsymbol{\alpha}$ computed in Section 2.2.1 as the inputs.

The inputs $\boldsymbol{\alpha}$ of the hidden layer, arranged as a column vector, are multiplied by the weight vectors $\boldsymbol{W}^h$ of the neurons (row vectors), added to a bias constant b^h, and then fed to a hyperbolic tangent activation function to obtain the output a of the neuron. For the i-th neuron:

$$a_i = \tanh(\boldsymbol{W}_i^h \boldsymbol{\alpha} + b_i^h) \tag{2.15}$$

Figure 2.3 (a) shows the block diagram of a sigmoid neuron. The output layer receives in input the outputs $\boldsymbol{a} = \left[a_1 \dots a_{n_h}\right]^\top$ of the hidden layer; they are weighted by the vectors $\boldsymbol{W}^o$, added to the bias b^o and fed to the linear saturation activation function to compute the output of the MLP $\boldsymbol{k} = [k_1 \dots k_{n_k}]$. For the i-th neuron:

$$k_i = \begin{cases} 0, & \text{if } \boldsymbol{W}^o \boldsymbol{a} + b^o < 0 \\ \boldsymbol{W}^o \boldsymbol{a} + b^o, & \text{if } 0 \le \boldsymbol{W}^o \boldsymbol{a} + b^o \le 1 \\ 1, & \text{if } \boldsymbol{W}^o \boldsymbol{a} + b^o > 1 \end{cases} \tag{2.16}$$

The block diagram of a linear saturated neuron is shown in Figure 2.3 (b), while Figure 2.4 shows the overall arrangement of the MLP. The particular choice of the output activation function reflects the bounds of the output of the FDI problem: the components of the fault vector $\boldsymbol{k}$ are normalized between 0 and 1. Additionally, the sharp transition of the output activation function permits to cut to zero the estimated fault condition when the dynamical response of the system is close to the nominal one, reducing the risk of false positive fault detections.

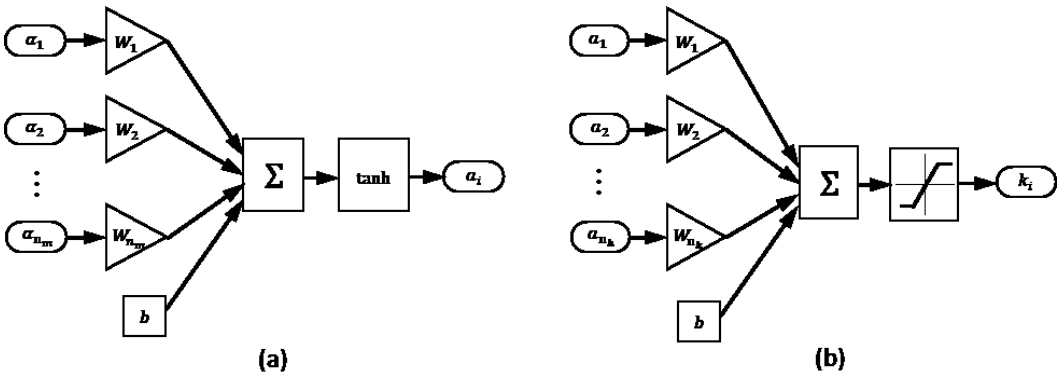

Figure 2.3: Block diagram of a sigmoid neuron (a) and linear saturated neuron (b)

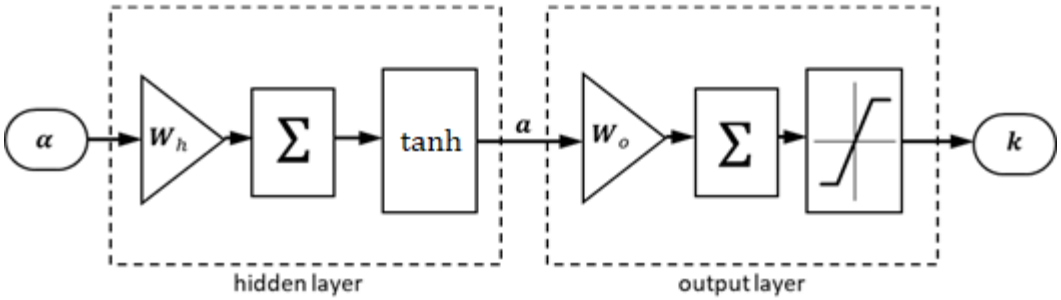

Figure 2.4: Arrangement of the single hidden layer Multi-Layer Perceptron (MLP) employed for FDI

The weights and biases of the neurons are determined during training, to minimize a performance function in consisting in the mean squared error between the outputs of the networks and the targets, leveraging a Levenberg-Marquardt backpropagation algorithm [82]. The Levenberg-Marquardt algorithm is an optimization algorithm meant to approach the second-order convergence speed while avoiding the direct computation of the Hessian matrix of the problem. We define a vector $\boldsymbol{x}$ containing in its elements the weight and bias parameters of each neuron of the network; $\boldsymbol{k}^t$ is the target, i.e. the expected output of the network for the training set, while $\boldsymbol{k}^e$ is the obtained output. Being the performance function a sum of squares, the Hessian matrix H of the errors with respect to the weight and biases can be approximated as:

$$\mathrm{H} \approx \mathrm{J}^\top \mathrm{J} \tag{2.17}$$

and the gradient of the performance function is:

$$\boldsymbol{g} = \mathrm{J}^\top \boldsymbol{e} \tag{2.18}$$

where $\boldsymbol{e} = \boldsymbol{k}^t - \boldsymbol{k}^e$ and J is the Jacobian matrix, which can be evaluated with a standard backpropagation method [83]. This approach is less expensive than the direct computation of the Hessian. At each iteration, the algorithm uses the approximated Hessian to update $\boldsymbol{x}$ in a quasi-Newton form:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - [\mathrm{H} + \mu \mathrm{I}]^{-1} \boldsymbol{g} = \boldsymbol{x}_i - [\mathrm{J}^\top \mathrm{J} + \mu \mathrm{I}]^{-1} \mathrm{J}^\top \boldsymbol{e} \tag{2.19}$$

where the scalar parameter μ is decreased after each successful iteration (i.e. after a reduction of the performance function). The training is stopped when either the maximum number of epochs is reached or the performance gradient decreases below a threshold. The result of the offline step of the FDI phase is the trained MLP model for $\boldsymbol{k}(\boldsymbol{\alpha})$, to be employed online for real-time fault detection. The use of the low dimensional representation of the system's response provided by the POD coefficients $\boldsymbol{\alpha}$ in place of the full dimensional $\boldsymbol{y}$ reduces the number of neurons needed in the hidden layer and simplifies the training process. In addition, $\boldsymbol{\alpha}$ has an advantage over the compressed signal $\hat{\boldsymbol{y}}$, as the Gappy POD step inherently checks the plausibility of the observations by comparison with the information contained in the POD modes, and mitigates the effect of local measurement errors.

2.3.2 Online

The MLP model trained offline is employed to speed up real-time FDI. The network receives as an input a new set of POD coefficients, associated with the map of system behavior observed and compressed online with the optimal compression mask $\hat{\boldsymbol{x}}$. The POD coefficients are estimated via Gappy POD, with the procedure described in Section 2.2.2.

The estimate of the current health condition of the system $\boldsymbol{k}^e$ is computed as the output of the Multi-Layer Perceptron, employing Equations (2.15) and (2.16). This strategy allows to perform the FDI task in a few milliseconds, much faster than the several minutes required by model-based fault detection techniques [84]. Additionally, the combination of supervised machine learning with the two-step compression, which allows embedding physics-based domain knowledge into the compressed representation of the system observations, permits to retain a good accuracy of the estimate.

The output of the online FDI is the real-time estimate of the system health $\boldsymbol{k}^e$, which will be employed as a starting condition for the estimation of Remaining Useful Life.

2.4 Estimation of Remaining Useful Life (RUL)

The last phase of the PHM process is the actual estimation of the Remaining Useful Life. The input for the process is the current health condition determined by FDI, employed as a starting point for the evaluation of a model of damage propagation.

The approach discussed in this work is the extension to system health monitoring of the damage tolerant design strategy for fatigue of structures. In the field of structural health monitoring, the components are inspected periodically in search of cracks. Since the rate of propagation of cracks in metal and composite structure is known and well described by physics based models, the next inspection is planned before the existing cracks reach a critical length; if no cracks are detected during the inspection, they are assumed to be just below the sensitivity of the employed equipment.

The extension of this approach to systems poses two issues:

- The higher complexity of the monitored equipment makes accurate inspection impractical in periodical maintenance; then, this study proposes to replace, at least in part, manual inspections with the automatic, real-time FDI process described in Section 2.3.
- The heterogeneous disciplines that rule the propagation of faults results in the difficulty (or often in the impossibility) to determine an accurate physics-based model for damage propagation. To address this obstacle, the proposed methodology uses an adaptive model for damage propagation, which is updated in real-time according to the observed time-history of the health condition.

Leveraging the definition of Remaining Useful Life of Equation (1.2), a model of damage propagation is employed in the form of a state-space dynamical representation. The model is integrated numerically, starting from the current fault combination determined by FDI as the initial condition, and accounting for the entire known time history of faults through a dynamic estimator filter. The model is adaptive and leverages a simple system identification algorithm to tune itself to match the observations. A function for the assessment of health condition [29, 37] is employed as a stopping criterion for the integration. When the estimated health condition reaches a value that is no more compatible with the requirements of the system, the corresponding integration timestep is assumed as the RUL estimate: indeed, it is the remaining time after which the equipment will no longer meet the required performances. The function for assessment of the health condition is physics-based and implies a significant computational burden. To enable real-time evaluation, a Support Vector Machine (SVM) is trained offline as a surrogate assessment function.

2.4.1 Offline

The function for health condition assessment $\phi_a(\boldsymbol{k})$ behaves as a binary classifier: it simulates the response of the system under the effect of the fault combination $\boldsymbol{k}$ and determines whether or not the applicable performance requirements are met by the equipment, assigning to $\boldsymbol{k}$ a binary output in the form of a "healthy" or "faulty" label. For some simple application, the simple comparison of the fault vector with a threshold may be enough; however, this is usually not acceptable to deal with the combined effects of multiple fault modes affecting the equipment at the same time. More complex assessment function quickly become impractical to evaluate in real-time. For example, a viable option for the health assessment of an actuator is to evaluate its transfer function with an iterative simulation at variable frequency of the command: this results in computational times of several seconds or more. To enable real-time evaluation of the assessment function, this study proposes the use of a surrogate function in the form of a Support Vector Machine (SVM).

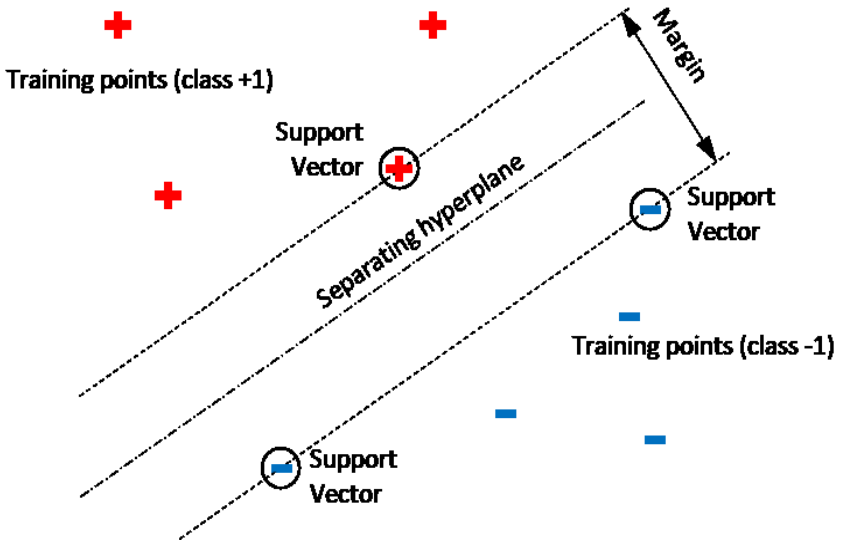

Figure 2.5: Geometrical interpretation of a linear Support Vector Machine binary classifier.

SVMs [85, 86] are algorithms that leverage supervised machine learning to perform an efficient classification of the input. Goh et al. [87] discuss the use of SVM for image classification. In [88], Leng et al. propose a binary tree classifier that employs SVMs to perform large scale classification of data. In [89], a similar method for monitoring fatigue damage of airframes is proposed. Fan et al. [90] employ SVMs to obtain efficient surrogate modelling of fluid-dynamics problems.

To train a surrogate assessment function, we assemble a training set with the matrices $\mathrm{K} = \left[\boldsymbol{k}_1^\top, \dots, \boldsymbol{k}_{n_s}^\top\right]^\top$ and $\Phi = \left[\phi_1, \dots, \phi_{n_s}\right]$ discussed in Section 2.1. In the standard linear formulation, given a set of training points $\boldsymbol{k}_i$, each defined in $\mathbb{R}^{n_k}$, and their classes $\phi_i = \pm 1$, the SVM seeks an optimal hyperplane in $\mathbb{R}^{n_k}$ to separate the two classes. The equation of a generic hyperplane in $\mathbb{R}^{n_k}$ is:

$$f(\boldsymbol{k}) = \boldsymbol{k}^\top \boldsymbol{\beta} + b = 0 \tag{2.20}$$

where $f(\boldsymbol{k})$ is a cost function, $\boldsymbol{\beta}$ has the same dimensionality as $\boldsymbol{k}$ and b is a scalar bias. The goal of the training process for the SVM is to find the best separating hyperplane, that is, the one that results in the largest margin between the two classes $\phi = \pm 1$. A geometrical interpretation of a linear SVM is provided in Figure 2.5.

The vector $\boldsymbol{\beta}$ is normal to the hyperplane, and the optimal margin length between the two classes is $2/\|\boldsymbol{\beta}\|$. Then, finding the best separating plane is equivalent to find the gradient $\boldsymbol{\beta}$ and bias b of the hyperplane that minimize $\|\boldsymbol{\beta}\|$, subject to the set of constraints $\phi_i f(\boldsymbol{k}_i) \geq 1$. The training algorithm solves this optimization as a quadratic programming problem, using the method of Lagrange multipliers, introducing the positive coefficients $c_1 \dots c_n$, and resulting in the objective function:

$$L_P = \frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{\beta} - \sum_i c_i(\phi_i f(\boldsymbol{k}_i) - 1)\, \boldsymbol{g} = \mathrm{J}^\top \boldsymbol{e} \tag{2.21}$$

which is equivalent to the dual problem:

$$\min_i \left(\frac{1}{2} \sum_i \sum_j c_i c_j \phi_i \phi_j k_i^\top k_j - \sum_i c_i \right) \boldsymbol{g} = \mathrm{J}^\top \boldsymbol{e} \tag{2.22}$$

Our application does not allow to separate the classes of the training dataset with a linear boundary. In these cases, the problem is addressed by introducing a nonlinear kernel to map the input vectors $\boldsymbol{k}$ to an auxiliary space where a linear boundary exists. The dual formulation is modified as follows:

$$\begin{gathered} \min_i \left(\frac{1}{2} \sum_i \sum_j c_i c_j \phi_i \phi_j \mathrm{G}_{ij} - \sum_i c_i \right) \\ \text{s.t.} \sum_i c_i \phi_i = 0 \\ c_i \geq 0 \end{gathered} \tag{2.23}$$

and subject to the Karush-Kuhn-Tucker complementarity conditions [91, 92]. G is the Gram matrix of the predictor vectors $k_1 \ldots k_{n_s}$ using the nonlinear kernel function:

$$\mathrm{G}_{ij} = \langle \psi(\boldsymbol{k}_i), \psi(\boldsymbol{k}_j) \rangle \tag{2.24}$$

where $\langle \cdot \rangle$ denotes an inner product. After the training, new input points are classified according to the sign of the score function:

$$\hat{\phi}_a(\boldsymbol{k}) = \mathrm{sign} \left(\sum_i c_i \phi_i \langle \psi(\boldsymbol{k}), \psi(\boldsymbol{k}_i) \rangle + b \right) \tag{2.25}$$

where c_i are the Lagrange multipliers optimized during training. The function $\hat{\phi}_a(\boldsymbol{k})$ is the surrogate function for the assessment of the health condition $\boldsymbol{k}$, to be employed in the online RUL estimation procedure.

2.4.2 Online

Remaining Useful Life is estimated online by integrating a model of damage propagation in the form of a state-space dynamical model, which expresses the evolution in time of the health condition of the system, employing the surrogate assessment function $\hat{\phi}_a(\boldsymbol{k})$ as a stopping criterion. A block diagram of the online procedure is provided in Figure 2.6.

The integration starts at time $t_0 = 0$, corresponding to the oldest known health condition k_0 measured by the first FDI. The integration from t_0 to the current time t_{now} (that is, the time coordinate associated to the last FDI) accounts for the known time history of the fault vector $\boldsymbol{k}(t)$ in order to filter out uncertainties in fault detection and tune the model of damage propagation. Indeed, one of the most important issues in system prognostics is that accurate physics-based descriptions of the fault growth rate are not commonly available.

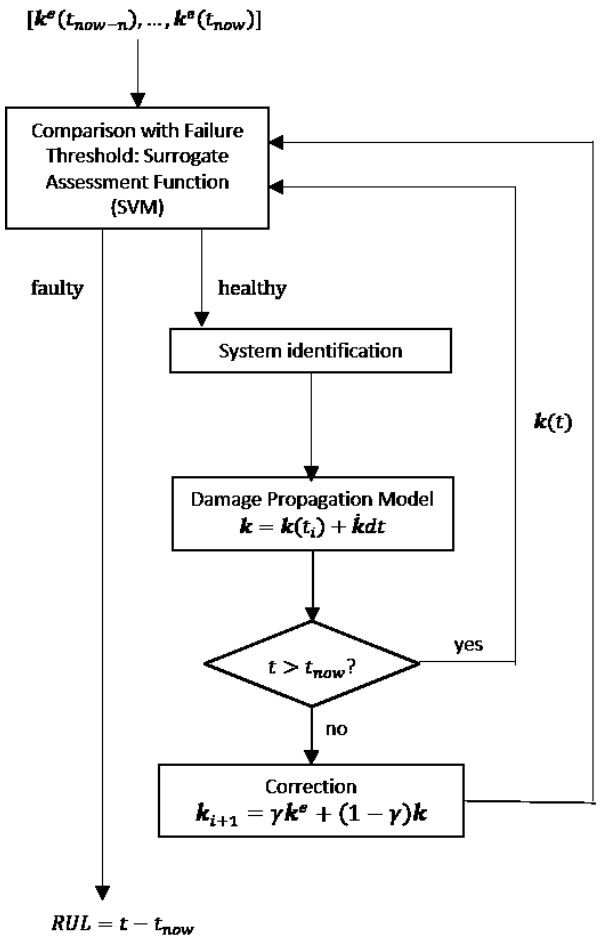

Figure 2.6: Schematic flow chart of the proposed RUL estimation real-time procedure

The proposed approach addresses this criticality by dynamically adapting a state space model to the obsevations. The general formulation of a state-space model is:

$$\begin{cases} \dot{\boldsymbol{k}} = \mathrm{A}\boldsymbol{k} + \mathrm{B}\boldsymbol{u} \\ \boldsymbol{j} = \mathrm{C}\boldsymbol{k} + \mathrm{D}\boldsymbol{u} \end{cases} \tag{2.26}$$

Where A is the state matrix, B is the control matrix, C is the output matrix, D is the feedthrough matrix, $\boldsymbol{k}$ is the state, $\boldsymbol{j}$ is the observation, and $\boldsymbol{u}$ is the input. For the application to RUL estimate addressed in this study, we can set $\boldsymbol{k} = \left[k_1 \dots k_{n_k}\right]^\top$ as the health condition of the system, and $\boldsymbol{u} = \left[u_1 \dots u_{n_u}\right]^\top$ as the environmental and operating conditions; the observation $\boldsymbol{j}$ can be considered equal to the state $\boldsymbol{k}$ (that is, the system health condition): then, we can neglect the second equation.

The state and control matrices may be derived from physics-based knowledge of the system. However, in this case a large uncertainty is usually associated to the state-space model. Therefore, the matrices shall be estimated from observed data, allowing a more precise prediction of the evolution of faults. The elements of A and B can be computed from the equation:

$$\kappa_i a_i = \dot{\kappa}_i \quad \text{for } i = 1 \dots n_k \tag{2.27}$$

where κ is a matrix containing the observed states k and inputs u of the last n timesteps:

$$\kappa_i = \begin{bmatrix} k_1(t_{now-n}) & \cdots & k_{n_k}(t_{now-n}) & u_1(t_{now-n}) & \cdots & u_{n_u}(t_{now-n}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k_1(t_{now}) & \cdots & k_{n_k}(t_{now}) & u_1(t_{now}) & \cdots & u_{n_u}(t_{now}) \end{bmatrix} \quad (2.28)$$

a_i is a column vector with the elements of the i-th row of A and the i-th row of B:

$$a_i = \left[A_{i,1} \dots A_{i,n_k}, B_{i,1} \dots B_{i,n_u}\right]^\top \quad (2.29)$$

and $\dot{\kappa}_i$ is a column vector containing the time history of the i-th element of the state derivative $\dot{\boldsymbol{k}}$:

$$\dot{\kappa}_i = \left[\dot{k}_i(t_{now-n}) \dots \dot{k}_i(t_{now})\right]^\top \quad (2.30)$$

At each integration timestep, the matrices A and B are computed by solving the linear systems of Equation (2.27). The number of timesteps n considered for the estimation shall be high enough to filter out uncertainties in the observations (i.e. the errors of the FDI process), while remaining low enough to allow real-time evaluation. In any case, n shall be larger than $n_k + n_u$, i.e. the sum of the number of elements of the state and control vectors: otherwise the system is under determined; a number of timesteps larger than $n_k + n_u$ is allowed since Equation (2.27) can be solved in the least squares sense.

After the state and control matrices A and B are identified, the state-space model is employed for two purposes: as dynamical estimator filter for the computation of the next fault condition, and as a predictor to extrapolate the future time evolution of the fault condition to determine the system RUL.

The fault condition at the next timestep is estimated by fusing the information from FDI and the state-space system, that operates as a filter based on a dynamical observer. The method is similar to Kalman filtering, but does not make assumptions about the linearity of the system or the variance of the observations; these assumptions represent a limitation of traditional Kalman filters, as highlighted in [93]. The state is updated as a weighted sum of the prediction of the state-space model and the observation of the FDI procedure:

$$\boldsymbol{k}_{i+1} = \gamma \boldsymbol{k}^e(t_{i+1}) + (1-\gamma)[\boldsymbol{k}_i + (\mathrm{A}_i \boldsymbol{k}_i + \mathrm{B}_i \boldsymbol{u}_i)\Delta t] \quad (2.31)$$

where $\gamma \in (0,1)$ is a scalar weight parameter, $\boldsymbol{k}^e(t_{i+1})$ is the fault condition measured by the FDI procedure according to Section 2.3, and the term $\boldsymbol{k}_i + (\mathrm{A}_i \boldsymbol{k}_i + \mathrm{B}_i \boldsymbol{u}_i)\Delta t = \boldsymbol{k}_i + \dot{\boldsymbol{k}}_i \Delta t$ is the fault condition predicted by the integration of the model. This procedure can be employed where the observations $\boldsymbol{k}^e$ are available, that is, for $t \leq t_{now}$. Since future fault condition are not measurable, the propagation of the state for $t > t_{now}$ is performed by the state space model alone:

$$\boldsymbol{k}_{i+1} = \boldsymbol{k}_i + (\mathrm{A}\boldsymbol{k} + \mathrm{B}\boldsymbol{u})\Delta t \quad (2.32)$$

At each time step t_i of the numerical integration, the surrogate assessment function $\hat{\phi}_a(\boldsymbol{k})$ trained offline determines whether the equipment is still able to operate under the effect of the fault combination $\boldsymbol{k}(t_i)$. When a "faulty" condition is detected by the assessment function at time t_f, the integration is stopped, and the

difference between the failure time and the current time is assumed as the RUL estimate:

$$RUL = t_f - t_{now} \tag{2.33}$$

The proposed methodology permits to achieve a good accuracy in RUL prediction, even if the rate of propagation of the damage from its incipient state at t_0 to the actual failure at t_f is not known. This is often the case for complex mechatronic systems, where heterogeneous components described by different disciplines coexist and work together, sometimes interacting in ways that are difficult to predict analytically. Additionally, this method has a lower computational cost than comparable approaches available in literature (e.g. those based on particle filtering [40, 94]) and can be executed in real-time on limited hardware resources. The RUL estimate is deterministic: an uncertainty associated to the prediction can be estimated through a Monte-Carlo analysis. To do so, the RUL prediction process is repeated iteratively. At each iteration, a random noise is superimposed to the observed fault conditions $\boldsymbol{k}^e$. This noise is sampled each time over the error distribution of the FDI, easily assessed offline over a validation dataset. As a result, an empirical characterization of the uncertainty distribution associated to RUL prediction is obtained.

Chapter 3

Application

Electromechanical Actuators (EMAs) are emerging as a novel technology for current and next generation flight control systems. As they convert directly electrical power into the mechanical power needed to drive the flight control surfaces [95, 96], they permit to adopt the More Electric [97] and All Electric Aircraft [98] system architectures. A widespread integration of EMAs in aircraft systems would eliminate the need for a centralized hydraulic system and would result in a reduction of the overall aircraft empty weight, with benefits in terms of fuel consumption and operating costs, as highlighted by Garcia Garriga et al. [99].

Electromechanical actuators replace the local hydraulic circuit of EHAs with a mechanical transmission [100]. A typical arrangement of an EMA is shown in Figure 3.1: the electric motor and its Power Drive Electronics is coupled to the user and external load by a reducer. Commonly the transmission includes an ordinary or planetary gearbox coupled to a device for conversion from rotary to linear motion, either a ball-screw or a roller-screw. Lead screws are usually avoided for their high friction, low efficiency and poor reliability. Alternatively, a high gear ratio reducer (such as a compound planetary, harmonic, or cycloidal drive) can be connected directly to the user through a rotary output shaft. In several applications, a rotary user is preferred since the transmission is more compact and allows a better power density. Additionally, the ballscrew is usually the first component of the transmission that is damaged in case of overload. Avoiding to convert the output to linear motion, reliability and robustness of the system are significantly improved.

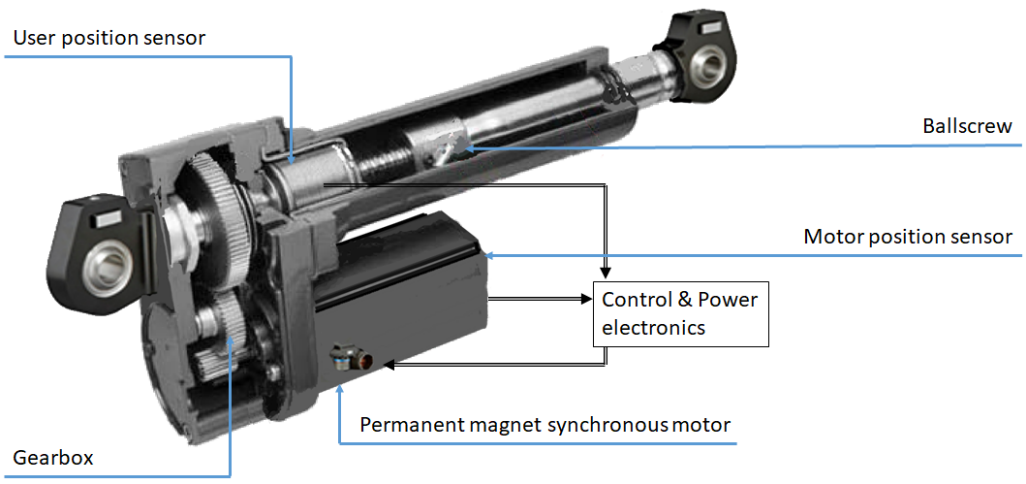

Figure 3.1: Cross section of an Electromechanical Actuator highlighting its main components

The electrical machines most commonly employed for aircraft EMAs are Brushless Direct Current (BLDC) motors or Permanent Magnet Synchronous Motors (PMSMs): those are chosen for their high power to weight ratio, and for their high reliability [101].

EMAs have some peculiar characteristic that differentiate them from other actuation systems. The absence of a hydraulic circuit, either centralized or local, eliminates all the issues related to the management of hydraulic fluid throughout the whole lifecycle of the product. Usually, aircraft-grade hydraulic fluids are toxic, pollutant or flammable; during maintenance, the system needs to be drained before disassembly and purged after the intervention, and a specialized maintenance crew is usually needed. Additionally, hydraulic components commonly feature very strict tolerances and small calibrated passageways, that are easily clogged if the fluid is accidentally contaminated. Therefore, maintenance interventions on EMAs are much easier, as discussed by Cronin [102].

EMAs are easily scalable to very small, low power and low weight applications: indeed, they are extensively employed for flight controls off small UAVs, with some actuators weighing down to few grams. Such miniaturization is not easily attainable by hydraulic systems.

The placement of sensors on the electrical machine, which are used for closed loop control of the motor current and speed, enables to integrate health monitoring functions in a smart actuator; the particular architecture of EMAs allows for a much more accurate and reliable fault detection than, for example, EHs.

On the other hand, some disadvantages are also related to Electromechanical devices. First, their power density is much lower than that of hydraulic systems, although the use of rare-earth permanent magnet is somehow reducing this penalty for advanced EMAs. Actuators based on hydraulic power rely on the working fluid as a heat sink; EMAs do not have this option, and can easily overheat if they operate in off-nominal conditions; then, thermal control of the equipment is of critical importance and must be accurately designed.

The issue of EMAs that most limits their diffusion in flight-critical aerospace applications is that their transmission is prone to mechanical jamming: that is, as a result of its most probable failure modes, the transmission gets locked in position. This condition may cause the loss of the associated aerodynamic surface, and control of the aircraft can become difficult [103]. Some configurations that allow redundancy of the mechanical transmission are available [104, 105, 106], but they all imply a significant increase in complexity and weight of the system.

Hence, the health monitoring task is of great importance for Electromechanical Actuators. An accurate and robust PHM framework would allow to overcome the aforementioned limitations in reliability often associated to EMAs and ease the early integration of such technologies into new aircraft designs. However, EMAs constitute a challenging application for Prognostic methodologies. The operation of electromechanical systems depends on the interaction of several components whose behavior is described by heterogeneous disciplines, such as mechanics, electronics, control theory, fluid dynamics and heat transfer. In aerospace applications, EMAs

often experience operating and environmental conditions that are hardly repeatable and predictable: as opposed to an industrial actuator, that will likely follow the same motion profile for thousands or millions of cycles, in a semi-controlled environment, an aerospace flight control servomechanism will face a different time history for each mission. All these aspects contribute to the complexity of the physics describing the propagation rate of wear and damage, as well as their effect on the behavior of the system as a whole. As a result, PHM strategies for these systems are still an open field of research.

3.1 High Fidelity (HF) Model

A High Fidelity model of an EMA was developed as a simulated test bench to collect reference data of the actuator operation, in different working conditions, and under the effect of multiple fault modes. The model is a physics-based, lumped parameters simulation characterized by a very high level of detail. The general architecture of the model is shown in Figure 3.2, and reflects the subsystem and components hierarchy of common hardware EMAs. Specifically, the model includes the following subsystems:

- The Actuator Control Electronics (ACE) model implements the control law employed to compare the commanded position with the feedback signals of measured position and speed, in order to determine the torque command to the motor.
- The model of Power Electronics has two main functions: (1) evaluate the commutation sequence of the motor phases, as a function of the current angular position of the rotor, and (2) close the current/torque control loop, applying to the stator coils the voltage needed for the motor to produce the commanded torque.
- The Electromagnetic model of the motor evaluates the magnetic coupling between rotor and stator, to determine dynamically the current, torque and back-EMF for a given input voltage, also accounting for the effect of possible fault precursors.

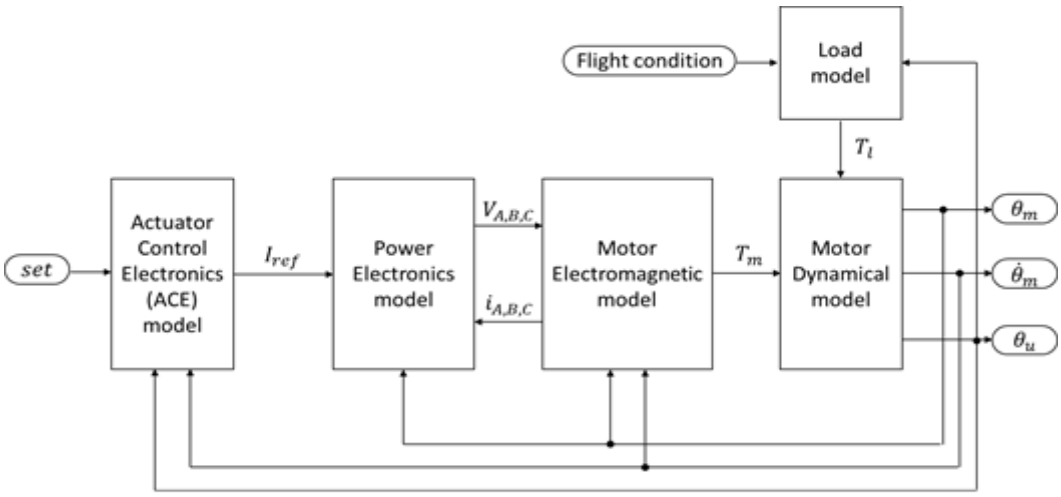

Figure 3.2: Block diagram of the HF model

- The Dynamical model of the motor and transmission computes the position and velocity of the actuator, under the combined effects of motor torque and external loads.
- Eventually, the Load model estimates the hinge moment by integrating a linearized model of the longitudinal aircraft dynamics.

The following sections describe in detail the implementation of the subsystems of the EMA in the high fidelity simulation model.

3.1.1 Controller Model

The Controller model, whose block diagram is shown in Figure 3.3, computes the control law of the EMA, to compute the torque and current requested to the electrical machine. The control law reflects common implementations on industrial and aerospace hardware, with a proportional position loop and a Proportional-Integral-Derivative (PID) velocity loop. Specifically, the controller accepts as an input the position or velocity setpoint, the measured motor speed, and the measured user position. Although more advanced control techniques are available in literature, such as State-space controllers, Fuzzy logic, or Neural Networks, PID are still the standard implementation for most industrial and aerospace applications, since they provide better robustness, especially when dealing with significant uncertainties in the behavior of the controlled plant.

The speed-position mode switch allows to choose among a position control mode and speed control mode. In position control mode, the setpoint is compared to the user position measured by a transducer. The position error is multiplied by a proportional gain to determine a velocity setpoint. In speed control mode, the position loop is disabled and the velocity setpoint is provided externally. In both cases, the velocity setpoint is limited by a saturation accounting for the maximum speed achievable by the motor, and the resulting command speed is compared to the actual motor speed signal. Then, a velocity error is fed to a PID controller to determine the required motor torque.

The PID includes a filter on the derivative branch and an anti-windup logic, as shown in Figure 3.4. The derivative filter is a low-pass, first order transfer function that allows to reduce the high frequency noise on the error signal, which would be

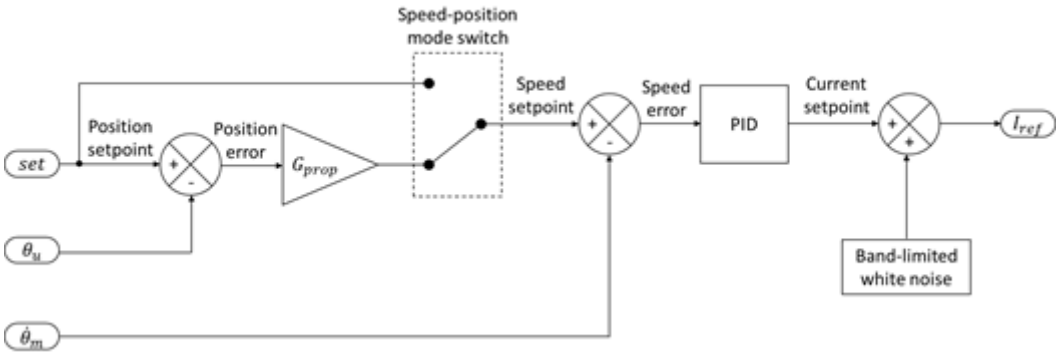

Figure 3.3: Block diagram of the Controller subsystem

otherwise amplified by the derivative. The transfer function of the derivative branch is then:

$$\frac{y_D}{e} = \frac{G_D s}{\tau_D s + 1} \tag{3.1}$$

where G_D is the derivative gain and τ_D is the characteristic time of the filter. The anti-windup logic is needed for the integrative branch, to mitigate the excess overshoot usually associated to the interaction of integrative control with a nonlinear plant characterized by a limited maximum rate of change. Several arrangements for anti-windup logic are available in literature; the algorithm implemented in this model temporarily disables the integral contribution if at least one of the following conditions is met:

- The error grows larger than a tolerance band, or
- The output of the PID reaches its saturation.

The output of the PID controller has the dimensions of a reference torque for the motor. This torque signal is divided by the nominal back-EMF coefficient of the motor, to determine the required stator current. The current command is limited by a saturation accounting for the peak current that can be handled by the motor, and routed to the Commutation and Power Electronics subsystem.

To simulate the effect of electromagnetic interference on the signal transmission lines, a band-limited white noise is added to the reference current signal.

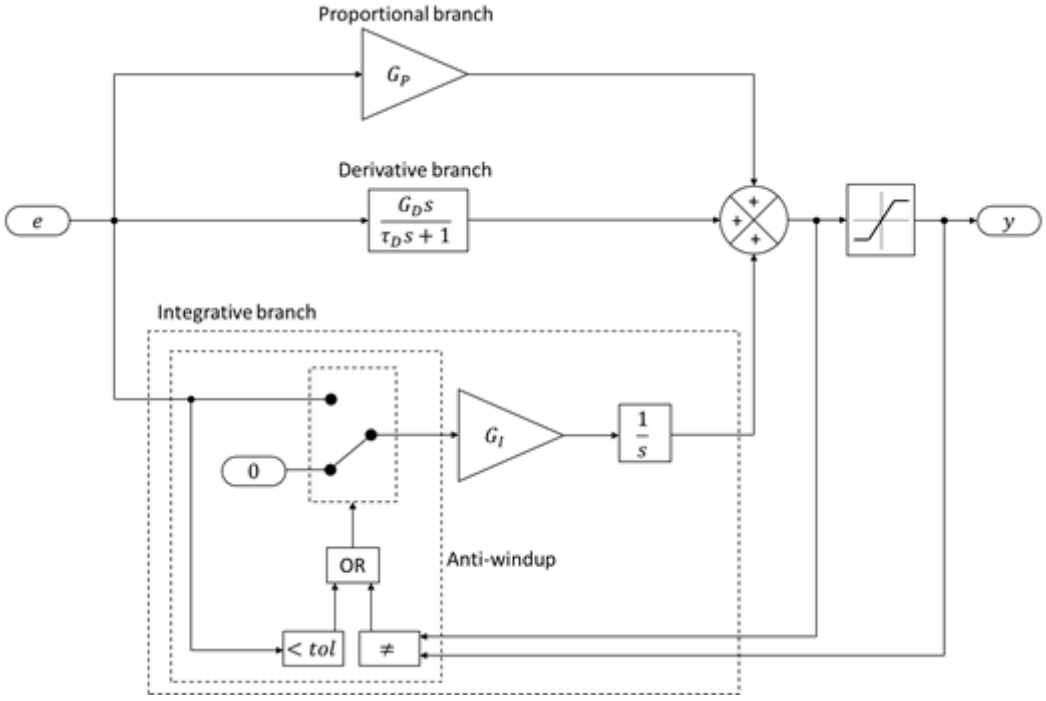

Figure 3.4: PID controller as implemented in the HF EMA model

3.1.2 Power Electronics Model

The model of power electronics determines the motor phase commutation sequence and actuates the current control for each of the motor phases. The model is slightly different if considering an EMA based on BLDC or PMSM electrical machine.

3.1.3 Brushless Direct Current (BLDC) motor

A Brushless Direct Current (BLDC) motor is a permanent magnet machine characterized by a trapezoidal waveform of the back-EMF of each phase. Voltage is applied to two out of three phases at a time, with a square waveform. The rotor position is sensed by three Hall sensors, with a resolution of $60°/P$, where P is the number of pole pairs. Figure 3.5 (a) shows the typical arrangements of the phases and Hall sensors for a 2 pole pairs configuration. Each Hall sensor outputs a Boolean 1 when it is over a magnetic north, and a Boolean 0 when it is over a magnetic south. The signals of the Hall sensors are plotted against the rotor position in Figure 3.5 (b); additionally, the required phase currents are shown.

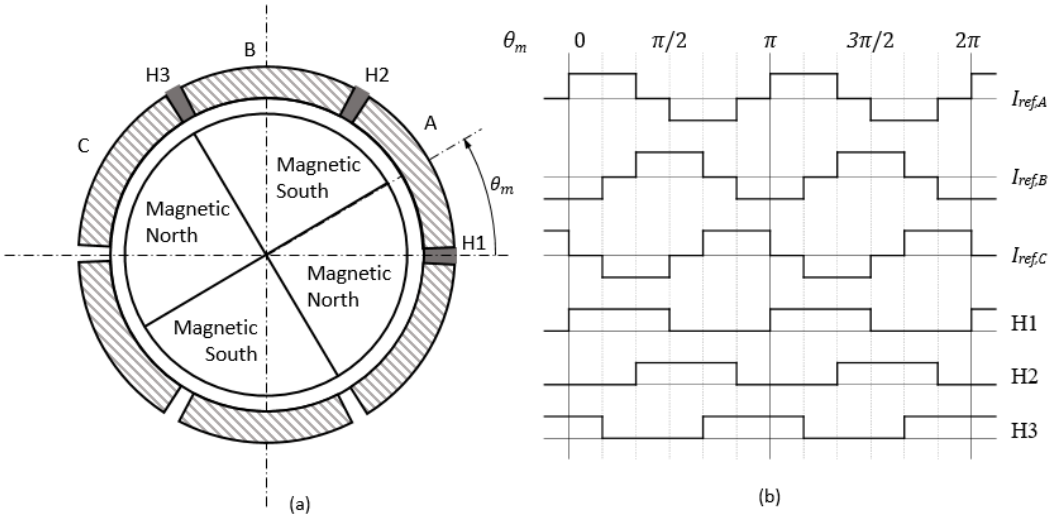

Figure 3.5: (a) Typical phase and Hall sensors configuration of a 2 pole pairs BLDC motor; (b) Readings of the Hall sensors and phase commutation sequence.

The reference currents (i.e. the phase currents required for a given torque command and a given rotor position) can be expressed as a function of the current setpoint I_{ref} and the readings of the Hall sensors H1, H2, H3:

$$\begin{cases} I_{ref,A} = I_{ref}(\mathrm{H2} - \mathrm{H1}) \\ I_{ref,B} = I_{ref}(\mathrm{H3} - \mathrm{H2}) \\ I_{ref,C} = I_{ref}(\mathrm{H1} - \mathrm{H3}) \end{cases} \tag{3.2}$$

The reference currents are compared with the measured phase currents by a hysteresis controller, which computes the actual activation signals for the switches of the three-phase bridge. The output of the hysteresis controller is a vector of three Boolean elements, each of which commutes from 0 to 1 if:

$$I_{ref,j} - i_j > b_h \tag{3.3}$$

and commutes from 1 to 0 if:

$$I_{ref,j} - i_j < -b_h \tag{3.4}$$

where b_h is the amplitude of the hysteresis band, for $j = A, B, C$. This signal is routed to the three-phase bridge, and commands the activation of the six switches. Specifically, when the output of the j-th hysteresis controller is high, the corresponding phase is connected to supply voltage; otherwise, it is connected to ground. A block-diagram of the Power Electronics model for the BLDC motor is shown in Figure 3.6.

3.1.4 Permanent Magnet Synchronous Motor (PMSM)

A Permanent Magnet Synchronous Motor (PMSM) is conceptually similar to a BLDC machine, but the polar expansions of the stator and the permanent magnets on the rotor are arranged to produce a sine wave back-EMF on each of the stator phases. The rotor position is measured either by a resolver or an absolute encoder, with a resolution at least in the order of $1°/P$.

The behavior of a PMSM is studied by introducing three different reference frames to describe the electrical angle, as shown in Figure 3.7:

- $\alpha - \beta$ axes are fixed with respect to the stator, the α axis being aligned with the axis of symmetry of the electrical phase A, and the β axis is offset by 90° electrical to form a right-handed frame. The angle Φ is used as a polar coordinate to describe the angles along the stator, starting from the α axis.
- $d - q$ axes are a reference frame rotating with the rotor. The d axis is aligned with a north pole of the rotor, and the q axis is 90° electrical in advance. The angle ξ is used as polar coordinate to describe the angles along the rotor, starting from the d axis.
- The three-phase reference frame with axes A, B, and C aligned with the respective stator phases. The A axis of this reference frame coincides with the α axis.

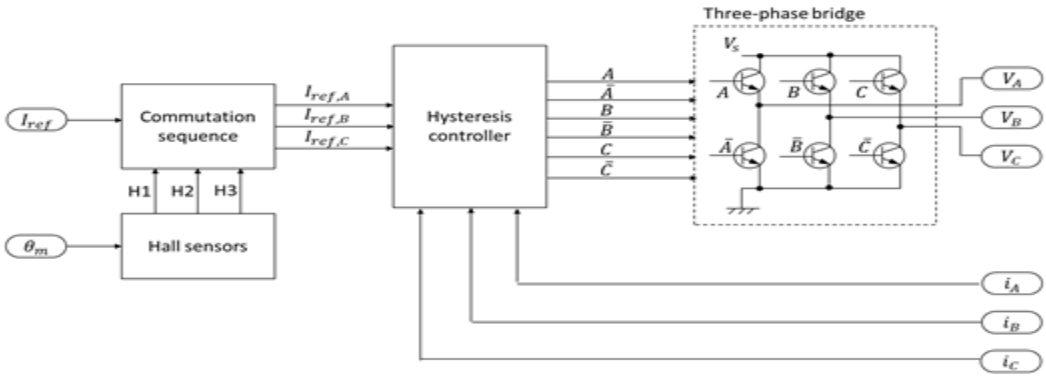

Figure 3.6: Block diagram of the Power Electronics model configured for the BLDC motor

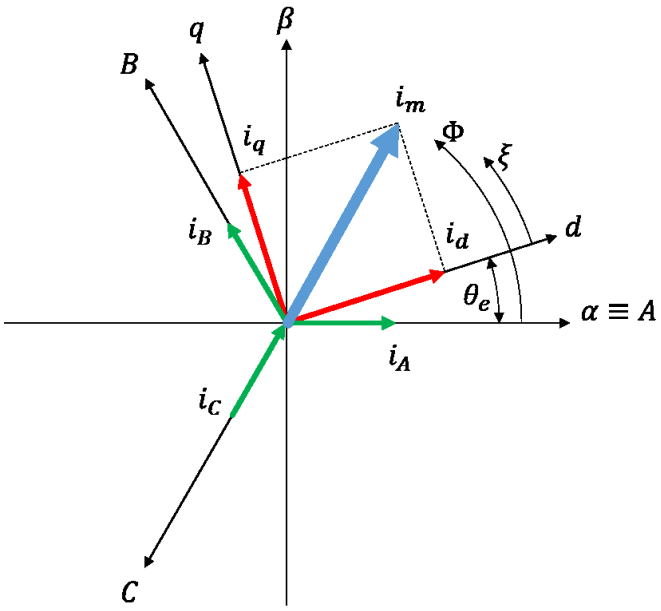

Figure 3.7: Reference frames for the Clarke-Park transformations

The angle between the $\alpha - \beta$ frame and the $d - q$ frame is θ_e. Two coordinate changes allow to switch between the three reference frames. Specifically, the Clarke transformation allows to convert the current and magnetic flux vectors expressed in the three-phase reference frame to the $\alpha - \beta$ reference frame. The Park transformation allows to convert the current and magnetic flux vectors expressed in the $\alpha - \beta$ reference frame to the $d - q$ reference frame.

The current vector expressed in the $\alpha - \beta$ reference frame has the form:

$$\boldsymbol{i} = i_\alpha + j i_\beta \tag{3.5}$$

where the bold denotes a vector quantity, and j is the imaginary unit. Similarly, in the three-phase reference frame the same current vector is expressed as:

$$\boldsymbol{i} = i_A + e^{\frac{j2\pi}{3}} i_B + e^{\frac{j4\pi}{3}} i_C \tag{3.6}$$

Since axis α id aligned with axis A, and axes B and C are $\pm 120°$ apart, the following two identities hold:

$$i_\alpha = \frac{2}{3}\left(i_A - \frac{1}{2} i_B - \frac{1}{2} i_C\right) \tag{3.7}$$

$$i_\beta = \frac{2}{3}\frac{\sqrt{3}}{2}(i_B - i_C) \tag{3.8}$$

Expressing those two equations in matrix form yields to the formulation of the Clark transformation:

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = [B]\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{3.9}$$

where $[B]$ is the Clarke matrix. Similarly, the current vector expressed in the $d - q$ reference frame has the form:

$$\boldsymbol{i} = i_d + j i_q \tag{3.10}$$

and the transformation between this form and the $\alpha - \beta$ frame is:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = [A] \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{3.11}$$

where $[A]$ is the Park matrix. In order to produce the maximum torque with minimum current, in usual operating conditions the PMSM power electronics command a stator current in quadrature with respect to the permanent magnet rotor. A direct current component may be commanded to reduce the effective electromagnetic coupling of the motor and the back-EMF produced at high speed, in order to reach rotation speed that would otherwise result in a back-EMF higher than the supply voltage.

In the model, as shown in Figure 3.8, this condition is not covered by the commutation logic, and the current setpoint I_{ref} from the control electronics is routed directly to i_q. This is acceptable if the required performances are compatible with the supply voltage and nominal back-EMF coefficient of the motor. The current setpoints for the three phases I_{refA}, I_{refB}, I_{refC} are evaluated through inverse Park and inverse Clarke transformations. Then, a hysteresis controller (implemented as per Section 3.1.2.1) commands the switches of the three-phase bridge to supply the required voltage to the stator.

3.1.5 Electromagnetic Model of the motor

The Electromagnetic model of the motor computes the torque and back-EMF produced by the electrical machine. The electromagnetic coupling between rotor and the stator phases is accounted for by three back-EMF coefficients k_A, k_B, k_C. Those are defined as the derivative of magnetic flux concatenated with each phase, with respect to the rotor angle θ_m. The back-EMF coefficients are computed considering the different distribution of magnetic field and windings for BLDC and PMSM machines (Figure 3.9):

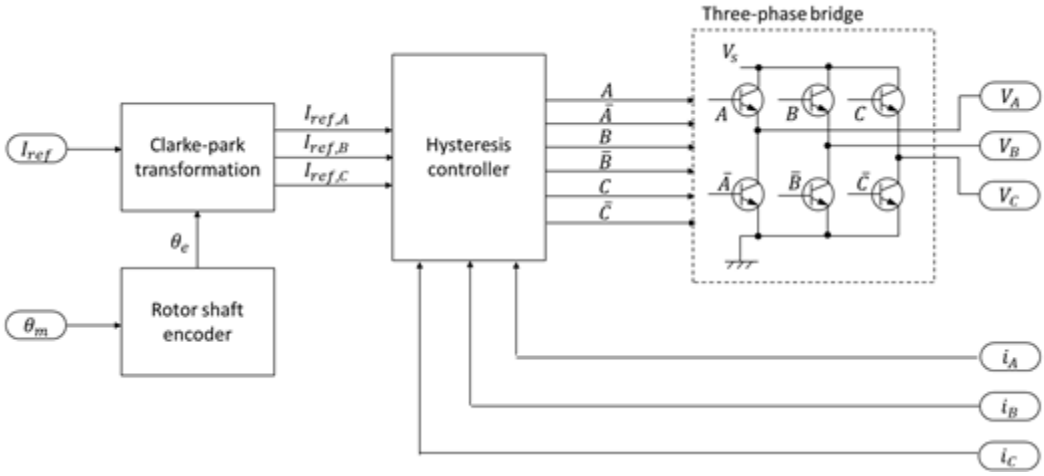

Figure 3.8: Block diagram of the Power Electronics model configured for the PMSM

- For the BLDC motor, a lookup table is employed to generate three trapezoidal waves, multiplied by the nominal back-EMF coefficient from the motor datasheet.
- For the PMSM, three sine waves, 120° apart from each other, are computed and multiplied by the nominal back-EMF coefficient.

The coefficients are then modified according to Section 3.1.6 to simulate the electrical fault modes. To evaluate the phase currents, a three-phase RL circuit is employed, as shown in Figure 3.10. The circuit is connected with a star arrangement and computed by Simscape, solving the following set of equations for each integration timestep:

$$i_A + i_B + i_C = 0 \tag{3.12}$$

$$V_j - k_j \omega = R_j i_j + L_j \frac{di_j}{dt} \tag{3.13}$$

for $j = A, B, C$. The resistance and inductance of each phase, R_j and L_j respectively, are the nominal values from the motor datasheet, modified to account for potential faults.

The currents, alongside with the respective back-EMF coefficients, are employed to compute the motor torque. Assuming a linear superposition of the contributions of each phase, the total motor torque is given by the sum of the torques produced by each of the three phases:

$$T_m = \sum_{j=A,B,C} i_j k_j \tag{3.14}$$

In addition, the torque is limited by a saturation accounting for the maximum magnetic flux through the stator polar expansions.

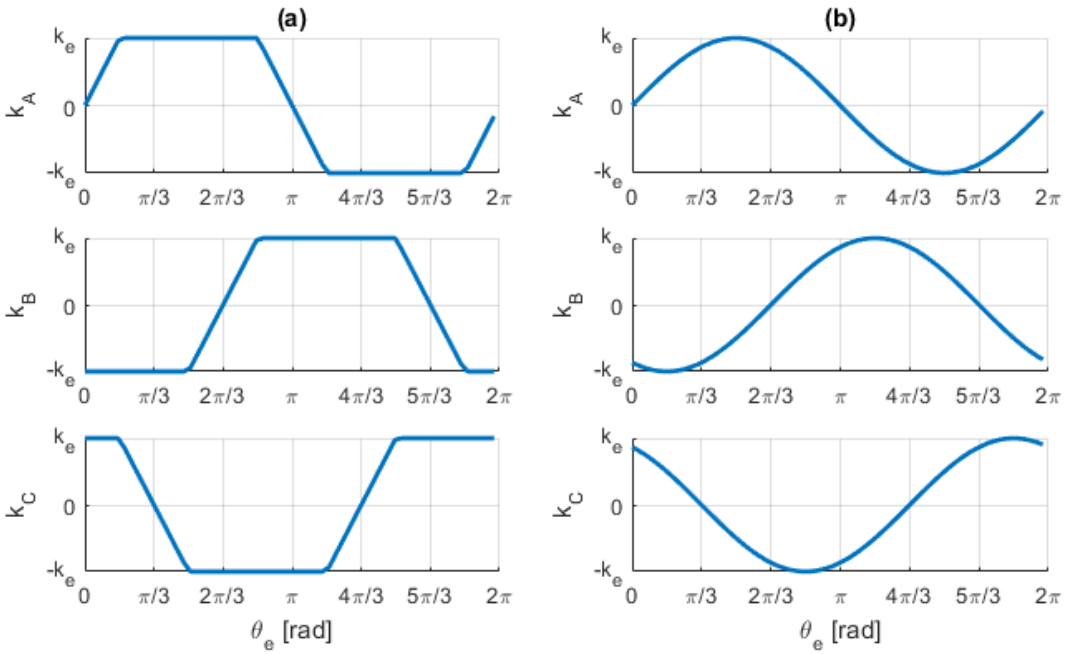

Figure 3.9: Phase back-EMF coefficients for BLDC (a) and PMSM (b)

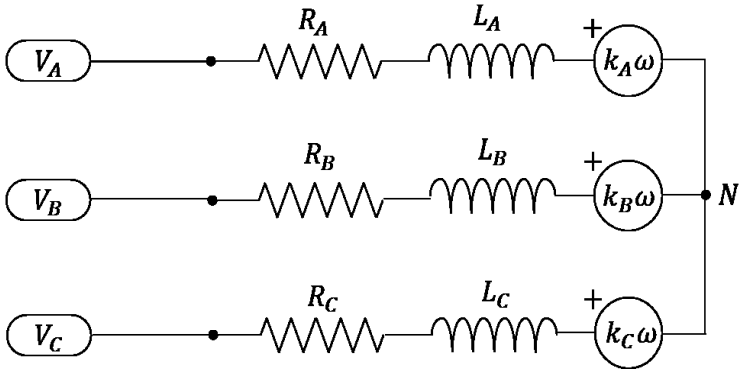

Figure 3.10: Equivalent circuit of the motor

3.1.6 Dynamical Model of motor and transmission

The dynamical model of motor and transmission computes the positions of motor and user, receiving as an input the motor torque and external load torque. The core of the model is a second order dynamical representation of the system, described by the equation:

$$T_m - T_l = J_m \frac{d^2\theta_m}{dt^2} + C_m \frac{d\theta_m}{dt} \tag{3.15}$$

where J_m and C_m are the inertia and damping of the motor-user assembly, expressed in the reference of the motor shaft. In addition, the model accounts for a number of non-linear phenomena affecting the behavior of the actuators, such as endstops, dry friction, and backlash. The viscous friction coefficient C_m is dependent on speed, to account for the Stribeck effect and potentially other nonlinear effects.

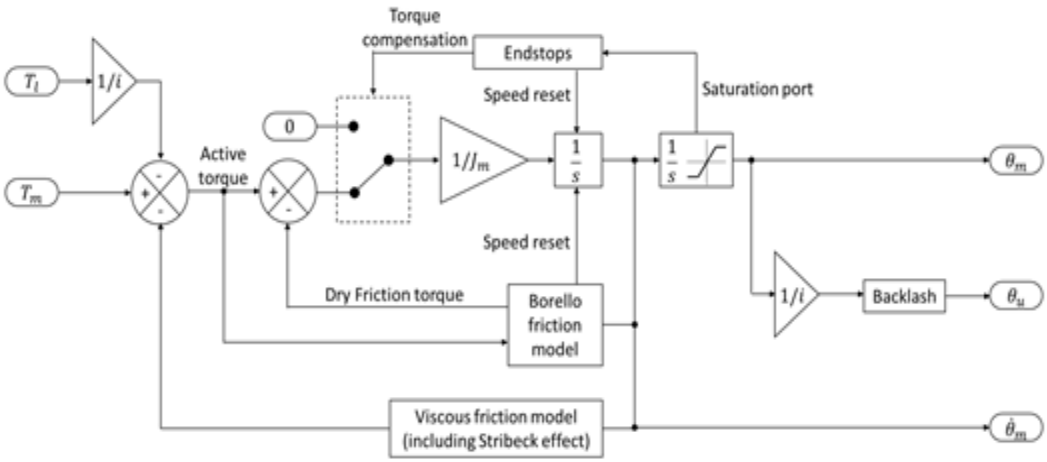

Figure 3.11: Block diagram of the motor-transmission dynamical model

As shown in Figure 3.11, the collision with endstops is detected by a saturated position integrator. When a saturation is detected, the following actions are performed:

- The velocity integrator is reset to zero
- If the total torque acting on the system (including motor torque, external load, and viscous and dry friction) is aligned with the endstop (i.e. pushes the actuator against the endstop), it is overwritten to zero.

Dry friction is simulated with the Borello model [107]. This is a numerical implementation of the Coulomb friction model, intended to overcome some limitation of alternative formulations (i.e. the saturated hyperviscous, Quinn, and Karnopp friction models). Specifically, to correctly simulate dry friction, a model shall be able:

- To apply a force of constant modulus (equal to the dynamic friction value) against the direction of motion, when the system is in motion;
- To act as a constraint, with a reaction that can grow up to the static friction value, usually higher than the dynamic friction one, when the system is stopped.

While the dynamic behavior is simple to implement numerically, static friction may cause unintended interactions with the numerical integration. Specifically, the integration method may skip the time instant when velocity crosses zero, failing to apply the static friction condition. This issue is usually addressed by adding a dead band near zero velocity (Quinn [108] and Karnopp models [109]), or by simulating static friction as a very large viscous action (hyperviscous saturated model, [110]). However, both these models require the introduction of non-physical parameters (i.e. dead band amplitude, viscous force) that need to be calibrated to work with a given model and a given integration method. Additionally, some models (e.g. hyperviscous) fail to keep the speed to zero in the static friction condition. The Borello model of dry friction addresses these points by adding a zero-crossing detection routine to the velocity integrator. When the system speed changes direction, the velocity integrator is reset to zero for one timestep. As a result, it is possible to apply correctly the static friction condition.

Backlash is modelled with a hysteresis band on the user shaft position. Specifically, the behavior of the Backlash model is summarized by Equation (3.16):

$$\theta_u(t) = \begin{cases} \dfrac{\theta_m(t)}{i} + BLK, & \text{if } \theta_u(t-dt) - \dfrac{\theta_m(t)}{i} > BLK \\ \dfrac{\theta_m(t)}{i} - BLK, & \text{if } \theta_u(t-dt) - \dfrac{\theta_m(t)}{i} < -BLK \\ \theta_u(t-dt), & \text{otherwise} \end{cases} \tag{3.16}$$

that is, the user position is constrained within a band of width $2BLK$ from the motor position reduced to the slow shaft. If the user position at the last timestep $\theta_u(t-dt)$ already lies inside the backlash band, it is not updated at the current timestep (i.e. the user does not move). This model yields acceptable results if the main source

of backlash is assumed to be the connection of the user shaft position sensor: under this assumption, the inertia and load downstream the mechanical play are negligible, and do not affect the overall system response. A more detailed model would require a multibody simulation of the mechanical transmission, accounting for the multiple degrees of freedom of each individual moving part.

3.1.7 Load Model

The load on the actuator is simulated assuming to install it in the elevator control of an aircraft. In particular, the dynamical model of the F-16 fighter aircraft is employed, as available from Stevens [111] . For the scope of this work, only the longitudinal linearized dynamics is considered, expressed in the state-space form:

$$\begin{cases} \dot{x} = [A]x + [B]u \\ y = [C]x + [D]u \end{cases} \tag{3.17}$$

where x is the state vector, y is the output vector, u is the control vector, $[A]$ is the state matrix, $[B]$ is the control matrix, $[C]$ is the output matrix and $[D]$ is the feedthrough matrix. Considering only longitudinal dynamics, the state vector is $x = [\Delta V\ \Delta\alpha\ \Delta\theta\ \Delta q]^{\mathrm{T}}$, where Δ denotes variations from the equilibrium condition, V is the flight speed, α is the angle of attack, θ is the pitch angle, and q is the pitch rate, and the control vector is $u = [\Delta\delta_t\ \Delta\delta_e]^{\mathrm{T}}$, where δ_t is the throttle command and δ_e is the elevator deflection. For the considered application, y is a scalar containing the hinge moment on the elevator, required to evaluate the load on the actuator. The matrices $[A]$ and $[B]$ of the state space model are derived from [111], and linearized for a trimmed condition at sea level at 153m/s (550km/h). The matrices $[C]$ and $[D]$ are evaluated combining thin airfoil theory and Prandtl lifting line model, from available images of the F-16 all moving tail. The numerical values of the matrices used for the state-space model are the following:

$$A = \begin{bmatrix} -1.9311 \cdot 10^{-2} & 8.8157 & -32.170 & -0.57499 \\ -2.5389 \cdot 10^{-4} & -1.0189 & 0 & 0.90506 \\ 0 & 0 & 0 & 1 \\ 0 & -0.82225 & 0 & -1.0774 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0754 \cdot 10^{-4} & 0.17370 \\ 0 & -2.1499 \cdot 10^{-3} \\ 0 & 0 \\ 3.9644 \cdot 10^{-6} & 0.17555 \end{bmatrix} \tag{3.18}$$

$$C = [-5.9978 \quad -8.9812 \cdot 10^{3} \quad 0 \quad -2.6102 \cdot 10^{2}]$$

$$D = [0 \quad 9.4643 \cdot 10^{3}]$$

The computed load is multiplied by a gain to account for the geometry of the linkage between the actuator and control surface, and fed back to the EMA model. This way, the model is able to simulate an operation close to the actual load condition experienced in flight by an FCS actuator, in order to demonstrate the applicability of the proposed diagnostic and prognostic approach in real-time.

3.1.8 Simulation of faults

In order to be employed as a simulated test bench for diagnostic algorithms, the EMA model shall be able to reproduce the behavior of the actuator accounting for the effect of a set of fault modes, both in their incipient stage and in system failure condition. Specifically, in this work the following failure modes are considered:

- Variation of friction
- Variation of backlash
- Partial short circuit of a motor phase
- Static eccentricity of the motor
- Drift of the controller proportional gain

For this case study, those are selected among the most common failure modes affecting electromechanical systems, according to the available literature [103, 112, 113, 114]. Alternatively, when applying the proposed procedure to a newly designed system, the selection of failure modes to be considered may be driven by the Reliability, Availability, Maintainability & Safety (RAMS) documentation of the equipment, such as a Failure Modes Effects & Criticality Analysis. The faults considered in this study are analyzed in the following sections, and their effect on the actuator response is discussed.

3.1.9 Friction

An increase of friction may result from the degradation of mechanical elements of the transmission of the actuator, such as spalling of the gear teeth sides or bearing rolling surfaces. This fault mode is easily simulated by varying the static and dynamic friction torque parameters within the Borello model (Section 3.1.4). The increase of friction from its nominal value determines a worsening of the positioning accuracy, and may result in a reduced margin of stability, stick-slip phenomena, or a limit cycle, interacting with the integrative contribution of the control law.

As shown in Figure 3.12, increasing the value of dry friction from the nominal 5% to 15% of the motor stall torque results in three distinct effects on the no-load response to a step command.

- An increase delay while starting from standstill;
- A reduction of the no-load speed, seen as a decreasing slope of the constant-speed phase of the actuation;
- A decrease of positioning accuracy resulting from a larger static error.

Figure 3.13 shows the response to a ramp command with a low slope, in presence of an increasing amount of dry friction. With a relatively low static friction (i.e. for $T_f/T_{m,max} = 0.1$) the output position follows the setpoint smoothly, although with a small static error. Increasing the dry friction coefficient, a stick-slip phenomenon appears, where the system repeatedly starts, overshoots the speed

setpoint, and stops under the effect of friction. This behavior translates into vibration and noise of the transmission.

Figure 3.14 shows the interaction between the integrative branch of the PID controller and an increasing amount of dry friction, resulting in a limit cycle. As the system overshoots the position setpoint, the integrative contribution of the PID winds up until the system starts up and tries to compensate the static error, but overshoots the setpoint in the opposite direction; the cycle repeats until the setpoint changes.

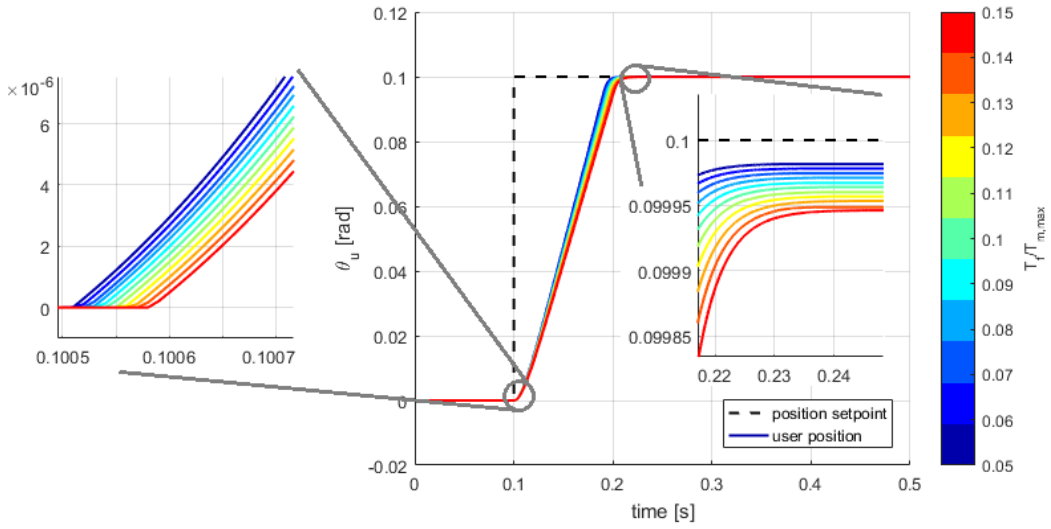

Figure 3.12: Effect of increasing friction on the step response.

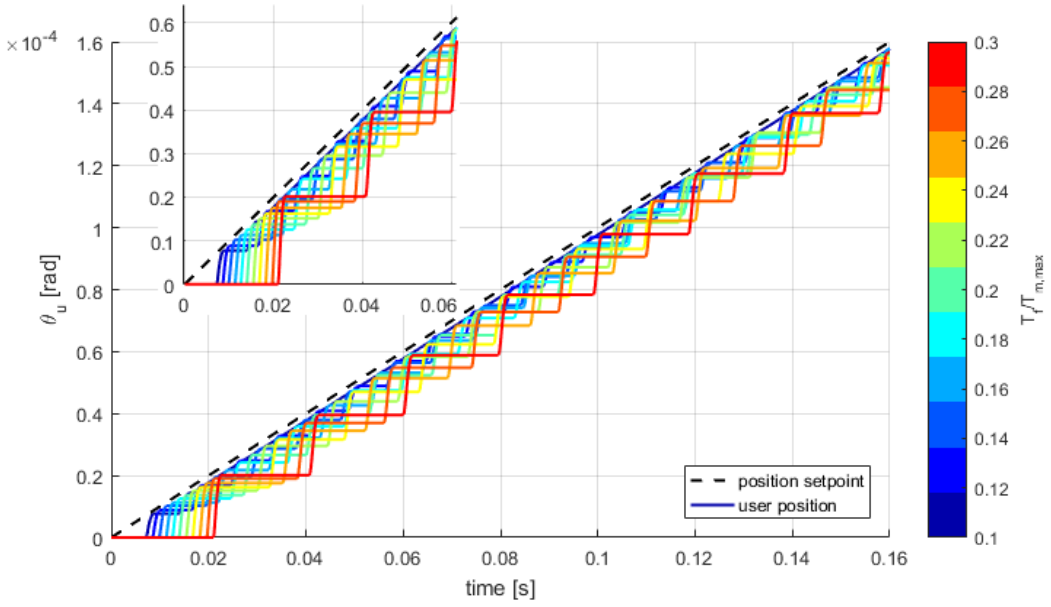

Figure 3.13: Stick slip on ramp response caused by an increasing amount of dry friction

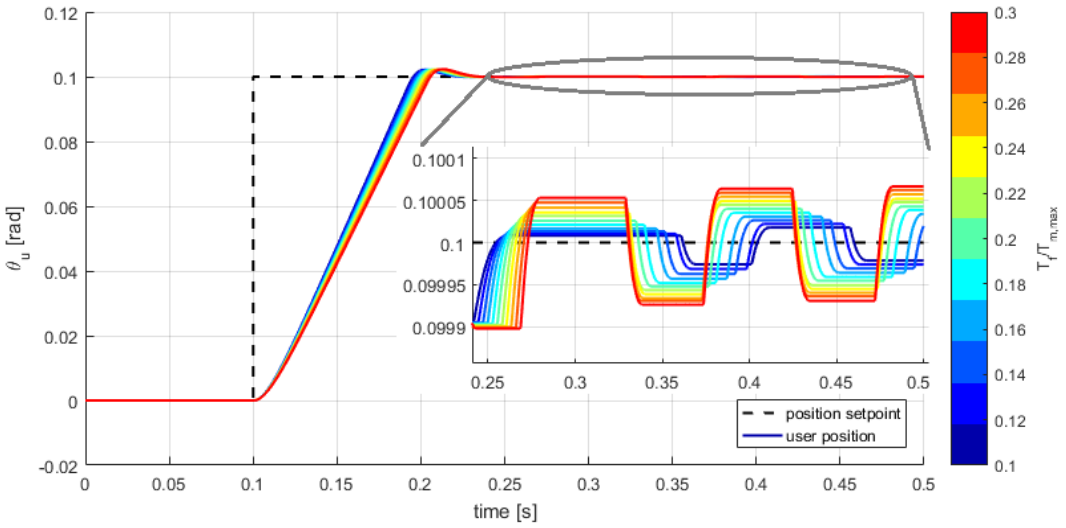

Figure 3.14: Interaction of integral controller and dry friction resulting in a limit cycle.

3.1.10 Backlash

Backlash is the mechanical play between components of a transmission. On an actuator transmission, it may interact with the control law producing limit cycles or reducing the system stability and accuracy. The interaction between backlash and controller can be avoided if the position sensor is installed upstream the mechanical play, but aerospace actuators commonly require a feedback on the user shaft to achieve an absolute position reference. For these reasons, the components of an electromechanical actuator are designed with a small interference, in order to give the transmission a preload and virtually no backlash. However, wear and degradation of contact surfaces of sliding and rolling elements of the transmission, such as gear teeth, or bearing rollers, may result in the increase of backlash affecting the actuator.

In this work, the backlash fault is simulated by varying the hysteresis amplitude BLK within the dynamical model of the motor and transmission, as expressed by Equation (3.16).

Figure 3.15 shows the effect of an increasing backlash amplitude on the response of the actuator to a sine wave position command. The user position is affected by a delay during the inversion of motion; conversely, when the user moves in a constant direction the control system is able to compensate even for quite large backlash amplitudes. A more visible effect is experienced by the motor current: when the direction is reversed, the transmission to the user disengages for a while due to the mechanical play. Then, the controller sees an increasing error and overcompensates, causing an oscillation. When the system engages back in the opposite direction, the position error decreases and the oscillation in current dampens out.

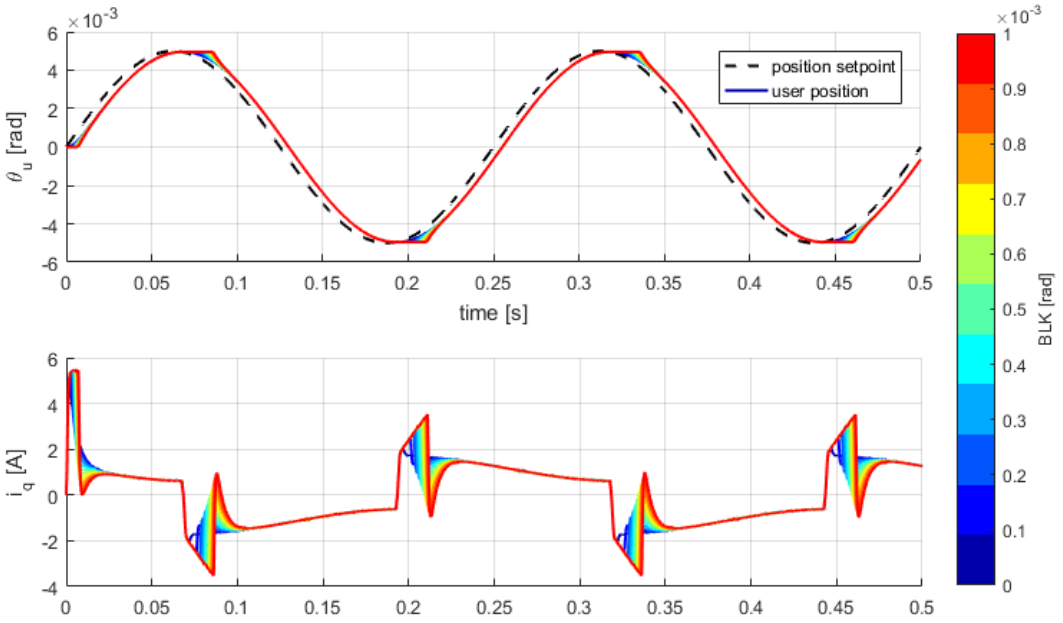

Figure 3.15: Response to a sine wave position command: effect of increasing backlash on user position (top) and motor current (bottom).

3.1.11 Partial Short Circuit

Short circuit of the windings is a common failure mode of most electrical machines. In a brushless motor, either PMSM or BLDC, each phase of the stator is manufactured with a number of windings of enameled copper wire. The polymeric coating of the wire is intended to insulate electrically the windings from each other, but has inherently a limited tolerance to high temperature. Occasionally, for example due to unexpectedly unfavorable operating conditions, the maximum temperature for the wire insulation may be exceeded locally, resulting in the short circuit of a winding with the next one. Depending on the extent and location of the damage, three classes of failure modes may be identified:

- Coil-to-coil short circuit, when two coils of the same phase come in contact
- Phase-to-phase short circuit, when the damage is located between two coils belonging to different phases
- Phase-to-ground short circuit, when a phase comes in contact with the motor case or with the iron core of the stator.

The latter two cases result immediately in a complete failure of the motor, since one or more phases are lost. The coil-to-coil short is commonly characterized by a more progressive evolution, since the short circuit currents involved are smaller and result in lower heating. Additionally, this fault mode is the most common of the three, since typically most of the windings are surrounded by windings belonging to the same phase; usually, only a small fraction of the windings is next to another phase or to the motor case. For these reasons, only the coil-to-coil partial short circuit is considered for this work.

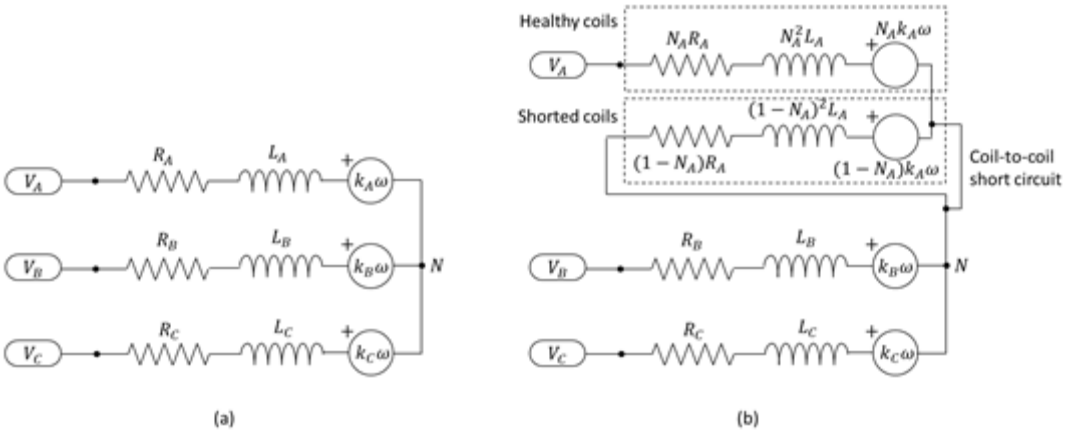

Figure 3.16: Equivalent circuit of a healthy stator (a) and one affected by a partial short circuit of phase A (b).

The equivalent circuit of Figure 3.16 is employed to model the effect of a partial short circuit fault on the operation of the motor. Defining $0 \leq N_A \leq 1$ as the fraction of healthy windings of phase A, since the faulty fraction of the phase is bypassed by the short circuit, the effective resistance of the phase will be:

$$R_A = N_A R_m \tag{3.19}$$

where R_m is the nominal phase resistance of the motor. The same method cannot be applied directly to the phase inductance, since the healthy and faulty fractions of the coil cannot be assumed to be in series and magnetically isolated. The inductance of a solenoid is proportional to the square of the number of turns. Then, we can assume that the effective inductance of the phase is proportional to N_A^2:

$$L_A = N_A^2 L_m \tag{3.20}$$

where L_m is the nominal phase inductance of the motor. The magnetic coupling with the rotor is proportional to the magnetic flux crossing the windings of each phase. Then, the back-EMF coefficient of phase A is proportional to N_A:

$$k_A = N_A k \tag{3.21}$$

This results both in a reduction of back-EMF seen by the damaged phase, and on a reduction of the torque produced by the same phase. An additional effect of the short circuit is a damping torque contribution produced by parasite currents flowing in the damaged fraction of the phase. Specifically, the short-circuited windings have a resistance of:

$$R_{A,S} = (1 - N_A) R_m \tag{3.22}$$

and is subject to a back-EMF:

$$E_{A,S} = (1 - N) k\omega \tag{3.23}$$

resulting in a parasite current:

$$i_{A,S} = \frac{E_{A,S}}{R_{A,S}} = \frac{k\omega}{R_m} \tag{3.24}$$

This current produces waste heat P_W in the winding and results in a braking torque $T_{A,S}$ proportional to the velocity of the motor:

$$P_W = i_{A,S}^2 R_{A,S} = (1 - N_A) \frac{k^2 \omega^2}{R_m} \tag{3.25}$$

$$T_{A,S} = (1 - N_A) \frac{k^2 \omega}{R_m} \tag{3.26}$$

Then, the presence of a partial phase short circuit has two effects on the operation of the motor: a reduced torque produced by the damaged phase (both for the reduction of magnetic coupling with the stator and for the braking contribution resulting from parasite currents) and an increased power dissipation, resulting in higher operating temperatures. As a result, the no-load speed of the motor is reduced, and a ripple can be detected in the current signal due to the unbalance between the phases. Figure 3.17 shows this effect of a partial short circuit of increasing magnitude on the current signal.

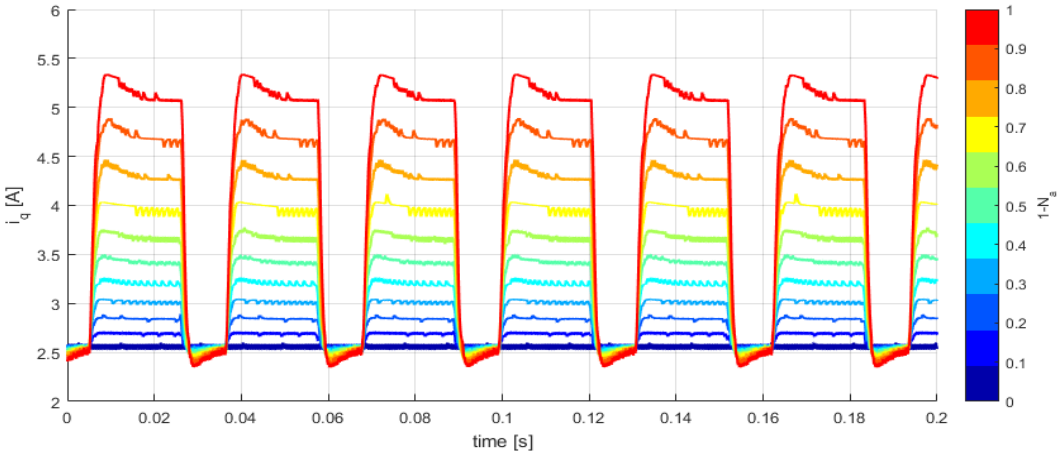

Figure 3.17: Motor current for constant speed, constant load actuation and increasing partial short circuit of phase A.

3.1.12 Rotor Eccentricity

Rotor eccentricity is the effect of a misalignment of the bearings supporting the shaft, or of the geometric tolerances of the shaft itself. In general, the axis of rotation of the rotor, the axis of symmetry of the stator, and a principal axis of inertia of the rotor may be slightly offset from each other. The misalignment between the axis of symmetry of the stator and the axis of rotation of the rotor is defined as static eccentricity, while that between the axis of rotation and principal axis of inertia of the rotor is the dynamic eccentricity. The first contribution causes a non-uniform distribution of the air gap width over the revolution, which is fixed with respect to the stator. On the other hand, dynamic eccentricity causes a distribution of the air-gap which moves with the rotor; additionally, the mass unbalance with respect to the axis of rotation is a source of vibration. Dynamic eccentricity is easily detected with a dedicated accelerometer, measuring the vibration at the frequency of the motor rotation. Conversely, static eccentricity has no immediate effect on vibrations, but can be detected from the ripples produced on currents and voltages of the motor. Then, this work focuses on modelling static eccentricity only.

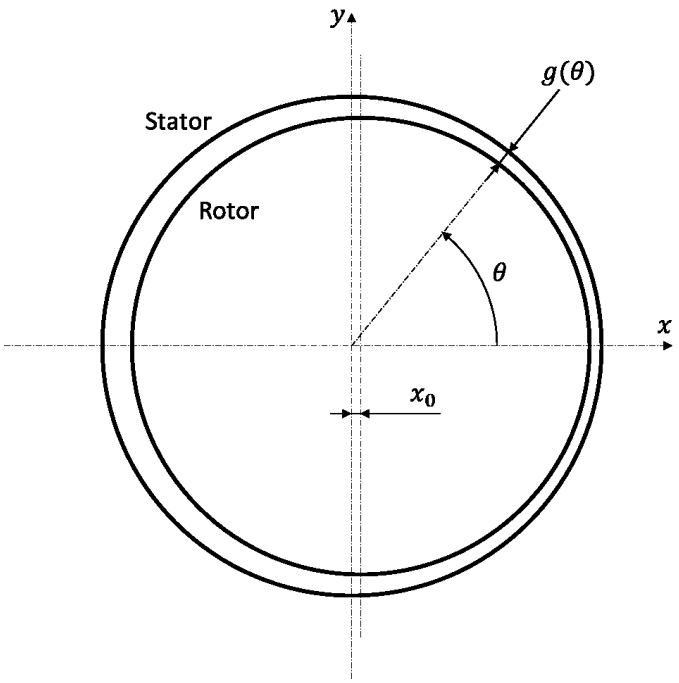

Figure 3.18: Distribution of the air gap in presence of a static rotor eccentricity

Figure 3.18 proposes a diagram of the rotor and the stator in presence of a static eccentricity aligned with the x axis. With a Cartesian reference frame centered on the rotor, the edges of rotor and stator are described respectively by the equations:

$$x^2 + y^2 = r_r^2 \tag{3.27}$$

$$(x - x_0)^2 + y^2 = r_s^2 \tag{3.28}$$

where r_r is the radius of the rotor and r_s is the radius of the stator. This translates in polar coordinates:

$$\rho = r_r \tag{3.29}$$

$$\rho^2 - 2\rho x_0 \cos\theta + x_0^2 - r_s^2 = 0 \tag{3.30}$$

The local air gap for a given direction $g(\theta)$ is the distance between the rotor and the inner wall of the stator, that is, the difference in the radial coordinate:

$$\begin{aligned} g(\theta) &= x_0 \cos\theta - \sqrt{1 - \left(\frac{x_0}{r_s}\right)^2 \sin^2\theta} - r_r \\ &\cong x_0 + r_s\left[1 - \frac{1}{2}\left(\frac{x_0}{r_s}\right)^2 \sin^2\theta\right] - r_r \cong x_0 \cos\theta + g_0 \end{aligned} \tag{3.31}$$

where $g_0 = r_s - r_r$ is the nominal air gap, i.e. the air gap when the eccentricity $x_0 = 0$. In the previous equation, the Taylor expansion $\sqrt{1+x} \cong 1 + x/2$ was employed, since x_0 is small compared to r_s. Additionally, the term in x_0^2/r_s is negligible with respect to x_0. Introducing the non-dimensional eccentricity parameter ζ, the air gap can be expressed in the form:

$$g(\theta) = g_0(1 + \zeta \cos\theta) \tag{3.32}$$

The effect of rotor eccentricity on the distribution of magnetic flux density is evaluated with the approach proposed by [115, 116]. The air gap seen by a magnetic pole of the rotor is assumed to be constant and equal to the actual air gap found on the axis of symmetry of the pole. This avoids to run a computationally expensive electromagnetic FEM to determine the instantaneous back-EMF coefficients of the motor phases. By solving the magnetic circuit composed by the stator, rotor and air gap, it is possible to compute the magnetic flux $\psi_i(\theta_m)$ experienced by the i-th phase of the stator; the derivative of this quantity with respect to the rotor angle $d\Psi_i(\theta_m)/d\theta_m$ is equivalent to the back-EMF coefficient for that phase. As a result, the back-EMF coefficients are corrected as follows:

$$k_1 = k_{1,0}(1 + \zeta \cos(\theta_m + \phi)) \tag{3.33}$$

$$k_2 = k_{2,0}(1 + \zeta \cos(\theta_m + \phi + 2\pi/3)) \tag{3.34}$$

$$k_3 = k_{3,0}(1 + \zeta \cos(\theta_m + \phi - 2\pi/3)) \tag{3.35}$$

Figure 3.19 and Figure 3.20 show the effect of static rotor eccentricity on the response of the system. Specifically, Figure 3.19 is referred to an eccentricity of varying amplitude and fixed phase, aligned with the stator phase A. Figure 3.20 shows the effect of an eccentricity of constant amplitude (equal to half of the nominal air gap) and a varying phase. In both cases, the simulations were performed assuming a constant speed, constant load actuation. The presence of rotor eccentricity produces a noticeable ripple in current, which can be leveraged for fault detection.

The model employed to predict the effect of rotor eccentricity on the behavior of the electrical machine is quite simplified, and does not account for local variations of the magnetic flux. A more detailed simulation would require a complete 3D electromagnetic Finite Element (FE) analysis; however, the prognostic methodology proposed in this work is not constrained to use a particular formulation, and can be employed as well by switching the eccentricity model to a more accurate one.

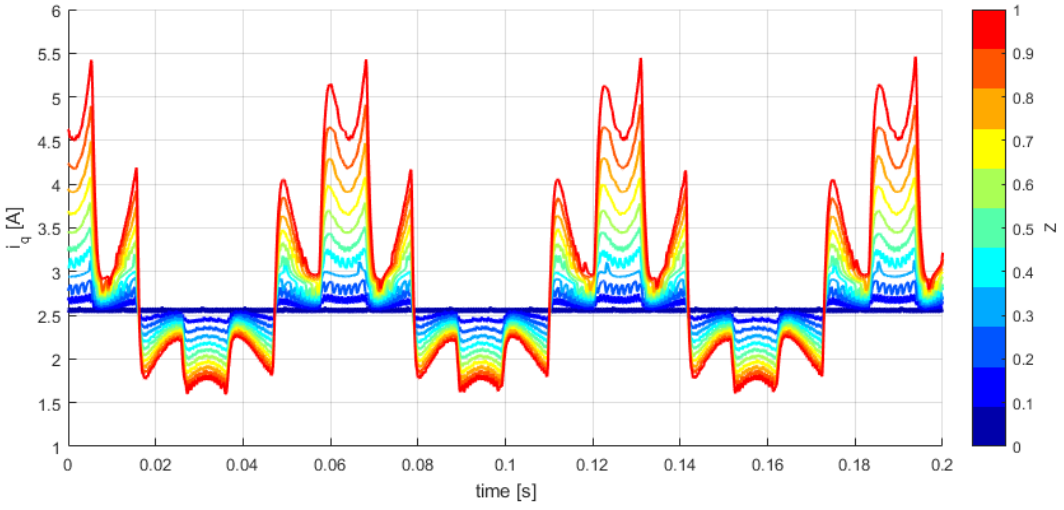

Figure 3.19: Motor current for steady state operation, in presence of static eccentricity of increasing magnitude

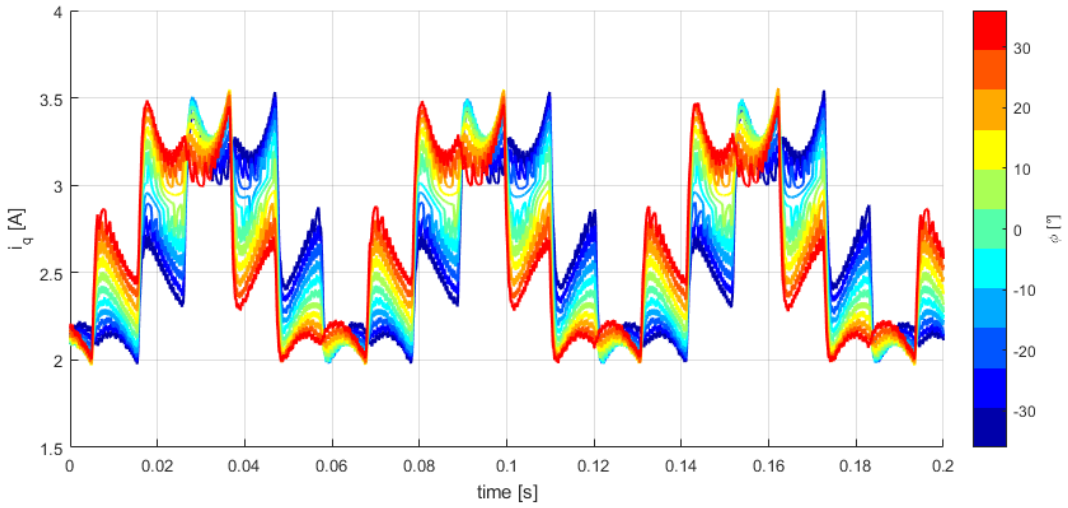

Figure 3.20: Motor current for steady state operation, in presence of static eccentricity of constant amplitude and varying phase.

3.1.13 Proportional Gain Drift

A drift of the position loop proportional gain is considered in this study as an additional fault mode. Although this may be the effect of the variation of electrical characteristics in an analogue controller, it does not usually appear in modern digital electronics. However, even if this behaviour is not completely representative of a physical fault mode of a modern actuator, it is implemented in this work in order to validate the proposed strategy with a larger number of possible system failures.

Figure 3.21 shows the effect of a variation of the position loop proportional gain on the step response of the actuator. The green curve corresponds to the nominal gain G_{P0}. The blue curve result from a progressive reduction of the gain down to 50% of the nominal value: the response becomes slower and more damped, as the rise time and the static error increase. The red and yellow curves are caused by an increase of the nominal gain up to 150%; in this case, the stability of the system is reduced and the overshoot increases.

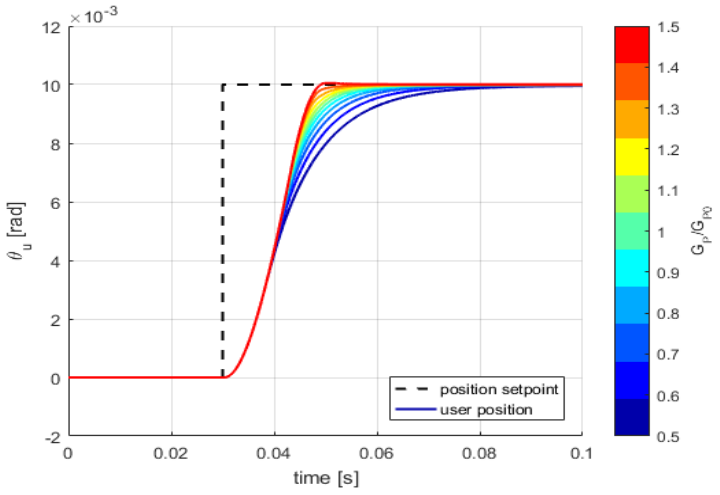

Figure 3.21: Effect of a variation of the position loop proportional gain on the step response of the actuator.

3.2 First Low Fidelity (LF1) Model

The High Fidelity model described in Section 3.1 allows to simulate the behavior of an electromechanical actuator accounting for the physical interactions of its subsystems and components. The result is a detailed emulation of the operation of the equipment, although obtained through a lumped parameter model. Therefore, the HF model can be employed as a simulated test bench to assess the performance of diagnostic and prognostic strategies, or to collect ground truth data for training machine learning algorithms within the aforesaid prognostic strategies. However, the computational burden required by the HF model is significant: on a common laptop PC, the computational time of the HF model lies almost two orders of magnitude above the simulated time interval. Thanks to the lumped parameter formulation, this is much faster than simulations employed in other fields of engineering requiring the solution of Partial Derivative Equations (PDEs), such as Finite Elements Methods (FEM) or Computational Fluid Dynamics (CFD). However, the computational effort is still high and not suitable for real-time or nearly real-time evaluation, which is usually required for onboard monitoring algorithms.

These reasons motivate the development of lower fidelity simulations of the system, able to run in a fraction of the computational time required by the HF model, while retaining an acceptable level of accuracy in simulating both nominal and faulty operation of the servoactuator. In the proposed methodology, the LF1 model is employed to collect the matrix of informative behavior maps for the training dataset. The model LF2 (Section 3.3) is used to compute the dynamical response of the actuator within the assessment of the system health condition, during the evaluation of the Remaining Useful Life (as described in Section 2.4).

A first simplified model is implemented by replacing the three-phase models of the inverter and motor (Sections 3.1.2 and 3.1.3) with an equivalent single-phase formulation.

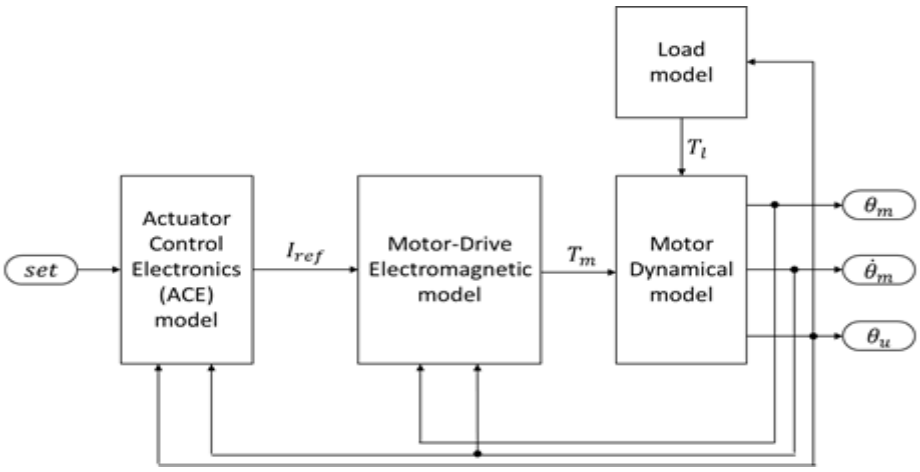

Figure 3.22: Layout of the LF1 model

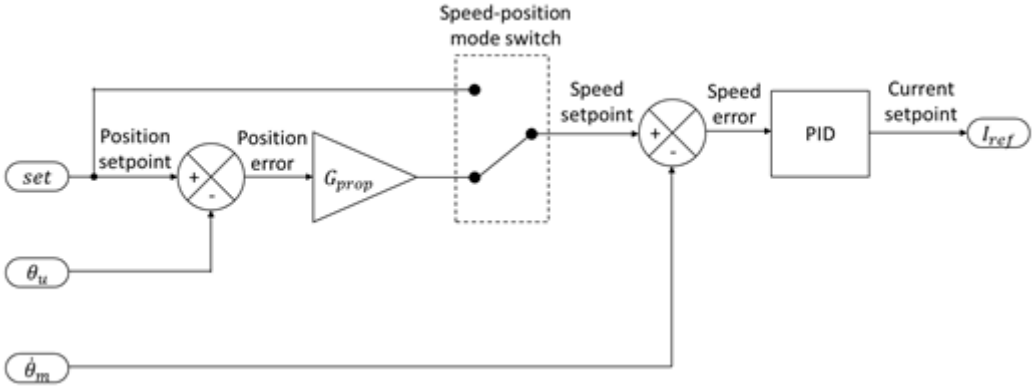

Figure 3.23: Block diagram of the controller subsystem.

The layout of the LF1 model is shown in Figure 3.22, and includes the following subsystems:

- The controller compares the position setpoint with the measured position and velocity, and computes a torque/current setpoint I_{ref}
- The electromagnetic model accepts as an input the current setpoint and rotor position to evaluate the torque generated by the motor, accounting for the bounds resulting from the limited supply voltage, magnetic flux saturation, and the possible presence of short circuit or eccentricity faults.
- The mechanical model of the motor and transmission computes position and velocity of the rotating assembly, accounting for the non-linearities resulting from dry friction, backlash and mechanical end-stops.

These subsystems are described in detail in the following sections.

3.2.1 Controller model

The controller model shares a similar architecture to its HF counterpart, as shown in Figure 3.23. The position setpoint is compared to the measured position, and the error, through a proportional gain, results in a velocity setpoint. Then, PID controller evaluates the current setpoint I_{ref} for the electromagnetic model subsystem.

With respect to the HF model, the PID controller has the simplified transfer function:

$$I_{ref} = G_P e_v + \frac{G_I e_v}{s} + G_D e_v s \tag{3.36}$$

and does not integrate any anti-windup or derivative filtering strategies. The dynamic behavior of the controller will therefore be slightly different from the HF model, but the discrepancy will only emerge in a relatively small fraction of the whole operating envelope of the actuator. Then, the simplified model can be used as a surrogate of the HF model for monitoring purpose or for collecting the training dataset for FDI, assuming to exclude from the diagnostic analysis the operating condition that lie outside the applicability region of the LF1 model. Additionally,

the additive noise on the current setpoint, implemented in the HF model, is neglected in the LF1 one.

3.2.2 Equivalent single-phase EM model

The first LF model employs a single-phase formulation with an overall behavior equivalent to the three-phase HF model of the power electronics and motor. The electromagnetic model is represented schematically by the block diagram of Figure 3.24.

The current setpoint I_{ref} is compared to the actual current flowing in the motor, and a simplified on-off control applies ground or supply voltage to the motor. The stator is modeled as a RL circuit, with governing equation:

$$V_m - k\omega = Ri_m + L\frac{di_m}{dt} \tag{3.37}$$

where $V_m = \pm V_s$ is the voltage applied to the stator, V_s is the supply voltage, k is the back-EMF coefficient, ω is the angular rate of the rotor, R and L are the equivalent single-phase resistance and inductance of the stator respectively, and i_m is the motor current. The simple control law for the current loop is expressed by:

$$V_m = \begin{cases} V_s, & i_m < I_{ref} \\ 0, & i_m = I_{ref} \\ -V_s, & i_m > I_{ref} \end{cases} \tag{3.38}$$

In order to obtain from the LF1 model a behavior comparable to the HF model, the parameters k, R and L are set as follows:

$$\begin{cases} R = 2R_s \\ L = 2L_s \\ k = k_v \end{cases} \tag{3.39}$$

This way, the supply-to-ground resistance of the motor is preserved, as well as the maximum stator current and RL characteristic time $\tau_F = L/R$. Additionally, the LF1 model shares the same current-torque characteristic of the HF model, by using the same nominal back-EMF coefficient.

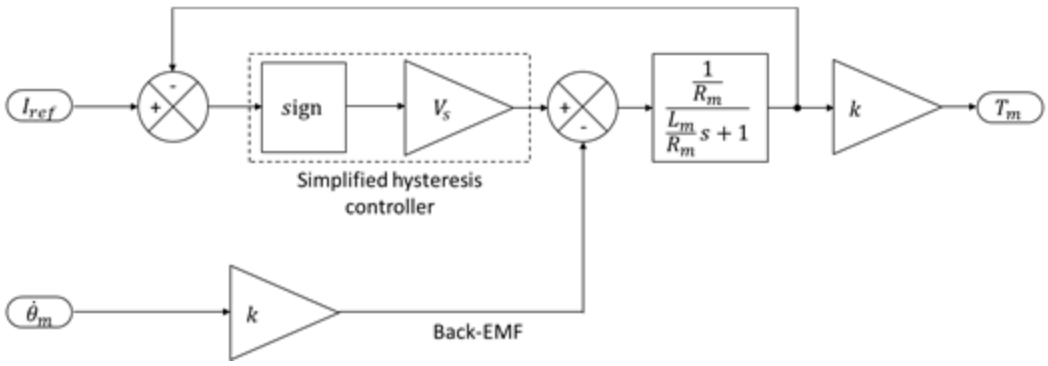

Figure 3.24: Block diagram of the LF1 Electromagnetic model

The motor torque T_m is computed starting from the stator current, and limited by the saturation of magnetic flux, which is expressed by a maximum torque value $T_{m,max}$:

$$T_m = \begin{cases} -T_{m,max}, & T_m < -T_{m,max} \\ ki_m, & -T_{m,max} \leq T_m \leq T_{m,max} \\ T_{m,max}, & T_m > T_{m,max} \end{cases} \tag{3.40}$$

The equivalent single-phase electromagnetic model significantly reduces the computational time of the simulation. In the HF model, the solution of the three-phase circuit requires an iterative procedure to be run at each time step, since the Y-connected phases may suffer from an imbalance due to the presence of faults and Y-Δ conversions cannot be employed. Conversely the LF model allows for a direct solution of each time step.

On the other hand, the single-phase formulation complicates the implementation of electrical faults affecting the separate phases of the full model in a different way, such as partial short circuit or eccentricity. Additionally, a suitable strategy to obtain comparable signals for the motor currents and voltages from the two models is necessary. These two issues are addressed in the following sections.

3.2.3 Equivalent current of the HF model

In order to compare the three-phase currents and voltages of the HF model to the single-phase formulation of the LF model, a conversion strategy is required. In this work, this problem is addressed by defining an equivalent single-phase current signal I_{3eq} as a combination of signals from the HF model. The definition of I_{3eq} varies between the BLDC and PMSM models.

In the BLDC model, an equivalent current is defined as the current flowing through each of the two phases that are active at any time. Since the currents are equal in modulus and have opposite signs, only their modulus is retained. This signal is computed within the HF model by evaluating the following expression at each timestep:

$$I_{3eq} = \frac{1}{2}(|i_A| + |i_B| + |i_C|) \tag{3.41}$$

Note that the $1/2$ coefficient is needed since only two phases out of three have non zero current at any time. This current signal has the same behavior of, and can be compared to, the modulus of the single-phase current computed by the LF model.

In the PMSM model, the I_{3eq} signal is equivalent to the quadrature component reconstructed from the measured stator currents. To recover the quadrature current component from the measured three-phase currents, the Clarke-Park transformations are employed:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = [A] \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = [A][B] \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{3.42}$$

where:

$$[A] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.43}$$

is the Park matrix, and:

$$[B] = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{3.44}$$

is the Clarke matrix. Note that, in the HF model, the reconstructed quadrature and direct currents do not necessarily coincide with the commanded current components described in Section 3.1.2.2, due to the dynamics of the hysteresis controllers and RL circuits. The quadrature current computed above is proportional to the motor torque, and can be compared to the single-phase current of the LF model.

The hysteresis control of the current loops produces a significant ripple that is not easily reproducible (with the same amplitude, frequency and phase) by an LF model. Since this may lower the accuracy of the comparison between the HF and LF models within a monitoring algorithm, the current signal is low-pass filtered. Specifically, a third order filter was chosen, with transfer function:

$$\frac{y}{x} = \frac{1}{(\tau_F s + 1)^3} \tag{3.45}$$

This filter allowed to reduce significantly the amount of noise on the current signals, while retaining the information related to the operating condition of the motor and the possible presence of fault modes. As a side effect, the filter introduces a delay that may cause a discrepancy between the LF and HF models. To compensate for this delay, the same filtering is applied to the LF model, in order to keep the two signals synchronized.

3.2.4 Shape functions for simulation of faults

The equivalent single-phase model of the motor requires modeling the effect of some of the failure modes in a non-physical manner. Specifically, partial phase short circuit and static rotor eccentricity, as described in Sections 3.1.6, affect each phase in a different way.

A first cut approach for modelling this behavior consists in averaging the effect over the three phases: this way, the resistance, inductance and back-EMF coefficient of each phase are modified statically, as the mean value of the three phases:

- In presence of partial short circuit, the equivalent single-phase resistance is $R = 2/3(R_A + R_B + R_C)$, and the equivalent single-phase inductance is $L = 2/3(L_A + L_B + L_C)$; the back-EMF coefficient of the LF model is $k = (k_A + k_B + k_C)/3$.
- Static eccentricity cannot be simulated with this approach since, the resistance and inductance are not modified; the back-EMF coefficient could

be averaged over one revolution and for the three phases, but the sinusoidal distribution of the air gap only results in a ripple with null average.

This approach proved to be highly inefficient and not suitable to produce an acceptable surrogate of the HF model. Figure 3.25 compares the motor current of the HF and LF models, in nominal condition and under the effect of partial short and eccentricity faults; the previously mentioned, revolution-averaged approach for fault simulation is employed for the LF model. Clearly, this approach is not acceptable: the characteristic ripple produced by faults is not modelled. The partial short circuit results only in a net increase of current, while eccentricity cannot be modelled at all.

A more accurate model of the electrical faults is implemented leveraging a modulation of the motor parameters as a function of the rotor angular position; this solution was initially implemented in [117] for BLDC motors, and extended to PMSM in [118]. Specifically, the motor resistance, inductance and back-EMF coefficient are modulated over the rotor revolution by shape functions sensitive to faults, intended to mimic the response of the HF model.

The shape function for the partial phase short circuit fault is determined considering the active phases for a given rotor position. For the BLDC motor, the commutation sequence yields the following expression for the modulating function ϕ_{SC}:

$$\phi_{SC} = \begin{cases} \dfrac{N_A + N_B}{2}, & -\dfrac{\pi}{6} + n\pi < \theta_e \leq \dfrac{\pi}{6} + n\pi \\ \dfrac{N_B + N_C}{2}, & \dfrac{\pi}{6} + n\pi < \theta_e \leq \dfrac{\pi}{2} + n\pi \\ \dfrac{N_A + N_C}{2}, & \dfrac{\pi}{2} + n\pi < \theta_e \leq \dfrac{5}{6}\pi + n\pi \end{cases} \tag{3.46}$$

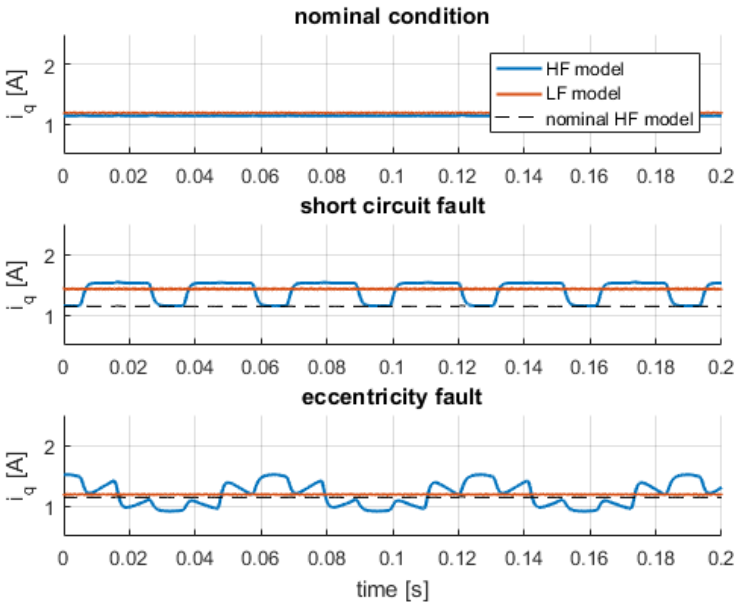

Figure 3.25: Comparison between the HF model LF1 model with the first cut approach for the simulation of electrical faults

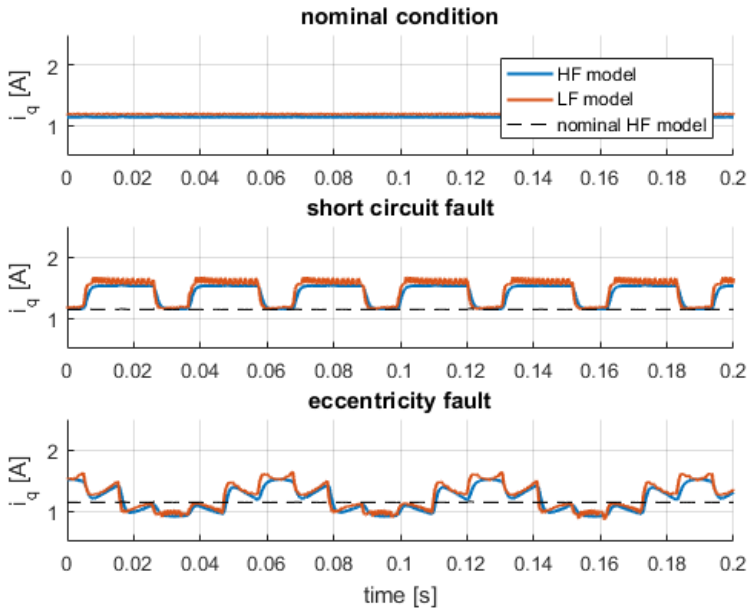

Figure 3.26: Dynamical response of the HF and LF models in nominal conditions (top). Dynamical response of the HF and LF models, in presence of 50% short circuit on phase A (middle). Dynamical response of the LF and HF models, in presence of static eccentricity of 50% the nominal air gap width, aligned with the axis of symmetry of phase A (i.e. $\boldsymbol{\zeta = 0.5}$, $\boldsymbol{\psi = 0}$) (bottom).

Then, ϕ_{SC} assume the average value of the fractions N of healthy windings associated with the phases that are active for any given rotor angle θ_e.

In the case of the PMSM model, all three phases are active at any time, with a different fraction of the total current setpoint. Hence, the modulating function assumes the following form, found by trial and error and standing that $\phi_{SC} \leq 1$ and $\phi_{SC} = 1$ when all phases are healthy (i.e. $N_{A,B,C} = 1$):

$$\phi_{SC} = \frac{2}{3}\left[N_A \sin^2\theta_e + N_B \sin^2\left(\theta_e - \frac{\pi}{3}\right) + N_C \sin^2\left(\theta_e + \frac{\pi}{3}\right)\right] \tag{3.47}$$

Figure 3.26 shows the behavior of this approach. In Figure 3.26 (top), the response of the LF model in nominal conditions is shown to be unaffected by the shape functions. The middle graph of Figure 3.26 shows the response of the LF and HF models in presence of 50% short circuit on phase A. Comparing this to Figure 3.25 (middle), clearly the proposed approach is able to capture the effect of short circuit fault with high accuracy, and without increasing significantly the computational burden.

As regards static eccentricity, a similar approach is adopted. For the BLDC motor, the expression of the modulating function is found by trial and error in the form of:

$$\phi_E = 1 - \zeta_{LF}[\cos(\theta_e + \psi) + \sigma(6\theta_e - \pi)\sin(\theta_e + \psi)] \tag{3.48}$$

where $\sigma(x)$ is the sawtooth function:

$$\sigma(x) = \frac{x}{2\pi} - \text{floor}\left(\frac{x}{2\pi}\right) \tag{3.49}$$

ψ is the angular displacement between the axis of symmetry of phase A and the location of the minimum air gap width, and ζ_{LF} is a function of the normalized eccentricity amplitude ζ. A linear regression, explained in deeper detail in Section 3.4, allowed to determine the relationship $\zeta_{LF} \approx 0.42\,\zeta$. The step discontinuities of ϕ_E, introduced by the sawtooth function, reflect the abrupt commutations happening in the operation of BLDC motors. The same approach applied to the PMSM yields:

$$\phi_E = 1 - \zeta_{LF} \cos(\theta_e + \psi) \tag{3.50}$$

The comparison between the HF and LF models is shown in the bottom graph of Figure 3.26, with the effect of static eccentricity of 50% the nominal air gap width, aligned with the axis of symmetry of phase A. A significant increase in accuracy can be noticed with respect to the simplified formulation of Figure 3.25 (bottom).

3.2.5 Mechanical Model

The mechanical model is a second order dynamical model, implemented in a similar way as that of the HF model. The behavior of the model is described by the differential equation:

$$T_m - T_l = J_m \frac{d^2\theta_m}{dt^2} + C_m \frac{d\theta_m}{dt} + K_m \theta_m \tag{3.51}$$

where J_m, C_m and K_m are the inertia, viscous friction coefficient and stiffness of the system, respectively. The nonlinearity of Dry Friction, Backlash and endstops are considered as in Section 3.1.4.

Differently from the HF model described in Section 3.1.4, the viscous contribution is linear (i.e. C_m is a constant) and the elastic component $K_m\theta_m$ is introduced to integrate the load model: Equation (3.17) expresses the external load as a function of the flight condition $\boldsymbol{x}$ and the position of the control surface:

$$\Delta M_h = [C] \begin{bmatrix} \Delta V \\ \Delta\alpha \\ \Delta\theta \\ \Delta q \end{bmatrix} + [0 \quad D_{12}] \begin{bmatrix} \Delta\delta_t \\ \Delta\theta_u \end{bmatrix} \tag{3.51}$$

However, considering simulations with a short duration compared to the characteristic time of the aircraft modes (which is in the order of some seconds), the contribution of the flight condition is almost constant. Then, the load on the actuator can be expressed as the sum of a constant contribution depending on the flight condition (included in the T_l term of Equation (3.51)) and an elastic contribution $K_m\theta_m = D_{12}\theta_u/i$, where i is the transmission ratio.

3.3 Second Low Fidelity (LF2) Model

The LF1 model described in Section 3.2 leverages a single-phase equivalent circuit for the motor (instead of the three-phase HF model) and does not require iterative solution of the stator circuit. Therefore, the number of operations to be

performed at each timestep is small, and the computational burden is greatly reduced with respect to the HF model.

However, an even more simplified model is desirable for some applications. Specifically, the assessment function for the estimation of Remaining Useful Life requires an iterative execution of a dynamical simulation of the system: consequently, the HF and LF1 models involve computational times which are impractical even for offline computations. The minimum timestep of the LF1 model is limited by the time constant τ_{RL} of the RL circuit which emulates the electrical behavior of the stator coils:

$$\tau_{RL} = \frac{L}{R} \tag{3.52}$$

In most practical applications, the inductance of the stator is very small compared to the resistance, resulting in a small time constant and a fast response of the stator circuit. Then, the integration timestep must be very small, usually at least one order of magnitude smaller than the time constant, to achieve acceptable numerical accuracy. This translates in a rather long computational time for the model.

As τ_{RL} is small, a second LF model (LF2) has been developed assuming that the inductance of the stator is negligible. This allowed to increase the timestep and speed up the computation; under the right conditions, the loss of accuracy is acceptably small. The model was initially presented in [119] for the BLDC model; the extension to PMSM is trivial.

3.3.1 Simplified model for the current loop to increase integration time step

The electromagnetic subsystem of the LF1 model, shown in the block diagram of Figure 3.24 and described in Section 3.2.2, behaves as a current control loop following the current setpoint I_{ref}. During normal operations, current loop produces a delay in the order of τ_{RL}. As the motor speed increases, so does the back-EMF, until the supply voltage is no more enough to keep the commanded current flowing in the motor. In this saturation condition, the stator current is a function of the back-EMF, supply voltage and stator resistance.

A block diagram for the electrical subsystem of the second LF model is shown in Figure 3.27. Neglecting the effect of stator inductance, the equivalent circuit of the motor is subject to the following governing equation:

$$V_m - k\omega = Ri_m \tag{3.53}$$

When the supply voltage is sufficient to overcome the back-EMF, the current control loop achieves a quasi-instantaneous regulation of i_m, so that the motor current is equal to the current setpoint I_{ref}:

$$i_m = I_{ref} \tag{3.54}$$

In saturation, the maximum and minimum net voltages (including the effect of back-EMF) that can be applied to the stator are respectively:

$$V_{max} = V_S - k\omega \tag{3.55}$$

$$V_{min} = -V_S - k\omega \tag{3.56}$$

where V_S is the supply voltage. Then, the stator current is bounded between:

$$i_{max} = \frac{V_{max}}{R} \tag{3.57}$$

and:

$$i_{min} = \frac{V_{min}}{R} \tag{3.58}$$

This behavior is achieved by the dynamical saturation block shown in Figure 3.27. Note that this block is available in Simulink, and allows to define variable upper and lower bounds for the output.

3.4 Assessment function

Model LF2 is employed within the assessment function $\phi_a(\boldsymbol{k})$ [37] that permits to evaluate the performance of the system affected by the fault combination $\boldsymbol{k}$ and determine if it still meets its requirements. The assessment function runs the LF2 model in order to compute:

- *The no-load speed of the actuator*: this is done by sending a step command large enough to saturate the position control loop of the system.
- *The stall load*: to do so, the system is actuated at a low, constant speed (in order to limit inertial effects) while the load increases linearly with time. The stall load is the minimum load needed to stop the actuator.
- *The stability margins*: the LF model is ran iteratively in open-loop mode, with sine wave commands of varying amplitude, to build its open-loop Bode plot. Then, the phase margin is the distance from the system's phase to -180° when the gain is 0dB; the gain margin is the gain of the system corresponding to a phase of -180°.

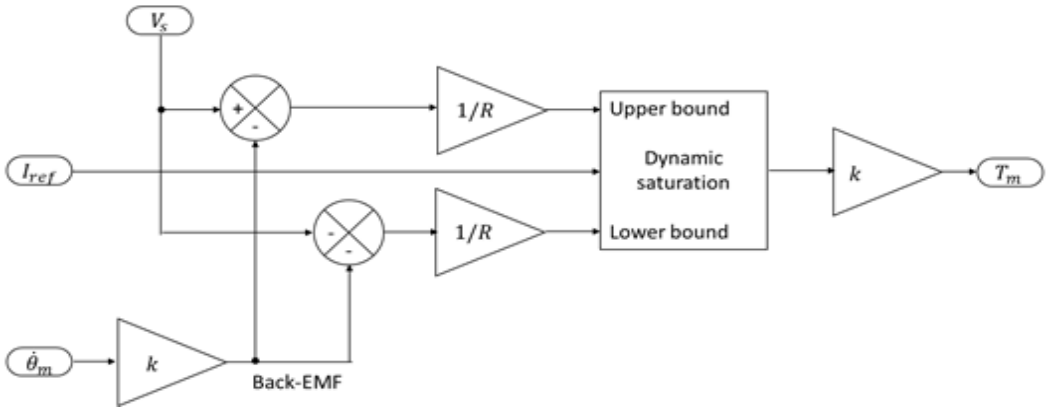

Figure 3.27: Electrical subsystem of the second LF model, highlighting the dynamical saturation block employed to replace the current control loop of the first LF model. The rest of the model is in common with the first LF model (Section 3.2).

These performance parameters are then compared to a set of requirements. If all the requirements are met, the assessment function associates a "healthy" label to the system; otherwise, a "faulty" label. The whole process requires approximately 50 evaluations of the LF2 model, for a computational time in the order of 40 seconds.

Chapter 4

Estimation of actuator load

Diagnostic and prognostic algorithms analyze the response and behavior of a system in order to determine its health condition and estimate its Remaining Useful Life (RUL). However, the behavior of any system is influenced by a multitude of factors other than the system health. Mainly, we can refer to those as the operating and environmental conditions: the formers are strictly related to the particular operating scenario of the equipment (for example, the command time history for an actuator); the latters are external factors for which a limited or null control authority is available (such as vibration and external temperature).

Among those factors, the external load has a particularly significant impact on the performance of actuation devices. Specifically, actuator load influences the dynamical response of the system, especially when its magnitude gets close to the stall force of the actuator; this effect can be confused with a fault in the equipment, e.g. an increase of friction between mechanical elements of the transmission. Additionally, external load affects the rate at which degradation of components propagates: a high load causes increased internal forces, pressures, friction and temperatures, reducing the expected life of the system.

For some actuation devices, external load has a repeatable time history in nominal conditions. For example, this is often the case for servoactuators employed within the industrial automation context. Those systems are usually subject to a periodical time history regarding both command and external loads. Therefore, the effect of operating conditions can be neglected when monitoring the dynamical behavior of the system, and discrepancies from the nominal response can be usually ascribed to faults of the equipment.

On the other hand, servosystems employed in vehicles, including aircraft flight control actuators, are usually subject to unpredictable operating conditions. The command time history depends on the input from the pilot or the flight control computer, while the external load is determined mainly by the aerodynamic flow. For health monitoring purposes, the command time history can be easily logged, since on a fly-by-wire system it is already in the form of a digital signal; its knowledge can be leveraged to improve the estimate of the condition of the actuator. Conversely, the external load is not commonly available since it is not measured by a sensor. Usually, this prevents the execution of advanced heath monitoring algorithms in real-time. Several approaches to PHM [120, 40, 121] propose to measure the dynamical response of servoactuators during either pre-flight checks or planned maintenance interventions. In this condition, the wind speed is null and so is the actuator load: hence, the behavior of the system is not influenced by

external factors and can be employed to estimate faults. Real-time processing is usually reserved to properly called diagnostic algorithms, which aim to identify complete failures of the system in order to manage redundancies [122, 123]. However, as highlighted by [124, 125], the adoption of PHM strategies in real-time would provide significant benefits for maintenance planning and logistic management of the fleet, as well as allow a dynamic reconfiguration of the mission accounting for the residual capability of the system.

Ideally, three alternative techniques could be employed to estimate the external load on flight control actuators.

The most direct approach consists in the installation a dedicated sensor to measure load, namely a load cell in series with the actuator. This method provides a good accuracy, and requires limited signal conditioning and post processing to obtain a useful measurement: a Wheatstone bridge arrangement of the sensitive elements of the load cell, followed by an amplifier, low-pass filter and analog-to-digital converter is usually all the hardware required for the operation of the sensor. A load-cell installed between the actuator and aerodynamic control surface would need to bear all the force exchanged between the two components, and would introduce a weak spot in the mechanical transmission. In case of mechanical failure of the load cell, the control surface would be disconnected from the actuation line, with obvious safety and reliability issues. As a result, direct measurement of the actuator load using a load cell is not commonly employed, except for airframes purposely instrumented for flight-testing.

Alternatively, the aerodynamic load can be estimated from the flight condition and an aerodynamic model of the aircraft. The model could leverage information from the air data system and inertial measurement unit in order to determine the relative wind speed and direction to the control surfaces; then the pressure distribution on the controls can be integrated to estimate the hinge moment. This approach has the advantage of not requiring additional hardware to be installed onboard, except the computational resources needed for processing, if the estimation is to be performed in real-time. As a drawback, it is difficult to achieve an acceptable accuracy with this method. Complex nonlinear and unsteady effects, such as turbulence, cannot be accounted for if the aerodynamic model employed for the estimation of hinge moment is too simplified: for example, this is the case of lumped parameter models based on aerodynamic derivatives. The employ of more detailed fluid dynamic models, such as those based on CFD, rapidly leads to computational times not suitable for real-time evaluation, while the uncertainty may be still unacceptable to estimate the effect of load on the actuator dynamical response.

A third possible approach suggests estimating the aerodynamic load on the flight control actuators starting from the deformation of the airframe. Aerodynamic and inertial loads result in a strain field on the structure, which can be measured with a set of strain gages. By placing the sensors in optimal locations [42, 50] and mapping the strain information to the loads, it is possible to determine the hinge moment on each control surface. The computation essentially implies the application of an inverse Finite Element Method (FEM), and the associated

computational burden is intermediate between the two methods described previously, whereas machine learning and reduced order modelling technique could help speeding up real-time computations. The accuracy is potentially good, and close to that achievable with a direct measurement, provided that the individual sensors are placed in strategic locations where the deformation is highly affected by the hinge moment and less by other load distributions. The areas near the mechanical connection between the actuator and airframe are usually good candidates for the placement of strain gages. However, this method requires the installation of a large number of strain sensors (at least in the order of tenths). Traditional resistive strain gages require individual wiring for each sensor, and relatively heavy equipment for signal conditioning and acquisition. Additionally, the length of electrical cables from the sensors to the acquisition equipment is potentially prone to electromagnetic disturbance and variation of resistance due to temperature: both these effects can alter the readings, reducing the accuracy of the system. These disadvantages result in this approach being employed only in flight-testing of newly developed aircraft. Extensive use on commercial flight is usually not applicable for health monitoring purposes.

Neither of the methods described above for the estimation of hinge moment is commonly viable for practical, large scale applications, due to the multiple issues related to accuracy, system complexity, cost and weight, safety and reliability. Although, innovative sensor technologies may allow for a viable approach to in-flight measurement of the airframe deformation, enabling the real-time estimation of actuator loads for health monitoring. The following Sections propose a possible algorithm for the estimation of hinge moment from structural strain, and describe optical strain sensors that overcome the main disadvantages of traditional strain gages.

4.1 Load reconstruction from deformation of the structure

An example of simplified algorithm for determining the actuator load starting from the FBG measurement is provided as a proof of concept for the feasibility of the method. More advanced techniques are currently under development and will be published in future works. A simple aerodynamic model (Section 4.1.1) of a wing with two control surfaces (an aileron and a flap) is employed in combination with a one-dimensional structural model (Section 4.1.2) to obtain the strain measurements and the hinge moment acting on the aerodynamic surfaces in different flight conditions. Then, Section 4.1.3 shows a simple strategy to estimate the hinge moment starting from the strain measurements.

4.1.1 Aerodynamic model

The distribution of lift and pitching moment on the wing is estimated with Prandtl's lifting line theory. Despite less accurate than a complete CFD model, this

method offers a simpler implementation and is considered suitable for the sake of this study.

The wing is modelled with a distribution of chord c, angle of attack α and airfoil (i.e. the lift and moment coefficients versus angle of attack) along the wingspan. The effect of the control surfaces deflection is modelled as a local variation of angle of attack. To solve the lift distribution numerically, the wing is discretized spanwise in wing elements, each of which is assumed to be subject to a uniform flow velocity and angle of attack. The lift generated by an individual element $L(y)$ can be expressed as a function of dynamical pressure and the local lift coefficient $C_L\big(y, \alpha(y)\big)$:

$$L(y) = \frac{1}{2}\rho V^2 (c(y)\, dy) C_L\big(y, \alpha(y)\big) \tag{4.1}$$

where ρ is the air density, V the relative air speed and the local angle of attack:

$$\alpha(y) = \alpha_0 + \alpha_g(y) + \alpha_c(y) + \alpha_i(y) \tag{4.2}$$

is the sum of the global angle of attack α_0, the contribution of the geometrical wing twist α_g, the contribution of the control surface deflection α_c, and the contribution of the induced speed $\alpha_i = V_i/V$. Alternatively, $L(y)$ can be written as a function of the circulation $\Gamma(y)$ across the wing element:

$$L(y) = \rho V \Gamma(y) \quad \Leftrightarrow \quad \Gamma(y) = \frac{L(y)}{\rho V} \tag{4.3}$$

Eventually, the distribution of circulation $\gamma = d\Gamma/dy$ allows computing the induced speed distribution across the wing span:

$$V_i(y) = \int_{-b/2}^{b/2} \frac{\gamma}{2\pi(y-s)} ds \tag{4.4}$$

The numerical solution is obtained by initializing $V_i(y)$ to a first guess distribution, and iterating Equations (3.1) to (3.4) until convergence is reached. The net aerodynamic load on the control surfaces L_a is assumed equal to the difference between the total lift distribution $L(y)$ and lift distribution computed without considering the contribution of the control surface.

The structural load on the wing is modeled as the superposition of the lift distribution of the wing without the control surfaces, applied at the aerodynamic center of the airfoil, and the distribution of pitching moment:

$$M(y) = \frac{1}{2}\rho V^2 (c^2(y)\, dy) C_M\big(y, \alpha(y)\big) \tag{4.5}$$

The lift on the aerodynamic surfaces is applied to the wing structure as concentrated loads L_{ac} at the spanwise positions corresponding to the location of hinges. Additionally, a concentrated torque $M_a = L_{ac} c_a/2$, where c_a is the chord of the control surface, is applied at the location of the actuator. This is equal to assume a uniform pressure distribution on the control surface, so that the resulting force is applied at half of its chord. Figure 4.1 shows the load distribution along the wing span for one particular flight condition taken as an example. A 10m span, 1m

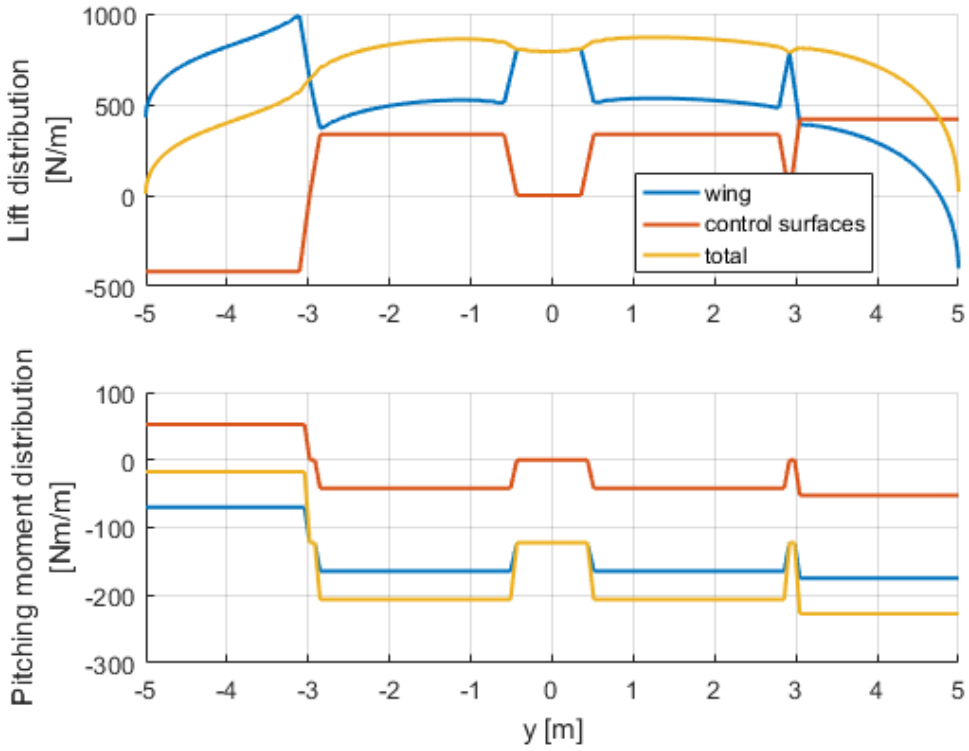

Figure 4.1: Distribution of aerodynamic load along the wing span

chord wing with a NACA 2412 airfoil has been considered, flying at 50m/s with 5° angle of attack, 5° aileron deflection, and 10° flap deflection. The total aerodynamic force is given by the superposition of contributions from the control surfaces and fixed wing.

4.1.2 Structural model

The deformation of the structure resulting from the aerodynamic loads L_y computed in Section 4.1.1 is evaluated with an ideal monocoque beam formulation. The cross section of the wing is modelled as a set of stringers and spar caps, bearing the normal stresses, and connected by planar skin panels that transfer shear stresses. A schematic representation of the wing cross section is provided in Figure 4.2. The shear stress T_z, bending moment M_x, and torsion moment M_t are computed with Euler's beam theory:

$$T_z(y) = \int_y^{b/2} L_y(s)ds \tag{4.6}$$

$$M_x(y) = \int_y^{b/2} sL_y(y-s)ds \tag{4.7}$$

$$M_t(y) = \int_y^{b/2} (M + M_a)(s)ds \tag{4.8}$$

where s is an auxiliary integration variable and $L_y = L + L_{ac}$ is given by the superposition of the distributed load of the wing and the concentrated load of the

control surface. No shear forces along the x direction, neither bending moments along the z direction are considered, as the effect of aerodynamic drag on the global deformation of the wing is deemed negligible. The resulting T_z, M_x and M_t for the example load case of Figure 4.1 is shown in Figure 4.3.

Navier's Equation for the stress distribution allows determining the normal stresses in the i-th stringer or spar cap:

$$\sigma_i = \frac{M_x J_z}{J_x J_z + J_{xz}^2} z_i + \frac{M_z J_x}{J_x J_z + J_{xz}^2} x_i \tag{4.9}$$

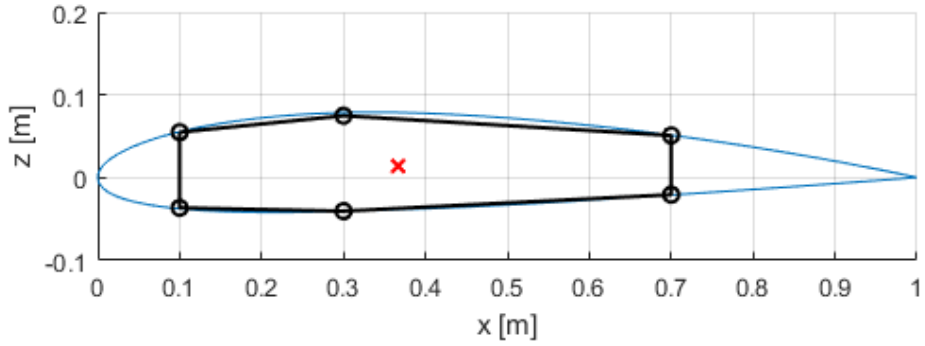

Figure 4.2: cross section of the wing in the ideal monocoque beam model. The blue contour is the NACA 2412 airfoil employed for this preliminary study; the black lines are the shear-bearing panels, the black circles are the stringer positions and the red cross is the center of gravity of the cross section.

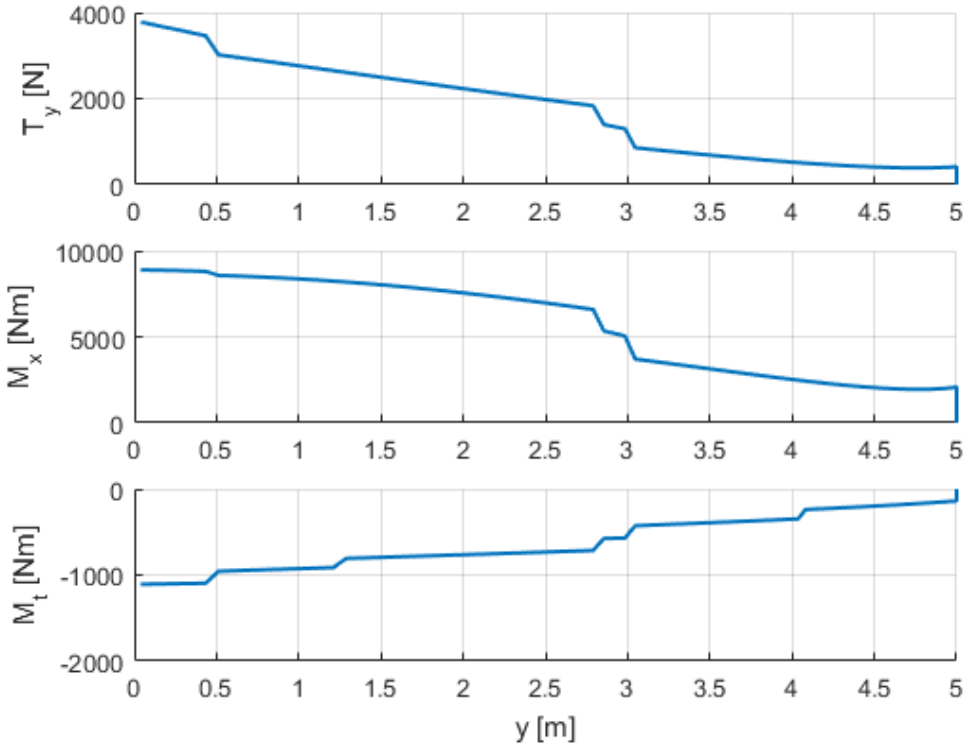

Figure 4.3: shear force, bending moment and torsion moment diagrams

where J_x, J_z and J_{xz} are the moments of inertia of the wing cross section. Since in the considered case study $M_z \approx 0$, the partial derivative along y of the normal stress σ_i is:

$$\frac{\partial \sigma_i}{\partial y} = \frac{T_z J_z}{J_x J_z + J_{xz}^2} z_i \tag{4.10}$$

The shear stresses in the skin panels and spar web can be computed by solving the system of equations including the equilibrium to translation along y of each stringer element and the equilibrium to rotation of the entire wing section.

$$\begin{cases} A_i \dfrac{\partial \sigma_i}{\partial y} = \sum_j \tau_j s_j \\ \sum_j \tau_j s_j A_j + M_t = 0 \end{cases} \tag{4.11}$$

where A_i is the area of the cross-section of the i-th stringer, s_j is the thickness of the j-th panel, and A_j is the area described by the j-th panel with respect to the origin of the reference frame. The distribution of normal and shear stresses on the wing are shown in Figure 4.4 for the example load case of Figure 4.1; the circles highlight a possible location of strain sensors, suitable to determine the actuator load from the airframe deformation.

4.1.3 Reconstruction of Hinge Moment from strain measurements

A simplified, linearized algorithm for estimating the hinge moment from the strain measurements is proposed here as a proof of concept for the feasibility of such method to determine the external load on flight control actuators.
The placement selected for strain sensors is shown in Figure 4.4. The driving criteria of this choice is to find locations where large variations of strain are

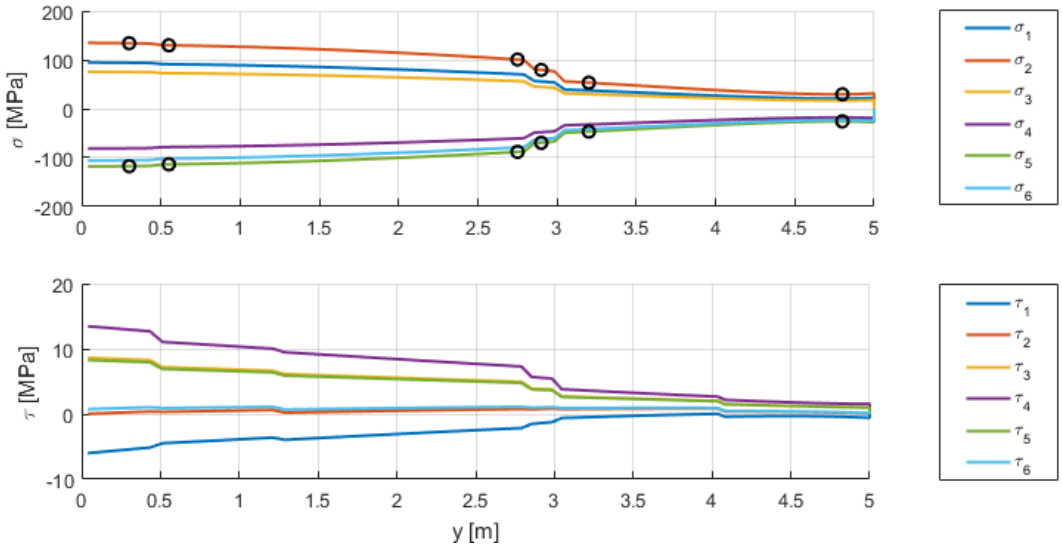

Figure 4.4: Distribution of normal (top) and shear stresses (bottom). The black circles in the top graph represent the location of 12 FBG sensors.

experienced with varying aerodynamic forces on the control surfaces. Additionally, the structural elements where strain sensors are installed shall be easily accessible to allow a simple manufacturing process. For these reasons, the sensors are located on the stringers, with a spanwise position close to the hinges of each control surface. A more detailed analysis may benefit from the placement of sensors near the interfaces between the airframe and flight control actuators, since most of the stress distribution in these areas is likely caused by the actuator load. However, the actuator mounts are accounted for in the structural model only as concentrated torque loads: the limited resolution of the simulation does not allow to predict accurately local stress concentrations, and measurements taken in correspondence of the servoactuators could not be estimated.

To find a relationship between the actuator load and strain measurements, a linear model is assumed with the formulation:

$$\mathrm{A}\boldsymbol{x} = \boldsymbol{\varepsilon} \tag{4.12}$$

$$\mathrm{C}\boldsymbol{x} = \boldsymbol{M}_h \tag{4.13}$$

where $\boldsymbol{x} = \left[q\ \alpha\ \omega_r\ \delta_f\ \delta_a\right]^\top$ is the flight condition, q is the dynamic pressure, α is the aircraft angle of attack, ω_r is the roll rate, δ_d is the flap deflection, δ_a is the aileron deflection, $\boldsymbol{\varepsilon}$ is a vector containing the strain measured by each sensor, and $\boldsymbol{M}_h = \left[M_f\ M_a\right]^\top$ is a vector containing the flap hinge moment M_f and the aileron hinge moment M_a. To determine the matrices A and C, a training set of 100 simulations with different flight conditions is collected. Then, the matrices are estimated to approximate in the least squares sense the known observations of the training set, with a process similar to that described in Section 2.4.2 for the fault propagation model. Given the simplified formulation proposed here, the matrices A and C could be computed analytically; however, in a real application scenario more detailed models would be required (e.g. a 3D Computational Fluid Dynamics (CFD) model for the aerodynamic loads and a Finite Elements Method (FEM) [126] for the structure), making an analytical approach infeasible. In addition, an effective hardware implementation of such in-flight load reconstruction techniques will need to account for the dynamical behavior of the wing structure [127].

The linear relationship described above allows to estimate the flight condition from the measurements $\boldsymbol{\varepsilon}$ by solving Equation (3.12); by substituting into Equation (3.13) the hinge moments can be determined. Figure 4.5 summarizes the result of this approach, tested on a validation set of 100 simulations (different from those of the training set). A random noise up to 5% of the signal amplitude has been added to the strain measurements of the validation set, in order to simulate the measurement errors due to vibrations and imperfect installation of sensors. Given the simple linearized formulation of the aerodynamic and structural models, the hinge moment estimate matches the actual one with good accuracy, usually within 2% of the actual load; the main contribution to the error is due to the noise simulating the measurement uncertainty.

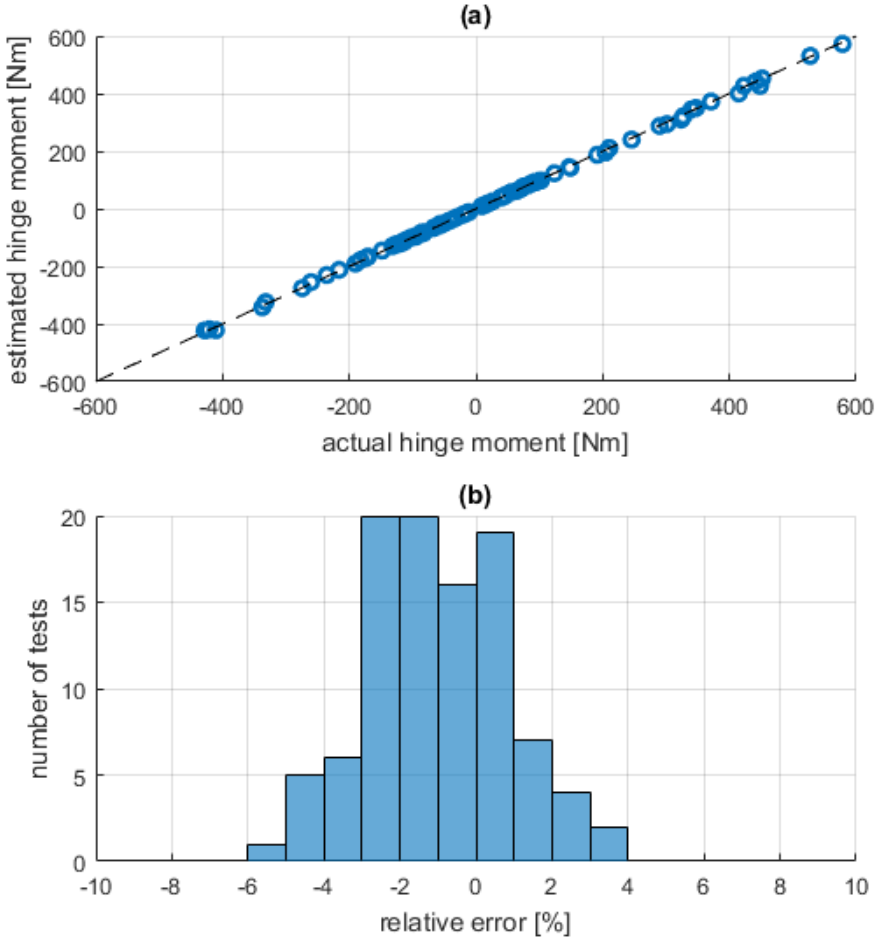

Figure 4.5: (a) comparison between actual and estimated hinge moment; (b) distribution of the relative error on the hinge moment estimate

4.2 Strain sensors based on Fiber Bragg Gratings (FBGs)

Several recently introduced technologies are available to employ optical fibers as sensors capable to measure various quantities. All these solutions rely on the correlation of an external quantity of interest (e.g. strain, temperature, humidity, concentration of chemicals) with some measurable characteristics of light transmitted through the sensing optical fiber (such as intensity, frequency and spectrum, phase, polarization).

Among these, interference (or Fabry-Perot) sensors [128] allow to measure the average value of an external quantity of interest across the entire length of the sensitive fiber; a point measurement is not usually possible with this arrangement. Intensity sensors rely on the variation of attenuation of light through the fiber, e.g. caused by mechanical bending. This solution allows for high sensitivity, but lacks a reference signal: if the attenuation of the fiber varies with aging, or the intensity of the emitting laser changes, the measurement drifts causing an error.

On the other hand, sensors based on Fiber Bragg Gratings (FBGs) [129, 130] offer highly accurate measurements of the local value of a quantity. The sensors are small and easily multiplexed, so a high spatial resolution can be achieved with low invasivity on the monitored system. For these reasons, this technology is investigated in this work as a possible candidate for the estimation of actuator load in flight, from measurement of the airframe deformation.

4.2.1 FBGs: Principle of operation

An optical fiber is a thin, layered wire, usually made of glass or polymers and employed for signal transmission. Starting from the outside, the fiber is composed by three layers with different properties, as shown in Figure 4.6:

- The coating is an external protective layer, usually made of polymeric materials although metals or ceramics can be used for specific applications, e.g. for resisting particularly harsh environmental conditions. The typical diameter of the coating is in the order of 250 microns.
- The cladding is a first glass layer, usually with a standard diameter of 125 microns. It is needed to direct and contain light inside the fiber.
- The core is the innermost layer of the fiber, and is made of glass with a higher index of refraction than the cladding. This way, light is trapped inside the core by total internal reflection at the core-cladding interface. For common, single-mode fibers the core has a diameter of 8 microns.

The refraction of light through a variation of refraction index is described by Snell's law:

$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2 \tag{4.14}$$

where α_1 and α_2 are the angles of the incident and refracted rays, as depicted in Figure 4.7, and n_1 and n_2 are the indexes of refraction of the two materials respectively. If $n_1 > n_2$, limit condition is found when the refracted ray should leave the interface at a right angle:

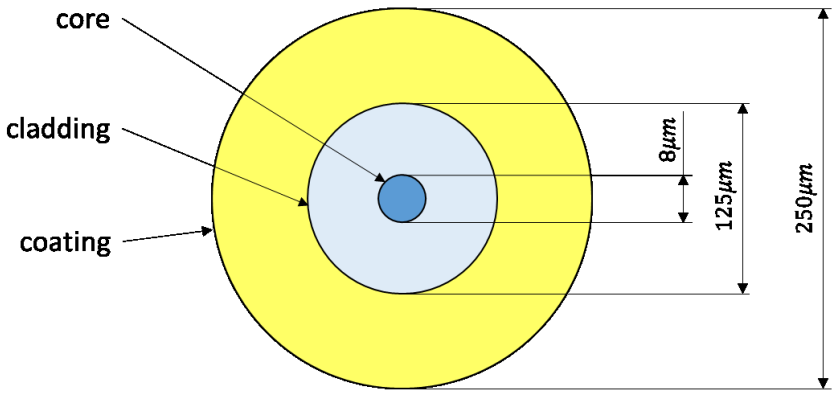

Figure 4.6: Typical cross-section of a common optical fiber for communications, highlighting the core, cladding and coating. The dimensions are referred to a standard, silica glass, single-mode fiber.

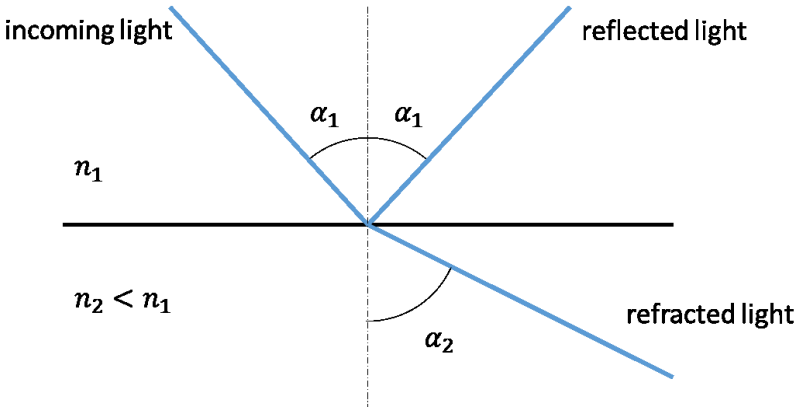

Figure 4.7: Path of the refracted and reflected rays in correspondence of a variation of refraction index, according to Snell's law.

$$\frac{n_1}{n_2}\sin\alpha_1 = 1 \tag{4.15}$$

In this condition, the α_1 angle assume the critical value $\alpha_{1,c}$. If the incident ray gets to the interface at an angle shallower than $\alpha_{1,c}$ (i.e. $\alpha_1 > \alpha_{1,c}$), it is reflected rather than transmitted. Optical fibers leverage this phenomenon, employing for the core a material with a higher refraction index than the cladding; this way, light travelling through the fiber is confined inside the core.

A Fiber Bragg Grating is a small section of the core (usually less than one millimeter in length) where the refraction index is modulated periodically [131, 132], with a spatial pitch Λ_g. When an electromagnetic wave crosses the grating, part of it is reflected in correspondence of each variation of the refraction index. If the incoming light assumes a specific wavelength, the reflected waves are in phase and combine via constructive interference, while the intensity of the transmitted wave decreases while crossing each interface. For any other wavelength, the reflected waves are out of phase and cancel out via destructive interference, while the transmitted wave keeps its intensity mostly unaltered.

The wavelength λ_B reflected by the grating is a function of its pitch Λ_g and its effective refraction index n:

$$\lambda_B = 2n\Lambda_g \tag{4.16}$$

As a result, when light with a given spectrum is sent through the grating, the transmitted spectrum lacks the wavelength λ_B, which is reflected back to the source (Figure 4.8).

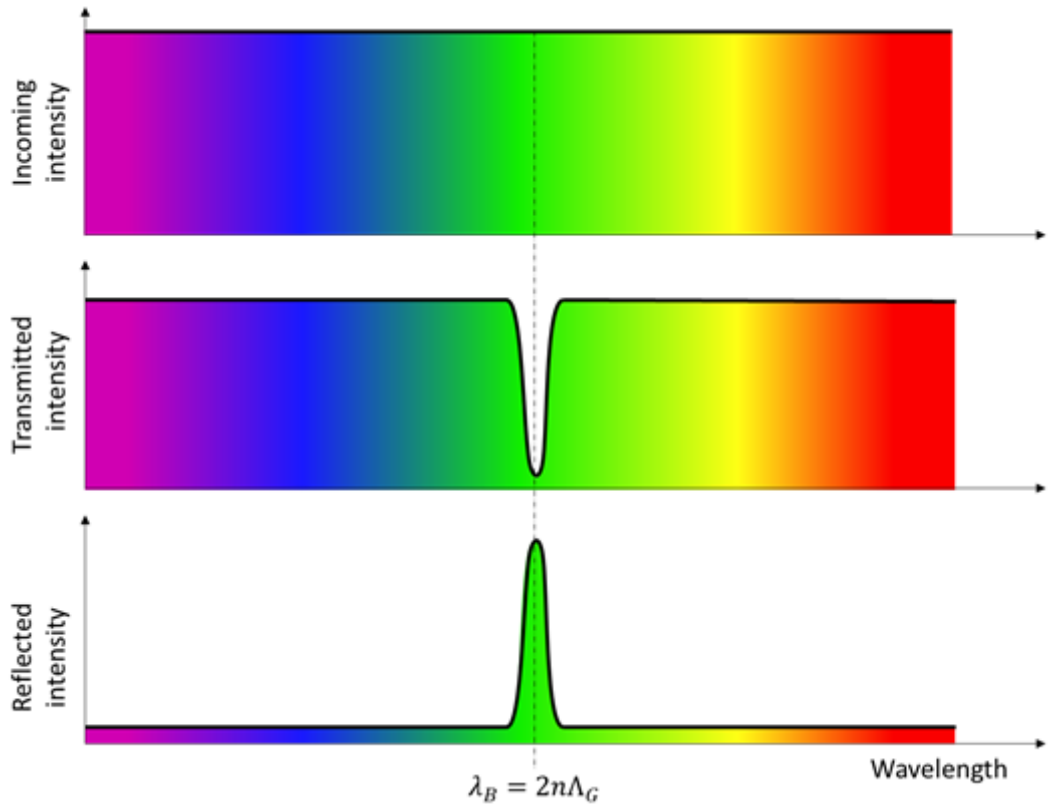

Figure 4.8: Transmitted and reflected spectra obtained with a Fiber Bragg Grating (FBG)

External phenomena can have an effect on the wavelength reflected by the FBG. Specifically, mechanical strain and temperature affect both the pitch of the grating (through mechanical deformation and thermal expansion respectively), and its refraction index. The dependency of the FBG wavelength from temperature variation ΔT and strain variation $\Delta\varepsilon$ is approximately expressed by:

$$\frac{\Delta\lambda_B}{\lambda_B} = (1 - p_e)\Delta\varepsilon + (\alpha_f + \zeta_f)\Delta T = C_\varepsilon \Delta\varepsilon + C_T \Delta T \tag{4.17}$$

where α_f is the thermal expansion coefficient of the fiber, ζ_f is its thermo-optic coefficient, and:

$$p_e = -\frac{1}{n}\frac{\partial n}{\partial \varepsilon} \tag{4.18}$$

is the photoelastic coefficient. The relationship between temperature, strain and wavelength can also be written in terms of a coefficient of strain C_ε and a coefficient of temperature C_T. As a result, the reading from an FBG sensor is a linear combination of the effects of strain and temperature. To obtain an accurate measurement of one quantity, either the other one must be controlled or its effect compensated in post processing.

For example, to measure temperature, the optical fiber can be installed in a way that its sensitive section, including the FBG, is left free-floating and not mechanically loaded, as shown in the schematic representation of Figure 4.9. This way, the deformation $\Delta\varepsilon$ is null, and temperature can be obtained as:

$$\Delta T = \frac{1}{C_T}\frac{\Delta\lambda_B}{\lambda_B} \tag{4.19}$$

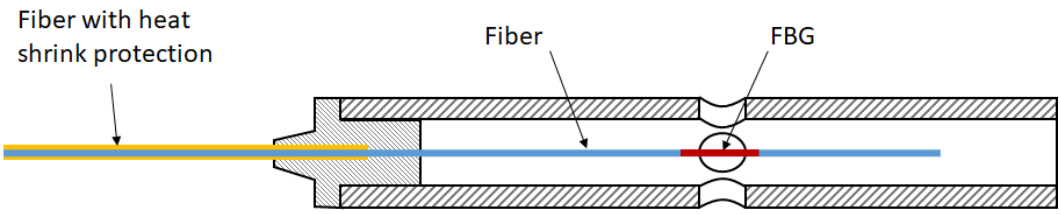

Figure 4.9: Example of FBG installation for temperature measurement.

If the FBG temperature sensor is bonded to a different material substrate, such as a component which mechanical deformation is known to be null, the expression for temperature is modified as follows:

$$\Delta T = \frac{1}{C'_T}\frac{\Delta\lambda_B}{\lambda_B} \tag{4.20}$$

where the new temperature coefficient C'_T accounts for the thermal expansion of the substrate α_m:

$$C'_T = \alpha_m + \zeta_f \tag{4.21}$$

Indeed, with this setup, we can assume the fiber stiffness to be negligible with respect to that of the underlying component. As a result, the expansion of the fiber is not free, but imposed by the thermal expansion of the substrate.

If the FBG is employed to measure strain, the simplest setup assumes that the temperature is constant (i.e. $\Delta T = 0$). This way, the expression for strain variation $\Delta\varepsilon$ can be easily derived from Equation (4.17):

$$\Delta\varepsilon = \frac{1}{C_\varepsilon}\frac{\Delta\lambda_B}{\lambda_B} \tag{4.22}$$

However, in most practical applications, temperature cannot be assumed constant. A possible method to compensate the thermal component of the FBG wavelength variation $\Delta\lambda_B/\lambda_B$ consists in installing a dedicated temperature sensor, either an additional FBG in the configuration of Figure 4.9, or a sensor based on other technologies, such as a thermocouple or a resistance thermometer [133]. Therefore, considering that temperature variation ΔT is known, and that thermal expansion is ruled by the substrate material rather than the fiber (so the coefficient C'_T is to be used instead of C_T), Equation (4.17) yields:

$$\Delta\varepsilon = \frac{1}{C_\varepsilon}\left[\frac{\Delta\lambda_B}{\lambda_B} - C'_T\Delta T\right] \tag{4.23}$$

An alternative method for thermal compensation implies to install two FBG sensors with different characteristics in the same measurement point, both bonded to the substrate, so that they are subject to the same strain and the same temperature. If the two sensors have different values for the strain and temperature coefficients, the system of equations:

$$\begin{cases} \dfrac{\Delta\lambda_{B1}}{\lambda_{B1}} = C_{\varepsilon 1}\Delta\varepsilon + C_{T1}\Delta T \\ \dfrac{\Delta\lambda_{B2}}{\lambda_{B2}} = C_{\varepsilon 2}\Delta\varepsilon + C_{T2}\Delta T \end{cases} \tag{4.24}$$

where the subscripts 1 and 2 refers to the two FBGs, has a unique solution $(\Delta\varepsilon, \Delta T)$ allowing a combined measure of temperature and strain. Compared to the previous method, which required to mount one of the sensors without loading it while at the same time ensuring a good thermal continuity with the substrate, the latter strategy allows for an easier installation. On the other hand, accuracy is limited by the difference in thermal and temperature coefficients of available sensors: if the two FBGs are too similar, the system (3.24) becomes ill-conditioned and the solution is highly affected by uncertainty in the measurement of wavelength.

During operations, the FBG sensors are connected to a transducing device, often referred to as the interrogator, able to measure the grating wavelength. As shown in Figure 4.10, the interrogator includes a narrow bandwidth, wavelength tunable infrared laser that produces the light required to read the sensors. Light generated by the laser passes through a three-way optical circulator and is sent through the optical fiber, to an array of FBG sensors. Light reflected back by the FBGs arrives to the optical circulator, and is routed to a photodetector. The wavelength emitted by the laser source is swept back and forth between a minimum and maximum value, with a given interrogation frequency. The signal from the photodetector is synchronized with the emitted frequency, and allows to reconstruct the reflection spectrum of the FBG. If a single FBG is present, the reflected spectrum will have a single peak centered on the grating wavelength; if multiple sensors are stacked on the same fiber, one peak will be detected for each FBG, provided that the individual gratings have different rest wavelengths. Commonly, if the laser power is sufficient, emitted light is split into multiple channels, each one with an individual photodetector, allowing to read a larger number of sensors.

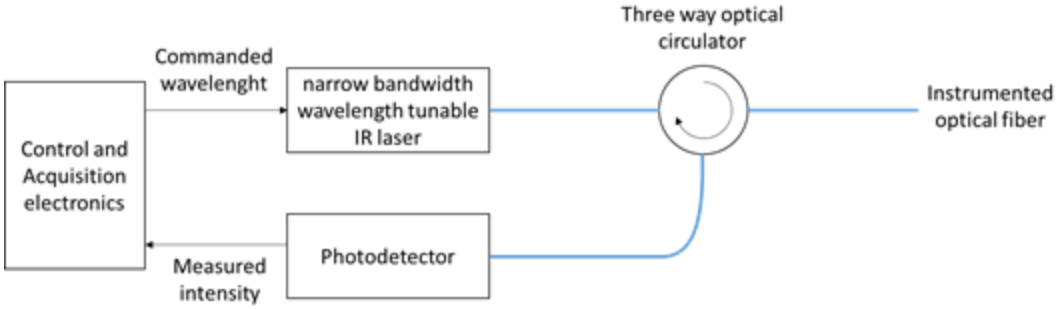

Figure 4.10: Block diagram of an FBG interrogator. The optical circulator is a device with $\boldsymbol{n}$ ports, arranged so that incoming light from the $\boldsymbol{i}$-th port is routed to the $(\boldsymbol{n}+\boldsymbol{1})$-th port, and incoming light to the last ($\boldsymbol{n}$-th) port is routed to the first port. The photodetector is a highly sensitivity photodiode that produces an output current proportional to the intensity of the incoming light. The photodetector is only sensitive to light intensity, and not to frequency, phase, or polarization.

Compared to traditional sensors for temperature and strain, FBGs offer several advantages, some of which are briefly listed below.

- Minimal invasivity of installation: sensors based on Bragg Gratings are much smaller than equivalent technologies. The sensor itself is only $125\ \mu m$ in diameter and less than a millimeter in length. Additionally, the sensors do not require individual wiring. Then, they can be installed where the poor accessibility would prevent the use of other sensing technologies [134].
- FBGs are totally insensitive to electromagnetic disturbance, since they operate in the infrared band: usually, their rest wavelength is in the range of 1400 to 1500 nm. Additionally, the characteristics of glass give FBGs a good robustness to aggressive chemicals [135]. This allows for the installation in harsh environments, [136, 137] near equipment that would usually jam the operation of traditional electrical sensors: for example, high resolution temperature measurements inside electrical machines can be performed using FBGs for health monitoring purposes.
- Multiple sensors can be easily stacked on a single optical fiber. Since an individual FBG reflects a specific wavelength and is transparent to the rest of the incoming spectrum, if several sensors are installed on the same fiber they can operate without interfering with each other, provided that their rest wavelengths are different and sufficiently spaced. The maximum number of sensors that can be stacked is determined by the wavelength range of the equipment used to acquire the measurements, and by the minimum spacing between the rest wavelength of each sensor, to avoid overlapping of the reflected waves through the whole measurement range. Usually, several tenths of sensors can be successfully employed on a single fiber, allowing to perform measurements characterized by high spatial resolution.
- Heterogeneous measurements can be performed by multiple sensors installed on the same optical fiber. For example, some of the sensors can be dedicated to strain measurement, while the rest is employed for thermal compensation. Additionally, several other quantities can be measured if converted to a mechanical deformation, such as acceleration, position, humidity, concentration of chemicals, vibrations.
- The low attenuation of light through the fiber allows to place the sensors in a remote location with respect to the single transducing equipment. Usually, a passive optical fiber can accurately transmit the signal with acceptable losses up to a distance of tenths of kilometers.
- FBGs provide high accuracy and repeatability both in temperature and strain measurements, compared to alternative technologies. The limiting factors are usually the installation and the resolution of the transducing equipment; currently accuracy up to 10^{-6} for strain and up to $0.1°C$ for temperature are easily achieved, while performances of cutting-edge equipment are constantly improving, as discussed by Hirayama and Sano [138] and Jung et al. [139].
- Due to their small dimensions, FBGs have a negligible heat capacity. This allows to obtain an almost instantaneous response as temperature sensors,

much faster than other technologies, as highlighted by Zhang et al. [140]. This characteristic is particularly useful when measuring the temperature of a flowing fluid.

Fiber Bragg Gratings also bring some disadvantages compared to traditional sensors.

- For strain measurements, the stiffness of bonding to the substrate is critical. FBGs have a much smaller surface area than traditional strain gages, so achieving a good adhesion with the material of the substrate may be difficult [141].
- Optical fibers are usually made of glass, so they are fragile and prone to accidental damages during manufacturing and installation.
- The path of the fiber carrying the sensors must be accurately planned to avoid sharp curves. The minimum radius of curvature to limit optical losses is in the order of some centimeters, while breaking of the fiber usually occurs for a curvature smaller than a centimeter.
- Thermal compensation is required to obtain accurate measures since, in most practical application, dependency of the FBG wavelength from temperature cannot be neglected.
- The technology has been introduced relatively recently, so the required equipment is still expensive compared to alternative sensors. Although, this situation may improve with time thanks to economies of scale

The next Sections describe the experimental activity carried out to validate the FBG technology for measurement of quantities of interest for diagnostic and prognostic analyses, specifically strain and temperature distributions. Some of the outcomes of these activities were presented in [142] and [143].

4.2.2 Calibration for strain measurement

A first experimental campaign was intended to determine the accuracy of an FBG-based system for measurement of strain. The next paragraphs describe the calibration of the sensors in presence of axial strain and bending, respectively.

4.2.3 Axial strain tests

A first activity consisted in calibrating FBG sensors for strain measurement, assessing their linearity and repeatability in combination with several alternative methods for securing the sensor to the measured component.

The setup shown in Figure 4.11 was employed to apply a controlled strain on the sensors. A section of optical fiber, including the FBG, was installed between two clamping blocks secured to a stainless steel breadboard. The considered part of fiber is characterized by uniform cross-section and uniform elastic behavior. The rest length of the fiber, measured inside the clamping blocks, is l_0. One of the clamping blocks includes a micrometric linear stage with a resolution of 10 μm;

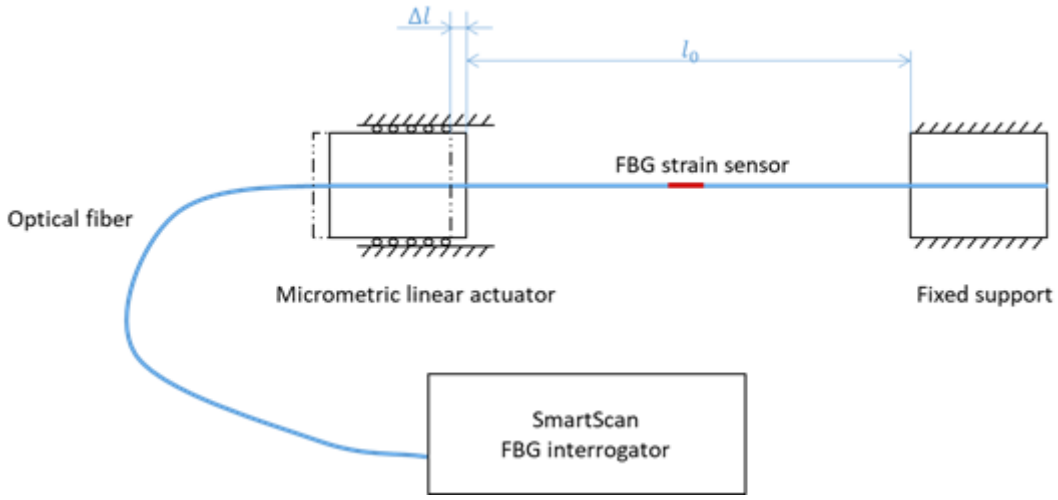

Figure 4.11: Setup for strain calibration and comparison of installation methods.

since the linear stage is manually operated and the positioning is achieved by rotating a knob against a dial, the actual repeatability may be higher than that. The displacement of the linear stage is defined as Δl. To exclude the effects of thermal expansion and thermo-optic behavior of the FBG, the experiment was conducted in a constant temperature environment.

The FBGs were connected to a Smart Scan interrogator, with a range of 40 nm (form 1528 nm to 1568 nm), a repeatability lower than 1pm, and a maximum frequency of 2.5 kHz (up to 25 kHz for a range reduced to 4nm). Data were sent to a laptop PC through an Ethernet connection, and logged to a text file.

The setup allowed applying a precise strain $\Delta\varepsilon = \Delta l/l_0$ to the FBG. The achievable resolution depends on the actual rest length of the fiber: a longer section allows obtaining smaller strain variations with the same linear displacement. As a baseline, an l_0 of 100mm results in a strain resolution of 10^{-4}, or $100\mu\varepsilon$.

With the experimental setup of Figure 4.11 several methods for the installation of FBGs were tested and compared.

- Mechanical clamping (with a protective rubber layer)
- Gluing (epoxy resin)

The main uncertainty in the measures is related to the behavior of the structure-to-fiber joint. The rest length l_0 is defined as in Figure 4.11, assuming that the deformation of the section of fiber laying on the clamping blocks is null. Actually, the structure-to-fiber joint has a finite stiffness, and is compliant to the deformation; so the actual rest length is slightly larger than l_0 and the measured strain will be lower than the expected value.

Figure 4.12 shows the results obtained for installation of FBGs via mechanical clamping. The clamping force is a compromise between the needs of avoiding to damage the fiber and preventing it to slip under load. The top graph compares the measured and expected strain time histories; at first, the FBG reading follows the imposed strain, although with an error due to the limited stiffness of the protective

rubber layer employed to transfer loads between the supports and optical fiber. As the strain increases (and so the force needed to deform the fiber), the clamping starts to slip; this behavior is visible starting from 100 seconds into the test. The rubber is permanently damaged after the slipping has occurred. The bottom graph of Figure 4.12 summarizes the relationship between the imposed and measured strain, with a first linear section and a plateau at about 800 microstrains (for the considered experimental setup). Clearly, this method of installation is not suitable for reliable strain measures, as high strain induces slipping and hysteresis in the response.

Much better results are obtained via gluing the optical fiber to the support with epoxy resin, as shown in Figure 4.13. Thanks to the stiffness of the glue being much higher than that of the rubber, the measured strain follows closely the imposed one, with a small relative error within 1%, up to high strain values. The good linearity of the response is highlighted in the bottom graph of Figure 4.13. A small viscous-elastic subsidence of the glue is shown in the detail of Figure 4.13 (top). This effect must be accounted for if very precise strain measurements are needed.

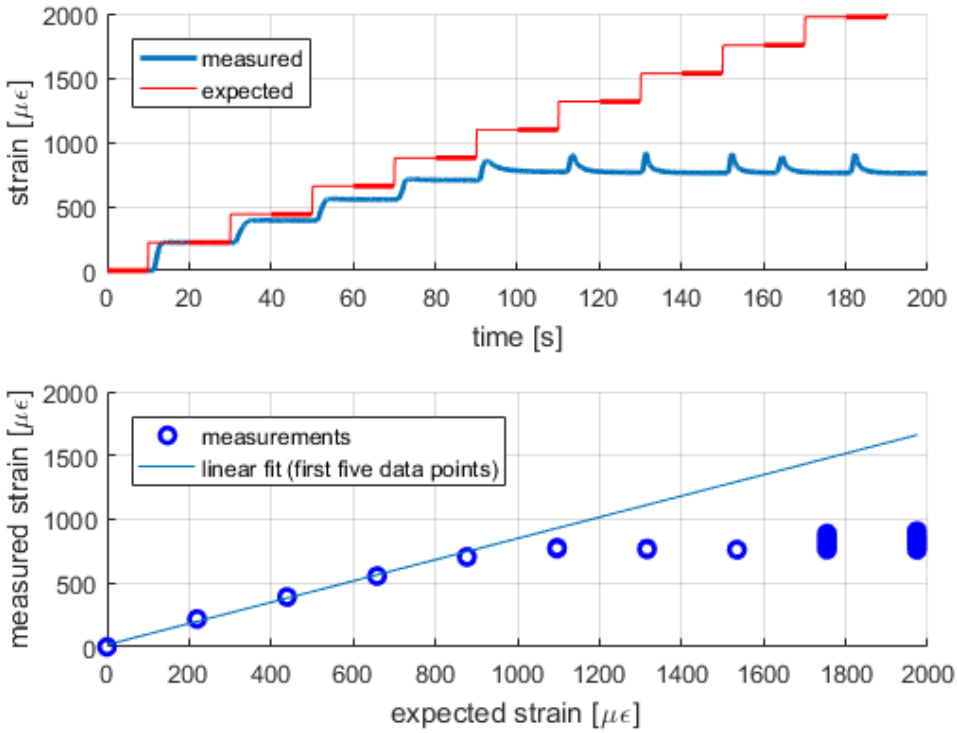

Figure 4.12: Results for mechanical clamping. (top) measured and imposed strains time history; (bottom) comparison between measured and imposed strains.

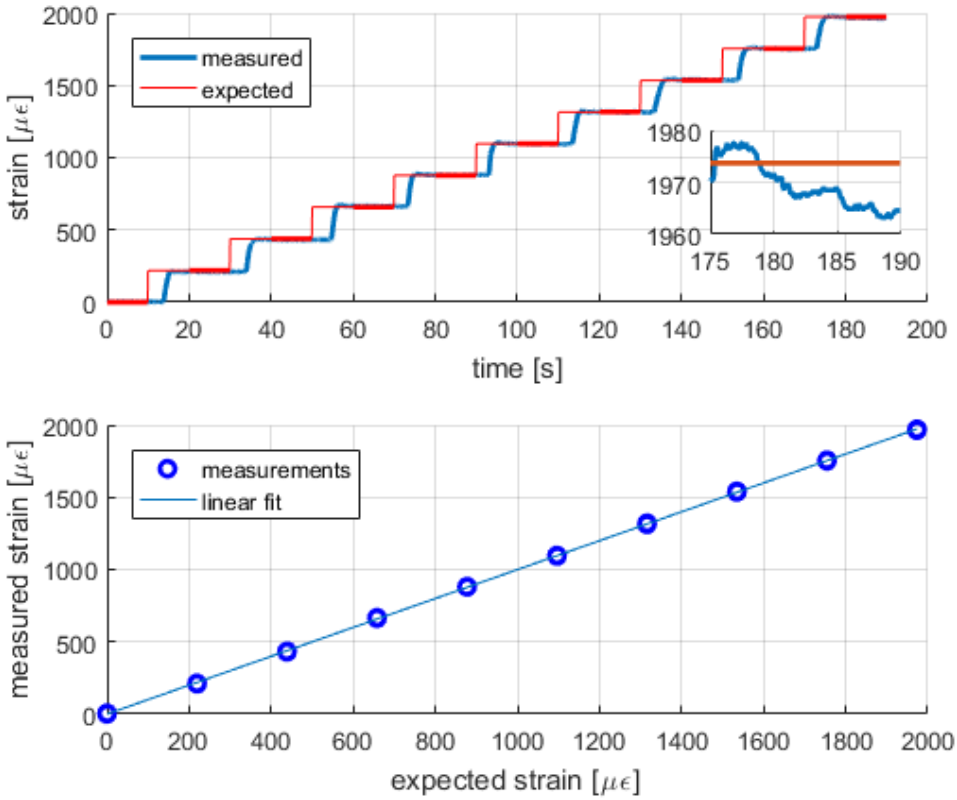

Figure 4.13: Results for gluing with epoxy resin. (top) measured and imposed strains time history; (bottom) comparison between measured and imposed strains.

4.2.4 Bending tests

Following the calibration, another experiment was carried out in order to compare different techniques for gluing the FBGs on structural components, with an arrangement closer to the final application than that employed previously.

A sample beam element with rectangular cross-section was installed between a fixed clamping block and a movable loading device, as shown in Figure 4.14. The loading device consists in a linear stage coupled to a geared DC servomotor, and allows to impose a displacement at the tip of the sample beam. As the beam has a constant section, the approximated displacement field $w(x)$ is easily obtained with Euler's beam theory:

$$w(x) = \frac{Fl^3}{6EI}\left(3\left(\frac{x}{l}\right)^2 - \left(\frac{x}{l}\right)^3\right) \tag{4.25}$$

where l is the length of the beam, EI its flexural rigidity, and F is the force applied by the load actuator. With the experimental setup employed for the measurement, F is not measured; additionally, although EI can be estimated from the nominal cross section of the beam and the material properties, it may be affected by a large uncertainty due to manufacturing tolerances. Conversely, the tip displacement is known, as it is generated by the load servoactuator; according to Equation (3.25), it can be expressed as:

$$w(l) = \frac{Fl^3}{6EI} \tag{4.26}$$

and then, Equation (3.25) can be rewritten as:

$$w(x) = \frac{w(l)}{2}\left(3\left(\frac{x}{l}\right)^2 - \left(\frac{x}{l}\right)^3\right) \tag{4.27}$$

The employed samples are aluminum and composite beams, with a 2mm by 20mm rectangular cross-section and a 300mm length. The FBGs are installed on the surface of the beam via gluing with epoxy resin. Several techniques were tested for the installation, including: gluing with and without vacuum curing, and with or without peel-ply. The FBGs are placed to measure the axial strain on the surface of the sample; this can be compared with the expected value estimated by Euler's theory:

$$\varepsilon_x = -z\frac{\partial^2 w}{\partial x^2} = -\frac{3h\,w(l)}{2l^2}\left(1 - \frac{x}{l}\right) \tag{4.28}$$

where $z = h/2$ is the distance from the elastic axis of the beam, equivalent to the half-thickness of the beam. Additionally, traditional resistive strain gages were installed near the FBGs to further validate the measurements.

The first sample involved sensors installed without vacuum curing. The fiber was subject to a controlled pre-tensioning achieved through a micrometric linear actuator, and then bonded on the test sample with epoxy resin. The FBG measurements can be considered reliable only with positive strain (i.e. in extension); the pre-tensioning is necessary for the FBG to measure strain both in compression. Figure 4.15 shows the results for this test sample. The measurement of the FBG overestimates the expected surface strain computed with Equation (3.28): this behavior can be explained considering that, in the employed setup, the combined thickness of the FBG and glue layer is not negligible with respect to the beam thickness. The measured strain lies within the tolerance band set by the maximum height of the glue, and the uncertainty is related to the actual position of the FBG within the glue layer (see Figure 4.16).

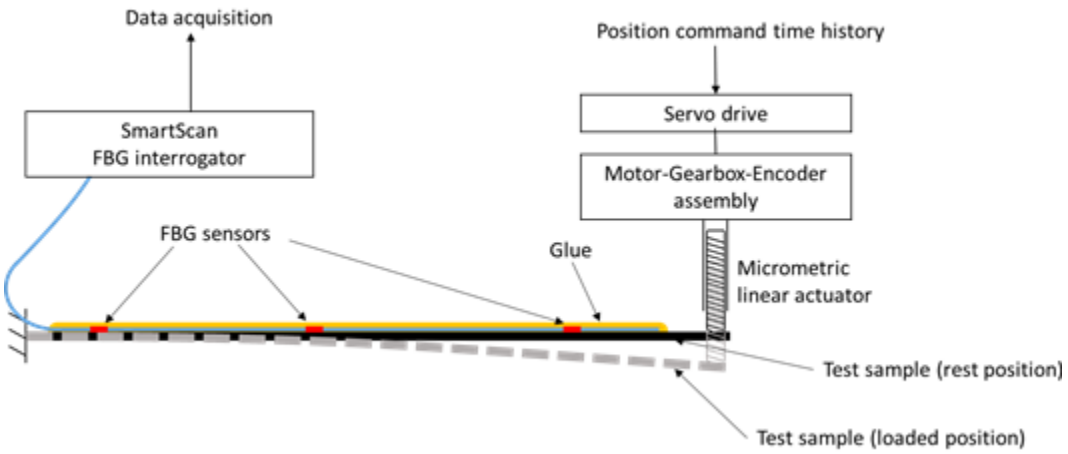

Figure 4.14: Experimental setup for the bending tests.

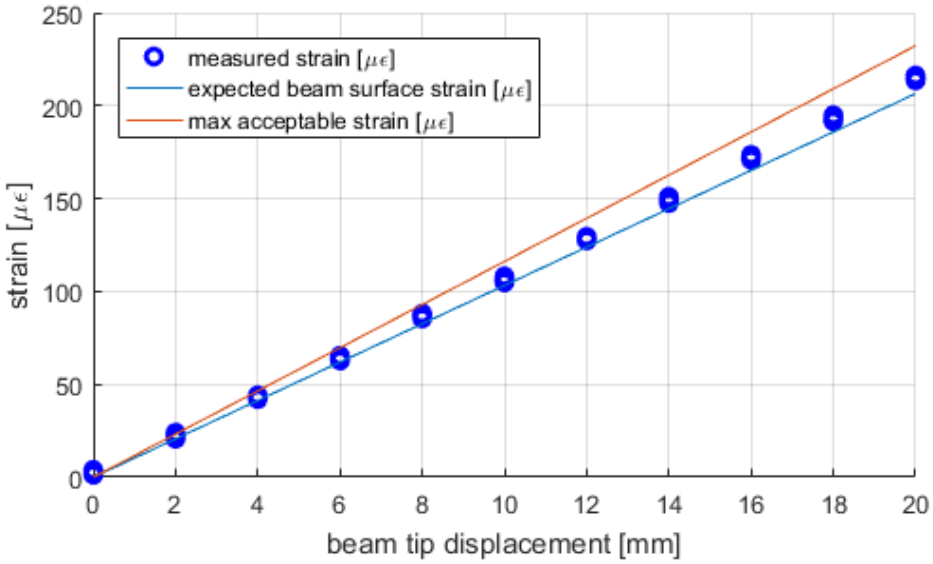

Figure 4.15: Response of the FBG sensors installed without vacuum curing. The blue line is the strain estimated by Equation (3.28) on the surface of the beam; the red line is the maximum strain possibly experienced by the sensors, accounting for the thickness of the glue layer.

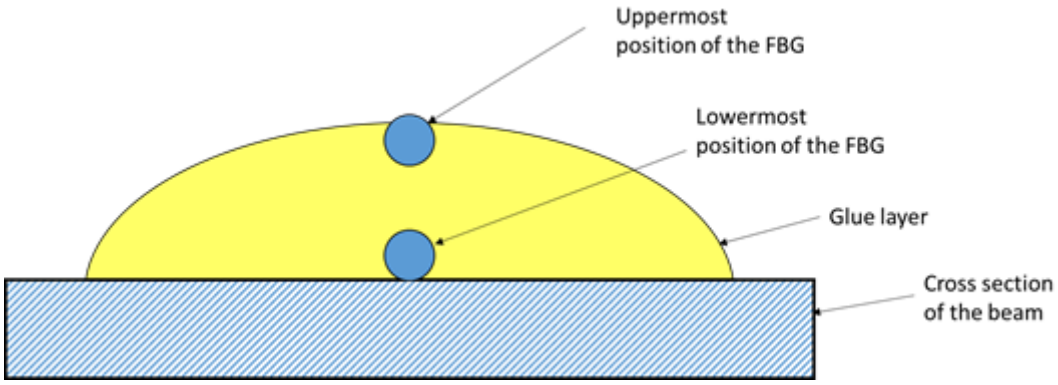

Figure 4.16: Uncertainty associated to the actual position of the FBG within the glue layer.

This uncertainty is inherently related to the strain measurement of a thin beam subject to bending deformation. A similar issue can be found with resistive strain gages, and the error is reduced when the measurement is performed on larger components, with smaller strain gradients.

In order to reduce the uncertainty associated with the glue layer height, a second sample was manufactured by vacuum curing the glue after the installation of the fiber. After pre-tensioning and gluing the FBG, a peel ply was set on the glue layer. Then, the sample was placed in a vacuum bag to obtain a uniform and thin glue layer, ensuring that the optical fiber is in contact with the surface of the sample. Figure 4.17 shows the results for this test; the surface strain is estimated with a smaller uncertainty and a better repeatability than what observed for the gluing without vacuum curing

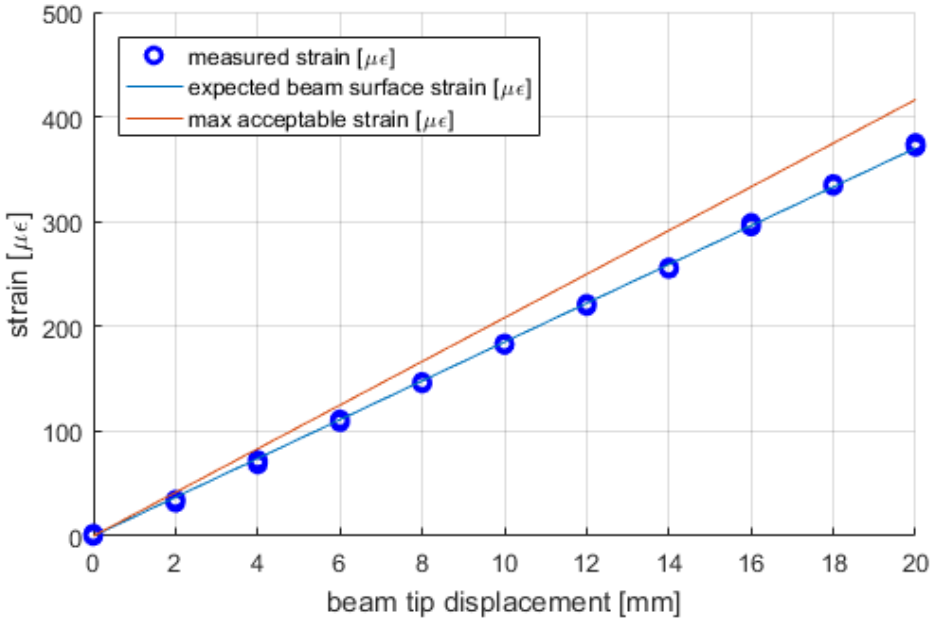

Figure 4.17: Results for the fiber with peel ply

A visual inspection of the sample highlighted how the peel ply removed most of the protective glue layer from above the optical fiber. To improve the robustness of the installed sensors to handling during manufacturing and assembly operations, a vacuum curing without peel ply was considered for a third sample; as shown in Figure 4.18, the performance is similar to the previous test; however, the process left a continuous glue layer above the optical fiber, resulting in a better protection from accidental damages.

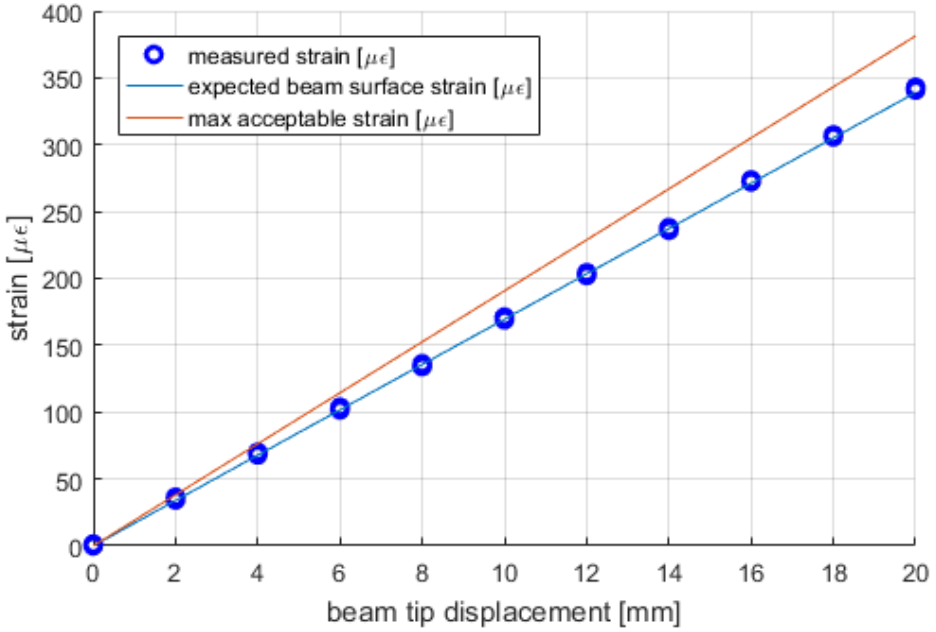

Figure 4.18: Results for the fiber without peel ply

4.2.5 Temperature measurement

The experiments described in the previous Sections were carried out in a constant temperature environment. A subsequent analysis was aimed at verifying the performance of FBGs as temperature sensors, and validating methods for thermal compensation.

To determine the behavior of the sensors in presence of temperature variations, an experimental setup was developed with the layout shown in Figure 4.19. A Peltier Cell provided a heat source with closed loop control, employing a PT100 resistance thermometer as feedback. Two fibers were installed side by side, each equipped with a single FBG: one fiber was left free-floating, so that mechanical load is null; the other was tensioned between two clamping blocks, with a setup similar to the axial strain tests described in Section 4.2.2. In order to guarantee that the three sensors (the PT100 and the two FBGs) remain at the same temperature, they were installed in a small container filled with water, with the heating element on the bottom. The convective heat transfer through the water ensures that, in steady state condition, the whole system is approximately at a common temperature.

Figure 4.20 shows the results of the experiment; the circles are the measured data points, while the continuous lines correspond to the theoretical wavelength-temperature relationship of Equation (4.17). The measures of the floating fiber are reported in Figure 4.20 (a); the slope of the curve is compatible with the predicted curve, while a small uncertainty may be ascribed to the local temperature gradients between the FBG and the PT100 resistance thermometer. The results for the pretensioned sensor, shown in Figure 4.20 (b), are obtained by applying a fixed strain to the fiber with the micrometric actuator. The measurements are in close accordance with the predicted behavior of the FBGs. A variation of temperature by 10°C produces a similar effect to a variation of 90 microstrains. Hence, a precise measurement of strain cannot neglect to employ some kind of thermal compensation.

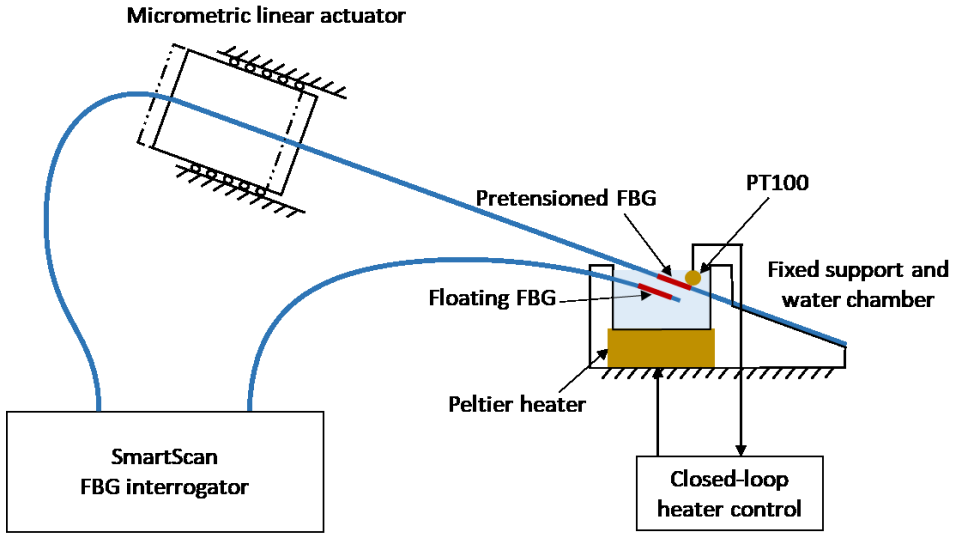

Figure 4.19: Experimental setup for temperature measurement and thermal compensation.

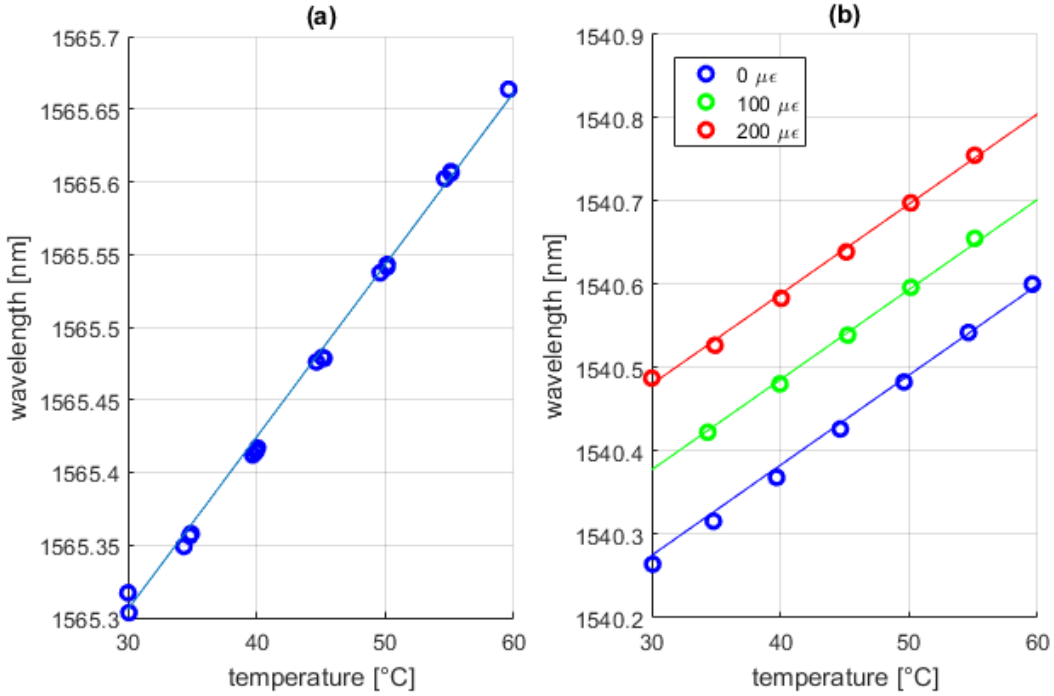

Figure 4.20: (a) Wavelength vs temperature characteristic for a free-floating FBG; (b) Wavelength vs temperature characteristic for a pretensioned FBG. In both cases, the circles are the measured data points, while the continuous lines represent the theoretical relationship of Equation (4.17).

4.2.6 Aging of the glue and compensation of temperature and humidity

In order to assess the long-term accuracy and reliability of measurements employing FBG sensors, an experimental campaign intended to determine the effect of environmental conditions and glue aging was run.

The test setup is shown in Figure 4.21. An optical fiber sample is glued and pretensioned between a fixed support and a translating one, which allows to apply a controlled and repeatable displacement. An DHT11 resistive temperature and humidity probe is placed close to the FBG sensor, and a PET enclosure covers both the FBG and resistive probe; this way, we can assume the two sensors to be subject to the same environmental conditions. Two glues were tested, specifically an epoxy resin and a cyanoacrylate.

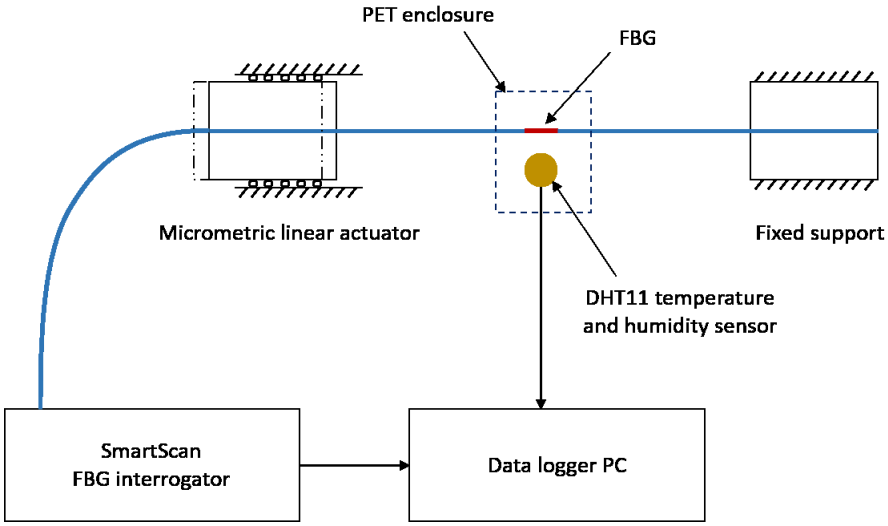

Figure 4.21: Experimental setup for characterization of glue aging

The test had the duration of approximately one month, in order to determine the repeatability associated to the strain measurement with a fixed displacement. Figure 4.22 shows the time evolution of the wavelength reflected by the FBGs, alongside with the ambient temperature and humidity recorded by the DTH11 probes during the experimental campaign. Figure 4.23 highlights a strong correlation between wavelength and environment temperature; the slope of the linear fit is compatible with the thermal expansion of the aluminum support of the fiber. In the considered experimental setup, as highlighted by Equation (4.20), the thermal dilation of the fiber itself has no effect on the FBG, since it is completely compensated by the displacement imposed by the supporting structure.

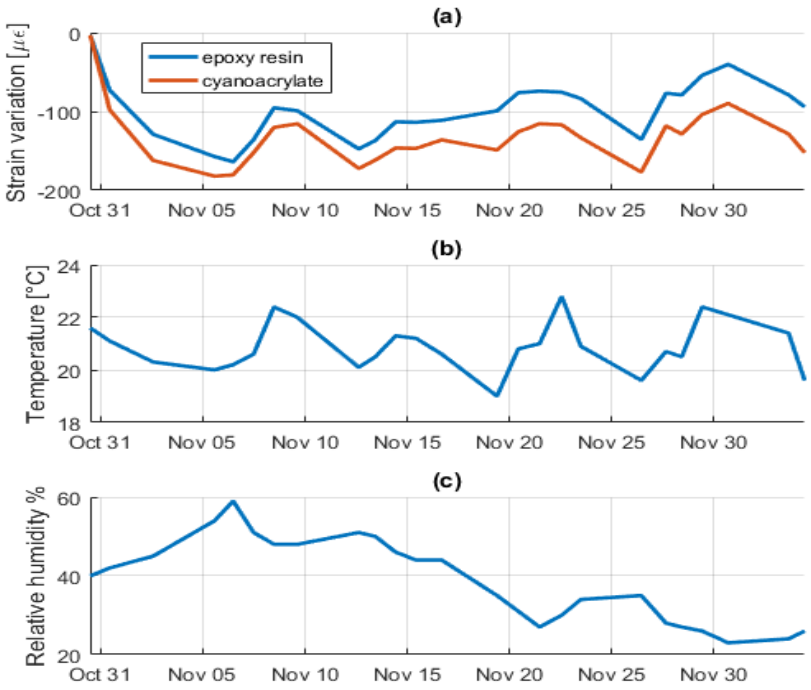

Figure 4.22: (a) Measured wavelength of the FBG; (b) ambient temperature; (c) ambient humidity.

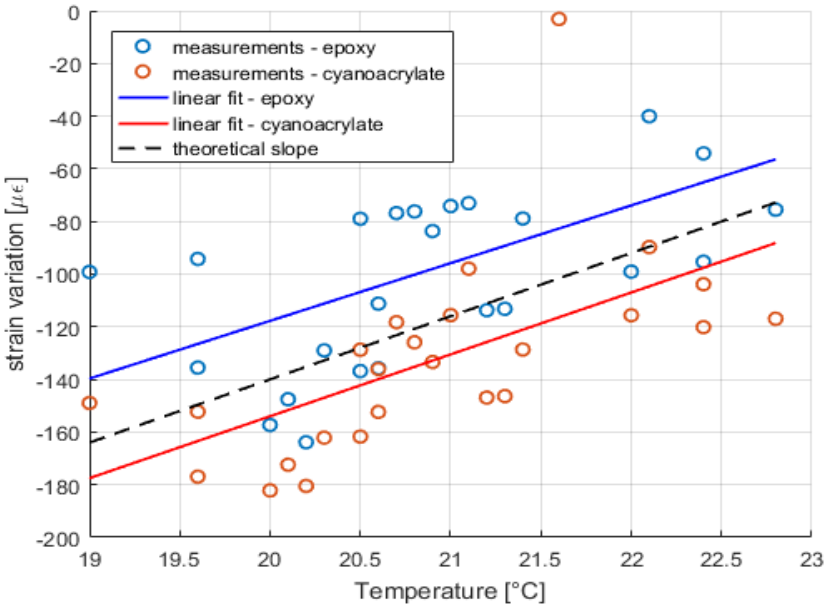

Figure 4.23: FBG wavelength vs ambient temperature

The variation of strain caused by temperature fluctuations can be compensated with Equation (3.17). The compensated wavelength is plotted against ambient humidity in Figure 4.24. The correlation of the two variables is small and not easily repeatable; a variation of the fiber axial strain associated with humidity fluctuations could be ascribed to a hygroscopic deformation of the epoxy glue employed to bond the fiber on the supporting structure. However, an accurate mathematical model of this behavior is difficult to derive analytically, and only a fitting curve can be employed to compensate for any effect of humidity.

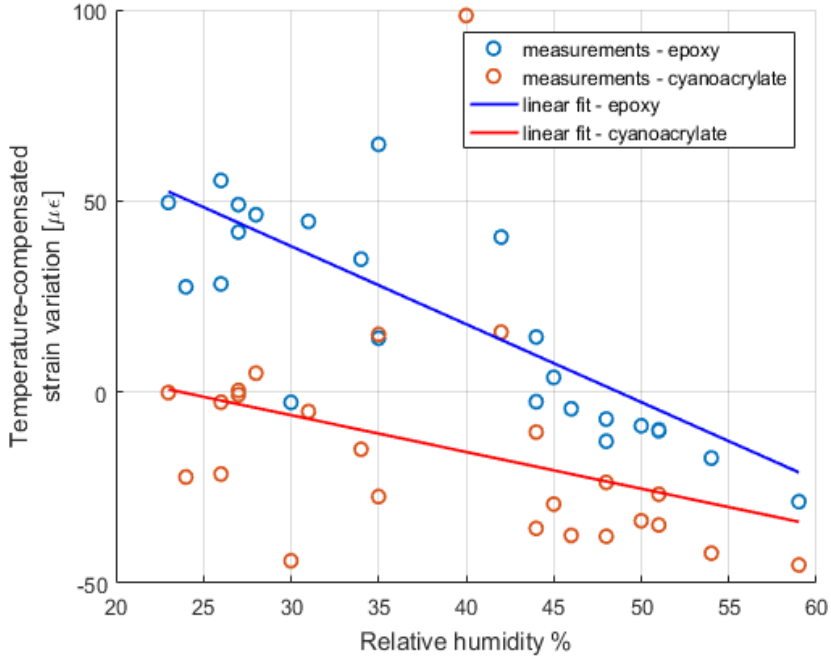

Figure 4.24: Temperature-compensated FBG wavelength vs ambient humidity

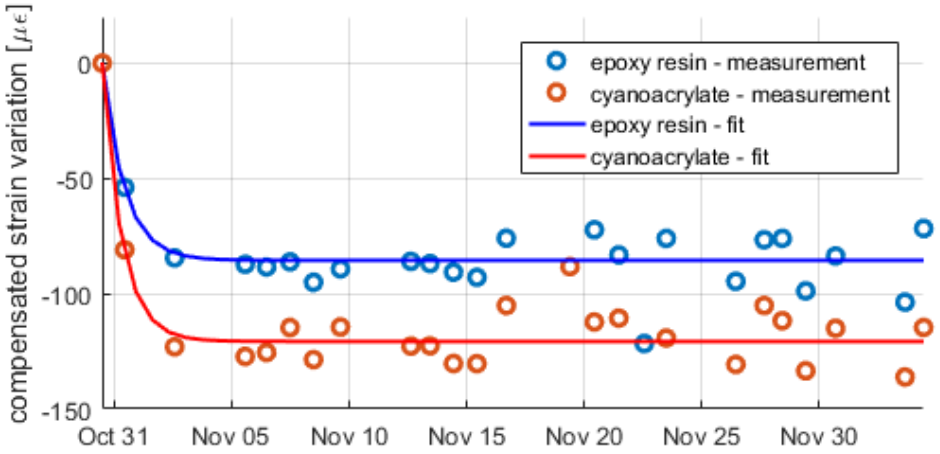

Figure 4.25: Temperature and humidity compensated FBG wavelength highlighting the viscous-elastic settling experienced during the aging of the glue bonding

The time-history of the temperature and humidity compensated FBG wavelength is shown in Figure 4.25. A decay of the measured wavelength of 1.8% and 2.8% of the initial pretensioning is observed for the epoxy resin and cyanoacrylate bonding respectively. The settling of the epoxy glue is completed after approximately one week, after which the measurements are stable.

The same experimental campaign was repeated with a better time resolution of the measures, by automating the acquisitions, and replacing the DHT11 temperature and humidity sensors with SHT85 ones, characterized by a higher accuracy and repeatability. Indeed, Figure 4.23 and Figure 4.24 reveal a significant uncertainty associated with the wavelength-temperature and wavelength-humidity relationships. At the same time, the deviations from the steady-state measurement reported in Figure 4.25 appear correlated for the two samples, suggesting a common disturbance to the temperature measurements.

Figure 4.26 shows the temperature and humidity trends for two additional sample FBGs. The time sampling was increased from once a day to once every 10 minutes; the SHT85 sensors allowed a measurement of temperature with a resolution down to 0.01°C, employing a moving average filtering.

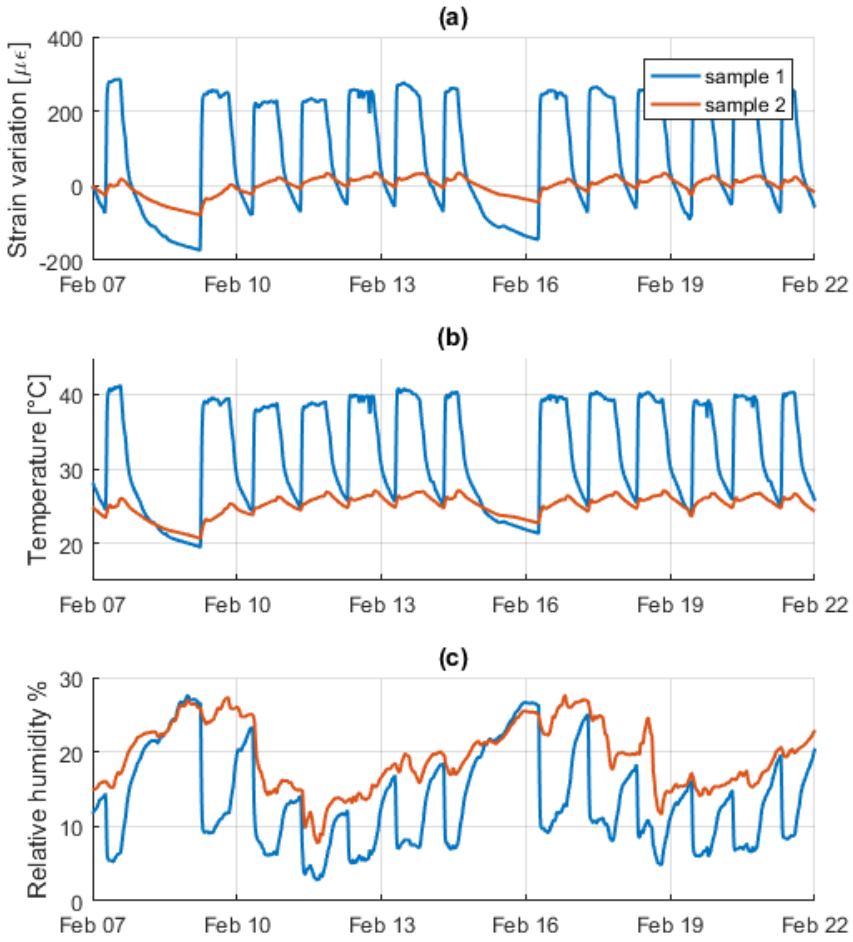

Figure 4.26: (a) Measured wavelength of the FBG for the second experimental campaign; (b) ambient temperature; (c) ambient humidity.

The wavelength-temperature correlation is shown in Figure 4.27; the slope of the fitting curve is consistent with the previous results and with the thermal expansion of the aluminum FBG support. A small but visible hysteresis is observed for the first sample, which is subject to larger and faster temperature variations. Figure 4.28 illustrates the correlation to humidity for the same two samples, similarly to Figure 4.24. For the second sample, a dependency of the FBG reading from ambient humidity is observed, compatibly with the previous results. However, this behavior is not repeatable for the first sample, which shows a large dispersion of the data and no noticeable correlation, and suggests that this behavior is due to the hygroscopic properties of the glue employed to bond the sensor to the structure, rather than those of the fiber's coating. Indeed, this effect appears highly dependent on the bonding, and hardly repeatable without a strict control of the process.

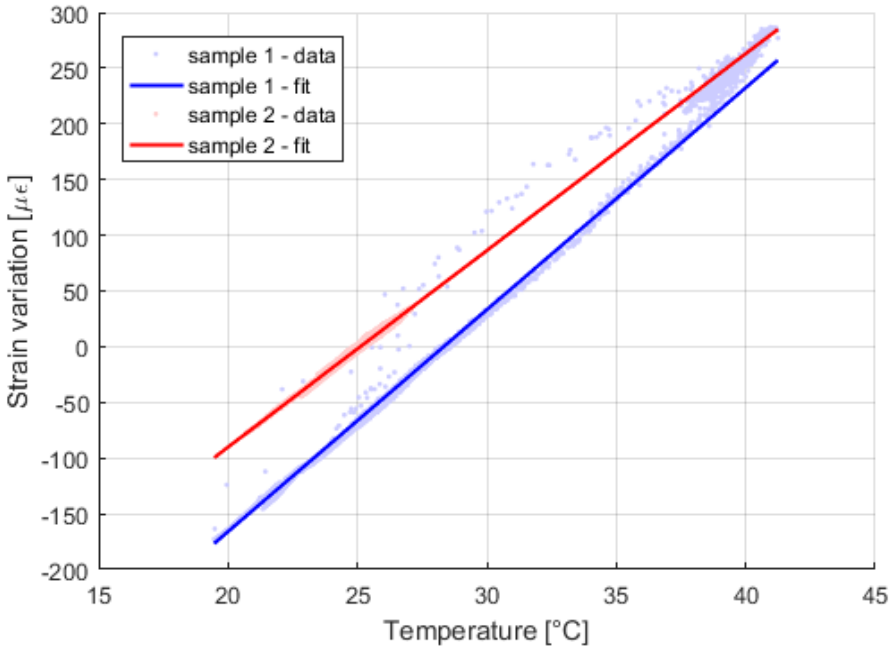

Figure 4.27: FBG wavelength vs ambient temperature for the second experimental campaign.

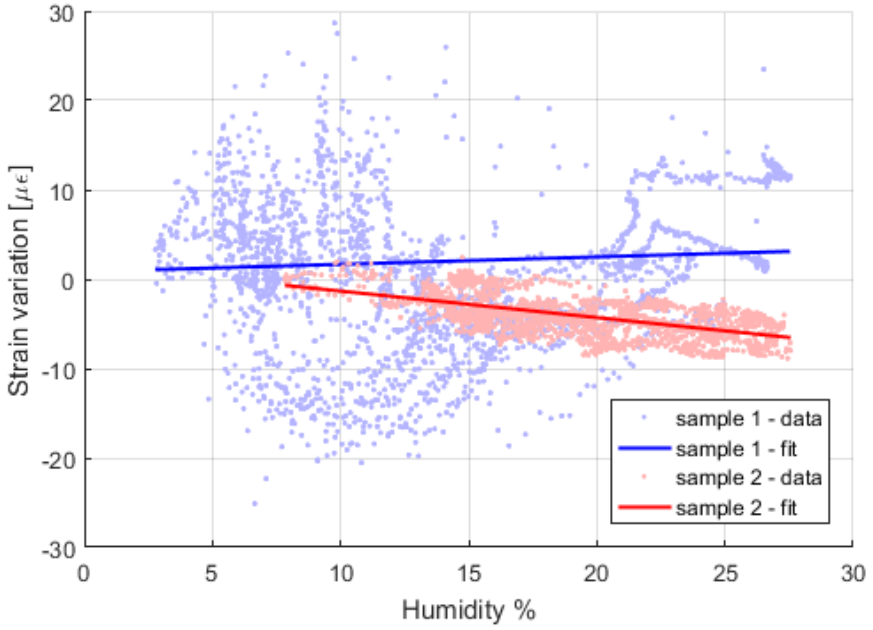

Figure 4.28: Temperature-compensated FBG wavelength vs relative humidity for the second experimental campaign.

4.2.7 Flight testing

A flight-testing campaign on a radio-controlled UAV was run to assess the feasibility and the impact at vehicle level of an FBG sensors network for evaluation of aerodynamic loads on flight control actuators [144]. The aircraft, provided by the ICARUS PoliTO students team and shown in Figure 4.29, featured electric propulsion, 20kg Maximum Take-Off Weight (MTOW), and 4m wingspan. The UAV was designed to participate to the 2017 edition of the Air Cargo Challenge organized by the University of Zagreb, and to carry an 8kg payload. It features a Carbon Fiber Reinforced Polymer (CFRP) composite modular structure; this allows to embed FBG sensor during the lamination of the airframe sections. The payload capability of this airborne platform is compatible with the installation of the sensors and acquisition systems, without the need to employ weight optimized equipment.

The FBG-based system for measurement of the airframe strain is similar to the Fiber Optic Strain Sensing (FOSS) system implemented on NASA's Odissey UAV, that has been employed in studies on active compensation of flutter oscillations [145]. The acquisition and telemetry system included a SmartScan FBG interrogator reading the measures of a network of sensors installed on the composite airframe. A Raspberry Pi board was connected to the interrogator with an Ethernet bus to log the data onto an SD card and send it to the ground control station. Additionally, the board was connected to the Inertial Measurement Unit (IMU) of the aircraft, in order to log the load factor experienced by the airframe. The interrogator requires a power supply of 1.0A at 9 to 36V; this was provided by a 6000mAh, 3S LiPo (11.1V) battery. The Raspberry Pi was powered by a 20000mAh USB power bank. The dedicated power supply for the system, independent from the aircraft propulsion, is able to provide an endurance in excess of 6 hours, significantly longer than the flight endurance of the aircraft. The oversizing has a twofold reason: on one hand, the battery is employed as a ballast to compensate to the aft displacement of the center of gravity caused by the installation of the interrogator; on the other hand, the LiPo battery employed for the interrogator was available as a spare of the main battery of the propulsion system.

(a) (b)

Figure 4.29: (a) Render of the UAV platform employed for flight testing; (b) picture of the aircraft.

For future flight tests, the USB power bank for the Raspberry Pi will be replaced with a DC-DC step-down converter, to use a single battery for the whole system and achieve a weight reduction of approximately 300g.

The weight budget of the onboard sensor acquisition and telemetry system is broken down in Table 4.1. The weight of the fiber and FBG sensors is negligible, lying in the range from 30g to 50g per kilometer of fiber. The interrogator is the heaviest component of the system: the specific model employed is not optimized for weight reduction and features a heavy stainless steel casing. Additionally, a significant mass saving can be achieved by downsizing the battery, or employing a power supply common with the propulsion system. The whole system adds up to a weight of 1955g: although significant for the weight budget of a small scale UAV, this is by no means demanding for the installation on larger platforms.

Table 4.1: Weight budget of the FBG acquisition system

Component	**Weight [g]**
SmartScan FBG interrogator	1100
Raspberry Pi	45
Interrogator Battery	419
USB power bank	360
FBG sensors	<1
Wiring and connectors	30
Total	**1955**

The FBG sensors were installed on the tail beam and wing of the aircraft; both sections of the airframe underwent static loading tests to verify the correct operation of the system. The tail beam was tested under actual flight conditions to determine the quality of the system, as well as the noise caused by aerodynamic turbulence and mechanical vibrations. The schedule of the flight test did not allow to fly the instrumented wing, as its manufacturing was still in progress; we are currently planning a flight acquiring the strain both from the tail and wing.

Figure 4.30 (a) and (b) show the placement of FBG sensors on the tail beam and wing respectively. The tail beam has 16 sensors divided in 4 channels; 11 sensors in 2 channels were placed on the main right wing spar (8 sensors between the upper skin and spar cap, and 3 sensors between the lower skin and spar cap). We planned to install 5 additional sensors on the lower spar cap, but an issue during the manufactured process prevented to connect their optical line to the interrogator. The sensors on the wing were equally spaced with a 200mm pitch, starting from the aircraft centerline. As the wing construction is modular and can be disassembled for transport, optical connectors were installed between the individual airframe sections. Out of the five sections of the wing, section 3 (the inboard one) and section 4 (the middle right one) were selected for FBG installation. Placing additional sensors on the lightly loaded section 5 (outboard right) would have resulted in a significant increase in complexity, as additional optical connectors would be

needed, while providing minimal useful data, since section 5 accounts for a small fraction of the total wingspan.

The experimental setup for the static test on the tail beam is shown in Figure 4.31. The front frame of the structure was clamped to the test stand, while a static load was applied at the connection interfaces with the empennages via a set of calibrated weights; a pure bending load or a combination of bending and torsion load could be applied, depending on the position of the weights with respect to the fuselage centerline. The measures of the FBGs were compared to the strain estimated by a Finite Elements (FE) analysis and to the readings of four Strain Gages (SGs) temporarily installed on the top skin of the airframe for further validation. Those were necessary to calibrate the FEM model to the actual

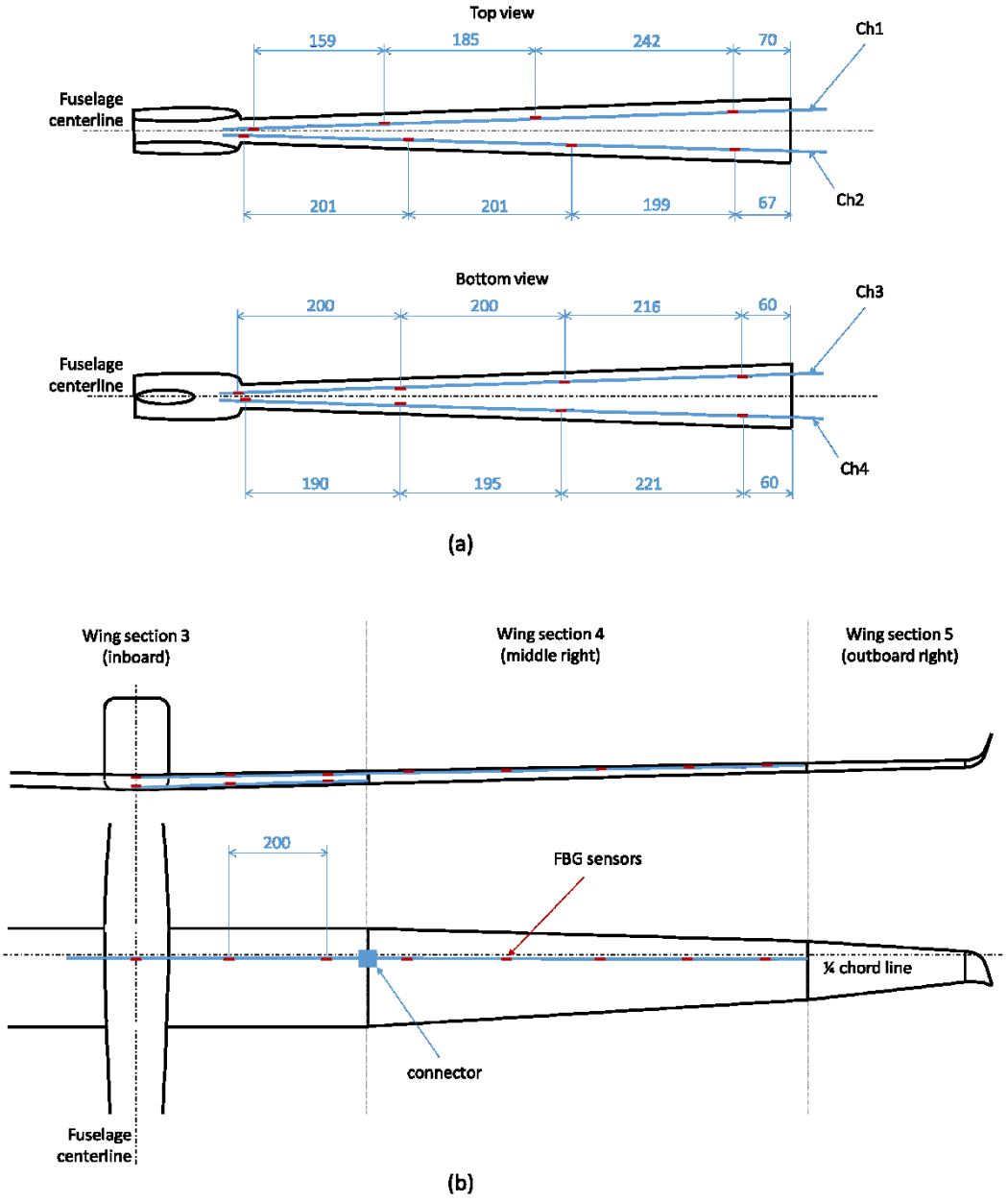

Figure 4.30: (a) Placement of FBG sensors on the tail beam; (b) placement of FBG sensors on the wing.

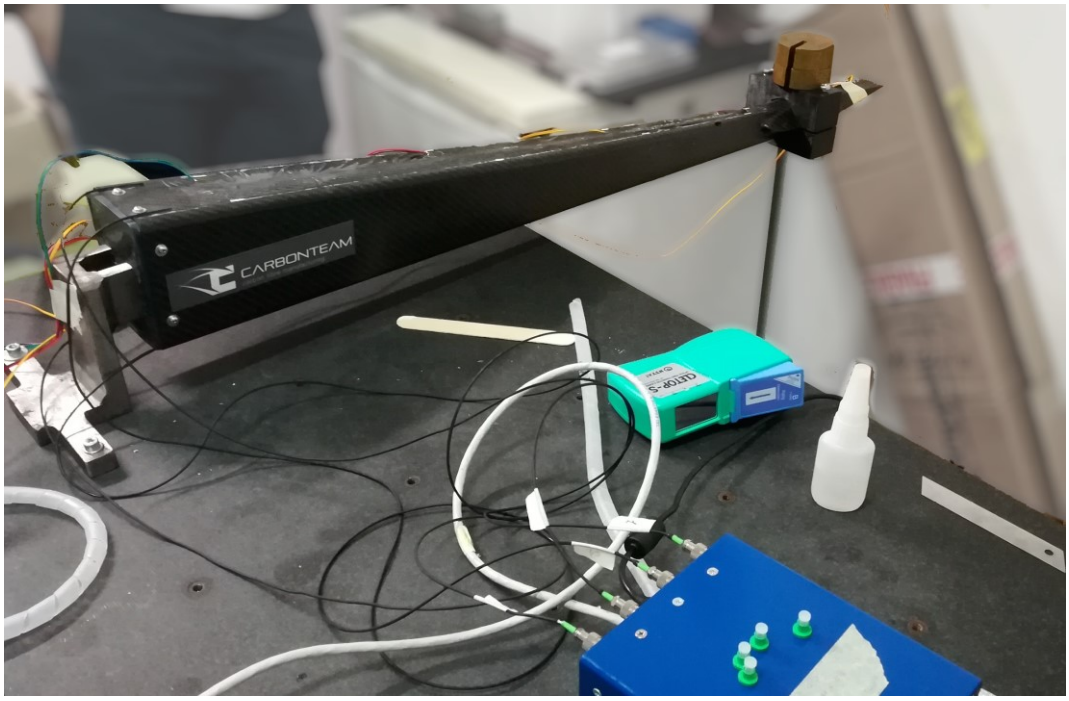

Figure 4.31: Experimental setup for the static loading test on the tail beam

mechanical properties of the materials, which are subject to uncertainties in the production process and not known accurately. The SGs were removed for the actual flight testing as their conditioning circuitry and acquisition equipment is too large and heavy to fit in the payload bay of the UAV.

Figure 4.32 shows the results of the static test on the tail beam. Most of the FBG measures match the SGs and the FE estimates. The FE analysis predicts a smooth strain distribution in the middle of the tail beam; due to the abrupt variations in the geometry, boundary conditions and load distribution, the forward and aft sections of the structure experience a more irregular strain distribution. The FBGs and SGs mostly match the prediction of the FE model with a small error, the only exception being the strain gage at the 476mm coordinate, which has a 13% error with respect to the FE estimate. This anomaly is probably due to either to a local manufacturing defect or to an imperfect SG bonding that fails to transfer the deformation from the structure to the transducer properly.

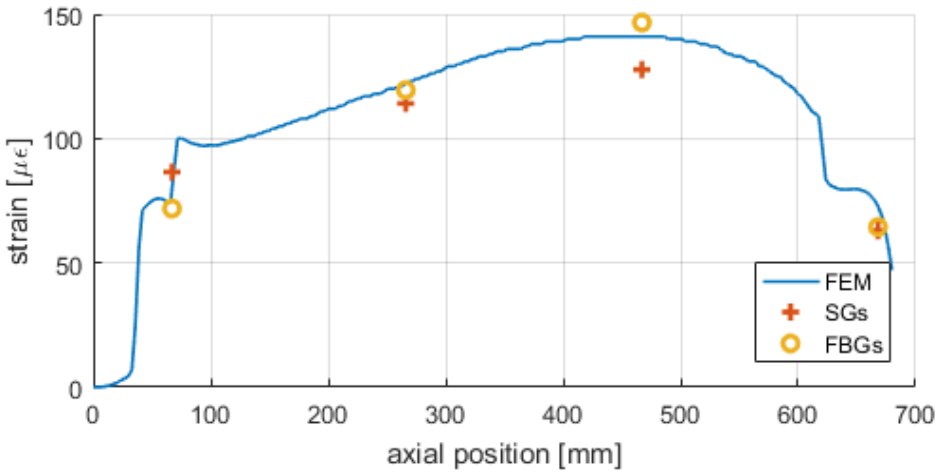

Figure 4.32: Results of static loading test on the tail beam

A similar test allowed to assess the operation of the sensor network installed on the wing. Figure 4.33 shows the experimental setup. The wing is clamped to the test stand by the root section, via a support simulating the wing-fuselage connection. The wing is installed upside down, so that the calibrated weights applied near the wing tip simulate a load in the same direction of lift.

The results of the test are shown in Figure 4.34; the FBG measures are compared with an FE analysis, while the mechanical properties of the material are now known from the test on the tail beam and no SGs are installed on the structure. The continuous curve on Figure 4.34 represents the axial strain on the top and bottom spar caps estimated by the FE model; the discontinuity corresponding to the junction between wing sections 3 and 4 is visible. As for the tail beam static test, the FBG measures mostly match the FE predictions; the largest discrepancy is found for the transducers installed near the wing-fuselage connection. This inaccuracy may be caused by the FE model considering the clamping perfectly stiff, while the real experimental setup can experience local deformations due to the flexibility of the support.

Figure 4.33: Experimental setup for the static loading test on the wing

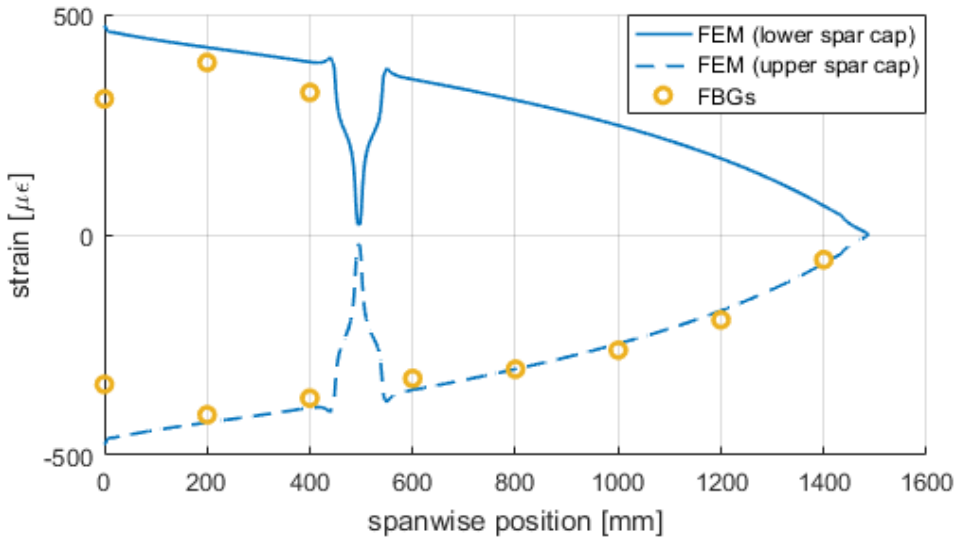

Figure 4.34: Results of static loading test on the wing

The flight test of the tail beam allowed to determine the quality of the measurements acquired on-field and to quantify the disturbances caused by external factors such as mechanical vibration and aerodynamic turbulence. Figure 4.35 and Figure 4.36 compare the load factor measured by the Inertial Measurement Unit (IMU) to the strains measured by the four FBG channels. The data are filtered with a 100ms moving average to smooth out the vibration caused by the propulsion system. A correlation between the strain measurement and load factor is visible, although the FBG signals are affected by a significant amount of noise and by a drift due to temperature variations. A first cut attempt to compensate for these effects is to compute an averaged measurement of the sensors installed at the same axial coordinate (Figure 4.37); specifically, the averaged strain ε_{avg} is computed as:

$$\varepsilon_{avg} = \frac{\varepsilon_1 + \varepsilon_2}{2} - \frac{\varepsilon_3 + \varepsilon_4}{2} \tag{4.29}$$

where the subscripts 1 and 2 denote the sensors installed on the top skin, and the 3 and 4 subscripts denote those installed on the bottom skin. This way, ε_{avg} is an approximation of the strain component caused by bending loads on the tail.

The tail beam is very stiff and does not flex significantly under normal flight loads. As a result, the strain produced by vibrations and thermal expansion is not a small component of the total strain field experienced during the flight test. From the comparison between Figure 4.32 and Figure 4.34, the wing experiences strains up to one order of magnitude larger than the tail; this may allow for a more accurate measurement with lower relative errors.

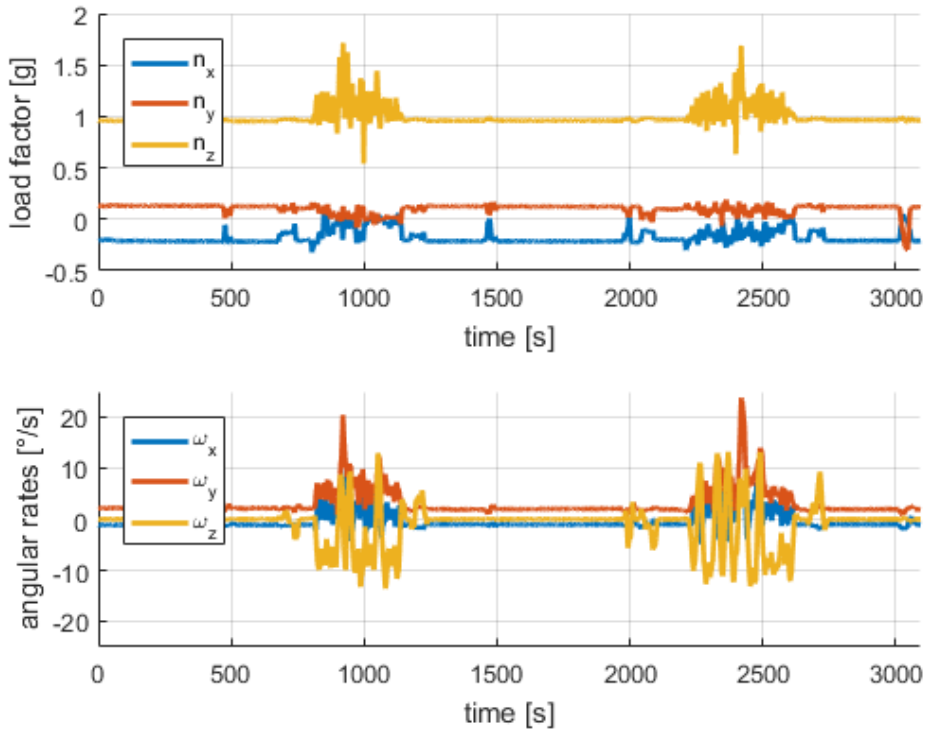

Figure 4.35: Load factor and angular rates measured by the Inertial Measurement Unit (IMU).

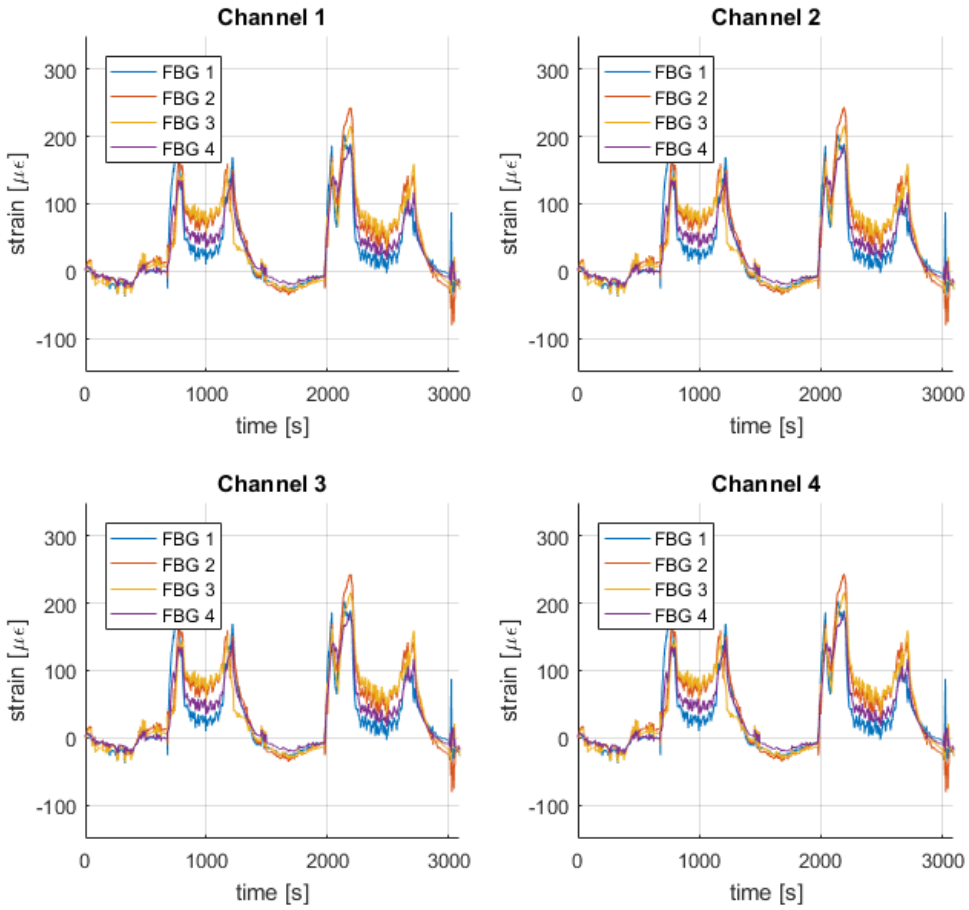

Figure 4.36: Strain measured by the four FBG channels

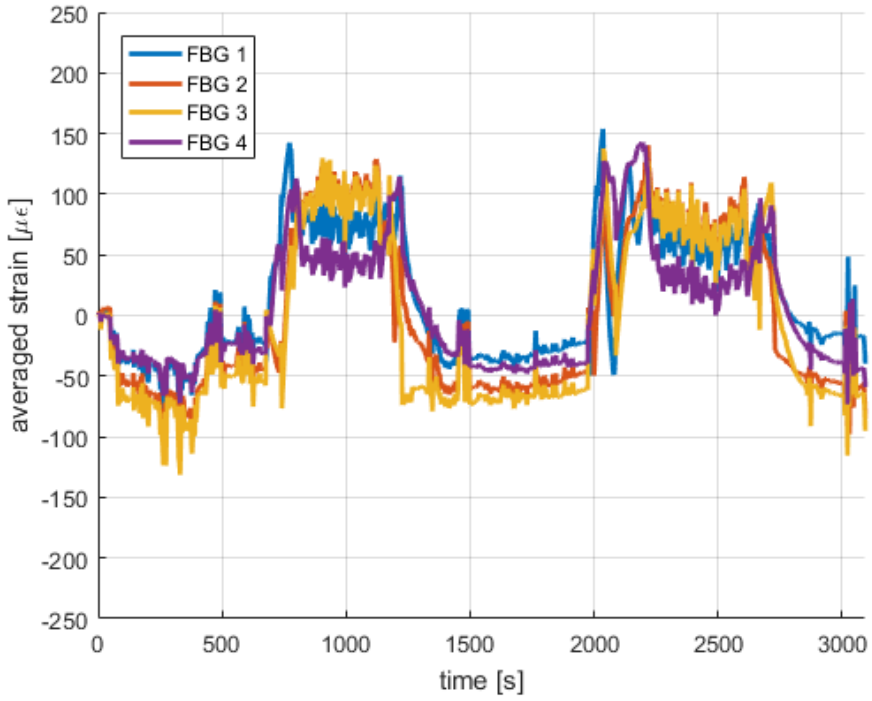

Figure 4.37: Averaged FBG measurements and load factor

Chapter 5

Validation of Physics-Based models

A test bench was designed to provide an experimental validation to the physics-based EMA models employed in this work and described in Section 3. The test bench aims to reproduce the behavior of an electromechanical actuator in different operating conditions, while featuring an open architecture to ease future modifications of the hardware and integration of additional sensors.

The layout of the test bench is shown in Figure 5.1. It reflects that of a common rotary EMA, including a PMSM with its power electronics module, a planetary gearbox to match speed and torque of the motor to the load, and a control unit to provide closed-loop position and speed control.

Off-the-shelf components, intended for industrial automation applications, were chosen for the electronics and PMSM. These share the same technology of aerospace EMAs, but are much cheaper and easier to source. The behavior for ground tests is assumed to be close enough to aerospace hardware for validation of the models.

The planetary gearbox is a custom design, intended to simulate a possible configuration suitable for primary and secondary flight controls, adapted for the integration of sensors, and featuring an adjustable backlash for the output shaft position sensor. Additionally, a custom assembly for the simulation of external loads on the motor was designed and manufactured.

5.1 Components of the EMA test bench

The following sections detail the main electromechanical components employed to set up the EMA test bench employed to validate the simulation models. In particular, Section 5.1.1 describes the motor and power electronics, Section 5.1.2 the mechanical transmission, and Section 5.1.3 the load simulation unit.

5.1.1 Motor

A commercial solution was employed for the motor and power electronics. Specifically, a PMSM and its driver supplied by Siemens and intended for applications in industrial automation were selected. The motor is a three-phase, star-connected permanent magnet synchronous machine and features four pole pairs. The driver is powered by the 400V three-phase industrial line for the power module, and by a 24VDC line for the logic section. An absolute encoder integral to the motor provides a position feedback for the motor shaft. This information is employed to synchronize the phase commutation and to close the velocity control loop within

the control unit. An industrial PC, running a simulated PLC, closes the position loop and manages the generation of the position setpoint. Additionally, the system provides a log of several variables, including phase voltages and currents, speeds, and positions. A basic datasheet for the motor is provided in Table 5.1.

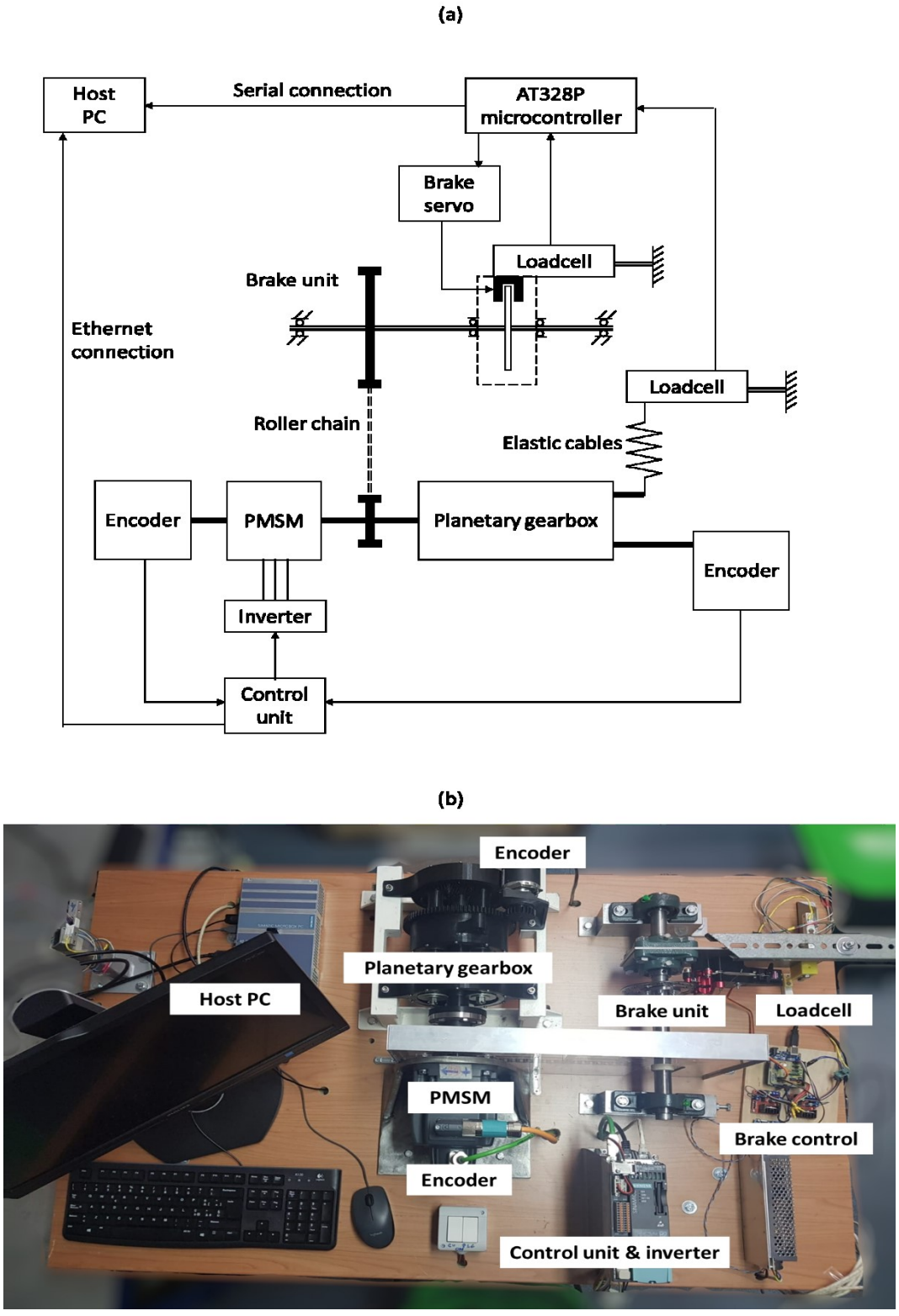

Figure 5.1: (a) Block diagram of the EMA test bench; (b) picture of the employed hardware.

Table 5.1: Datasheet of the motor

Quantity	Unit	Value
Torque gain	Nm/A	1.91
Phase resistance	ohm	2.75
Phase inductance	mH	30.5
Number of poles	--	8
Max current	A	10.7
Max voltage	V	380
Inertia	$Kgcm^2$	7.7

5.1.2 Gearbox

The test bench employs a compound planetary gearbox as a transmission between the motor and user shaft. The particular architecture designed for this application allows a high gear ratio within a limited size and with few moving parts, thus it is suitable for weight and size constrained aerospace actuation tasks [146].

Figure 5.2 (a) shows the general layout of the gearbox. The input shaft M carries the sun gear of the first stage reducer. The planet gears S1 engage with the sun gear and a fixed ring gear F. The satellite gears S2 are locked on a common shaft with S1, and engage with the output ring gear U. Since the transmission ratio between S1 and F is slightly different from that between S2 and U, the output shaft is dragged at a low speed as the satellites move. The first stage, including the sun gear M, satellites S1 and the fixed ring gear F, is mirrored on the other side of the output ring gear U. This way, the forces on each satellite are completely balanced, wear is reduced, and robustness to shock loads is improved. Additionally, the gearbox uses helical gears for the input stage (with opposed helix angles on the two sides), and herringbone gears for the output stage (including satellites S2 and output ring gear U): as a result, both the satellites and the output ring gear are completely supported by the helical teeth in the axial direction, and no planet carrier is needed.

Figure 5.2 (b) shows the distribution of velocity on the gears, and allows to compute the transmission ratio. The peripheral speed of the sun gear M is:

$$V_F = \omega_F r_F \tag{5.1}$$

and that of the output gear U:

$$V_U = \omega_U r_U = V_M \frac{r_{S1} - r_{S2}}{2r_{S1}} \tag{5.2}$$

where ω denotes an angular velocity and r denotes a radius. As a result, the transmission ratio is:

$$i = \frac{r_U}{r_M} \frac{2r_{S1}}{r_{S1} - r_{S2}} \tag{5.3}$$

The geometrical compatibility of the planetary gearing requires that:

$$(z_F - z_{S1})m_1 = (z_U - z_{S2})m_2 \tag{5.4}$$

$$z_F = z_M + 2z_{S1} \tag{5.5}$$

where m_1 and m_2 are the moduli of the input and output stages respectively. Then, if $m_1 = m_2$ the transmission ratio can be written as:

$$i = \frac{z_U}{z_M} \frac{2z_{S1}}{z_{S1} - z_{S2}} \tag{5.6}$$

Otherwise, by substituting Equations (5.4) and (5.5) into the (5.3) and considering that $r = zm$:

$$i = \frac{\frac{z_F}{z_M} + 1}{1 + \frac{z_F z_{S2}}{z_U z_{S1}}} \tag{5.7}$$

For the particular application of the test bench, the number of teeth reported in Table 5.2 were chosen, to achieve a gear ratio of 124.

Table 5.2: Numbers of teeth employed for the gearbox

z_M	21
z_{S1}	21
z_F	63
z_{S2}	20
z_U	62

The gearbox was built through Fused Deposition Modelling (FDM) additive manufacturing, employing a Poly-Lactic Acid polymer. This allowed to keep down the costs and time associated to manufacturing; the design phase was simplified as well accounting for the capability of additive manufacturing to deal with individual parts characterized by complex geometries.

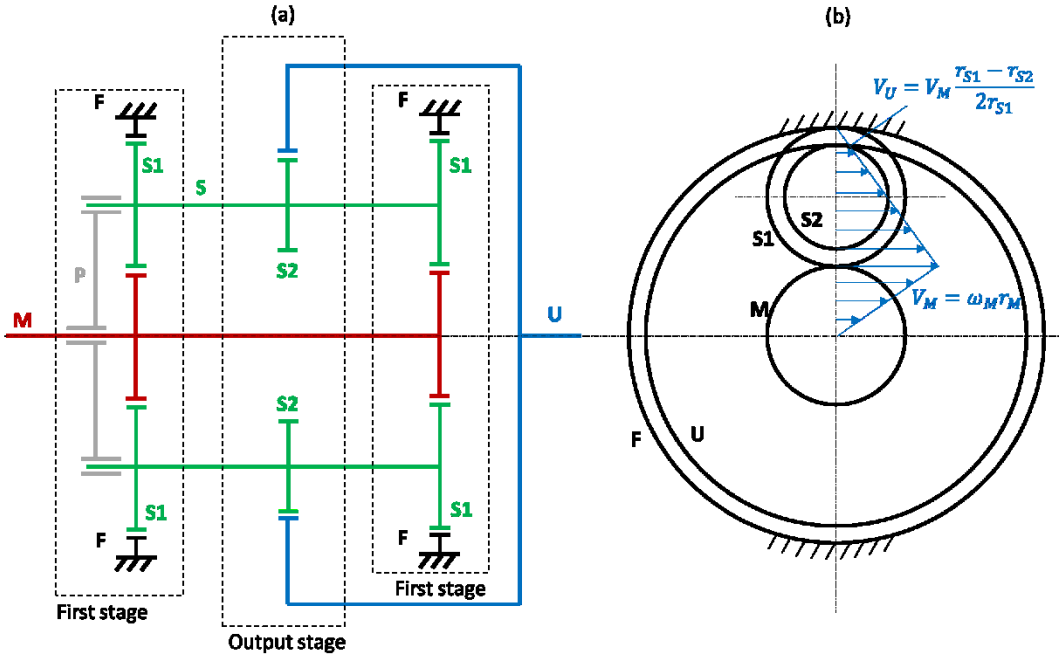

Figure 5.2: (a) Layout of the compound planetary drive. (b) Velocity distribution.

The use of FDM required to design each component accounting for the loose tolerances reachable by the process. A compromise was selected to minimize backlash while reducing interference between the meshing teeth. The rough surface finish resulting by 3D printing however resulted in a significant friction, which drove the sizing of the motor. At the same time, the strength of the material employed for manufacturing is limited, and did not allow to reach the stall load of the motor through the transmission. In order to measure the behavior of the motor under load, the braking unit described in Section 5.1.3 was developed.

A similar gearbox can be produced via traditional machining processes, achieving much higher strength and tolerances. However, the associated costs would be significantly higher, as geometries should be adapted to allow for clearances for tools and some individual parts would require a redesign: as an example, the internal herringbone teeth of the output ring gear would be difficult to obtain from a single part via subtractive manufacturing.

5.1.3 Brake unit

The gearbox is composed by low strength polymeric materials; its maximum allowable torque is a small fraction of the stall torque of the motor. Indeed, the motor is oversized in terms of stall torque to match the required inertia and to overcome the static friction of the transmission. As a result, external loads on the actuator cannot be simulated by applying a torque directly on the output shaft of the gearbox, as a torque large enough to produce visible effects on the motor operation would likely break the transmission. To simulate aerodynamic loads acting on the actuator, a braking unit is connected to the motor shaft through a roller chain, in parallel to the planetary gearbox.

The brake unit is shown in Figure 5.3: the chain drives a shaft parallel to the motor axis, on which a disc brake is mounted. The calliper is installed on a plate supported by two bearings on the shaft: this way, the whole assembly is free to rotate about the axis of the brake unit. This degree of freedom of the calliper is locked by a loadcell. The forces needed to keep the calliper assembly are measured by the cell, thus the braking torque can be determined accurately.

The brake is controlled in closed loop to follow a torque setpoint; an AT328P microcontroller reads the loadcell and commands a small servomotor to actuate the calliper until the desired torque is applied. At the same time, the microcontroller logs the measured torque to a PC.

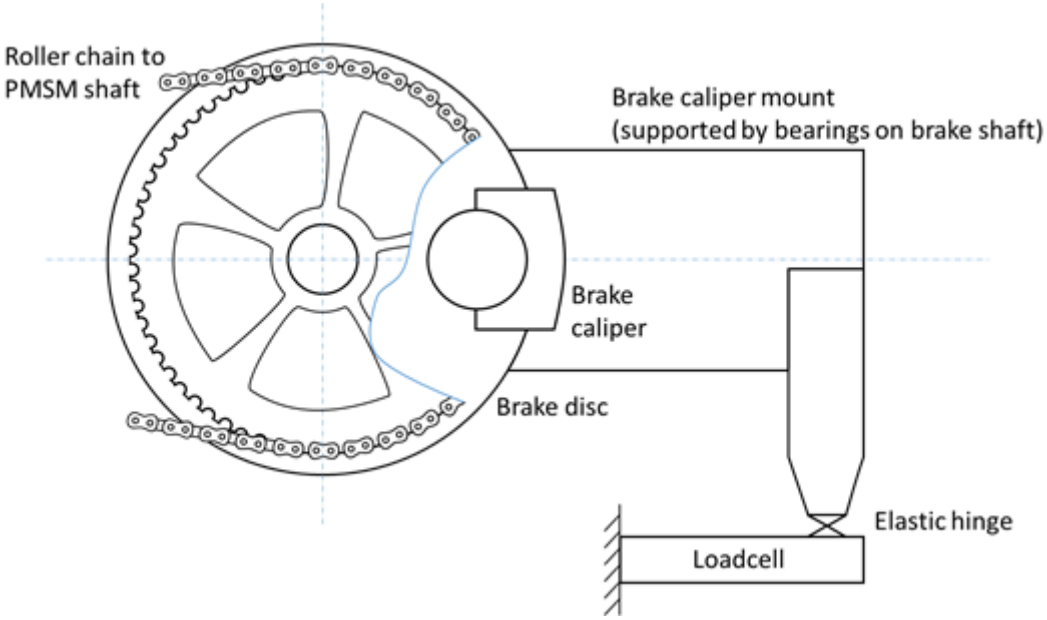

Figure 5.3: Brake unit of the test bench.

5.2 Characterization of components

Although most data regarding the motor and other commercial components employed for the test bench is available from the documentation, some parameters are not provided and require a direct measurement. Specifically, the following Sections discuss the measurement of the friction characteristics of the motor, gearbox and braking unit respectively.

5.2.1 Friction characteristic of the motor

A preliminary experiment was set up to measure the friction characteristic of the motor. The hardware employed for the test is shown in the block diagram of Figure 5.4. The PMSM is connected to a smaller DC gearmotor to drive it while unpowered. The gearmotor is supported by the input shaft of the motor: as a result, the stator has a single degree of freedom corresponding to the rotation around the rotor axis. This degree of freedom is locked by a load cell, installed offset from the motor axis by a lever with the arrangement shown in Figure 5.5. This way, the force sensed by the load cell is a proportional to the motor torque. The motor was driven by two L298N H-bridges, connected in parallel to deliver a sufficient current, and commanded by an AT328P microcontroller; the same controller provided a log of the load cell.

The speed of the motor is measured by its absolute encoder and logged by the control unit. Only the resolver is connected to the control unit; the stator coils are disconnected to prevent current from flowing through the motor phases, and guarantee that the motor does not draw any torque working as a generator. Even then, some torque may be lost due to parasite current in the iron of the stator. This loss is likely small since the involved speed is low; additionally it is present during

the normal motor operation as well, and can be assimilated into the viscous friction losses computed by the EMA models.

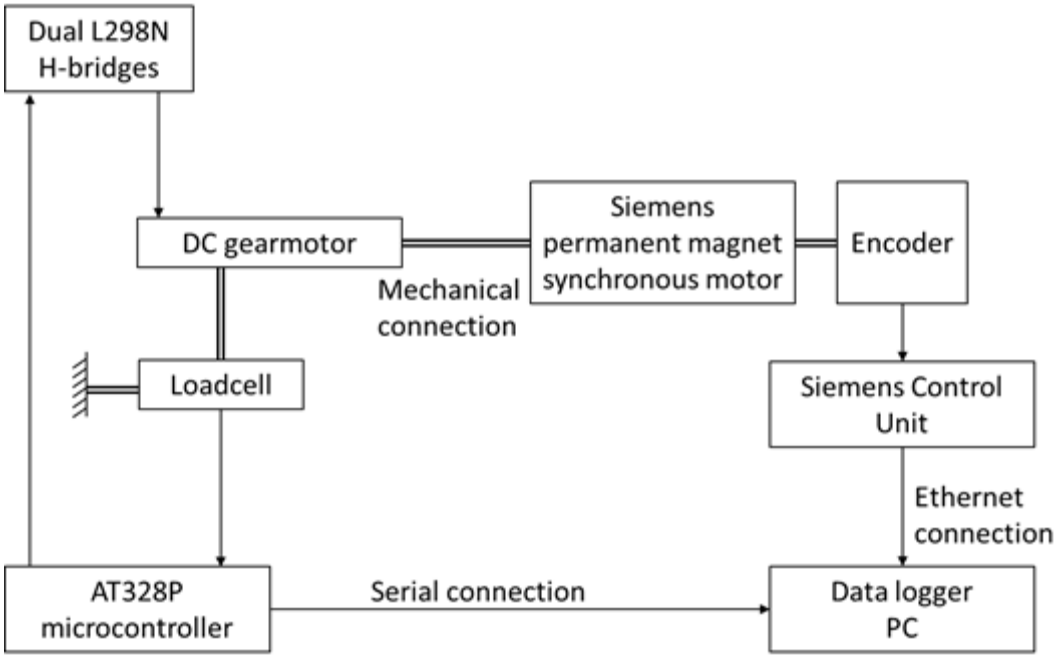

Figure 5.4: Block diagram of the experimental setup for measurement of motor friction characteristic.

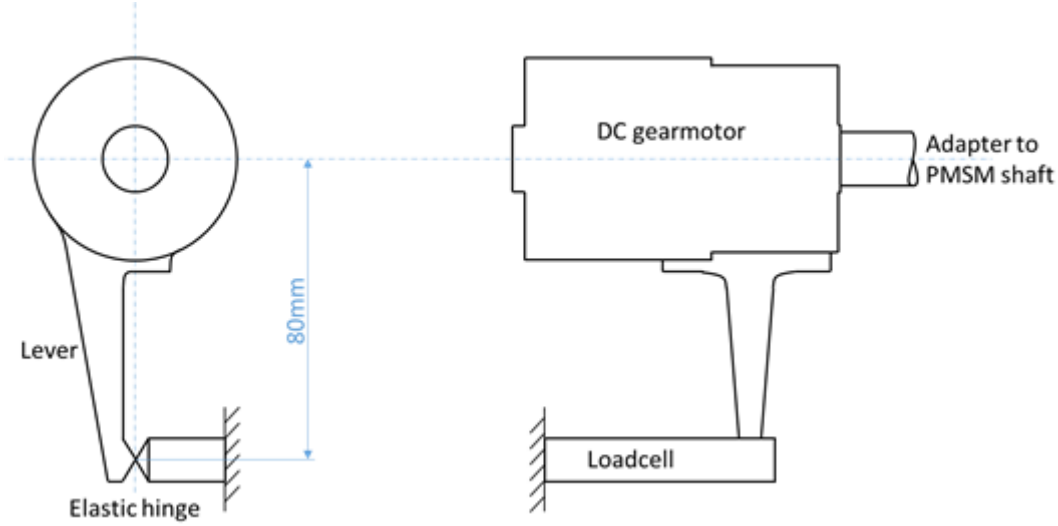

Figure 5.5: Arrangement of the load cell for the measurement of motor torque. The gearmotor is supported by the input shaft of the motor: as a result, its stator has a single degree of freedom corresponding to the rotation around the rotor axis. This degree of freedom is locked by a load cell, offset from the motor axis by a lever. This way, the force sensed by the load cell is a measure of the motor torque.

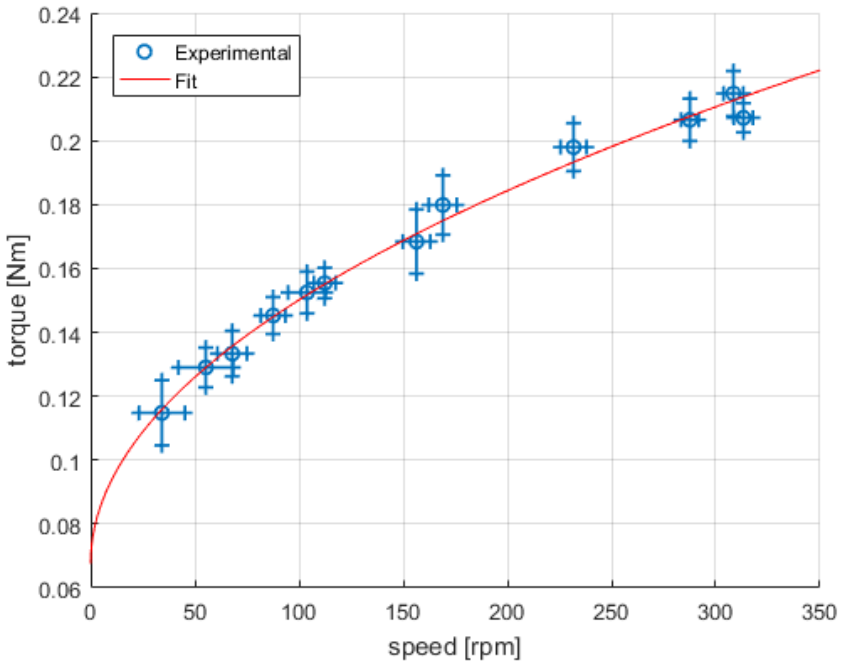

Figure 5.6: Friction torque versus speed characteristic of the motor

The gearmotor-PMSM assembly is driven at constant speed, so that the inertial torques are null. Several data points are measured at different velocities. Figure 5.6 shows the friction torque – speed characteristic of the motor. The blue data points are the measurements; the uncertainty is represented by the errorbars, set at three standard deviations from the average value. The points fit accurately with a square root curve (in red in Figure 5.6). This characteristic, slightly different from a linear damping, is in accordance with models for rolling friction available in literature [147]: in [148] a similar behavior is explained by local heating and displacement of the lubricant.

5.2.2 Efficiency and friction of the gearbox

A similar test allowed to measure the efficiency and preload friction of the gearbox. The test setup is shown in Figure 5.7, while Figure 5.8 provides a functional block diagram of the hardware components.

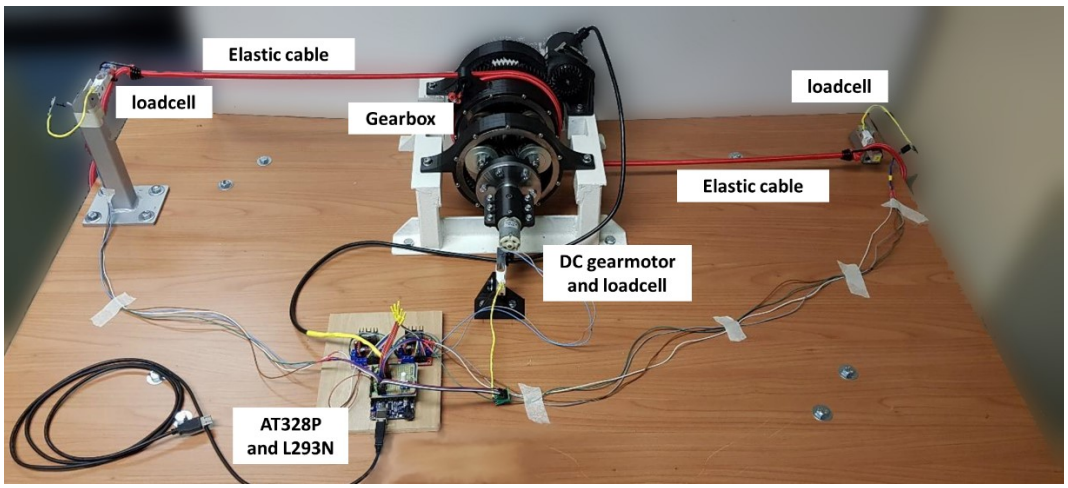

Figure 5.7: Arrangement of the test bench for gearbox characterization.

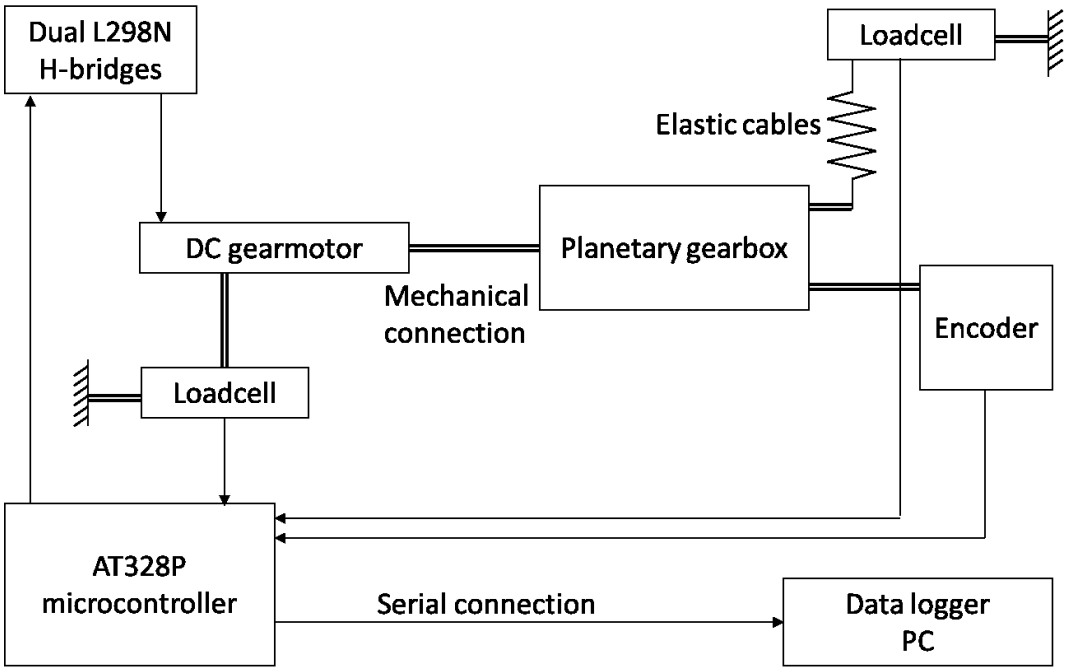

Figure 5.8: Block diagram of the experimental setup for the measurement of gearbox efficiency.

The gearbox was powered by the same DC gearmotor employed for the previous measurement. The load was provided by two elastic cables, with one end rolled around the output ring gear and the other clamped on the frame of the test bench. Three load cells were installed to measure the torque acting on the input and output shafts of the planetary drive. Specifically, two load cells, each with a 20kgf range, were placed in series with the load cords to measure the force applied to the ring gear; the third load cell was employed to measure the motor torque, with the arrangement of Section 5.2.1. Additionally, a position signal is provided by the output shaft encoder installed on the planetary drive. An AT328P microcontroller commanded the two L298N H-bridges driving the gearmotor, provided serial communication to a laptop PC, and acted as a data logger for the position and force

sensors. Table 5.3 summarizes the technical data of the hardware for this experimental campaign.

Table 5.3: datasheet of the hardware for gearbox characterization

Quantity	Unit	Value
Motor voltage	V	6
Motor max current	A	4
Max torque	Nm	2
Internal transmission ratio	--	85.184
Encoder resolution	ppr	80000
	°	0.0045
Full-scale of output shaft load cell	N	200
Accuracy of output shaft load cell	N	0.01
Full-scale of input shaft load cell	N	10
Accuracy of input shaft load cell	N	0.001
Transmission ratio	--	124
External radius of ring gear	mm	81
Offset of motor load cell	mm	80

The motor was controlled in closed loop with a constant speed command. As the output ring gear turned, the tension of the elastic cord increased and so the load torque opposing to the motion of the planetary drive. When the peak load was reached, the rotation was stopped and reversed; as a result, the elastic cords acted as an aiding load: this allowed to completely characterize the behavior of the gearbox and to measure both its direct and inverse efficiencies. An additional test was carried out without external load, in order to determine the preload friction of the reduction drive.

Figure 5.9 shows the log of the measures acquired during a single test. The input torque T_m is computed starting from the reading of the motor load cell F_{lc1}:

$$T_m = F_{lc1} r_m \tag{5.8}$$

where r_m is the offset of the motor load cell with respect to the motor axis. Similarly, the load torque T_u is computed from the measurements F_{lc2} and F_{lc3} of the two output shaft load cells:

$$T_u = (F_{lc2} + F_{lc3})\, r_u \tag{5.9}$$

where r_u is the external radius of the output ring gear. As shown in Figure 5.9, load torque increases nonlinearly with position: indeed, the elastic cords that provide load torque are characterized by a strongly nonlinear and hysteretic behavior. Input torque can be interpreted as the sum of a constant offset and a component proportional to load torque.

The efficiency of the gearbox is defined as the ratio between output and input power, excluding the friction torque caused by preload, that is, torque needed to spin the reducer with no load; the model shown in Figure 5.10 is taken as reference. According to Figure 5.10 (a) The efficiency in opposing load condition is:

$$\eta_D = \frac{\omega_u T_u}{\omega_m (T_m - T_f)} = \frac{T_u}{i(T_m - T_f)} \tag{5.10}$$

where T_f is the preload friction, defined as the average input shaft torque needed to spin the gearbox without any loads applied to the output shaft; only the measures associated to a speed opposed to the load direction are considered (i.e. those taken while loading the elastic cords). Similarly, the aiding load efficiency is computed as (Figure 5.10 (b)):

$$\eta_I = \frac{\omega_m (T_m + T_f)}{\omega_u T_u} = \frac{i(T_m + T_f)}{T_u} \tag{5.11}$$

where the only measures considered are those associated to a speed in the same direction of the load (i.e. while offloading the elastic cords).

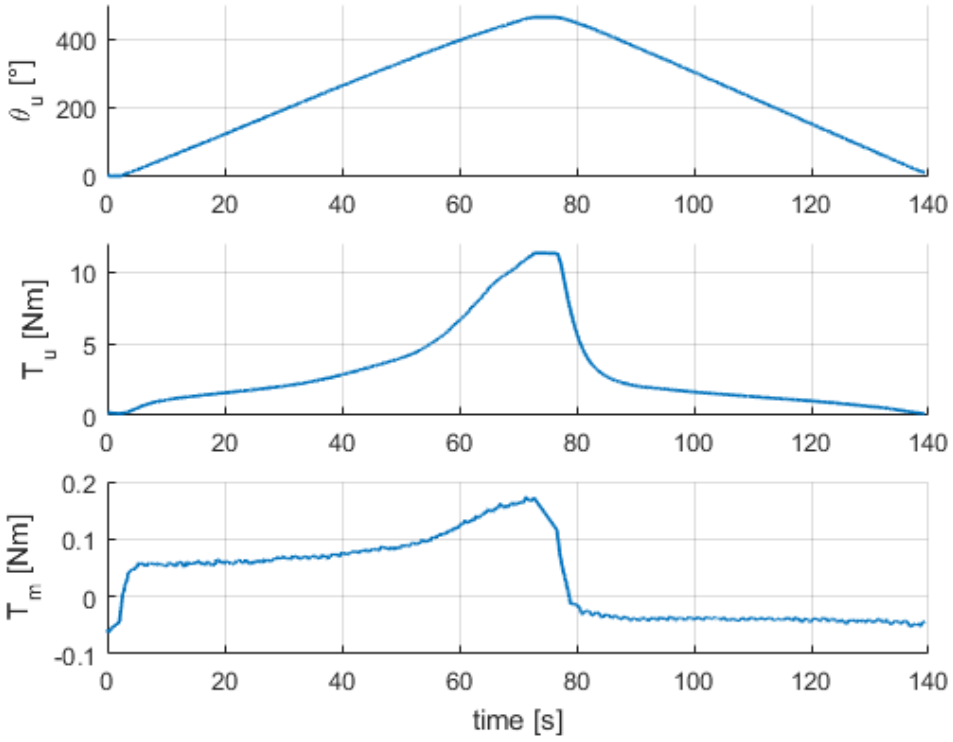

Figure 5.9: Log of measured user position (top), user torque (middle) and motor torque (bottom) for characterisation of the gearbox.

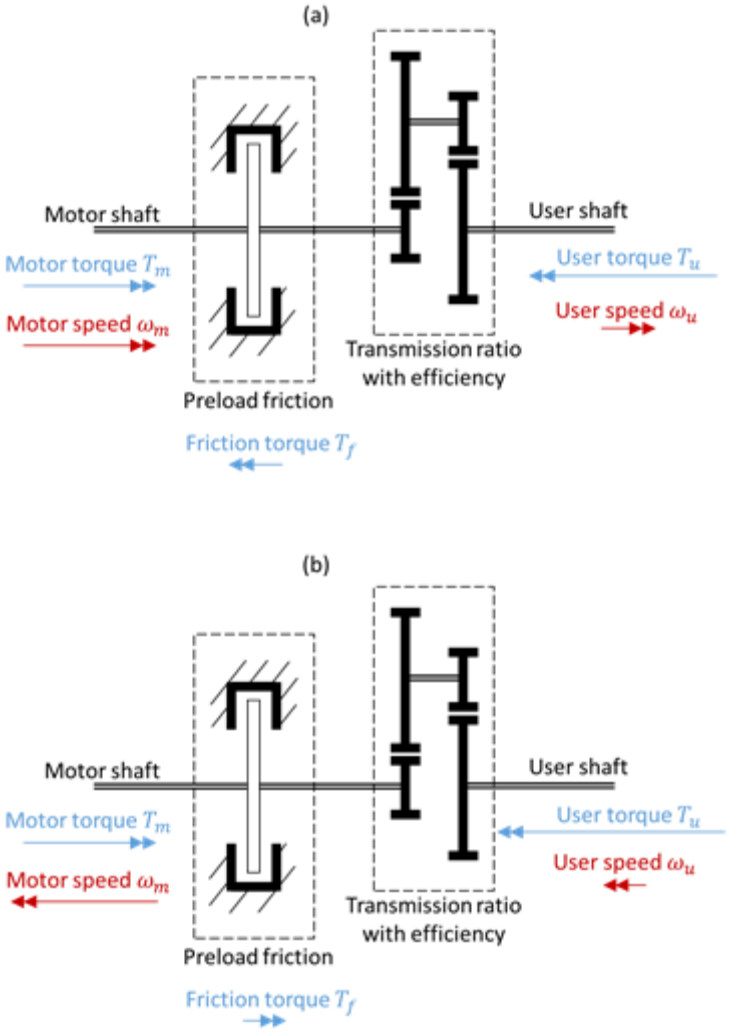

Figure 5.10: Model for computation of transmission efficiency. (a) Opposing load condition; (b) aiding load condition.

Figure 5.11 shows the motor torque as a function of the load torque. The black dashed line is the ideal case for unit efficiency, expressed by the equation $T_u = iT_m$. The points above this curve are associated to a positive speed, and correspond to the opposing load condition. The points below the unit efficiency curve are referred to a negative speed, i.e. to the aiding load operation. The efficiency in the two cases is related to the slope of the fitting line for the points of the graph. Specifically, for opposing load, starting from Equations (5.10) and (5.11), considering that T_f is a constant offset and that the motor and user torques are proportional (neglecting preload friction):

$$\eta_D = \frac{1}{i}\frac{\partial T_u}{\partial T_m} \tag{5.12}$$

and for aiding load:

$$\eta_I = i\frac{\partial T_m}{\partial T_u} \tag{5.13}$$

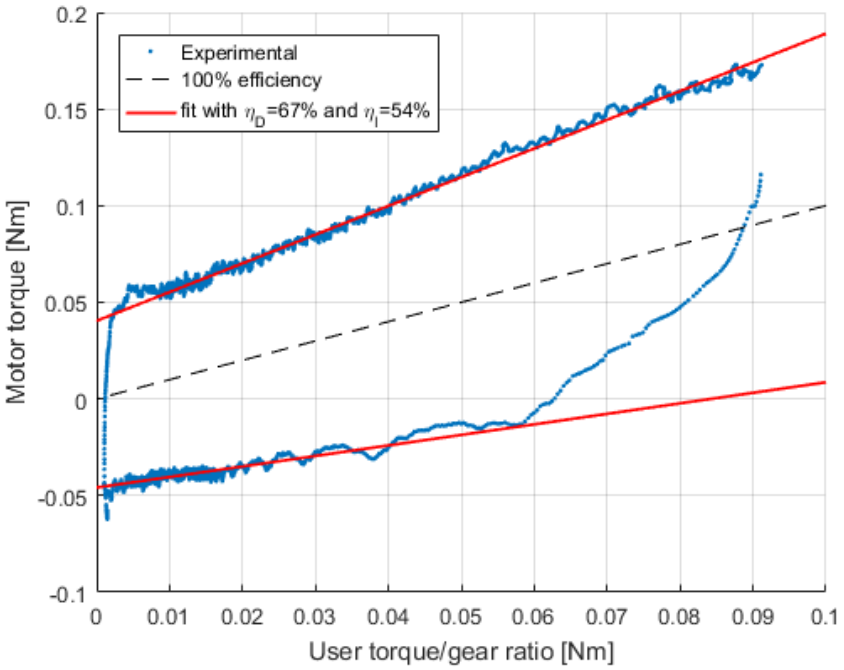

Figure 5.11: Motor torque vs user torque.

The red lines of Figure 5.11 represent the prediction of the efficiency model for 67% opposing load efficiency and 51% aiding load efficiency. The model fits accurately with most of the measured data points; the experimental values at the left end of the aiding friction curve are higher than the expected values: these measurements are acquired near the inversion of direction, when a non-negligible inertial torque is present. Similar considerations hold for the points at the rightmost end of the graph, obtained when the user torque is null and only the preload friction is observed.

The values obtained from this characterization for T_f, η_D and η_I are summarized in Table 5.4. These are well in accordance with the estimate described in [149, 150] and will be used in the HF and LF models.

Table 5.4: Measured efficiencies of the planetary gearbox.

Quantity	Unit	Value	Standard Deviation
Preload friction	Nm	0.045	0.002
Opposing load efficiency	--	67%	0.024
Aiding load efficiency	--	51%	0.132

5.2.3 Friction characteristic of the brake unit

The brake unit includes a roller chain driven by the motor shaft, a brake shaft supported by two self-aligning bearings, and a disc brake whose caliper is mounted on a floating plate, supported by a pair of bearings at one end and by a loadcell at the other end. As this assembly is a custom design, no datasheet is available regarding its friction characteristic and inertia. The moment of inertia is estimated

from the detailed CAD model of the unit, and reduced to the motor shaft according to the transmission ratio of each component. The friction characteristic is measured directly with a procedure similar to that employed for the motor (Section 5.2.1) and planetary drive (Section 5.2.2). As the sprocket of the roller chain is supported by the motor shaft, and its installation interferes with the adapter for the DC gearmotor and loadcell, the PMSM of the test bench is employed directly to move the assembly; its quadrature current i_q, reduced by the current drawn by the motor alone i_{q0}, gives a measured of the required torque T:

$$T_m = \frac{i_q - i_{q0}}{k_v} \tag{5.14}$$

where k_v is the torque gain of the motor. Figure 5.12 shows the measured torque-speed characteristic, reduced at the motor shaft, with the brake completely released. The blue circles are the averaged measures, while the errorbars are set three standard deviations away from the average values. The large uncertainty associated with the low speed measurements is due to the periodic fluctuations in torque produced by a small misalignment of the bearings supporting the brake shaft. The measured friction characteristic fits with a linear viscous friction, combined with a Stribeck effect which is evident at low speeds and can be modelled with a Gaussian curve. The global fitting curve T_{fb} is:

$$T_{fb} = T_{f0} + C_b\omega + a_1 e^{-\frac{x^2}{b_1^2}} \tag{5.15}$$

with $T_{f0} = 0.1843Nm$, $C_b = 1.0685 \cdot 10^{-3}Nms/rad$, $a_1 = 0.0425Nm$, and $b_1 = 5.995rad/s$.

The load torque seen by the motor can be ascribed almost completely to the four bearings installed on the brake shaft. However, the torque due to the two bearings of the caliper plate is read by the loadcell, and contributes to the total load signal fed to the Simulink models. Therefore, this contribution should not be considered in the friction characteristic of the brake assembly. As the four bearings

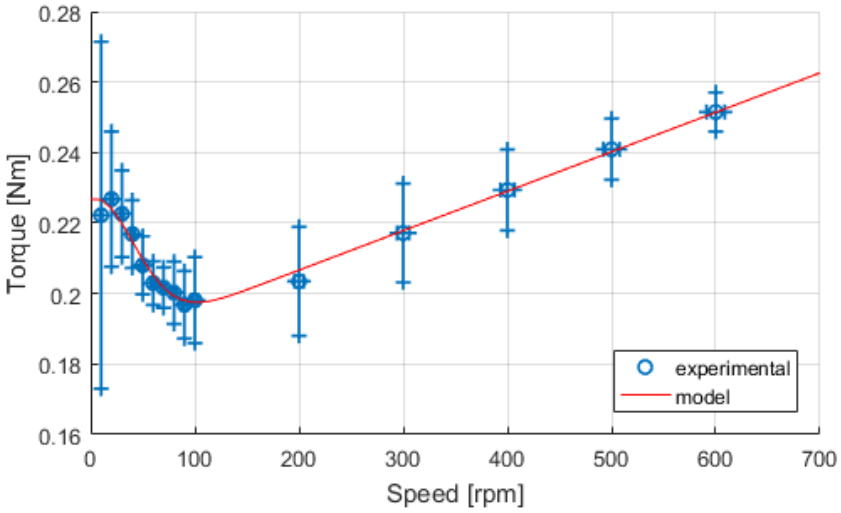

Figure 5.12: Motor speed vs brake shaft torque, brake released.

are equal, the friction characteristic translated into the EMA numerical models is that due to the two fixed bearings, assumed to be half of the curve shown in Figure 5.12.

5.3 Response to a step command

The test bench was initially compared to the high fidelity EMA numerical model (Section 3.1) as regards the response to step commands, both in speed control mode and in position control mode. This activity allowed to calibrate the model in order to reproduce as accurately as possible the output of the physical actuator.

5.3.1 Speed control mode

A step command of 300 rpm amplitude is given to the actuator in speed control mode. Figure 5.13 compares the speed setpoint and actual speed of the HF model (dashed lines) and the physical test bench (continuous lines). The commanded speed is the same for both systems. As shown by the red curves of Figure 5.13, the numerical model is able to accurately follow the dynamical response of the actuator.

Figure 5.14 shows the commanded and actual quadrature currents for the two systems. A small discrepancy is present before the start of the actuation, while the numerical model commands a null current while the physical bench has a small nonzero current, probably due to the interaction with static friction. The two systems are in accordance for the remaining part of the actuation, with the actual currents following their setpoints with a delay in the order of the electric characteristic time of the motor (equivalent to about 11ms).

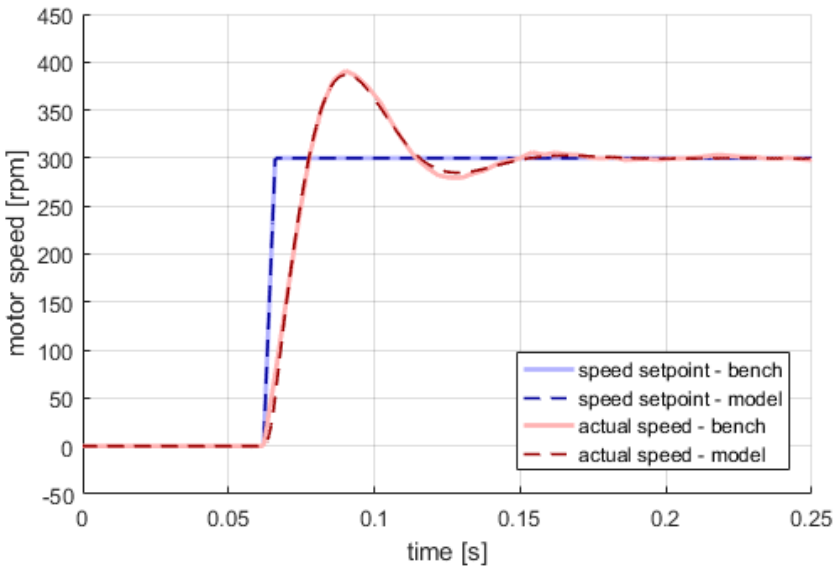

Figure 5.13: Speed setpoint vs measured speed, step command in speed control mode

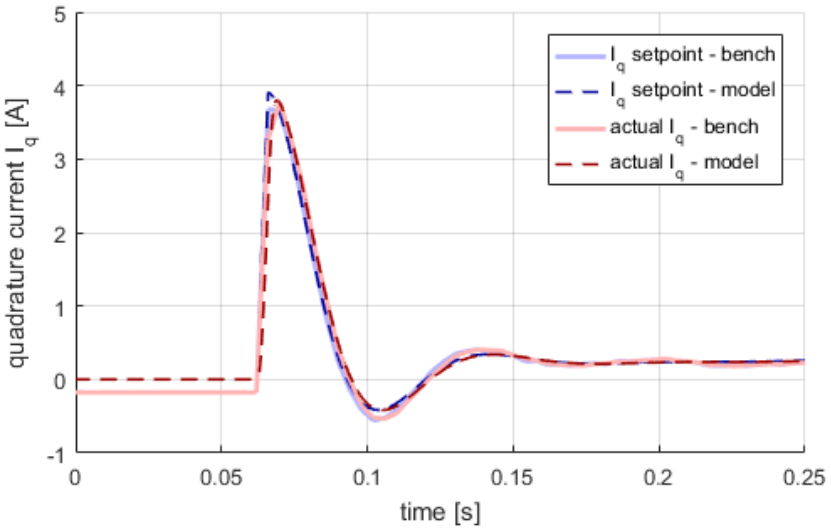

Figure 5.14: Quadrature current setpoint vs measured quadrature current, step command in speed control mode.

The measured direct currents are reported in Figure 5.15. The setpoint for both systems is zero, and both observe direct currents that are at least two orders of magnitude smaller than the respective quadrature components. In this case however, the model is not able to predict accurately the behaviour of the physical system: only the overall amplitude is reproduced correctly. This result is expected, since, the current control loop is slightly different between the two models: while the test bench employs a full d-q control with a dedicated PI regulator, the HF model has a simpler hysteresis controller on each of the three motor phases.

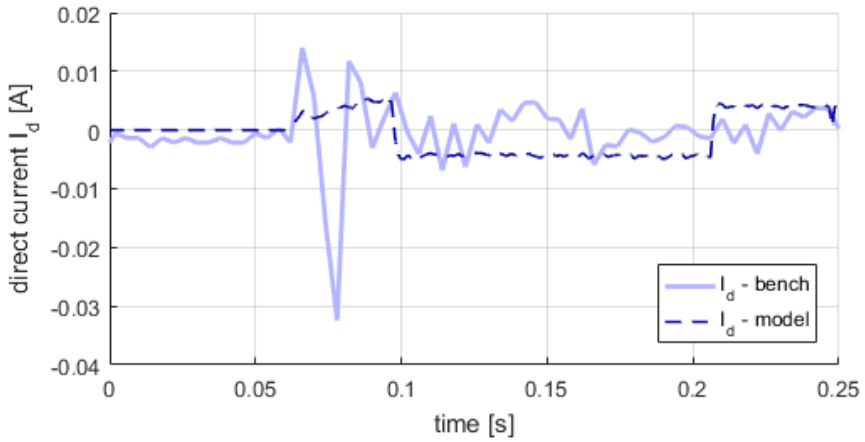

Figure 5.15: Direct current, step command in speed control mode.

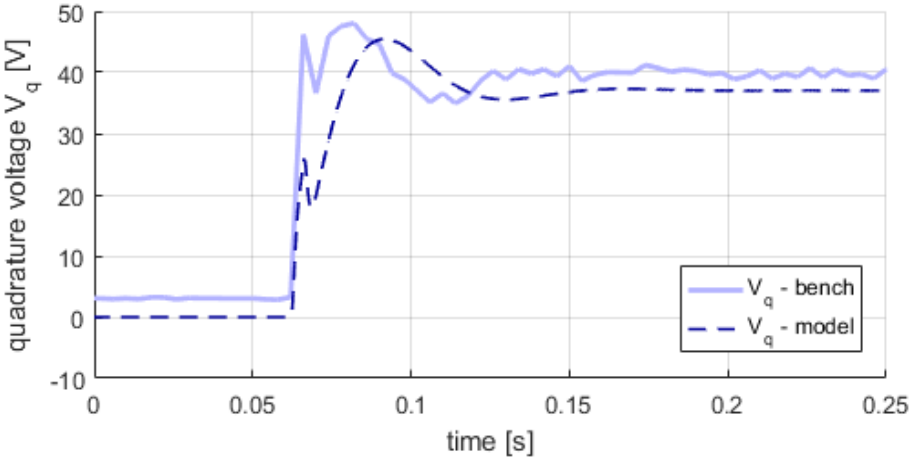

Figure 5.16: Quadrature voltage, step command in speed control mode.

Similar considerations hold for the quadrature voltage (Figure 5.16): the general shape of the waveform and the steady-state response are captured correctly, while the transient response has a higher deviation. Specifically, the d-q control of the test bench is faster to reach the steady-state value, with a slightly larger overshoot.

5.3.2 Position control mode

A similar test has been performed in position control mode, with a step command of 60° amplitude on the user shaft. Figure 5.17 shows the commanded position and the response of the two actuators. The position setpoint is not an ideal step, since this command would trigger the overcurrent protection of the physical actuator. The step setpoint is at first filtered by a speed and acceleration limiter, resulting in the blue curves of Figure 5.17. Both systems follow the command with a delay of approximately 60ms, and the two responses are almost indistinguishable.

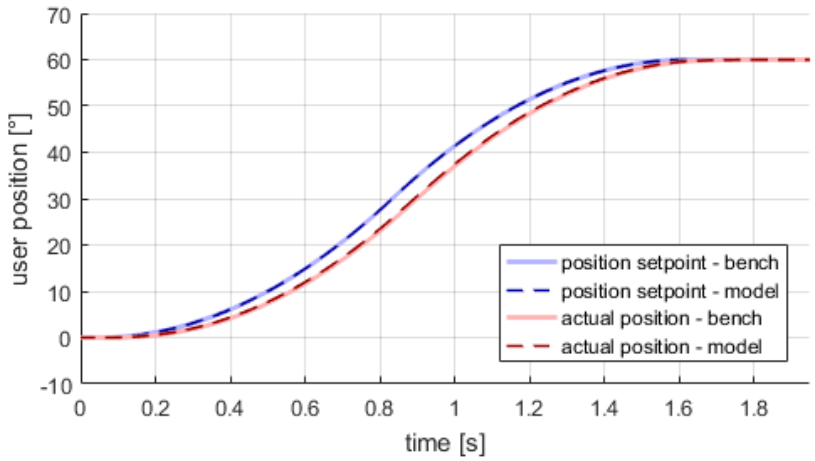

Figure 5.17: Position setpoint vs measured position, step command in position control mode.

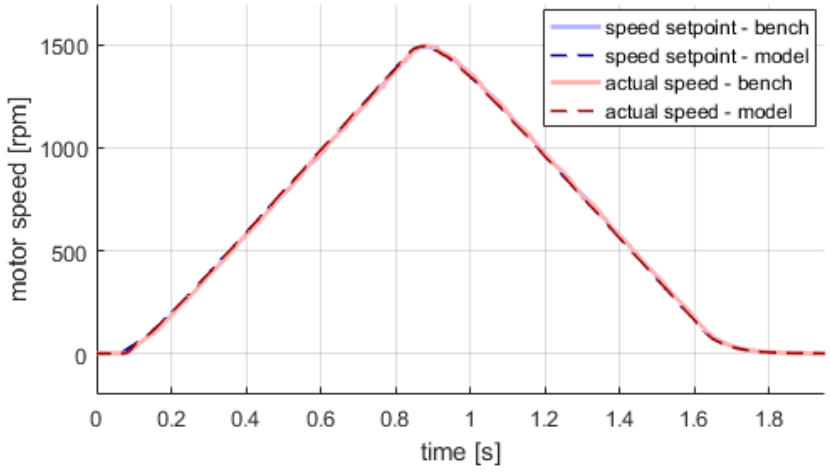

Figure 5.18: Speed setpoint vs measured speed, step command in position control mode.

A similar position error yields a similar speed setpoint for both systems, visible in Figure 5.18, as they share the same control logic. The triangular shape of the velocity curves reflects the operation of the acceleration limiter, set at 2000 rpm/s. Again, both systems are able to follow closely the commanded speed, with a very similar response.

Figure 5.19 compares the commanded and measured current components in the quadrature direction. The model follows the physical system with some errors; specifically, the low frequency components of the response are predicted accurately, while the high frequency ones are attenuated. The source of this discrepancy has been determined to lie in the hardware employed to acquire the external load on the physical actuator, and feed it in input to the numerical model. The sampling frequency of this system is limited to 80Hz, which is not sufficient to capture accurately the high frequency fluctuations in load, due for example to the alignment imperfections of the bearings of the brake shaft.

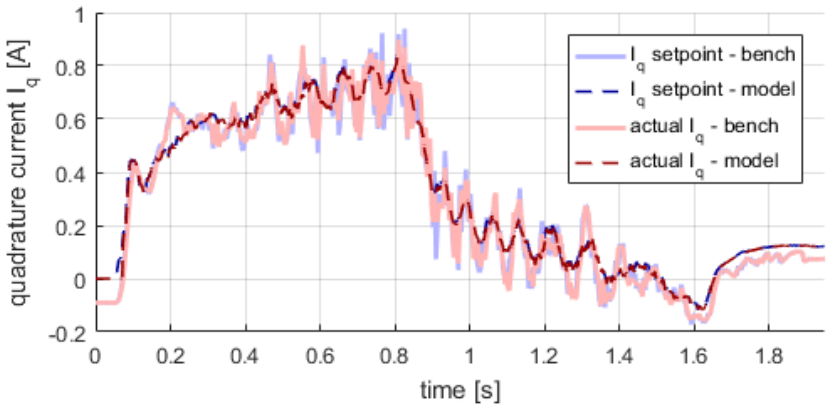

Figure 5.19: Quadrature current setpoint vs measured quadrature current, step command in position control mode.

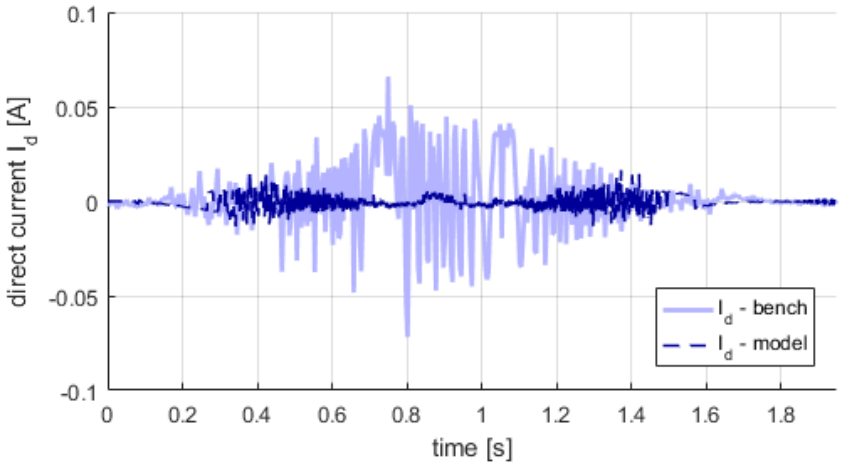

Figure 5.20: Direct current, step command in position control mode.

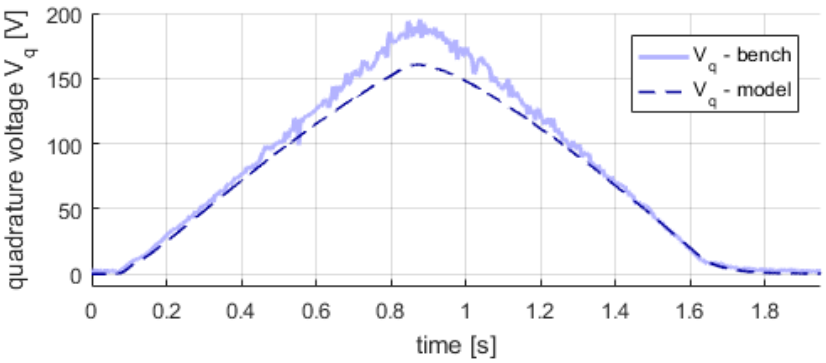

Figure 5.21: Quadrature voltage, step command in position control mode.

The current component along the direct axis is shown in Figure 5.20, while Figure 5.21 reports the voltage in the quadrature direction. For these quantities, the same considerations exposed for the speed control mode hold true. The systems feature a different control logic as regard the management of the direct current component and a different activation of the MOSFET switches inside the inverter, so a discrepancy in behaviour as regards these signals is expected.

5.4 Open-loop Bode plot

As a further validation of the HF model, the open-loop Bode diagram is computed for the model and compared to the response obtained experimentally. The HF model cannot be linearized without losing significant characteristics of its response; at the same time, the output signal from the test bench is sampled with a constant frequency. Therefore, analytical expressions for the model and bench outputs are not available, nor are they exact sine waves. To compute the gain and phase of the output y with respect to the input x for a given frequency $f = \omega/2\pi$, the Fourier coefficients for x and y are integrated numerically:

$$Re(x) = \frac{2}{t_s}\int_0^{t_s} x \sin \omega t \, dt \tag{5.16}$$

$$Im(x) = \frac{2}{t_s}\int_0^{t_s} x \cos \omega t \, dt \tag{5.17}$$

$$Re(y) = \frac{2}{t_s}\int_0^{t_s} y \sin \omega t \, dt \tag{5.18}$$

$$Im(y) = \frac{2}{t_s}\int_0^{t_s} y \cos \omega t \, dt \tag{5.19}$$

The amplitudes of the input and output signals are respectively:

$$A_{in} = \sqrt{Re(x)^2 + Im(x)^2} \tag{5.20}$$

$$A_{out} = \sqrt{Re(y)^2 + Im(y)^2} \tag{5.21}$$

and their phases can be written as:

$$\phi_{in} = \operatorname{atan}\frac{Im(x)}{Re(x)} \tag{5.22}$$

$$\phi_{out} = \operatorname{atan}\frac{Im(y)}{Re(y)} \tag{5.23}$$

The gain is obtained as the ratio between the output and input amplitudes, expressed in decibels:

$$G_{dB} = 20 \log_{10}\frac{A_{out}}{A_{in}} \tag{5.24}$$

and the phase is the difference between the output and input phases:

$$\phi = \phi_{out} - \phi_{in} \tag{5.25}$$

Figure 5.22 shows the open-loop Bode plot of the actuator. The continuous line is obtained through the HF model, while the points are the measures from the physical test bench.

As the system is strongly nonlinear, its behavior is dependent on the input amplitude. Therefore, three curves have been evaluated, for input amplitude of 0.01rad, 0.05rad and 0.2rad respectively, measured on the slow shaft. The dynamical response of the system is evaluated for frequencies from 0.1Hz to 100Hz; this range is limited by the capability of the hardware installed on the test bench, and specifically by the bandwidth available for communication between the actuator control unit and the host PC.

Clearly, the numerical simulations match correctly with the experimental data on most of the observed range in frequency and amplitude. The experimental curve corresponding to the highest amplitude (0.2 radians on the output shaft) is limited to a maximum frequency of about 10Hz. Above this value, the overcurrent protection of the test bench is triggered, while the HF model can be evaluated regardless of this limitation. A significant discrepancy between the model and

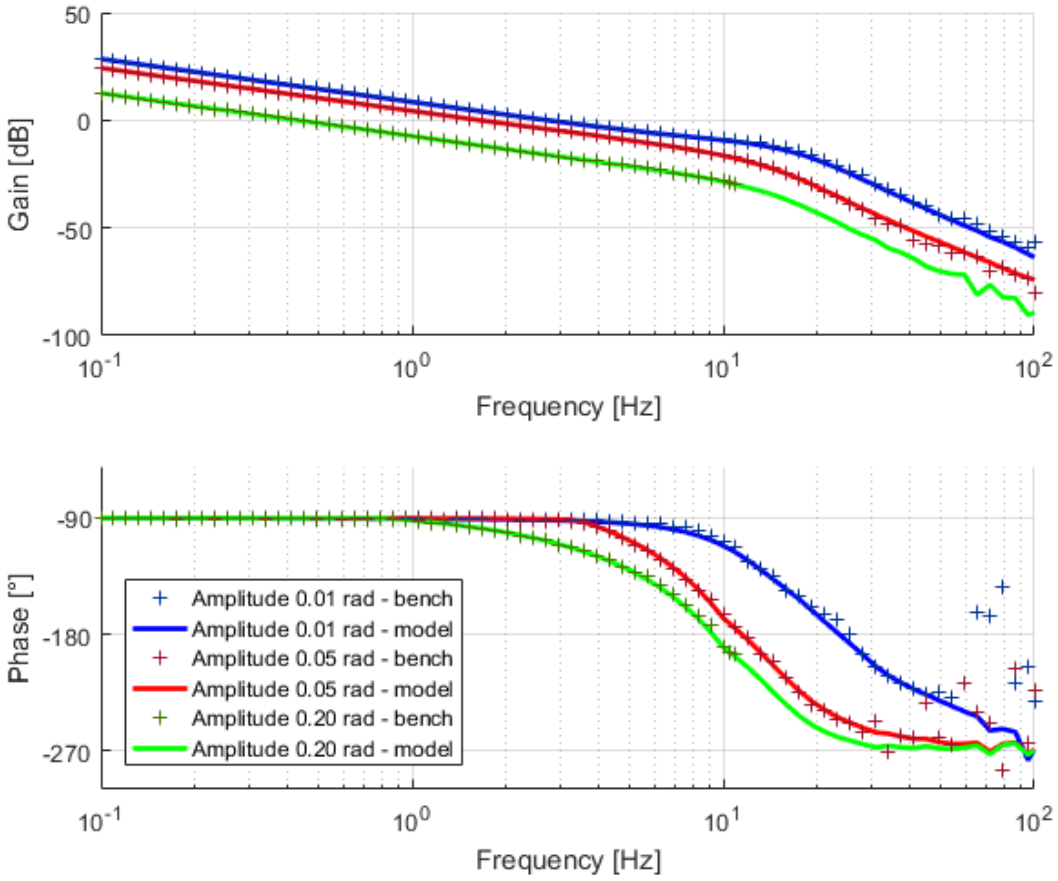

Figure 5.22: Open-loop Bode plot of the system

experimental results appears for frequencies above 30Hz: while the response of the numerical model evolves regularly, the physical test bench produces an unpredictable and noisy curve. This behavior can be ascribed to the different sampling frequencies of the two systems. The physical actuator is limited to a 250Hz sampling, while the model can record its output with a frequency as high as the integration frequency, set at 1MHz for numerical stability. Therefore, according to the Nyquist-Shannon theorem, significant aliasing phenomena may start to occur at one tenth of the sampling frequency, that is, 25Hz for the test bench.

The system is controlled in open-loop for position, and in closed-loop for velocity. As a result, at low frequency the system acts as an integrator. This is visible in the Bode plot, as the gain shows a slope of -20dB/dec while the phase is constant at -90°.

At around 10Hz the gain slope increases to -60dB/dec, while the phase gets to -270°. This is the result of the superposition of two contributions: a linear one, related to the closed-loop transfer function of the speed loop, and a nonlinear one, emerging from the combination of the speed and acceleration limits of the system.

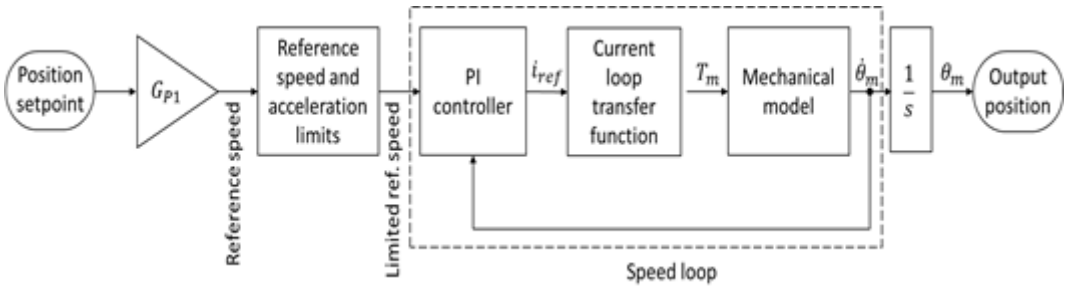

Figure 5.23: Simplified open-loop block diagram of the actuator

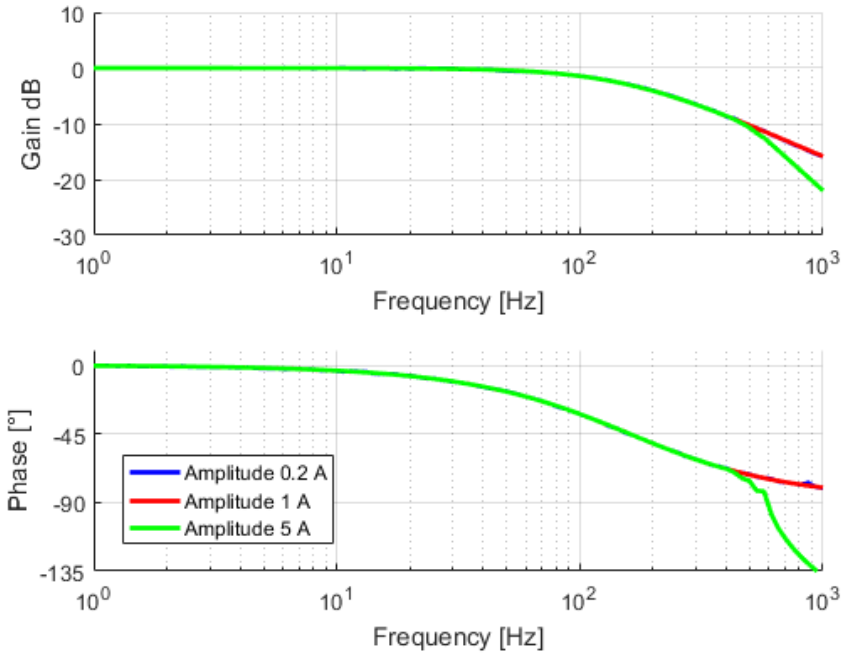

Figure 5.24: Bode plot of the current loop.

Figure 5.23 shows a simplified block diagram of the open-loop system. The transfer function of the current loop cannot be derived analytically, as its hysteresis control cannot be linearized. The bode plot of this block is obtained numerically and shown in Figure 5.24; in the considered frequency range (0.1Hz to 100Hz) the current loop can be assumed as instantaneous, since it shows a unit gain and zero phase.

The linearized transfer function of the speed loop can be then computed as:

$$\frac{y}{x} = \frac{G_{P2}s + G_I}{Js^2 + (C + G_P)s + G_I} \tag{5.26}$$

where G_{P2} and G_{I2} are the proportional and integral gains of the speed controller. With the parameters employed in the test bench, the numerator has a zero at $\omega = 100 rad/s = 15.9$Hz, while the denominator is a second order model with natural frequency $\sigma_n = 93{,}65\, rad/s = 14.9$Hz and damping $\zeta = 0.47$. Its Bode plot is reported in Figure 5.25: the response is close to that of a first order system with cutoff frequency around 15Hz.

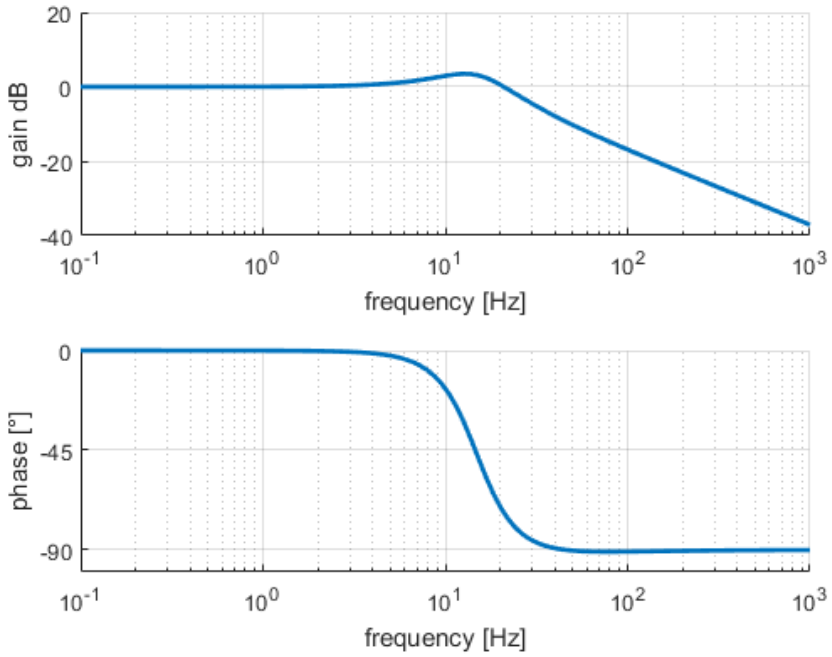

Figure 5.25: Bode plot of the speed loop.

The open-loop transfer function of the whole system is the product of the proportional gain of the position controller, the transfer function of the reference speed and acceleration limiter, the transfer function of the speed loop, and the integrator that converts the output speed into the output position.

The reference speed and acceleration limits have a nonlinear behavior that cannot be dealt with analytically. Their transfer function can however be computed numerically, similarly to that of the current control loop. The Bode plot for this block is shown in Figure 5.26. The behavior of this subsystem approximates that of a linear first order system whose cutoff frequency and static gain depend on the input amplitude. For the operating conditions found in the actuator, the cutoff frequency is in the order of 10Hz. This behavior is expected as for low frequency the acceleration limit is not active. For very high frequency, the output is a triangular wave whose amplitude is inversely proportional to the frequency, hence the slope of -20dB/dec.

By comparing the experimental Bode plot of the whole system (Figure 5.22) with those of its building blocks (Figure 5.24 to Figure 5.26), obtained either numerically or analytically, the measured response of the actuator can be explained as the superposition of the individual subsystems.

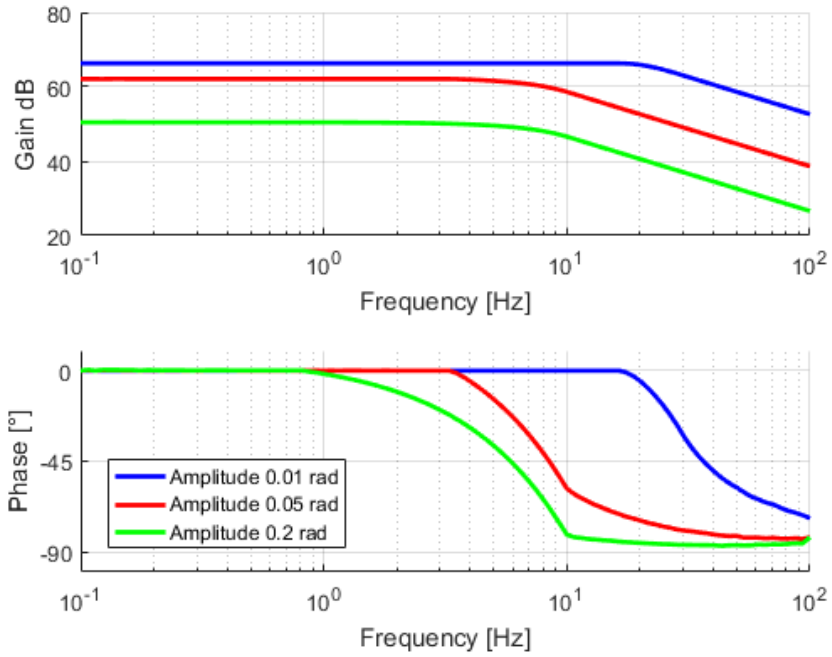

Figure 5.26: Bode plot of the reference speed and acceleration limiter.

Chapter 6

Validation of Machine Learning models

The following Sections discuss the preliminary results obtained offline and related to the validation of the machine learning models employed to speed up the computations online. Specifically, Section 6.1 motivates the choice of the Back-EMF coefficient as a monitored quantity of interest; Section 6.2 details the acquisition of the training and validation datasets for the surrogate models; Section 6.3 presents the preliminary results related to the Signal Acquisition and Compression task; Section 6.4 shows the preliminary results of the Fault Detection and Identification (FDI) task; Section 6.5 discusses the validation of the Support Vector Machine employed as a surrogate assessment function.

6.1 Choice of monitored signal

The choice of suitable monitored features is a key step of any diagnostic and prognostic methodology. The observed signals shall be sensitive to the considered fault modes, and only marginally affected (not affected at all, ideally) by externally induced variations in the system's operating conditions. In this study, the selection of the monitored signals is based on Expert Knowledge. This approach relies on legacy information on the technology family and improves the interpretability of the method; conversely, systematic feature selection algorithms may significantly ease the application of the prognostic framework to other systems.

A possible choice for a map of the system behavior to be monitored for FDI is the back-Electromotive Force (back-EMF) coefficient as a function of the rotor position. The employ of this quantity for detecting electrical faults was initially proposed in [151] and the feasibility of the method was demonstrated, in combination with machine learning, in [67].

The back-EMF coefficient map of a BLDC or PMSM machine can be estimated from available measurements, without adding dedicated sensors. Specifically:

- Phase currents are typically measured as a feedback signal to close the current-torque control loop
- Phase voltages are known without dedicated sensors, as they are a fraction of the supply voltage equal to the PWM duty cycle applied to each phase
- Rotor position is measured by Hall sensors, a rotary encoder or a resolver to synchronize the phase commutation sequence with the rotor position. In

some systems, this signal is employed for speed or position closed-loop control

- Rotor speed can be estimated as the time-derivative of rotor position, or measured by separated speed sensor.

The back-EMF coefficient $k(\theta)$ is reconstructed starting from the governing equation of the RL circuit that models the stator:

$$V(t) - k\big(\theta(t)\big)\frac{d\theta(t)}{dt} = Ri(t) + L\frac{di(t)}{dt} \tag{6.1}$$

where $V(t)$ is the voltage applied to the motor, $\theta(t)$ is the rotor position, $i(t)$ is the motor current, and R and L are the motor nominal resistance and nominal inductance, respectively. Equation (6.1) can be rearranged as follows to estimate the back-EMF coefficient:

$$k\big(\theta(t)\big) = \frac{V(t) - Ri(t) - L\frac{di(t)}{dt}}{\frac{d\theta(t)}{dt}} \tag{6.2}$$

The output of this step is the back-EMF coefficient $k\big(\theta(t)\big)$ sampled in time, with the acquisition frequency of the sensors. A resampling of the signal allows producing the map of back-EMF coefficient versus rotor position. To do so, a set of uniformly spaced values θ_i for the rotor angle θ is determined. Then, the points of $k(t)$ corresponding to a $\theta(t) \in [\theta_i - \delta,\ \theta_i + \delta]$ are selected, and their average is assigned to $k(\theta_i)$, for each θ_i. For the collection of the training and validation datasets, this process required about 1 second of computational time for 2 seconds of simulation time.

Although this resampling is quite computationally intensive, once an informative acquisition mask is determined as per Section 2.2 the acquisition of the sensors and the reconstruction of k can be performed in real time only for the informative points, triggered by the current rotor position.

Figure 6.1 shows the reconstructed back-EMF coefficient for varying electrical faults, commands, and external loads. As highlighted by Figure 6.1 (a), electrical fault modes have a significant and characteristic effect on the back-EMF coefficient map, which can be leveraged for FDI. On the contrary, the signal is insensitive to the operating conditions in terms of command time history and external load profile on the actuator: then, it is a suitable candidate for real-time health monitoring in actual flight conditions.

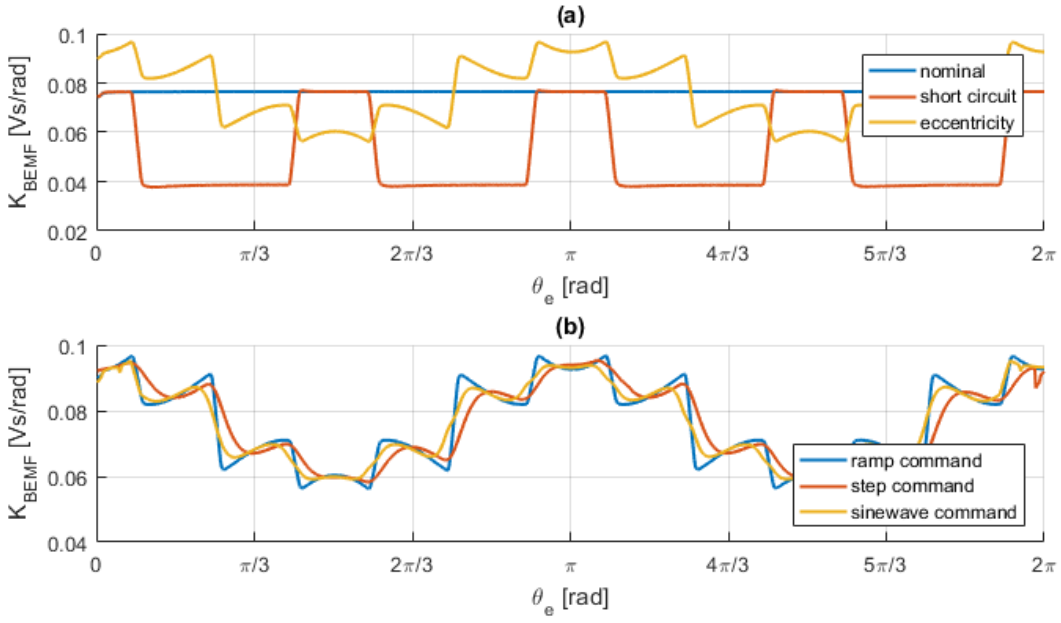

Figure 6.1: (a) Reconstructed map of the back-EMF coefficient $\boldsymbol{k}$ versus rotor angle $\boldsymbol{\theta}$ for different electrical faults. (b) Back-EMF coefficient reconstructed for different commands and loads: the resulting signal is mostly independent from the particular operating condition experienced by the actuation system.

The effect of mechanical faults on the signal is negligible and comparable to that of the external load. As a result, other quantities of interest need to be measured to detect these damages for a complete system health monitoring routine.

6.2 Acquisition of a training dataset

A training set and two validation sets are collected for the assessment of the machine learning models used for the PHM process. The datasets are computed with the physics-based models described in Section 3, and include the fault condition $\boldsymbol{k}$ and the corresponding curve of back-EMF coefficient versus rotor position, employed as the informative behavior map $\boldsymbol{y}(\boldsymbol{k})$.

The fault vector $\boldsymbol{k} = [k_1, k_2, k_3, k_4, k_5]$ is a five-element vector encoding the health condition of the system. Each element of $\boldsymbol{k}$ is normalized between 0 and 1. Two electrical fault modes are considered:

- A partial short circuit of the stator windings. This information is contained in the first three elements of the fault vector. The fault parameters k_1, k_2, k_3 express the fraction of healthy windings of each phase of the motor: $k_1 = 1 - N_a$, $k_2 = 1 - N_b$, $k_3 = 1 - N_c$, where N_a, N_b, N_c are the healthy windings of each phase.
- An eccentricity of the rotation axis of the rotor with respect to the axis of symmetry of the stator. This fault is expressed by the last two elements of $\boldsymbol{k}$, which are a function of the eccentricity amplitude ζ and phase ϕ. Specifically, the eccentricity is converted in Cartesian coordinates and normalized between 0 and 1: $k_4 = (1 + \zeta \cos\phi)/2$ and $k_5 = (1 +$

$\zeta \sin\phi)/2$. The switch to Cartesian coordinates permits to avoid a singularity in the nominal condition ($\zeta = 0$) where the phase is undetermined and can affect the accuracy and repeatability of fault detection.

The training set includes $n_s = 1000$ fault conditions sampled with the technique described in Section 2.1, and the associated back-EMF coefficient curves, computed with the LF model (Section 3.2). This number of training points was found in a previous study to be sufficiently representative of the system behavior.

A first validation set includes 100 fault conditions and the associated back-EMF coefficient curves, computed with the LF model. A second validation set includes 100 fault conditions and the associated back-EMF coefficient curves, computed with the HF model (Section 3.1).

The use of the LF model for the training set permits to allocate a reasonable computational time for the collection of the datasets. The HF model is employed for validation: the relationship between HF and LF models simulates that between the physical system and its physics-based digital twin. Indeed, in a field application, a sufficient amount of data for training cannot be collected with experiments alone, and some training points need to be computed with physics-based simulations. The use of a model with a different fidelity for validation permits then to determine the robustness of the proposed methodology to the structured and unstructured uncertainties in the simulations.

On the other hand, the first validation set is computed with the same LF model, to evaluate the behavior of the proposed methodology regardless the discrepancy between the HF and LF models.

6.3 Signal Acquisition, Feature extraction and Compression

At first, Proper Orthogonal Decomposition (POD) is applied to the back-EMF coefficient maps of the training set, as per Section 2.2. POD extracts the dominant modes that describe the variance of the data; the amount of dispersion explained by each POD mode is related to the respective eigenvalues. **Errore. L'origine riferimento non è stata trovata.** (a) shows the ordered eigenvalues λ_i in logarithmic scale, and **Errore. L'origine riferimento non è stata trovata.** (b) shows their cumulative sum $\sum_{j=1}^{n_m} \lambda_j \,/\, \sum_{j=1}^{n_s} \lambda_j$: clearly, the decay of eigenvalues is quite fast, and the first 11 modes are able to explain 99.9% of the training set information.

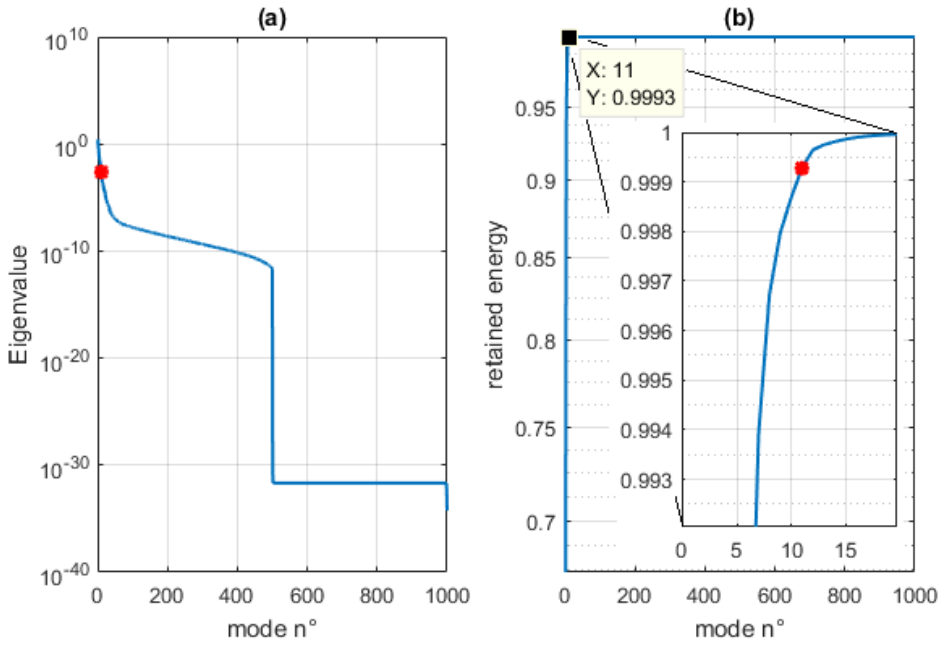

Figure 6.2: (a) Decay of the POD eigenvalues for the training set; (b) cumulative sum of the eigenvalues.

The second step of signal compression leveraged the training of Self Organized Maps on the first n_m modes, for $1 \leq n_m \leq 12$. Figure 6.3 shows the placement of the compression mask onto the first 12 POD modes. The mask obtained by training a SOM with the first n_m modes is plotted onto the n_m-th mode. The points of the masks tend to be placed near the local maxima and minima of the modes, which usually correspond to the most informative locations for measurement; this is particularly evident for a low number of retained POD modes. By increasing n_m, the mask gets closer to a uniform sampling, since the location of its points is a compromise between the optimization for each individual mode: this behavior may lead to the inclusion of misleading information in the acquisition and reduce the accuracy of FDI if an excessive number of POD modes are retained, as shown in [152].

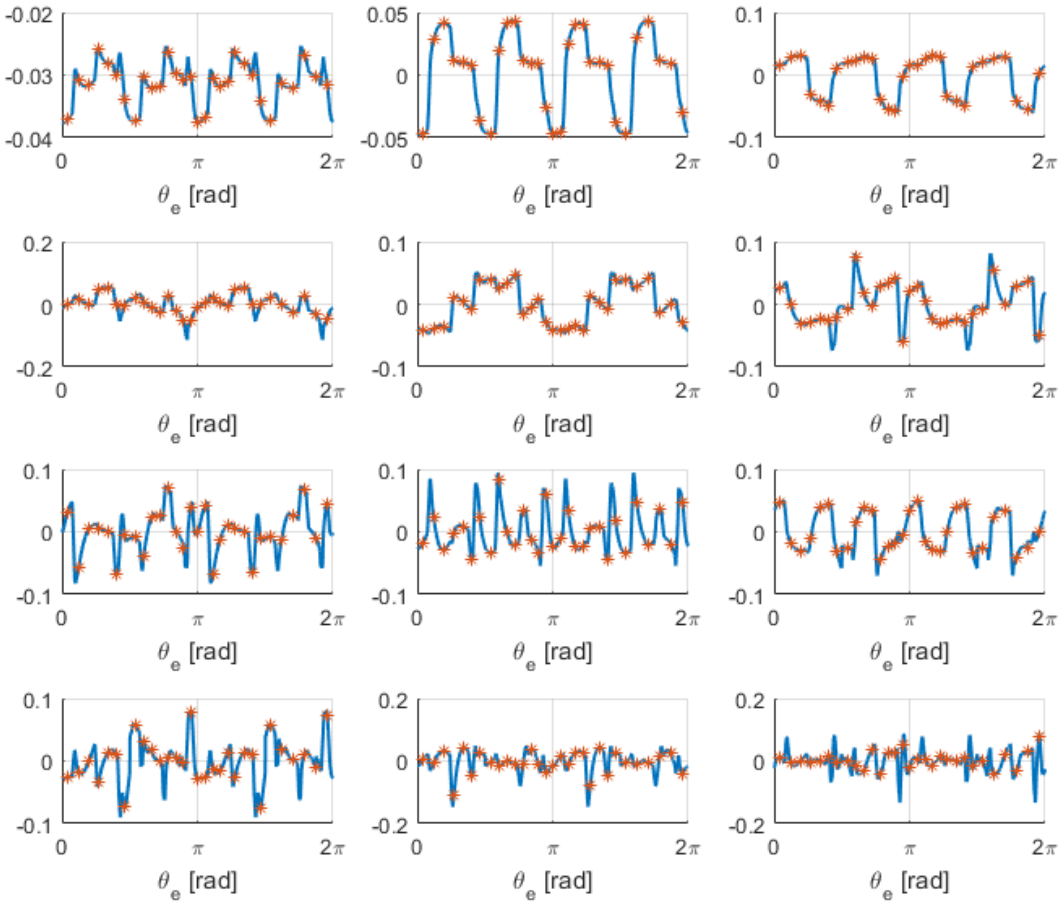

Figure 6.3: Placement of the sampling points of the compression mask on the first 12 POD modes. The mask obtained by training a SOM with the first $\boldsymbol{n_m}$ modes is plotted onto the $\boldsymbol{n_m}$-th mode.

Figure 6.4 shows the error associated with the recovery of the information of the full back-EMF curve via Gappy POD. Specifically, the boxplot of Figure 6.4 (a) shows the Normalized Root Mean Squared Error (NRMSE) on reconstruction of the back-EMF coefficient:

$$err_y = \frac{\sqrt{\frac{1}{n_e}\sum_{i=1}^{n_e}\left(y_{e,i}^2 - y_{o,i}^2\right)}}{\max(\boldsymbol{y}_o) - \min(\boldsymbol{y}_o)} \tag{6.3}$$

where $\boldsymbol{y}_o$ is the original back-EMF curve, in the form of an n_e-dimensional vector, and y_e is the one estimated by Gappy POD. The NRMSE on the POD coefficients is reported in Figure 6.4 (b):

$$err_\alpha = \frac{\sqrt{\frac{1}{n_m}\sum_{i=1}^{n_m}\left(\alpha_{e,i}^2 - \alpha_{o,i}^2\right)}}{\max(\boldsymbol{\alpha}_o) - \min(\boldsymbol{\alpha}_o)} \tag{6.4}$$

where $\boldsymbol{\alpha}_e$ are the POD coefficients estimated via Gappy POD, and $\boldsymbol{\alpha}_o$ are those estimated starting from the full-dimensional back-EMF curve. A suitable number of modes for the subsequent steps must be identified as a trade-off between the errors in signal reconstruction – that is a measure of the retained information – and

the error on POD coefficients – that is a measure of the accuracy of Gappy POD. The first POD coefficient is easily estimated, as the associated mode explains most of the variance of the signal; however, a single mode does not contain sufficient information to provide a good approximation of the signal, as observed through a high value of err_y. Increasing the number of modes, at first the error on signal reconstruction decreases, as the original curve can be approximated with higher precision. However, when the number of retained POD modes gets too large, the dispersion of the error increases, as misleading information starts to be included in the process. This behavior is visible in the last two boxplots of Figure 6.4 (a), corresponding to 10 and 11 POD modes. The error on POD coefficient estimation has a slightly different behavior, in that a low number of modes results in a small error, although often associated to a large dispersion. Indeed, in these conditions, Gappy POD is estimating the coefficients of the retained modes; the error on these coefficients, compared to the same quantities estimated starting from the whole signal, can be small in some cases. The dispersion lowers for a larger number of POD modes, and increases back when misleading information starts to be included in the reconstruction (a similar behavior to that of the error on signal reconstruction). A stable behavior is found, for the considered case study, for 7 to 9 POD modes.

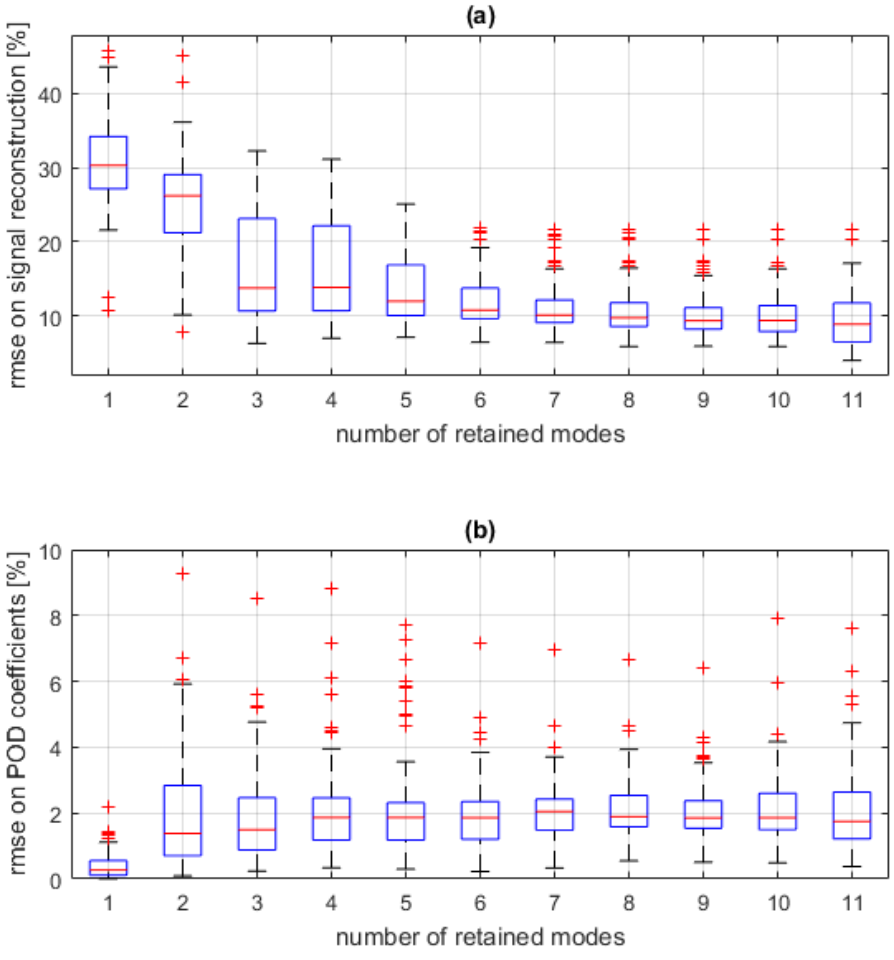

Figure 6.4: (a) Error on curve reconstruction via Gappy POD. (b) error on POD coefficients.

Figure 6.5 shows the U-matrix of the trained SOM in the case of $n_m = 9$ retained POD modes and $n_\omega = 30$ neurons, arranged in a 5-by-6 hexagonal topology. Figure 6.6 represents the component planes of the network. In this case study, the first input of the SOM is the angular coordinate θ_m, corresponding to the rotor angle of the BLDC motor. The SOM is not employed in this work as a proper clustering algorithm: after the training is complete, only the first component of the weight vector of each neuron is taken as an informative location where the signals are acquired and processed online. Since the initial sampling (Sections 2.1 and 6.2) included combination of different fault modes as well as single faults, the SOM does not identify completely distinct clusters, but rather a continuous set of partially overlapping ones.

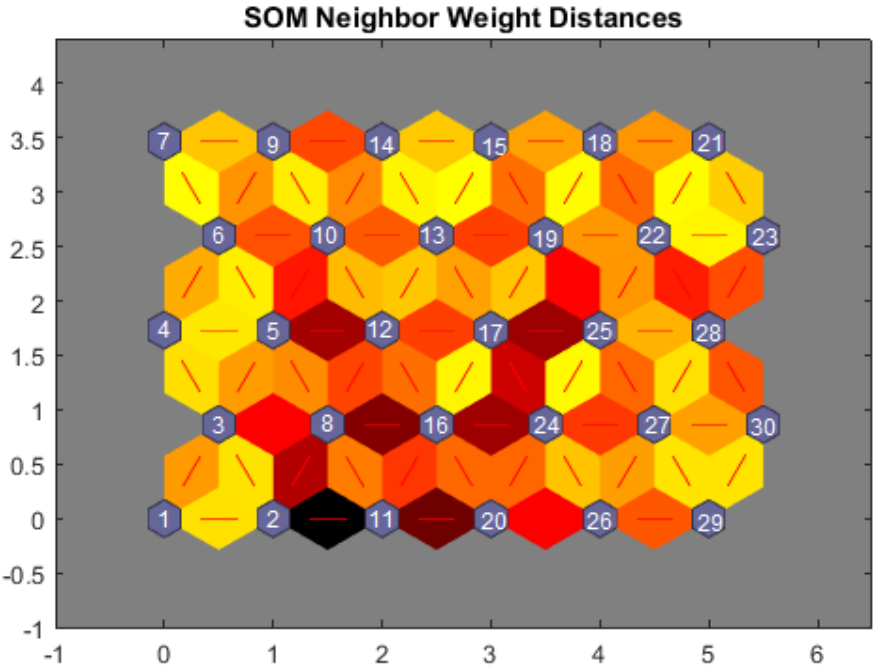

Figure 6.5: U-Matrix of the trained SOM for $\boldsymbol{n_m = 9}$. The neurons are labelled according to the coordinate of the associated sampling points.

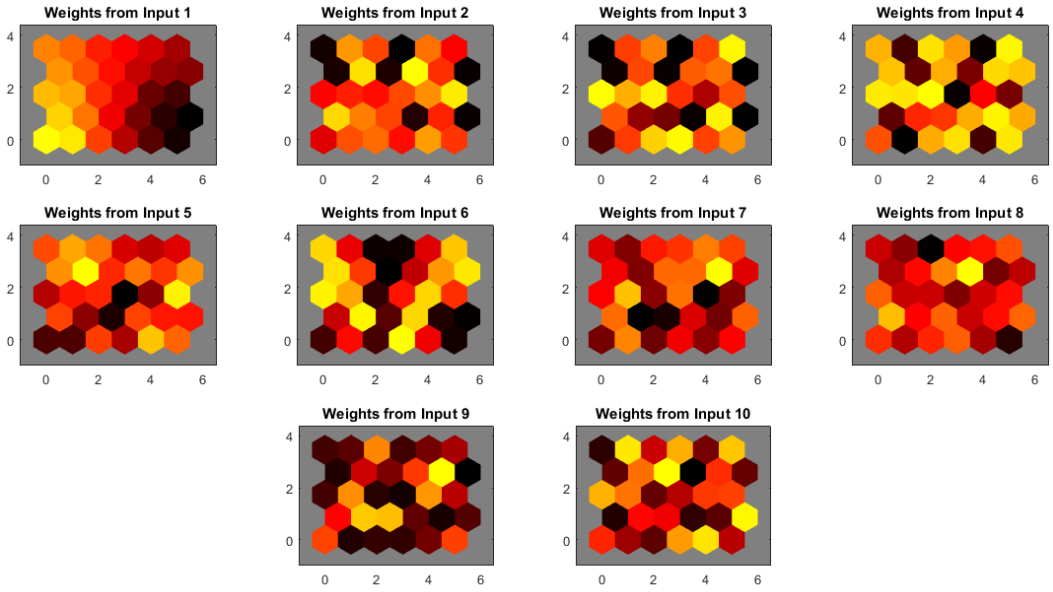

Figure 6.6: Component plane matrices of the trained SOM for $\boldsymbol{n_m = 9}$.

Figure 6.7 shows the reconstruction of a back-EMF coefficient curve of the second validation set (i.e. computed by the HF model) with an increasing number of POD modes. Increasing the number of modes, the curve is reconstructed more accurately, but the additional modes may include information that is misleading for the subsequent FDI.

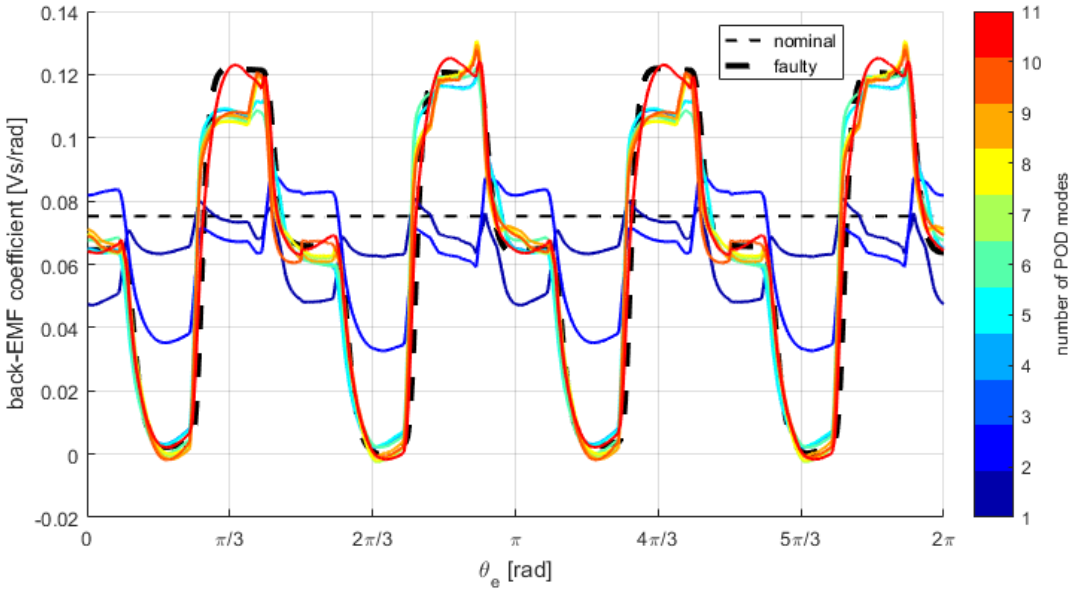

Figure 6.7: reconstruction of the back-EMF coefficient map with an increasing number of POD modes. A random fault combination of the validation set is considered, corresponding to $\boldsymbol{N_a = 0.640}$, $\boldsymbol{N_b = 0.055}$, $\boldsymbol{N_c = 0.972}$, $\boldsymbol{Z = 0.015}$, $\boldsymbol{\phi = -2.63°}$.

6.4 Fault Detection and Identification (FDI) through Multi-Layer Perceptron (MLP)

Fault Detection is performed, as per Section 2.3 by a Multi-Layer Perceptron (MLP). The neural network accepts in input the coefficients of the POD modes associated with the compressed signal and estimated by Gappy POD, and returns in output an estimate for the system health condition.

Two studies were performed to determine a suitable number of retained POD modes and a number of neurons for the hidden layer of the MLP. In both cases, the nets were trained on the training set described in Section 2.1, using the POD coefficients (determined as per Section 2.2) as the inputs, and the fault parameters as the targets. Both validation sets were employed to assess the performance of the networks: the first, computed by the same LF model as the training set, verifies that the training is adequately representative of the behavior of the system; the second, computed by the HF model, assesses the robustness of the FDI process against uncertainties in the models.

The first study employed a network with a fixed number of neurons and a number of inputs varying from 1 to 11 (i.e. from using only the coefficient of the first POD mode to the first 11 modes, corresponding to 99.9% of the original information). The networks have 5 output neurons, determined by the dimensionality of the output, and 15 neurons in the single hidden layer. Figure 6.8 summarizes the outcome of this study. In Figure 6.8 (a) and (b) the RMSE in fault detection are reported:

$$err_k = \sqrt{\frac{1}{n_k}\sum_{i=1}^{n_k}\left(k_{e,i}^2 - k_{o,i}^2\right)} \tag{6.5}$$

where $\boldsymbol{k} = [k_1, \dots, k_{n_k}]$ is the vector of fault parameters, the e subscript denotes the estimate performed resulting from FDI, and the o subscript denotes the target value. The error is not normalized, since the individual fault parameters are already normalized and allowed to range between 0 and 1. Predictably, the RMSE is lower for the first test set (Figure 6.8 (a)), in which the uncertainty associated to the physics-based models of the system has no effect. In this case, the test set is generated by the same model employed to compute the training set. The FDI error stabilizes in the order of 1% with a very small dispersion when considering 6 or more POD modes. As expected, retaining less than 5 modes leads to a large FDI error: the neural network is estimating 5 fault parameters independent from each other, and the problem becomes underconstrained if the dimensionality of the MLP input is 4 or less. The inclusion of up to 11 POD modes does not result in significant accuracy variations. The use of the second test set (Figure 6.8 (b)), computed by the High Fidelity model, results to a generally higher FDI error: here the discrepancy between the HF and LF models contributes to the overall accuracy of the fault estimation process. The FDI error in the range from 5% to 10% for 5 to 9 modes; 4 POD modes or less results in an underconstrained problem and a worse accuracy,

as in the previous case. Additionally, misleading information is included and the error increases when retaining 10 or more modes.

Figure 6.8 (c) shows the computational time in training for the MLP: being the number of neurons constant, training time is not affected significantly by the number of inputs. Figure 6.8 (d) reports the evaluation time, discriminating between the Gappy POD step and the MLP step: as expected, the computational time for Gappy POD increases with the number of retained modes, while the evaluation of the MLP is consistently between 5 and 10 milliseconds, being affected mainly by the number of neurons.

The second parametric study aims to determine a suitable number of neurons for the hidden layer of the network. The results are summarized in Figure 6.9. The number of POD modes is fixed at 9, as the previous studies assessed that this allows a good compromise between accuracy and computational time. Specifically, the study on the number of modes (Figure 6.8) showed that a good performance is obtained for 5 or 9 modes, while the preliminary assessment on the signal acquisition and compression (Figure 6.4, Section 6.3) highlights that the best representation of the system behavior is obtained for 7 to 9 modes.

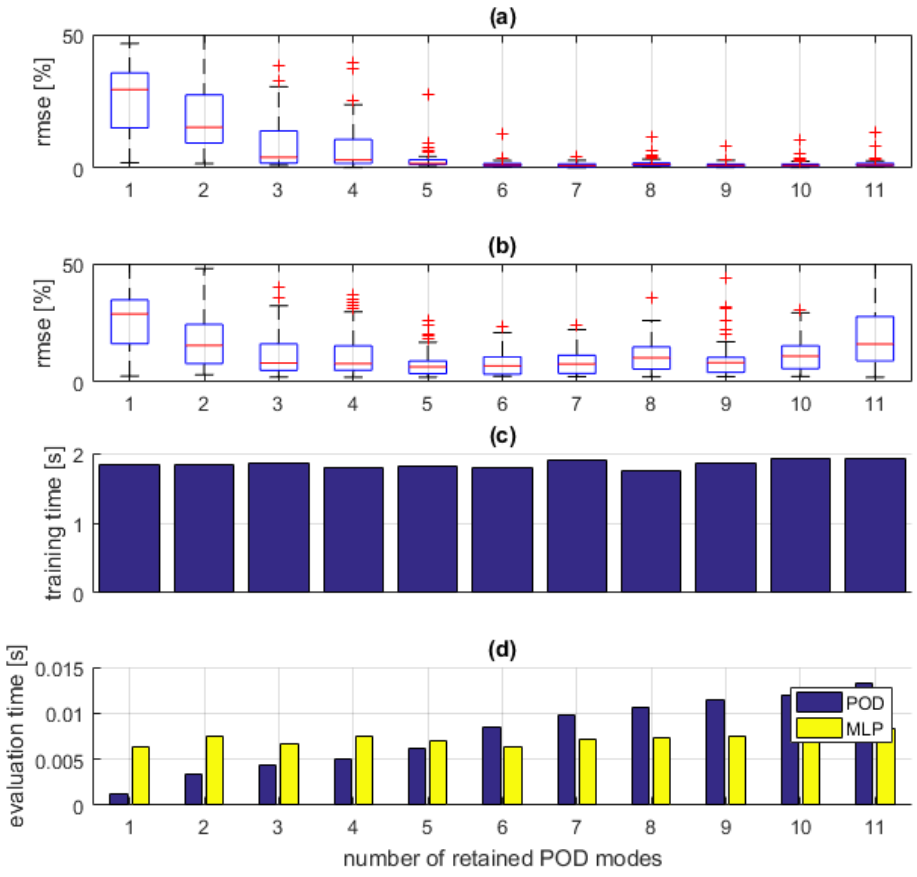

Figure 6.8: Study on the number of retained modes for FDI. (a) RMSE in fault identification, first validation set; (b) RMSE in fault identification, second validation set; (c) computational time in training; (d) computational time in evaluation.

Figure 6.9 (a) and (b) show the accuracy of the networks in terms of RMSE on the identified fault parameters. As in the previous case, the second test set (with the HF model) is associated to a larger, but still acceptable, error in identification. In both cases, a small number of neurons is not able to approximate correctly the behavior of the training data, and produces a large FDI error. The use of 8 or more neurons results in a consistently low average error, in the order of 1% for the LF test set, and usually below 10% for the HF test set. Increasing the hidden layer of the MLP does not lead to a significant overfitting up to 40 neurons, although the number and dispersion of the outliers starts to rise with more than 16 hidden layer neurons.

Figure 6.9 (c) and (d) report the computational time in training and in evaluation, respectively. An increasing number of neurons leads to a longer training, while the evaluation time is mostly unaffected: indeed, the upper limit for the number of neurons is quite low, and most of the evaluation time (in the order of milliseconds) is taken up by the overhead for calling the network. The dimensionality of the internal variables affects the global computational burden only marginally. Therefore, 14 neurons for the hidden layer are chosen as a good compromise between accuracy, overfitting and computational time.

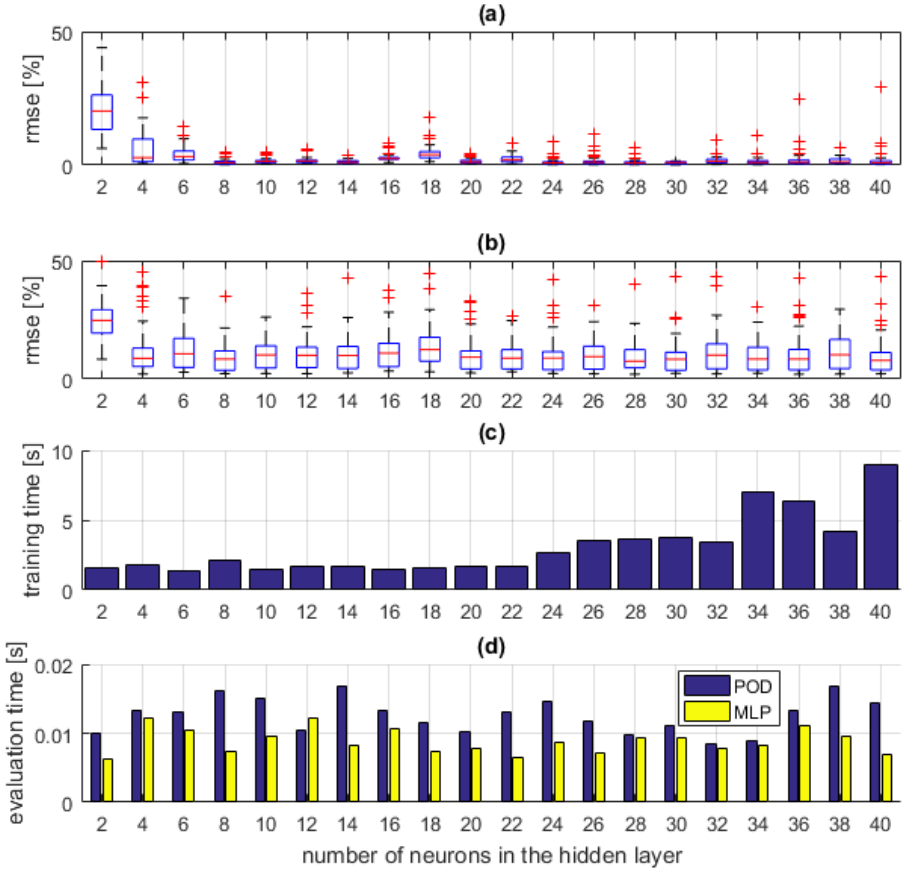

Figure 6.9: Study on the number of neurons in the hidden layer for FDI. (a) RMSE in fault identification, first validation set; (b) RMSE in fault identification, second validation set; (c) computational time in training; (d) computational time in evaluation.

The boxplot of Figure 6.10 shows the performance of the MLP with respect to the different considered failure modes. The short circuits on the three motor phases – encoded in k_1, k_2 and k_3 – are recovered with a median error lower than 1%; the dispersion is small, and only a few outliers have an error higher than 2%. As regards the eccentricity fault, represented by k_4 and k_5, the error associated with FDI has a slightly higher median, a smaller interquartile range and more numerous and spread out outliers, reaching values up to 7% in some cases. This behavior reflects the response of the physics-based models and the sensitivity of the considered features to the different fault modes. The effect of partial short circuit has a regular behavior and is relatively simple to model: this results in a stable and repeatable performance of the MLP. By contrast, the eccentricity fault produces a more irregular and nonlinear ripple on the back-EMF coefficient; this is more challenging to model with an MLP, and yields a larger number of outliers.

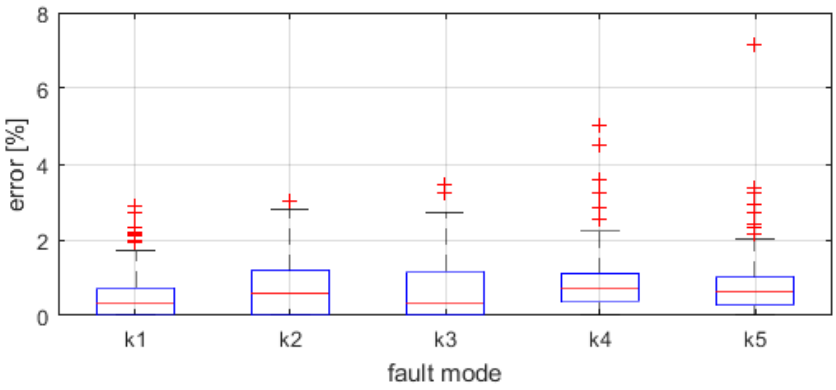

Figure 6.10: FDI error associated to each fault mode

6.5 Surrogate assessment function for RUL estimation

The Assessment Function ϕ_a described in Section 3.4 is replaced by a Support Vector Machine (SVM) to speed up online computations and to allow the real-time evaluation of the RUL estimation procedure described in Section 2.4.2. This strategy was initially proposed in [29, 30], while a parametric study on the SVM settings was discussed in [37].

The SVM is trained on the dataset described in Section 6.2; specifically, the fault combinations are employed in combination with the associated values of the full, model-based Assessment function. The validation set includes an additional 100 fault combinations and assessment function values, employed to determine the accuracy of the surrogate model. A parametric study was carried out to determine acceptable settings for the SVM training: most machine learning algorithm depend on a multitude of hyperparameters that affect the performance in training and evaluation; those parameters are strongly problem-dependent, and shall be calibrated for each specific application. For the SVM implementation discussed in this work, the most influential hyperparameter is usually the Kernel Scale k_s. This is a multiplier applied to the SVM input before feeding it to the kernel function; its value effectively determines a threshold to discriminate between the local variance of the function to approximate from the global shape of the training data. As a result,

the calibration of k_s provides a handle to tradeoff between the accuracy and generalization properties of the surrogate function.

Figure 6.11 shows the results of the parametric study in terms of accuracy. The 100 health conditions of the validation set include $h_t = 29$ "healthy" training points and $f_t = 71$ "faulty" training points. For each tested SVM, the global accuracy was evaluated as the ratio between correct classifications and the number of test cases:

$$A = 1 - \frac{h_f + f_f}{h_t + f_t} \tag{6.6}$$

where h_f is the number of missed detections (i.e. the full Assessment Function returned "faulty" while the surrogate returned "healthy"), and f_f is the number of false positives (i.e. the full Assessment Function returned "healthy" while the surrogate returned "faulty"). In addition, the ratio between missed detection and true positives was determined, that is equivalent to the fraction of misidentified "faulty" conditions:

$$MD = \frac{h_f}{f_t} \tag{6.7}$$

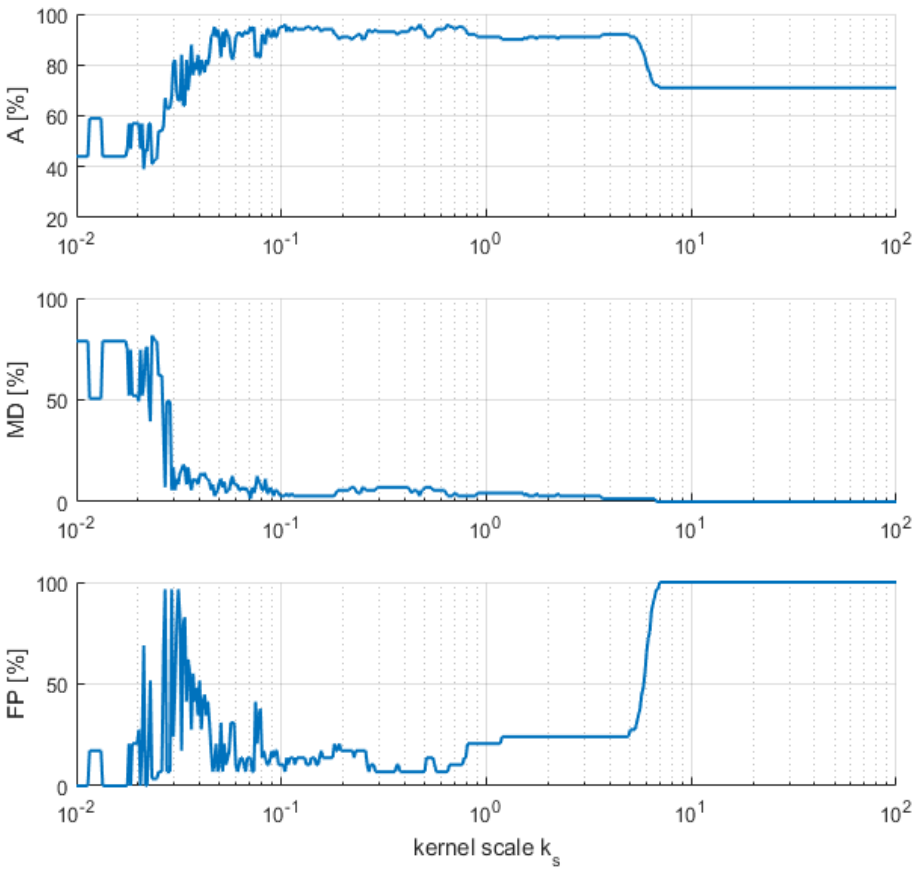

Figure 6.11: (top) accuracy of the SVM as a function of the kernel scale; (middle) fraction of missed detections; (bottom) fraction of false positives.

Similarly, the ratio between false positives and true negatives was computed:

$$FP = \frac{f_f}{h_t} \tag{6.8}$$

that is, the fraction of misidentified "healthy" conditions. Referring to Figure 6.11, a relatively stable performance emerges for values of k_s in a range from 10^{-1} to $5 \cdot 10^{1}$. This outcome suggests a good robustness of the SVM with respect to the particular choice of the training data and settings. The best performance is achieved for $k_s = 0.4735$, with an overall accuracy $A = 96\%$, two false positives (i.e. $FP = 6.90\%$) and two missed detections (i.e. $MD = 2.82\%$).

Figure 6.12 reports the computational time required by the SVMs in evaluation, for varying k_s. In most cases, the computational time is stable between 0.8 and 1.2 milliseconds; a higher dispersion is found for the k_s values where the accuracy has a large gradient (i.e. at the edges of the higher accuracy zone, for $k_s \approx 10^{-1}$ and for $k_s \approx 5 \cdot 10^{0}$). The improvement with respect to the full Assessment Function is of more than 4 orders of magnitude: the model-based approach required a computational time of about 40s (see Section 3.4). As a result, this surrogate modelling approach can be exploited to achieve real-time RUL estimation without requiring prohibitively high performing hardware resources.

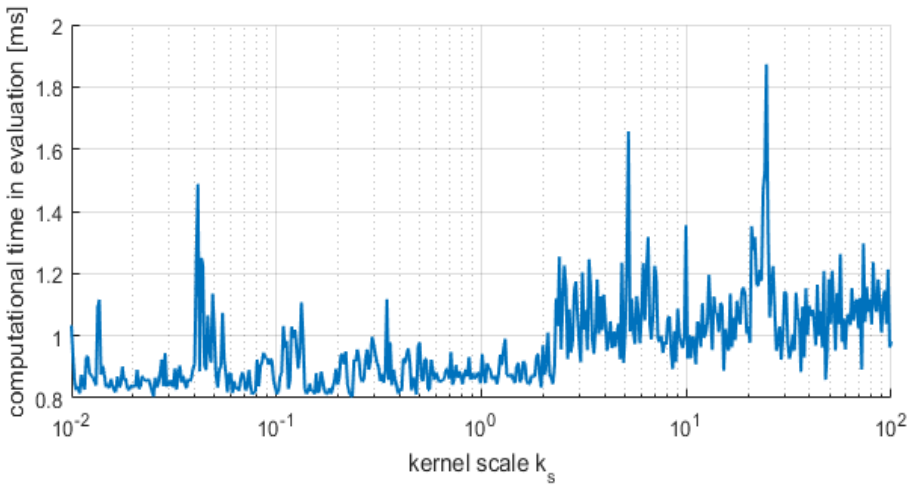

Figure 6.12: Computational time in evaluation for the SVM as a function of the kernel scale.

Chapter 7

Results

A validation set was collected to demonstrate the performance of the proposed methodology for failure prognosis. The dataset contains a total of 100 simulated runs to failure of the considered actuator. The time-history of the health condition is computed by integrating a model in the form:

$$\dot{\boldsymbol{k}}(t) = \mathrm{A}(t)\boldsymbol{k}(t) + \mathrm{B}\boldsymbol{u}(t) + \mathcal{N}(0,\sigma) \tag{7.1}$$

where $\mathrm{A}(t)$ is a state matrix, the term $\mathrm{B}\boldsymbol{u}(t)$ simulates the effect of operating and environmental conditions on damage growth rate, and $\mathcal{N}(0,\sigma)$ is an independent identically distributed Gaussian noise to represent the uncertainty inherently associated to fault propagation as a result of manufacturing defect and variability of operating conditions experienced by the equipment.

Within the validation set, 50 simulations are computed with a constant $\mathrm{A}(t)$, while in the others the state matrix is piece-wise constant in time. The variations of $\mathrm{A}(t)$ happen at random times and with random magnitude; however, the fault propagation rate is forced to be non-decreasing in time, as this is the most critical scenario possibly faced by a RUL estimation algorithm.

An example of propagation of faults, from a nominal condition to a complete failure, is shown in Figure 7.1. The interaction of the state-space model with the uncertainty term produces large deviations from the exponential growth of a purely linear model. The black vertical line represents the failure: a threshold is not represented as it is set by the requirements on the performance parameters, and not on the individual faults.

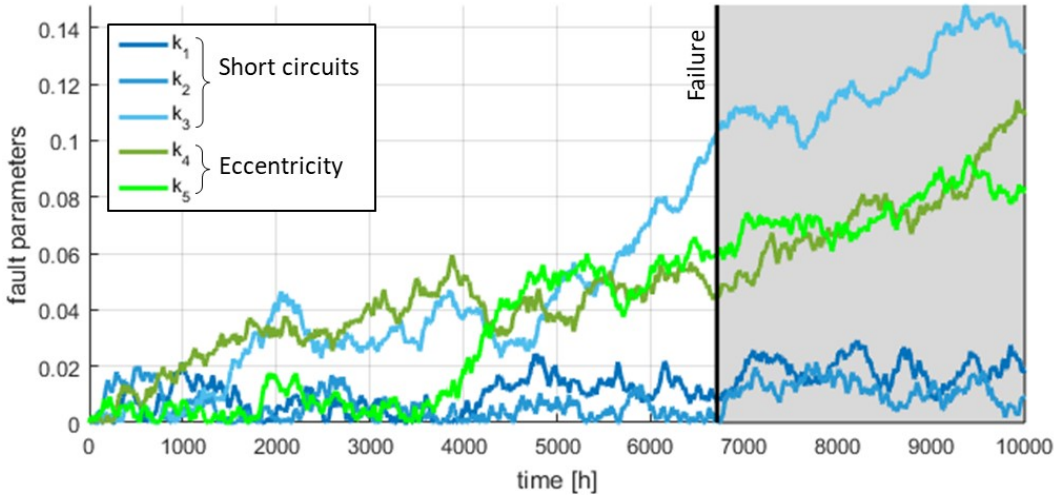

Figure 7.1: Example of simulation of a run to failure.

For each timestep of each simulation, an additive error is added to the instantaneous health condition of the system, sampled from the error distribution associated with the real-time FDI process. This error distribution is shown in Figure 7.2, highlighting the uncertainty associated with each component of the fault vector.

The uncertainty in FDI arises from two main sources: the additive white noise acting on the output of the dynamical model of the EMA (simulating the error in measurements) and the error associated to the evaluation of the MLP model.

The partial short circuits are characterized by smaller errors, while larger ones are recorded for rotor eccentricity. This behaviour mirrors the sensitivity of the specific monitored signal to each fault mode. An alternative solution would have evaluated the HF model to compute the response of the actuator for each timestep, and then ran the online FDI process to identify the faults; however, this way the creation of the validation set would have required an impractical amount of time, since the HF model would have been evaluated a large number of times. Additionally, both the online FDI and RUL estimation algorithms are completely deterministic: for a given run of the HF model, the RUL estimate is completely determined; its uncertainty is uniquely associated with the propagation of the disturbances in the system output, to the FDI, to the failure prognosis. As a result, an uncertainty distribution associated to the RUL estimate could not have been computed.

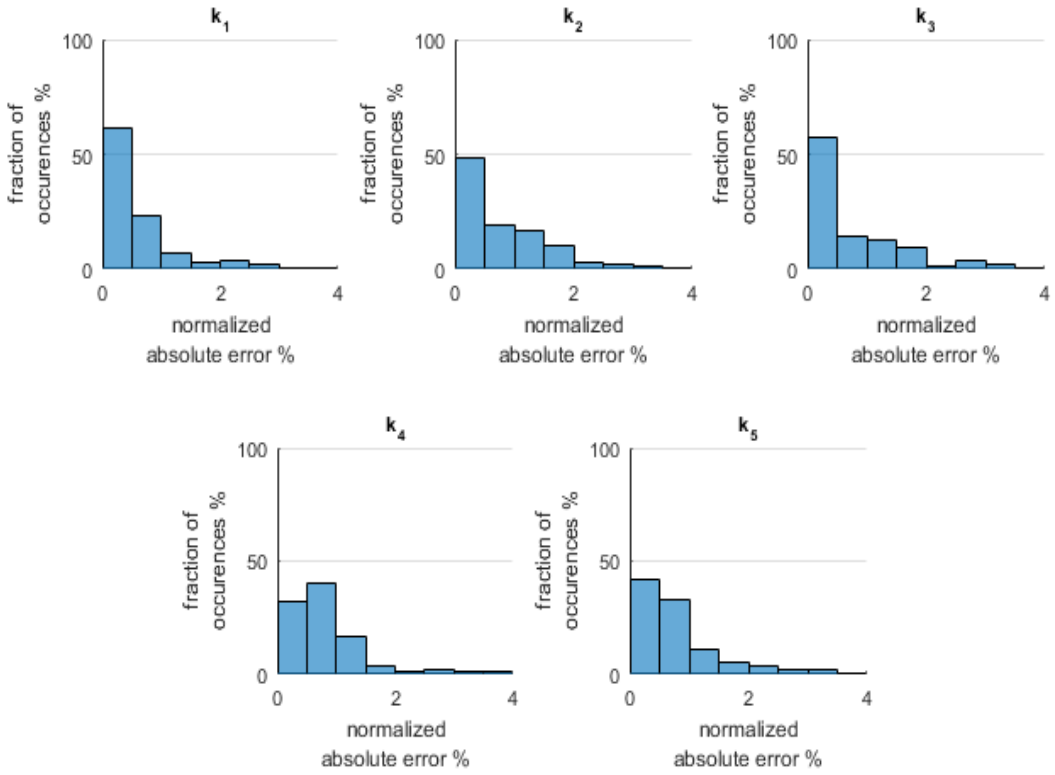

Figure 7.2: Normalized FDI error associated with each fault parameter.

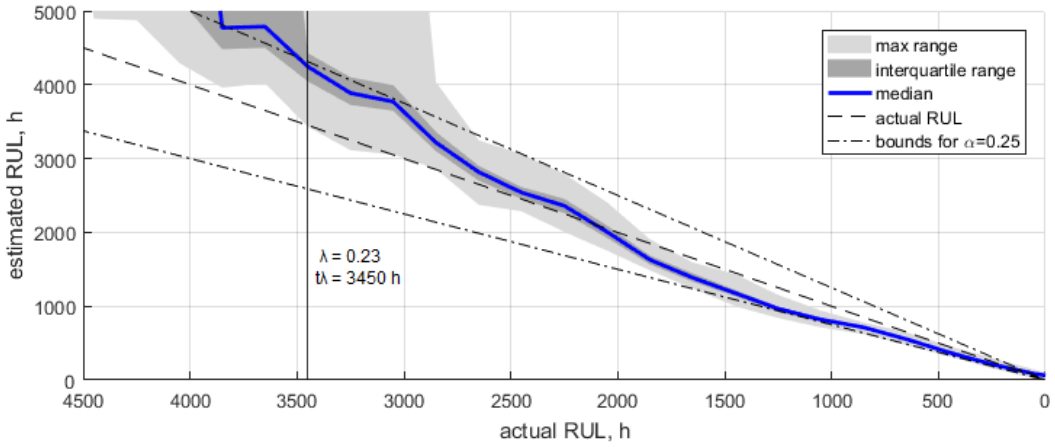

Figure 7.3 Comparison between actual and estimated RUL, with a time-invariant damage propagation model. The RUL is estimated every 200 hours, considering the FDI observations of the last 200 operating hours.

Figure 7.3 shows the evolution of the RUL estimate when the damage propagation model is constant in time, that is $dA(t)/dt = 0 \quad \forall t$. At the beginning, the algorithm overestimates the system RUL: indeed, the state and control matrices of Equation 2.26 are initialized to zero, and the estimated RUL is infinite.

The early estimates have a very large dispersion, as both the model of damage propagation and the measurement of the initial condition are affected by uncertainty. When the health of the system is close to nominal, a large contribution to the fault estimate produced by the FDI algorithm is due to the additive noise on damage propagation $\mathcal{N}(0, \sigma)$. Additionally, the FDI error is large compared to the actual faults.

The assimilation of additional observations of the fault propagation history allows to filter out errors inevitably affecting FDI, and to obtain less disperse and more robust predictions of the system life. As time runs, approaching to the right of Figure 7.3, the estimate of the remaining useful life improves in accuracy thanks to the efficient assimilation from real-time measurements. At first, only a rough value is available, but the expected failure is still far ahead, and a precise information is not needed yet. As the failure approaches, the estimate becomes more accurate, when it is required to plan corrective actions.

The $\alpha - \lambda$ metric [153] is employed to assess the RUL prediction algorithm by comparing the estimates to the specified α-bounds:

$$\alpha - \lambda \text{ Accuracy} = \begin{cases} 1, & \text{if} \quad \pi\left[\frac{RUL_e}{RUL_a}\right]_{-\alpha}^{+\alpha} \geq \beta \\ 0, & \text{otherwise} \end{cases} \tag{7.1}$$

where RUL_e is the RUL estimate, RUL_a is the actual RUL, and β is a minimum acceptable probability. The dashed-dotted line of Figure 7.3 represents the tolerance bounds around the actual RUL for $\alpha = 0.25$. The β-criterion is satisfied for $\beta = 0.5$. at $\lambda = 0.23$, that is for an actual RUL of 3450h. For many applications, this is a more than suitable advance to schedule corrective interventions in an effective manner.

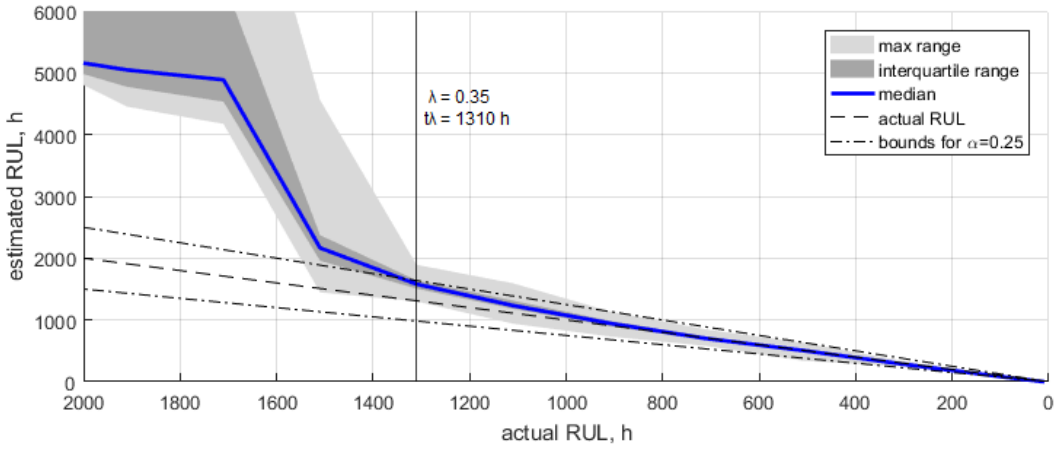

Figure 7.4: Comparison between actual and estimated RUL with a time-varying model. The reference damage propagation rate increases at $\boldsymbol{RUL \approx 1800h}$

Figure 7.4 shows the evolution of the RUL estimate for a test case where the damage propagates at a variable (increasing) rate, that is $d\mathrm{A}(t)/dt > 0 \quad \exists t$.

At the beginning, the actual RUL of the system is overestimated, since the current damage propagation rate is predicted to the complete failure. When the RUL is approximately 1800h, the fault growth rate rises: this event may simulate either an uncertainty in the model (e.g. a given component starts wearing faster after a critical threshold is reached, such as a hardened external layer is completely worn out) or an unpredictable change in the external conditions that affects the wear rate (such as, the lubricant of a sealed transmission may get contaminated and lose its properties).

The RUL estimation algorithm reacts to this change by adapting its prediction to the new observed time history of health conditions. As shown in the second part of Figure 7.4, shortly after the change in damage growth rate the dispersion on the estimation shrinks and the median RUL estimate moves closer to the actual one. The β-criterion for a minim probability $\beta = 0.5$ and a tolerance band defined by a relative uncertainty $\alpha = 0.25$ is satisfied at $RUL = 1310\mathrm{h}$, equivalent to $\lambda = 0.35$. This happens soon after the fault propagation rate has changed, with a small delay required to assimilate sufficient information and update the prediction. The result demonstrates that the proposed failure prognosis algorithm is able to adapt in real-time to variations in the behaviour of the monitored system.

Table 7.1 summarizes the error associated with the RUL prediction in the two test cases of Figure 7.3 and Figure 7.4. The early estimates are characterized by a large uncertainty, particularly in the case of a time-varying damage growth rate. As the actual RUL decreases, the algorithm is able to get the prediction within $\pm 20\%$ of the exact value. This error is coherent with most approaches available in literature, and is mainly due to the inherent variability associated to the fault growth.

The advantage introduced with the proposed approach is that the algorithm efficiently learns in real-time to compute predictions of the Remaining Useful Life, without relying on *a priori* physics-based knowledge about the rate of propagation of damages.

The whole online process runs in the Matlab environment, on a laptop PC with an i7-6500U processor and 8GB of memory. The computational time required for each FDI and RUL estimate ranges between 0.1s and 1s, which makes it suitable for on-board, nearly real-time execution.

Table 7.0.1: Median relative error in RUL estimate

actual RUL [h]	**time-invariant fault growth**	**time-varying fault growth**
3000	23.90%	243,64%
1500	-18,85%	50,11%
500	-17,04%	18,95%

Chapter 8

Conclusions

A comprehensive computational framework for fault detection and failure prognosis of dynamical assemblies was presented and discussed. The proposed methodology combines machine learning and reduced order modelling with physics-based numerical models to obtain compressed yet accurate representations of the behaviour of the system.

The entire process can be executed in real-time, dealing with the hardware constraint that are characteristic of on-board computations. Nevertheless, the proposed algorithm can learn a model of the damage evolution by assimilating online the observation of the instantaneous health status of the monitored system. As a result, the overall accuracy of the RUL prediction process is comparable to most approaches available in literature. The strength of the proposed methodology is its ability to predict the RUL in real-time, without requiring any prior knowledge about the physics of the phenomena affecting the actual propagation of faults, while at the same time retaining a suitable accuracy.

An experimental validation of the physical models employed as a case study was provided, limited for now to the nominal conditions. In addition, novel sensor technologies were considered and assessed experimentally; these will provide additional, useful information for the health monitoring algorithms without prohibitive increases in weight, complexity and costs of the aerospace vehicle.

Future developments on this research will include the experimental validation of the numerical models in faulty conditions, the integration of the additional information provided by optical sensors in the prognostic process, and the test of the entire real-time process in-flight. In addition, the potentiality to include a real-time estimation of the RUL prediction uncertainty is currently under investigation.

Appendix A

Experimental validation of gearbox efficiency models

The Simulink models of the electromechanical actuator described in Section 3 include a computation of the mechanical transmission efficiency according to Berri et al. [150]. The validation of these simplified formulations against higher fidelity models is available in literature [149]; the experimental data gathered in this work allowed an experimental verification of their results: a preliminary discussion is presented in Section 5.2.2 that compares the measured efficiency of the EMA test bench transmission to the values predicted by the models. This appendix provides additional data confirming the previous results.

A.1 Experimental setup

A simple, dedicated test set was developed for the purpose of validating the gearbox efficiency model. A schematic layout of the experimental setup is shown in Figure A.1 (a), while Figure A.1 (b) is a picture of the test bench.

The same gearmotor employed for the characterization of the main EMA test bench subassemblies, as described in Section 5, was used to drive the tested gearbox at a constant speed, while measuring the required torque with a loadcell. A known load torque was provided by a pair of calibrated weights hanging to a winch

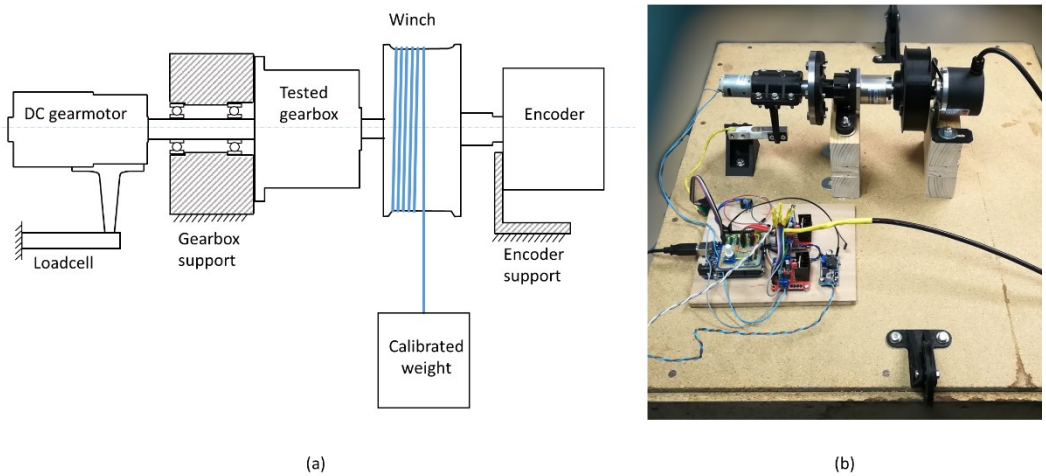

Figure A.1: (a) layout of the experimental setup for the gearbox characterization; (b) picture of the dedicated test bench

connected to the output of the gearbox; the load could be varied by changing the calibrated weights, while the speed was controlled through the voltage fed to the motor. A high resolution incremental encoder provided position and speed measurements on the output shaft of the gearbox.

Three off-the-shelf planetary gearboxes were tested, whose characteristics are summarized in Table A.1. All these gearings differ from each other by the number of stages, while sharing the same number of teeth and modulus. Specifically, the sun gear has 17 teeth, the ring gear 46, and the three satellites of each stage 14.

Table A.1: Main characteristics of the three tested gearboxes

	Gearbox 1	**Gearbox 2**	**Gearbox 3**
Gear ratio	3.71:1	13.73:1	50.89:1
Number of stages	1	2	3

The gearbox was driven at an approximately constant speed, to reduce the inertial components of the motor torque. The speed was changed by steps in order to acquire different data points. Each measurement consisted in two time series, containing the log of the encoder position and loadcell reading, respectively. The user torque is known, since it results from the calibrated weight times the radius of the winch.

The input shaft bearing of the tested gearbox supports the gearmotor. The loadcell constrains the gearmotor case with respect to the rotation about the rotor axis. As a result, the motor torque T_m can be computed from the loadcell reading F_{LC} as:

$$T_m = bF_{LC} \tag{A.1}$$

where b is the distance from the motor axis to the loadcell.

A.2 Results

The three gearboxes were tested with different loads and speeds. The use of a winch as a user allowed to explore both the aiding load (i.e. while lowering the calibrated weights) and opposing load (i.e. while lifting the weights) operating conditions of the mechanical transmission. As a results, the maps of Figure A.2 to Figure A.4 were obtained.

The measured motor torque T_m is plotted against user torque T_l for the opposing load (blue data points) and aiding load condition (red data points). The black dashed line represents the ideal case of unit efficiency, where the user torque is equal to the motor torque times the gear ratio τ. The measurements are offset from this ideal condition by a constant quantity, that is the friction caused by pre-loads, plus a contribution linear with respect to the load torque. This latter contribution is a measure of the gearbox efficiency.

The blue solid line fits the opposing load data points. Its slope dT_m/dT_l permits to estimate the gearbox efficiency in this condition η_D, as:

$$\eta_D = \left(\tau \frac{dT_m}{dT_l}\right)^{-1} \tag{A.2}$$

since in this condition the user torque can be written as $T_l = (T_m - F)\tau\eta_D$. The model described by [150] offers an estimation of the aiding load efficiency η_I as a function of the opposing load one η_D and the gear ratio:

$$\eta_I = \frac{2\eta_D\tau - \tau + 1}{\eta_D\tau - \eta_D + 2} \tag{A.3}$$

The red solid line of Figure A.2 to Figure A.4 describes the relationship between motor and load torques in opposing load conditions:

$$T_m = \frac{T_l}{\tau}\eta_I - F \tag{A.4}$$

The model is able to predict the observations with good accuracy; the measured motor torque in aiding load condition matches the model within the measurement uncertainty. This is particularly visible for the 2-stage gearbox (Figure A.3). For the 3-stage transmission (Figure A.4), the measurement uncertainty is higher because of the difficulty in measuring very low motor torques with high precision. In the single stage gearbox (Figure A.2), the measurement accuracy is limited by the high speed of the user shaft, which completes the stroke of the calibrated weights in a short time and allows acquiring fewer data points.

Table A.2 summarizes the efficiencies of the three tested gearboxes. As expected, the efficiency in opposing load is close to the single-stage efficiency, raised to the power of the number of stages. The aiding load efficiency decreases faster, as the number of stages increases and the transmission approaches to an irreversible condition. The error between the measured aiding load efficiency and that estimated by the model is below 5% for all the tested gearboxes.

Table A.2: Observed efficiencies of the three tested gearboxes

	Opposing load efficiency - experimental	**Aiding load efficiency - model**	**Aiding load efficiency - experimental**	**Error**
Gearbox 1	91.31%	90.85%	92.63%	-1.92%
Gearbox 2	79.16%	74.58%	72.85%	2.37%
Gearbox 3	68.76%	55.35%	58.24%	-4.95%

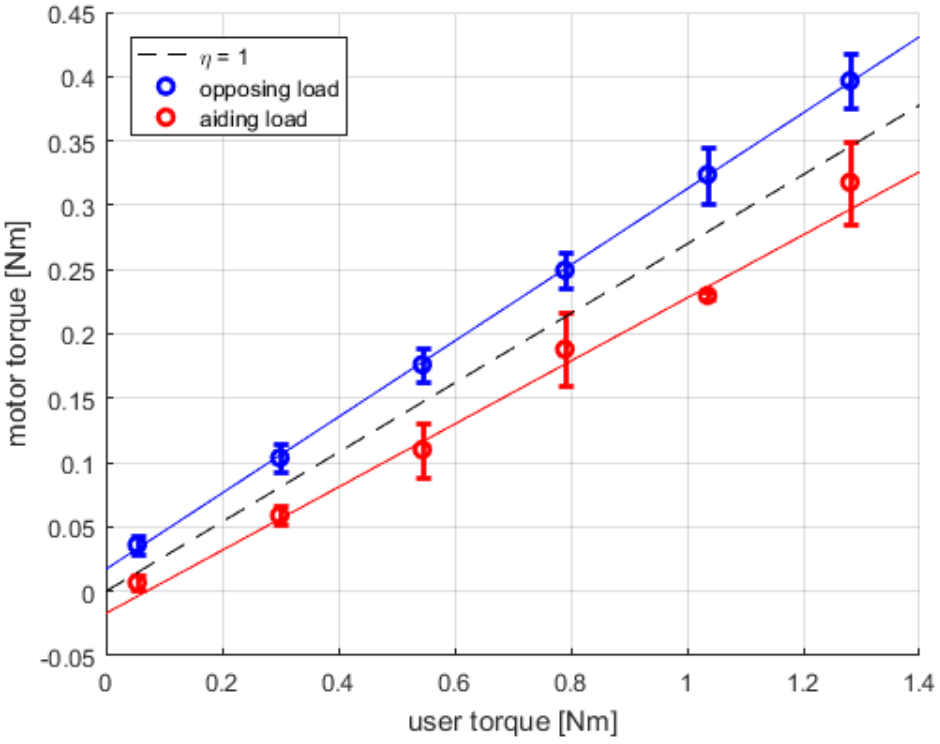

Figure A.2: Measured and predicted efficiency maps of the Gearbox 1 (3.71:1 gear ratio).

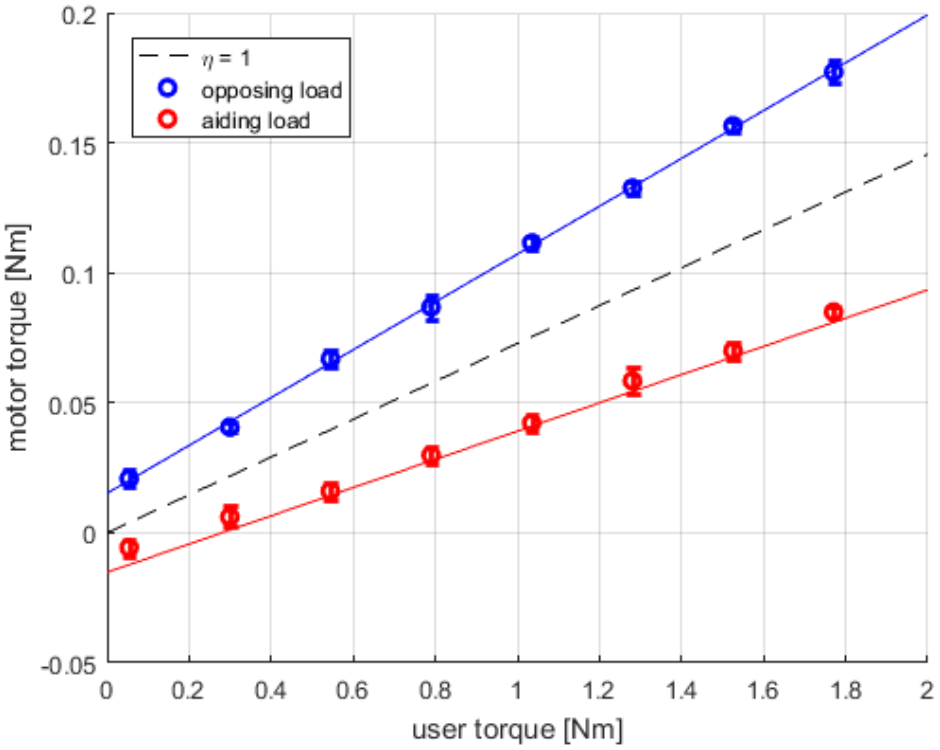

Figure A.3: Measured and predicted efficiency maps of the Gearbox 2 (13.73:1 gear ratio).

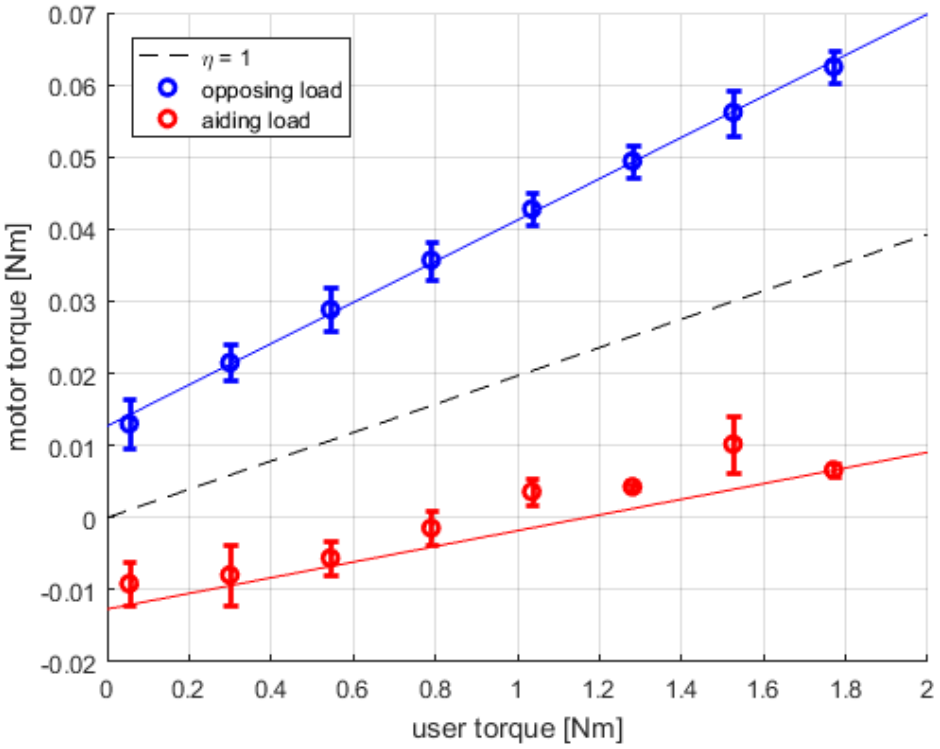

Figure A.4: Measured and predicted efficiency maps of the Gearbox 3 (50.89:1 gear ratio).

References

[1] Reliability data handbook. Universal model for reliability prediction of electronics components, PCBs and equipment, BSI British Standards, 2004.

[2] J. C. Chesley, Handbook of Reliability Prediction Procedures for Mechanical Equipment, USA Navy, Naval Surface Warfare Center, NSWC Carderock Division, 2011.

[3] R. Venkataraman, P. Seiler, M. Lukatsi and B. Vanek, “Reliability assessment of actuator architectures for unmanned aircraft,” *Journal of Aircraft,* vol. 54, no. 3, pp. 955-966, 2017.

[4] D. C. Garmendia, I. Chakraborty and D. N. Mavris, “Methods for evaluationg electrically actuated hybrid wingbody control surface layouts,” *Journal of Aircraft 52,* vol. 52, no. 6, pp. 1780-1790, 2015.

[5] X. Cui, S.Wang, Q. Li and J. Shi, “Reliability assessment of aircraft power and actuation system based on three-axis control,” in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, South Korea, 2015.

[6] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel and S. Poll, "Ensemble of optimized echo state networks for remaining useful life prediction," *Neurocomputing,* vol. 281, pp. 121-138, 2018.

[7] P. Polverino and C. Pianese, “Model-based prognostic algorithm for online RUL estimation of PEMFCs,” in *3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, Barcelona, Spain, 2016.

[8] R. Jihin, D. Söffker and N. Beganovic, “Integrated prognostic model for RUL estimation using threshold optimization,” *Structural Health Monitoring 2017: Real-Time Material State Awareness and Data-Driven Safety Assurance,* 2017.

[9] K. Pipe, “Practical prognostics for condition based maintenance,” in *International Conference on Prognostics and Health Management, IEEE*, Denver, CO, USA, 2008.

[10] Z. Lv, J. Wang, G. Zhang and H. Jiayang, “Prognostics health management of condition-based maintenance for aircraft engine systems,” in *IEEE Conference on Prognostics and Health Management (PHM), IEEE*, Austin, TX, USA, 2015.

[11] N. Zerhouni and K. Medjaher, “Health assessment, prognostics and remaining useful life - part A,” in *From Prognostics and Health Systems Management to Predictive Maintenance*, John Wiley & Sons, Inc., 2016, pp. 67-107.

[12] M. D. Sudolsky, “IVHM solutions using commercially-available aircraft condition monitoring systems,” in *IEEE Aerospace Conference, IEEE*, Big Sky, MT, USA, 2007.

[13] J. Dunsdon and M. Harrington, “The application of open system architecture for condition based maintenance to complete IVHM,” in *IEEE Aerospace Conference*, Big Sky, MT, USA, 2008.

[14] I. K. Jennions, Integrated Vehicle Health Management Perspectives on an Emerging Field, SAE International, 2011.

[15] A. Ladj, F. B. S. Tayeb and C. Varnier, “An integrated prognostic based hybrid genetic-immune algorithm for scheduling jobs and predictive maintenance,” in *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, 2016.

[16] J. Yan, “Maintenance planning and scheduling,” in *Machinery Prognostics and Prognosis Oriented Maintenance Management*, John Wiley & Sons, 2014, pp. 183-239.

[17] T. Tinga and R. Loendersloot, “Physical Model-Based Prognostics and Health Monitoring to Enable Predictive Maintenance,” in *Predictive Maintenance in Dynamic Systems*, Springer, 2019, pp. 313-353.

[18] K. O. Kolcio, “Model-Based Fault Detection and Isolation System for Increased Autonomy,” in *Integrated System Health Management for Space Systems - AIAA Space Forum (AIAA 2016-5225)*, Long Beach, CA, USA, 2016.

[19] D. Henry, J. Cieslak, J. Z. Torres, P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaard, H. Strauch, A. M. Giordano, M. D. Stefano,

C. Ott, M. Reiner, J. Jaworsk, E. Papadopoulos, G. Visentin, F. Ankersen and J. G. Fernandez, “Model-based fault diagnosis and tolerant control: the esa's e.deorbit mission,” in *18th European Control Conference (ECC)*, 2019.

[20] M. Battipede, M. D. L. Dalla Vedova, P. Maggiore and S. Romeo, “Model based analysis of precursors of electromechanical servomechanisms failures using an artificial neural network,” in *AIAA Modeling and Simulation Technologies Conference (AIAA 2015-2035)*, Kissimmee, FL, USA, 2015.

[21] T. Engelberth, D. Krawczyk and A. Verl, “Model-based method for condition monitoring and diagnosis of compressors,” in *51st CIRP Conference on Manufacturing Systems*, 2018.

[22] C. Shi, S. Wang, X. Wang and J. Wang, “Remaining useful life prediction based on particle filtering for high-speed pump in load sensing electro-hydrostatic actuator,” in *CSAA/IET International Conference on Aircraft Utility Systems*, Guiyang, China, 2018.

[23] B. Huang, Y. Di, C. Jin and J. Lee, “Review of data-driven prognostics and health management techniques: Lessions learned from phm data challenge competitions,” in *Machine Failure Prevention Technology*, Virginia Beach, VA, USA, 2017.

[24] O. Bektas, J. A. Jones, S. Sankararaman, I. Roychoudhury and K. Goebel, “A neural network framework for similarity-based prognostics,” *MethodsX,* vol. 6, pp. 383-390, 2019.

[25] P. Qian, X. Ma, D. Zhang and J. Wang, “Data-driven condition monitoring approaches to improving power output of wind turbines,” *IEEE Transactions on Industrial Electronics,* vol. 66, no. 8, pp. 6012-6020, 2019.

[26] M. D. L. Dalla Vedova, D. De Fano and P. Maggiore, “Neural network design for incipient failure detection on aircraft EM actuator,” *International Journal of Mechanics and Control,* vol. 17, no. 1, pp. 77-83, 2016.

[27] Y. Liu, Y. Chang, S. Liu and S. Chen, “Data-driven prognostics of remaining useful life for milling machine cutting tools,” in *IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2019.

[28] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang and R. X. Gao, “Deep learning and its applications to machine health monitoring,” *Mechanical Systems and Signal Processing ,* vol. 115, pp. 213-237, 2019.

[29] P. C. Berri, M. D. Dalla Vedova and L. Mainini, “Real-time Fault Detection and Prognostics for Aircraft Actuation Systems,” in *AIAA SciTech Forum (AIAA 2019-2210)*, San Diego, CA, USA, 2019.

[30] P. Berri, M. Dalla Vedova and L. Mainini, "Computational framework for real-time diagnostics and prognostics of aircraft actuation systems," *Computers in Industry,* vol. 132, p. 103523, 2021.

[31] P. Freeman, R. Pandita, N. Srivastava and G. J. Balas, “Model-based and data-driven fault detection performance for a small UAV,” *IEEE/ASME Transactions on Mechatronics,* vol. 18, no. 4, pp. 1300-1309, 2013.

[32] R. Venkataraman and P. J. Seiler, “Model-based detection and isolation of rudder faults for a small UAS,” in *AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, (AIAA 2015-0857)*, Kissimmee, FL, USA, 2015.

[33] F. Meng, T. Meyer, P. Thomas and J. Wenske, “Observer design and optimization for model-based condition monitoring of the wind turbine rotor blades using genetic algorithm,” in *Journal of Physics: Conference Series*, 2018.

[34] W. Booyse, D. N. Wilke and S. Heyns, “Deep digital twins for detection, diagnostics and prognostics,” *Mechanical Systems and Signal Processing,* vol. 140, p. 106612, 2020.

[35] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess and B. Wu, Intelligent Fault Diagnosis and Prognosis for Engineering Systems, John Wiley & Sons, Inc, 2006.

[36] R. Isermann, Fault-Diagnosis Applications, Springer Berlin Heidelberg, 2011.

[37] P. C. Berri, M. D. Dalla Vedova, P. Maggiore and G. Quattrocchi, “Model-based strategy and surrogate function for health condition assessment of actuation devices,” in *10th EASN Virtual International Conference on Innovation in Aviation & Space to the Satisfaction of the European Citizens*, 2020.

[38] L. Yongxiang, S. Jianming, W. Gong and L. Xiaodong, “A data-driven prognostics approach for rul based on principle component and instance learning,” in *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Ottawa, ON, Canada, 2016.

[39] S. Al-Dahidi, F. Di Maio, P. Baraldi and E. Zio, “Remaining useful life estimation in heterogeneous fleets working under variable operating conditions,” *Reliability Engineering & System Safety,* vol. 156, pp. 109-124, 2016.

[40] A. Mornacchi, G. Vachtsevanos and G. Jacazio, “Prognostics and health management of an electro-hydraulic servo actuator,” in *Annual conference of the Prognostics and Health Management Society*, San Diego, CA, USA, 2015.

[41] X. Li, X. Yang, Y. Yang, I. Bennett and D. Mba, “A novel diagnostic and prognostic framework for incipient fault detection and remaining service life prediction with application to industrial rotating machines,” *Applied Soft Computing,* vol. 82, p. 105564, 2019.

[42] L. Mainini, «Structural assessment and sensor placement strategy for self-aware aerospace vehicles,» *Structural Health Monitoring 2017: Real-Time Material State Awareness and Data-Driven Safety Assurance,* vol. 1, pp. 1586-1594, 2017.

[43] P. C. Berri, M. D. Dalla Vedova and L. Mainini, “Learning for predictions: Real-time reliability assessment of aerospace systems,” in *AIAA SciTech Forum*, 2021 (In press).

[44] P. Baraldi, F. Di Maio, P. Turati and E. Zio, "A modified Auto Associative Kernel Regression method for robust signal reconstruction in nuclear power plant components," in *European Safety and Reliability Conference (ESREL)*, 2014.

[45] P. Baraldi, E. Zio, G. Gola, D. Roverso and M. Hoffmann., "A procedure for the reconstruction of faulty signals by means of an ensemble of regression models based on principal components analysis.," in *ANS NPIC HMIT 2009 Topical Meeting*, Knoxville, Tennessee, USA, 2009.

[46] M. D. Morris and T. J. Mitchell, “Exploratory designs for computational experiments,” *Journal of Statistical Planning and Inference,* vol. 43, no. 3, pp. 381-402, 1995.

[47] A. Keane, A. Forrester and A. Sobester, Engineering Design via Surrogate Modelling: A Practical Guide, American Institute of Aeronautics and Astronautics, Inc., 2008.

[48] A. I. Forrester and N. W. Bresslo, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 462, no. 2071, pp. 2177-2204, 2006.

[49] F. A. C. Viana, G. Venter and V. Balabanov, "An algorithm for fast optimal latin hypercube design of experiments," *International Journal for Numerical Methods in Engineering,* vol. 2, pp. 135-156, 2010.

[50] L. Mainini and K. E. Willcox, "Sensor placement strategy to inform decisions," in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, Colorado, USA, 2017.

[51] P. C. Berri, M. D. Dalla Vedova and L. Mainini, "Diagnositcs of actuation systems faults from dynamic data," in *6th European Conference on Computational Mechanics (ECCM)*, Glasgow (UK), 2018.

[52] J. L. Lumley, "Atmospheric Turbulence and Radio Wave Propagation," in *The structure of inhomogeneous turbulent flows*, 1967, pp. 166-178.

[53] P. Holmes, J. L. Lumley and G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge Monographs on Mechanics, Cambridge University Press, 1996.

[54] L. Sirovich, «Turbulence and the dynamics of coherent structures. Part 1: Coherent structures,» *Quarterly of Applied Mathematics,* vol. 45, n. 3, pp. 561-571, 1987.

[55] M. Hinze and S. Volkwein, "Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control," *Lecture Notes in Computational Science and Engineering,* pp. 261-306, 2005.

[56] S. Walton, O. Hassan and K. Morgan, "Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions," *Applied Mathematical Modelling,* vol. 37, no. 20-21, pp. 8930-8945, 2013.

[57] K. E. Willcox and J. Peraire, “Balanced Model Reduction via the Proper Orthogonal Decomposition,” *AIAA Journal,* vol. 40, no. 11, pp. 2323-2330, 2002.

[58] G. Kerschen, J.-c. Golinval, A. F. Vakakis and L. A. Bergman, “The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview,” *Nonlinear Dynamics,* vol. 41, pp. 147-169, 2005.

[59] T. Kohonen and P. Somervuo, “Self-organizing maps of symbol strings,” *Neurocomputing,* Vols. 1-3, pp. 21-30, 1998.

[60] T. Kohonen, Self-organizing Maps, 3rd ed., Berlin, Heidelberg: Springer-Verlag, 2001.

[61] S. Kaski, «Data Exploration Using Self-Organizing Maps,» *Acta Polytechnica Scandinavica: Mathematics, Computing and Management in Engineering Series,* n. 82, 1997.

[62] A. Richardson, C. Risien and F. Shillington, “Using self-organizing maps to identify patterns in satellite imagery,” *Progress in Oceanography,* vol. 59, no. 2-3, pp. 223-239, 2003.

[63] M. Svensson, S. Byttner and T. Rognvaldsson, “Self-organizing maps for automatic fault detection in a vehicle cooling system,” in *4th International IEEE Conference Intelligent Systems*, Varna, Bulgaria, 2008.

[64] T. Aroui, Y. Koubaa and A. Toumi, “Clustering of the Self-Organizing Map based Approach in Induction Machine Rotor Faults Diagnostics,” *Leonardo Journal of Sciences,* vol. 15, pp. 1-14, 2009.

[65] A. Katunin, M. Amarowicz and P. Chrzanowski, “Faults diagnosis using self-organizing maps: A case study on the DAMADICS Benchmark problem,” in *Federated Conference on Computer Science and Information Systems (FedCSIS)*, Lodz, 2015.

[66] E. Oja, «Simplified neuron model as a principal component analyzer,» *Journal of Mathematical Biology,* vol. 15, n. 3, pp. 267-273, 1982.

[67] G. Quattrocchi, P. Berri, M. Dalla Vedova and P. Maggiore, "Innovative Actuator Fault Identification Based on Back Electromotive Force Reconstruction," *MDPI Actuators,* vol. 9, no. 3, p. 50, 2020.

[68] N. E. Murray and L. S. Ukeiley, “An application of Gappy POD,” *Experiments in Fluids,* vol. 42, no. 1, pp. 79-91, 2006.

[69] E. J. Candès, J. K. Romberg and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics,* vol. 59, no. 8, pp. 1207-1223, 2006.

[70] R. Everson and L. Sirovich, “Karhunen–Loève procedure for gappy data,” *Journal of the Optical Society of America A,* vol. 12, no. 8, p. 1657, 1995.

[71] P. Saini, C. M. Arndt and A. M. Steinberg, “Development and evaluation of gappy-POD as a data reconstruction technique for noisy PIV measurements in gas turbine combustors,” *Experiments in Fluids,* vol. 57, no. 7, 2016.

[72] K. Willcox, «Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition,» *Computers & Fluids,* vol. 35, n. 2, pp. 208-226, 2006.

[73] T. Bui-Thanh, M. Damodaran and K. E. and Willcox, “Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition,” *AIAA Journal,* vol. 42, no. 8, pp. 1505-1516, 2004.

[74] M. R. Hasan, L. Montier, T. Henneron and R. V. Sabariego, “Stabilized Reduced-Order Model of a Non-Linear Eddy Current Problem by a Gappy-POD Approach,” *IEEE Transactions on Magnetics,* pp. 1-8, 2018.

[75] M. D. Dalla Vedova, A. Germanà, P. C. Berri and P. Maggiore, “Model-Based Fault Detection and Identification for Prognostics of Electromechanical Actuators Using Genetic Algorithms,” *MDPI Aerospace,* vol. 6, no. 9, p. 94, 2019.

[76] M. D. L. Dalla Vedova, D. De Fano and P. Maggiore, “Neural Network Design for Incipient Failure Detection on Aircraft EM Actuator,” *International Journal of Mechanics and Control (JoMaC),* vol. 17, no. 1, pp. 77-83, 2016.

[77] M. T. Hagan, H. B. Demuth, M. H. Beale and O. D. Jesús, Neural Network Design, Boston, MA, USA: PWS Publishing, 1996.

[78] D. W. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of the Society for Industrial and Applied Mathematics,* vol. 11, no. 2, pp. 431-441, 1963.

[79] Y. Li, G. Tang, J. Du, N. Zhou, Y. Zhao and T. Wu, “Multilayer Perceptron Method to Estimate Real-World Fuel Consumption Rate of Light Duty Vehicles,” *IEEE Access,* vol. 7, pp. 63395-63402, 2019.

[80] S. Li, D. C. Wunsch, E. O’Hair and M. G. Giesselmann, “Comparative Analysis of Regression and Artificial Neural Network Models for Wind Turbine Power Curve Estimation,” *J. Sol. Energy Eng. Nov 2001, 123(4): 327-332,* vol. 123, no. 4, pp. 327-332, 2001.

[81] Ł. Jedliński and J. Jonak, “Early fault detection in gearboxes based on support vector machines and multilayer perceptron with a continuous wavelet transform,” *Applied Soft Computing,* vol. 30, pp. 636-641, 2015.

[82] M. Hagan and M. Menhaj, “Training feedforward networks with the Marquardt algorithm,” *IEEE Transactions on Neural Networks,* vol. 5, no. 6, pp. 989-993, 1994.

[83] S. E. Fahlman, “An Empirical Study of Learning Speed in Back-Propagation Network,” Technical Report CMU-CS-88-162, 1999.

[84] M. D. Dalla Vedova, P. C. Berri and S. Re, “Metaheuristic Bio-Inspired Algorithms for Prognostics: Application to on-Board Electromechanical Actuators,” in *3rd International Conference on System Reliability and Safety (ICSRS)*, Barcelona, Spain, 2018.

[85] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[86] N. Stanevski and D. Tsvetkov, “Using Support Vector Machine as a Binary Classifier,” in *International Conference on Computer Systems and Technologies*, 2005.

[87] K. S. Goh, E. Chang and K.-T. and Cheng, “SVM binary classifier ensembles for image classification,” in *Proceedings of the tenth international conference on Information and knowledge management - CIKM'01*, 2001.

[88] Q. W. S. a. S. D. Leng, “Construction of Binary Tree Classifier Using Linear SVM for Large-Scale Classification,” in *International Conference on Robots & Intelligent System (ICRIS), IEEE*, 2018.

[89] L. Shenbo and L. Zhou, “Fatigue crack monitoring of aerospace structure based on lamb waves and binary tree support vector machines,” *Journal of Vibroengineering,* vol. 19, no. 5, pp. 3271-3282, 2017.

[90] H.-Y. Fan, G. S. Dulikravich and Z.-X. Han, “Aerodynamic data modeling using support vector machines,” *Inverse Problems in Science and Engineering,* vol. 13, no. 3, pp. 261-278, 2005.

[91] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Proceedings of 2nd Berkeley Symposium*, 1951.

[92] W. Karush, Minima of Functions of Several Variables with Inequalities as Side Constraints, Chicago, IL, USA: (M.Sc. thesis). Dept. of Mathematics, Univ. of Chicago, 1939.

[93] M. G. Michailidis, M. Agha, M. J. Rutherford and K. P. Valavanis, “A Software in the Loop (SIL) Kalman and Complementary Filter Implementation on X-Plane for UAVs,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 2019.

[94] A. Nesci, A. De Martin, G. Jacazio and M. Sorli, “Detection and prognosis of propagating faults in flight control actuators for helicopters,” *MDPI Aerospace,* vol. 7, no. 3, p. 20, 2020.

[95] M. Hussien, “Power Electronics Application for More Electric Aircraft,” in *Recent Advances in Aircraft Technology*, InTech, 2012.

[96] Y. Deng, S. Y. Foo and I. Bhattacharya, “ “Regenerative electric power for More Electric Aircraft,”,” in *IEEE SouthEastCon*, Lexington, KY, USA, 2014.

[97] R. Quigley, «More Electric Aircraft,» in *Proceedings of the Eighth Annual IEEE Applied Power Electronics Conference and Exposition*, 1993.

[98] M. Howse, “All-electric aircraft,” *Power Engineer,* vol. 17, no. 4, p. 35, 2003.

[99] A. Garcia Garriga, S. S. Ponnusamy and L. Mainini, “A multi-fidelity framework to support the design of More-Electric Actuation,” in *2018 Multidisciplinary Analysis and Optimization Conference, AIAA Aviation Forum, (AIAA Paper 2018-3741)*, 2018.

[100] J.-C. Maré, “Electro-mechanical Actuators,” in *Aerospace Actuators 2*, John Wiley & Sons, Inc., 2017, pp. 171-217.

[101] J. Weimer, “The role of electric machines and drives in the more electric aircraft,” in *IEEE International Electric Machines and Drives Conference, 2003. IEMDC'03.*, Madison, WI, USA, 2003.

[102] M. J. Cronin, “All-electric vs conventional aircraft - The production/operational aspects,” *Journal of Aircraft,* vol. 20, no. 6, pp. 481-486, 1983.

[103] T. Missala, “Electromechanical Actuators – Selected Safety-Related Problems,” *Recent Advances in Automation, Robotics and Measuring Techniques,* pp. 175-186, 2014.

[104] L. Stridsberg, “Low Weight, Highly Reliable Anti-Jamming Device for Electromechanical Actuators,” *SAE Technical Paper Series, SAE International,* 2005.

[105] F. Annaz, «Architecture philosophy in high integrity electromechanical actuators,» in *5th International Symposium on Mechatronics and Its Applications, IEEE*, Amman, Jordan, 2008.

[106] Y. Hussain, S. Burrow, P. Keogh and L. Henson, “A Review of Techniques to Mitigate Jamming in Electromechanical Actuators for Safety Critical Applications,” *International Journal of Prognostics and Health Management,* vol. 9, 2018.

[107] L. Borello and M. D. L. Dalla Vedova, “Load dependent coulomb friction: a mathematical and computational model for dynamic simulation in mechanical and aeronautical fields,” *International Journal of Mechanics and Control (JoMaC),* vol. 7, no. 1, pp. 19-30, 2006.

[108] D. Quinn, “A new regularization of Coulomb friction,” *Journal of Vibration and Acoustics,* vol. 126, no. 3, pp. 391-397, 2004.

[109] D. Karnopp, “Computer Simulation of stick slip friction in mechanical dynamic systems,” *Journal of Dynamic Systems, Measurement, and Control,* vol. 107, pp. 100-103, 1985.

[110] L. Borello and M. D. L. Dalla Vedova, “Dry Friction Discontinuous Computational Algorithms,” *International Journal of Engineering and Innovative Technology (IJEIT),* vol. 3, no. 8, 2014.

[111] B. L. Stevens, F. L. Lewis and E. N. Johnson, Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd Edition, Wiley, 2016.

[112] E. Balaban, P. Bansal, P. Stoelting, A. Saxena, K. Goebel and S. Curran, "A diagnostic approach for electro-mechanical actuators in aerospace systems," in *Proceedings of the 2009 IEEE Aerospace Conference*, Big Sky, MT, USA, 2009.

[113] E. Balaban, A. Saxena, K. Goebel, C. Byington, S. Bharadwaj and M. Smith, "Experimental Data Collection and Modeling for Nominal and Fault Conditions on Electro-Mechanical Actuators," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, San Diego, CA, USA, 2009.

[114] E. Balaban, A. Saxena, S. Narasimhan, I. Roychoudhury, K. Goebel and M. Koopmans, "Airborne Electro-Mechanical Actuator Test Stand for Development of Prognostic Health Management Systems," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, Portland, OR, USA, 2010.

[115] D. Belmonte, M. D. L. Dalla Vedova and P. Maggiore, "Electromechanical servomechanisms affected by motor static eccentricity: Proposal of fault evaluation algorithm based on spectral analysis techniques," in *Safety and Reliability of Complex Engineered Systems (Proceedings of ESREL 2015)*, 2015.

[116] D. Belmonte, M. D. L. Dalla Vedova and P. Maggiore, "Prognostics of Onboard Electromechanical Actuators: A New Approach Based on Spectral Analysis Techniques," *International Review of Aerospace Engineering (I.RE.AS.E),* vol. 11, no. 3, pp. 96-103, 2018.

[117] P. Berri, M. Dalla Vedova and P. Maggiore, "A Smart Electromechanical Actuator Monitor for New Model-Based Prognostic Algorithms," *International Journal of Mechanics and Control (JoMaC),* vol. 17, no. 2, pp. 19-25, 2016.

[118] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore and F. Viglione, "A simplified monitoring model for PMSM servoactuator prognostics," in *9th EASN International Conference on "Innovation in Aviation & Space"*, Athens, Greece, 2019.

[119] P. C. Berri, M. D. L. Dalla Vedova and P. Maggiore, "A Simplified Monitor Model for EMA Prognostics," in *8th EASN-CEAS International*

Workshop on "Manufacturing for Growth & Innovation", Glasgow, UK, 2018.

[120] A. De Martin, G. Jacazio, A. Nesci and M. Sorli, "In-depth Feature Selection for PHM System's Feasibility Study for Helicopters' Main and Tail Rotor Actuators," in *Proceedings of the European Conference of the PHM Society*, 2020.

[121] A. De Martin, G. Jacazio and M. Sorli, "Enhanced particle filter framework for improved prognosis of electro-mechanicalflight controls actuators," in *Proceedings of the European Conference of the PHM Society*, 2018.

[122] K. Ahlstrom, J. Torin, K. Fersan and P. Nobrant, "Redundancy management in distributed flight control systems: experience & simulations," in *Digital Avionics Systems Conference*, 2002.

[123] P. Goupil, J. Boada-Bauxell, A. Marcos, E. Cortet, M. Kerr and H. Costa, "AIRBUS efforts towards advanced real-time Fault Diagnosis and Fault Tolerant Control," in *IFAC proceedings volumes*, Cape Town, South Africa, 2014.

[124] H. M. Elattar, H. K. Elminir and A. M. Riad, "Towards online data-driven prognostics system," *Complex & Intelligent Systems volume 4, pages271–282(2018),* vol. 4, pp. 271-282, 2018.

[125] L. Tang, E. Hettler, B. Zhang and J. DeCastro, "A Testbed for Real-Time Autonomous Vehicle PHM and Contingency Management Applications," in *Annual Conference of the Prognostics and Health Management Society*, 2011.

[126] F. Tornabene, N. Fantuzzi and M. Bacciocchi, "Finite Elements Based on Strong and Weak Formulations for Structural Mechanics: Stability, Accuracy and Reliability," *International Journal of Engineering & Applied Sciences,* vol. 9, no. 2, pp. 1-21, 2017.

[127] D. Lisitano, J. Slavič, E. Bonisoli and M. Boltezăr, "Strain proportional damping in Bernoulli-Euler beam theory," *Mechanical Systems and Signal Processing,* vol. 145, p. 106907, 2020.

[128] T. Yoshino, K. Kurosawa, K. Itoh and T. Ose, "Fiber-Optic Fabry-Perot Interferometer and its Sensor Applications," *IEEE Transactions on Microwave Theory and Techniques,* vol. 30, no. 10, pp. 1612-1621, 1982.

[129] J. Santos and F. Farahi, Handbook of Optical Sensors, CRC Press, 2018.

[130] D. Ahuja and D. Parande, "Optical sensors and their applications," *Journal of Scientific Research and Reviews,* vol. 1, no. 5, pp. 060-068, 2012.

[131] J. Habel, T. Boilard, J. S. Frenière, F. Trépanier and M. Bernier, "Femtosecond FBG written through the coating for sensing applications," *MDPI Sensors,* vol. 17, no. 11, p. 2519, 2017.

[132] D. Grobnic, S. Mihailov, C. W. Smelser and H. Ding, "Sapphire Fiber Bragg Grating Sensor Made Using Femtosecond Laser Radiation for Ultrahigh Temperature Applications," *IEEE Photonics Technology Letters,* vol. 16, no. 11, pp. 2505-2507, 2004.

[133] N. Tanaka, Y. Okabe and N. Takeda, "Temperature-compensated strain measurement using fiber Bragg grating sensors embedded in composite laminates," *Smart Materials & Structures,* vol. 12, no. 6, pp. 940-946, 2003.

[134] W. Chen, R. Gassino, Y. Liu, A. Carullo, G. Perrone, A. Vallan and D. Tosi, "Performance assessment of FBG temperature sensors for laser ablation of tumors," in *Proceedings of the IEEE International Symposium on Medical Measurements and Applications*, 2015.

[135] A. Theodosiou, D. Pugliese, E. Ceci Ginistrelli, N. Boetti, D. Janner, D. Milanese and K. Kalli, "Femtosecond Laser Written Plane-by-Plane Bragg Grating Sensors in Bioresorbable Phosphate Optical Fibres," *Journal of Lightwave Technology,* vol. 99, p. 1, 2019.

[136] S. J. Mihailov, "Fiber Bragg Grating sensors for harsh environments," *Sensors,* vol. 12, no. 2, pp. 1898-1918, 2012.

[137] S. J. Mihailov, D. Grobnic, C. Hnatovsky, R. B. Walker, P. Lu, D. Coulas and H. Ding, "Extreme Environment Sensing Using Femtosecond Laser-Inscribed Fiber Bragg Gratings," *MDPI Sensors,* vol. 17, no. 2, p. 2909, 2017.

[138] N. Hirayama and Y. Sano, "Fiber Bragg grating temperature sensor for practical use," *ISA Transactions,* vol. 39, no. 2, pp. 169-173, 2000.

[139] J. Jung, H. Nam, B. Lee, J. O. Byun and N. S. Kim, “Fiber Bragg grating temperature sensor with controllable sensitivity,” *Applied Optics,* vol. 38, no. 13, pp. 2752-2754, 1999.

[140] D. Zhang, J. Wang, Y. Wang and X. Dai, “A fast response temperature sensor based on fiber Bragg grating,,” *Measurement Science and Technology,* vol. 25, no. 7, 2014.

[141] Y. Zhang, L. Zhu, F. Luo, M. Dong, R. Yang, W. He and X. Louet, “Comparison of Metal-Packaged and Adhesive-Packaged Fiber Bragg Grating Sensors,” *IEEE Sensors Journal,* vol. 16, no. 15, pp. 5958-5963, 2016.

[142] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore and T. Scolpito, “Feasibility study of FBG-based sensors for prognostics in aerospace applications,” in *Journal of Physics: Conference Series*, 2019.

[143] P. C. Berri, D. V. M. D. L. and P. Maggiore, “Experimental Comparison of Fiber Bragg Grating Installation Techniques for Aerospace Systems,” in *9th EASN International Conference on Innovation in Aviation & Space*, Athens, Greece, 2019.

[144] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore, C. Corsi, A. Laudani and C. Secci, “Study of packaging and installation of FBG sensors for monitoring of aircraft systems,” in *30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference (ESREL2020 PSAM15)*, Venice, Italy, 2020.

[145] H. Boussalis, K. Valavanis, D. Guillaume, F. Pena, E. U. Diaz and J. Alvarenga, “Control of a Simulated Wing Structure with Multiple Segmented Control Surfaces,” in *2013 21st Mediterranean Conference on Control & Automation (MED)*, Platanias-Chania, Crete, Greece, 2013.

[146] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore and G. Riva, “Design and Development of a Planetary Gearbox for Electromechanical Actuator Test Bench through Additive Manufacturing,” *MDPI Actuators,* vol. 9, no. 2, p. 35, 2020.

[147] I. Geonea, N. Dumitru and I. Dumitru, “Experimental and theoretical study of friction torque from radial ball bearings,” in *IOP Conf. Series: Materials Science and Engineering*, 2017.

[148] SKF, “Using a friction model as an engineering tool,” *Evolution technology magazine from SKF,* 8 june 2006.

[149] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore and A. Manuello Bertetto, “Evaluation of Aiding and Opposing Load Efficiency for an Actuator Planetary Drive,” *International Journal of Mechanics and Control (JoMaC),* vol. 21, no. 1, pp. 113-120, 2020.

[150] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore and A. Manuello Bertetto, “Simplified models for mechanical transmission efficiency with opposing and aiding loads,” *International Journal of Mechanics and Control (JoMaC),* vol. 20, no. 2, pp. 135-139, 2019.

[151] G. Quattrocchi, P. C. Berri, M. D. L. Dalla Vedova and P. Maggiore, “Back-EMF Reconstruction for Electromechanical Actuators in Presence of Faults,” in *30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference (ESREL2020 PSAM15)*, Venice, Italy, 2020.

[152] L. Mainini and K. E. Willcox, “Data to decisions: Real-time structural assessment from sparse measurements affected by uncertainty,” *Computers & Structures,* vol. 182, pp. 296-312, 2017.

[153] A. Saxena, J. Celaya, B. S. S. Saha and K. Goebel, "Metrics for Offline Evaluation of Prognostic Performance," *International Journal of Prognostics and Health Management,* vol. 1, no. 1, pp. 2153-2648, 2010.