



ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR



Doctoral Dissertation
Doctoral Program in Mechanical Engineering (33rd cycle)

Advanced Diagnostic for Fly-by-Wire Flight Control Systems

Rocco Gentile

* * * * *

Supervisors

Prof. Sorli Massimo, Supervisor
Prof. Jacazio Giovanni, Co-supervisor
Ing. Francesco Marino, Industrial Supervisor

Doctoral Examination Committee:

Prof. Vertechy Rocco, Referee, University of Bologna
Prof. Zobel Pierluigi Beomonte, University of L'Aquila

Politecnico di Torino
August 12, 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.

.....

Rocco Gentile
Turin, August 12, 2021

Summary

The Fly-By-Wire technology (FBW) is adopted as primary flight control actuation system for both legacy aircraft and new-generation ones. In particular, several studies have been conducted over the failure modes of these systems, that are considered both safety critical from a technical perspective and a profitable subject of investigation for all the Maintenance Repair and Overhaul (MRO) companies in aviation industry. The actual testing procedures are typically time-consuming and they need to be upgraded in line with the new Industry 4.0 indications, such as automation and data-collection.

In this framework, this thesis reports the results that have been obtained during a three-years research project in collaboration with Lufthansa Technik, with the final goal to automate and improve the diagnostic accuracy with innovative Machine Learning techniques. The development of a new testing protocol has required the collection of results from different research activities.

The first fundamental step has been the design and implementation of an autonomous testing through a series of concatenated modular signals. These signals include a set of predefined shapes that can be adapted per different types of tested servo-commands, with the goal to collect a set of Health-Features (HF) per each critical unit sub-component. Each signal shape has been designed in order to cover most of the traditional prescribed tests in order to cut the testing time of more than one third.

The collected signals are then automatically stored and analysed by an autonomous feature-extraction algorithm based on signal processing techniques. This software represents the testing protocol core, since it has been tuned in order to extract all the possible information from the measurements that can be used to track any anomalous behaviour of the unit sub-components. During the experimental campaign in the LHT shops in Hamburg, several set of measurements have been collected and analysed in order to refine and validate the HFs extraction techniques from the signals. Furthermore, in order to improve the actual data collection process, an automatic data-collection framework has been built to store correctly all the measurement channels and test results from the experimental campaign. In particular, the extracted HFs from the measured channels are systematically collected in a unique final report that has the double goals of firstly informing the

technician about the tested unit health status, and secondly building-up a database of test reports.

All the information from these reports are in fact automatically collected in a unique database which lay the foundation for both any statistical analysis and the training and testing operation for the Diagnostic Module algorithm. This last part of the testing protocol is based on a Supervised Machine Learning algorithm which has to automatically recognize the health status of each unit part from the collected HFs. Firstly, per three target components, a sub-set of optimal HFs have been extracted according to their statistical relevance. Afterwards, these subset of data have been used to tune the algorithms hyperparameter over a specific training set. In this way, the algorithms have been trained to correctly associate the assigned failure labels, and define multi-dimensional decision boundaries for any further classification of new incoming data. The classification performance are then compared by defining specific metrics that can be calculated according to wrongly and correctly recognized classes.

In conclusion, the advanced diagnostic framework includes an autonomous and adaptable testing procedure, an advanced signal postprocessing software for feature extraction, a solid framework of data collection in terms of measurements and reports and a classification algorithm for failure identification. These core blocks definitely improve the quality of failure recognition for these component and at the same time the productivity of the repairing workshop with innovative techniques.

Acknowledgements

I would like to acknowledge my academic advisors, Prof. Massimo Sorli and Prof. Giovanni Jacazio. Furthermore, I would like to express my deepest gratitude to the whole T-CE 2Z Lufthansa Technick Department for their great professional and personal support during my stay in Hamburg.

In particular, it is my privilege to thank ing. Francesco Marino, Michael Burke, Oliver Ritter and Andreas Formella for their kind help and co-operation throughout my research period.

Last but not least, I owe a deep sense of gratitude to my dearest loved ones: my parents Luigi and Maria Grazia, my sister Paola and my better half Laura. Your love and encouragement have been essential during these years away from home.

*To my loving family
and to Laura*

Contents

List of Tables	XI
List of Figures	XII
1 Introduction	1
1.1 Historical Background	2
1.1.1 Prognostic and Advanced Diagnostic in Aeronautic Industry	3
1.1.2 Economic forecast of MRO market	6
1.2 Project Motivation	10
2 Flight Control Systems	15
2.1 General Overview	15
2.2 A320 FBW Architectures	17
2.3 A320 Elevator	21
2.4 A320 Aileron	28
2.5 A320 Spoiler	31
2.6 Position Transducers	35
2.6.1 Demodulation Techniques	37
3 Experimental Setup	43
3.1 Prototyping Laboratory Test bench	43
3.1.1 Hydraulic Test Bench	44
3.1.2 Measurement System	46
3.1.3 Testing Software	47
3.1.4 Collaborative Robot	49
3.2 Test Field Test bench	51
3.2.1 Digital control loops	51
4 Automatic Entry Test	55
4.1 Traditional OEM Maintenance Protocol	55
4.2 Automatic Procedure	57
4.3 Modular Signal Generation	59
4.3.1 A320 Elevator signal	62

4.3.2	A320 Aileron signal	75
4.3.3	A320 Spoiler signal	77
4.4	Testing Framework	83
5	Health Features Extraction and Collection	89
5.1	HyDiag software structures	90
5.2	Measurements data management	91
5.3	HyDiag software structure	94
5.4	Sequences Loading and Pre-processing	97
5.4.1	extraction_channel.m function	98
5.4.2	Ratiometric Demodulator function	101
5.4.3	Signal Denoising function	101
5.4.4	Synchronous Demodulator function	108
5.4.5	findchangeslope function	109
5.5	Sequences Signals HFs Extraction	114
5.5.1	Preliminary Sequences	117
5.5.2	Pressure Adjustment Sequences	121
5.5.3	Leakages Sequences	135
5.5.4	Automatic Sequences	143
5.6	Hydiag Report	172
6	Data Collection and Diagnostic Module	177
6.1	HyDiag Reports Database Building	178
6.2	Failure labels assignment	181
6.2.1	Servovalve Failures	182
6.2.2	Recentering-Device Failures	184
6.2.3	Mode Switching Valve Failures	186
6.2.4	Failure Labels assignment criteria through HFs	186
6.3	Features Selection	192
6.4	Classification Algorithms	202
6.4.1	Logistic Regression for Classification	203
6.4.2	K-Nearest Neighbour for Classification	205
6.4.3	Support Vector Machine for Classification	207
6.4.4	Multi-Layer Perceptron for Classification	211
6.5	Classifiers Comparison and Results	213
6.5.1	Dataset split and Classifier Tuning	216
6.5.2	Classification Results	221
7	Conclusion	233
7.1	Summary	233
7.1.1	Automatic and Scalable testing procedure	234
7.1.2	Automatic Health-Features extraction from experimental data	235

7.1.3	Data collection of experimental results and test reports . . .	235
7.1.4	Failure recognition through ML algorithms	236
7.2	Future Development	237
A	Synchronous Demodulation	239
A.1	Synchronous Demodulation - Integral Formulation	239
A.1.1	Nominal conditions: no phase-shift between signals	239
A.1.2	Phase-shift between signals	241
A.1.3	Noise effect	242
B	Health Features limits reduction	249
B.1	Historical Database	249
B.2	Data cleansing and pre-processing	251
B.2.1	Data cleansing with <code>db_build_histodb_fix.py</code>	251
B.3	Sweet-spots identification in limits range	254
	Nomenclature	259
	Bibliography	263

List of Tables

2.1	A320 Servo-commands overview	18
2.2	A320 Elevator Components	22
2.3	A320 Aileron Components	29
2.4	A320 Spoiler Components	32
4.1	Elevator preliminary sequences info	64
4.2	Elevator pressure adjustments sequences info	67
4.3	Elevator leakages sequences info	69
4.4	Elevator automatic sequences info	74
4.5	Aileron automatic sequences info	78
4.6	Spoiler pressure adjustments sequences info	81
4.7	Spoiler leakages sequences info	82
4.8	Spoiler automatic sequences info	84
5.1	Preliminary Sequences HFs Recap table	121
5.2	Pressure Sequences HFs Recap table	122
5.3	Leakages Sequences HFs Recap table	136
5.4	Fully Automatic Sequences HFs Recap table	144
5.5	HyDiag Report structure in .csv format	174
6.1	EHSV Failures Labels	189
6.2	RD Failures Labels	192
6.3	MSV Failures Labels	192
6.4	Feature Selection Methods Comparison	194
6.5	Selected HFs from ANOVA F-score per each component classification	202
6.6	Classifiers tuned parameters	220
6.7	Classifier Performance Metrics	224
B.1	pn_cmm_limits.csv structure	253

List of Figures

1.1	RUL prediction scheme	4
1.2	ISO 13374-1 processing framework	5
1.3	PHM technique for product development	6
1.4	Economic Forecast of MRO market	8
1.5	Aircraft Data Management Value Chain	9
1.6	Heavy Maintenance procedures in LHT facility	10
1.7	Normalized Elevator TSI distribution per SLE	11
1.8	SLE percentages for A320 actuators in LHT shops during 2019	12
1.9	A320 FBW units TSI distribution after the repairing events in the LHT shops (from 2017 to 2020)	13
2.1	Aircraft Principal Axes	16
2.2	FBW Architecture	17
2.3	A320 Flight Control Systems	17
2.4	A320 Flight Control Computers Scheme	19
2.5	A320 Hydraulic Systems Layout	20
2.6	A320 Hydraulic Supply to FCS	20
2.7	A320 Elevator Installation Scheme	21
2.8	A320 Elevator Main Subcomponents	23
2.9	A320 Elevator Working Modes	24
2.10	A320 RD Scheme	25
2.11	RD Force-Position characteristic curve	26
2.12	RD Current-Position characteristic curve	27
2.13	Scheme of a jet-deflector EHSV	28
2.14	A320 Aileron Installation Scheme	29
2.15	A320 Aileron Illustrated	30
2.16	A320 Aileron Working Modes	31
2.17	A320 Spoiler Installation Scheme	32
2.18	A320 Spoiler Illustrated	33
2.19	A320 Aileron Working Modes	34
2.20	LVDT electric scheme [41]	36
2.21	LVDT linear characteristic	37
2.22	LVDT configurations	38

2.23	Synchronous Demodulator	39
2.24	Lock In Amplifier Demodulator	40
2.25	6- wires RVDT electric scheme	41
2.26	Ratiometric Demodulator	41
3.1	H-Lab configuration	44
3.2	H-Lab Test Bench Hydraulic Scheme	45
3.3	H-Lab Measurement System	46
3.4	Workstation GUI panel	48
3.5	UR10 Robotic Arm for the 31075 Elevator adjustment	50
4.1	Standard Maintenance Protocol	56
4.2	HyDiag Testing Protocol	58
4.3	Signal generation code	61
4.4	Elevator complete excitation signal	63
4.5	Elevator weight evaluation	64
4.6	Elevator Preliminary Sequences	65
4.7	Elevator Pressure Adjustment Sequences	66
4.8	Elevator Leakages Sequences	68
4.9	Elevator Automatic Sequences	70
4.10	Aileron complete excitation signal	75
4.11	Aileron weight evaluation	76
4.12	Spoiler complete excitation signal	77
4.13	Spoiler Pressure Adjustment Sequences	80
4.14	Spoiler Leakages Sequences	82
4.15	Spoiler Automatic Sequences	83
4.16	HyDiag Test Framework (<i>LHTSmart Factory</i>)	84
4.17	H-Lab vs Test Field Procedure	85
4.18	Hydiag Testing Scheme	87
5.1	Hydiag SW versions flowchart	91
5.2	HyDiag software PV version data management flowchart	92
5.3	Hydiag SW GUI	95
5.4	High-level DBG software flowchart	96
5.5	<code>extraction_channel</code> function flowchart	100
5.6	<code>denoising</code> function flowchart	103
5.7	Time series inputs channels normalization and PSD analysis	105
5.8	Window Functions	106
5.9	Magnitude Squared Coherence Coefficient	106
5.10	Magnitude Squared Coherence Peaks	107
5.11	Filtered signal with <code>denoising</code> function	108
5.12	Change point detection (example with Sequence n.15)	111
5.13	<code>stab_finder</code> example of Sequence n.15 Current signal (zoom)	112
5.14	Change point detection (example with Sequence n.7)	115
5.15	<code>move_finder</code> example of Sequence n.7 Position signal (zoom)	116

5.16 Elevator <i>Sequence</i> 1	118
5.17 Aileron <i>Sequence</i> n.1	119
5.18 Spoiler <i>Sequence</i> n.1	119
5.19 INC sensor fixing	120
5.20 Elevator <i>Sequence</i> n.3	124
5.21 Aileron <i>Sequence</i> n.3	124
5.22 <i>Sequences</i> n.3 for both Elevator and Aileron	124
5.23 Elevator <i>Sequence</i> n.4	126
5.24 Elevator <i>Sequence</i> n.5	126
5.25 Elevator <i>Sequences</i> n.4 and 5	126
5.26 Aileron <i>Sequence</i> n.4	127
5.27 Aileron <i>Sequence</i> n.5	127
5.28 Aileron <i>Sequences</i> n.4 and 5	127
5.29 Elevator <i>Sequence</i> n.6	129
5.30 Step State Level Analysis	131
5.31 EHSV Steps Analysis	132
5.32 EHSV LVDT Elbow Detection	133
5.33 Elevator <i>Sequence</i> n.7	135
5.34 <i>Sequences</i> n.8 for all PFCS	138
5.35 Elevator <i>Sequence</i> n.9	139
5.36 <i>Sequences</i> n.9 Oscillation Frequency Spectrum	140
5.37 Spoiler <i>Sequences</i> n.9	141
5.38 Elevator <i>Sequences</i> n.10	142
5.39 Elevator <i>Sequences</i> n.12	143
5.40 Elevator <i>Sequences</i> n.12 MSV spectrum analysis	144
5.41 Elevator <i>Sequences</i> n.13 command	148
5.42 Elevator <i>Sequences</i> n.13 RD characteristic	149
5.43 Elevator <i>Sequences</i> n.13 RD characteristic: x_0 and I_0 evaluation	150
5.44 Elevator <i>Sequences</i> n.13 RD characteristic: deadband identification	151
5.45 Elevator <i>Sequences</i> n.13 RD characteristic: deadband identification (zoom)	151
5.46 Elevator <i>Sequences</i> n.14	153
5.47 Elevator <i>Sequences</i> n.15	154
5.48 Elevator RD eccentric adjustment	155
5.49 Elevator <i>Sequences</i> n.15: SNR and f_{peak} extraction from PSD analysis	156
5.50 Elevator <i>Sequence</i> n.16	158
5.51 Elevator <i>Sequence</i> n.16 Hysteresis Plot	158
5.52 Elevator <i>Sequence</i> n.17	159
5.53 Aileron <i>Sequence</i> n.17	160
5.54 Elevator <i>Sequence</i> n.18	160
5.55 Elevator <i>Sequence</i> n.19	161
5.56 Aileron <i>Sequence</i> n.19	162

5.57 Elevator <i>Sequence</i> n.20	162
5.58 MSV LVDT Elbow Detection	163
5.59 Elevator <i>Sequence</i> n.21	164
5.60 Elevator <i>Sequence</i> n.22	164
5.61 Elevator <i>Sequence</i> n.23	165
5.62 Aileron <i>Sequence</i> n.23	166
5.63 Spoiler <i>Sequence</i> n.23	166
5.64 Elevator <i>Sequence</i> n.23 resolution plot	167
5.65 Elevator <i>Sequence</i> n.24	168
5.66 Elevator <i>Sequence</i> n.25	169
5.67 Elevator <i>Sequences</i> n.26 command	170
5.68 Elevator <i>Sequences</i> n.26 EHSV characteristic	171
5.69 Elevator <i>Sequences</i> n.26 EHSV HFS evaluation	171
5.70 HyDiag Report generation flowchart	173
5.71 Folder organization for both data and reports	174
6.1 Diagnostic Module Parts	178
6.2 HyDiag Reports building process	179
6.3 SLE Dataframe in pivot form	180
6.4 HyDiag Database analysis	181
6.5 EHSV structure	183
6.6 RD Bell-Crank degraded surface	185
6.7 RD Control Shaft degraded surface	185
6.8 RD Bell-Crank degraded surface	186
6.9 Failure labels assignment flowchart	187
6.10 HyDiag Database reshape	188
6.11 Target Component Classes populations	191
6.12 Feature Selection Methods	193
6.13 Fisher Linear Discriminant for class separation	196
6.14 F-Distributions shapes depending on its degrees of freedom	197
6.15 HFS Correlation Heatmap	199
6.16 Target component HFS ranked according to $-\log_{10}(p)$	201
6.17 Sigmoid Function	203
6.18 Logistic Regression probability contours	204
6.19 KNN Classification examples depending on K	206
6.20 Support Vector Machine Classifier	207
6.21 SVM Classification examples depending on C	208
6.22 SVM Classification with different Kernels	210
6.23 SVM with RBF Kernel: C and γ influence	210
6.24 MLP structure	212
6.25 MLP Classifier tuning α	214
6.26 Classification Flowchart	215
6.27 Best Practice Guideline to avoid overfitting	217

6.28	Cross-Validation Technique	218
6.29	<code>GridSearchCV</code> strategy	219
6.30	Confusion matrix and related performance measures	221
6.31	EHSV target component: Classifiers Confusion Matrices	226
6.32	RD target component: Classifiers Confusion Matrices	227
6.33	MSV target component: Classifiers Confusion Matrices	228
6.34	Two Principal component over example dataframe	229
6.35	Data points distribution per each target component	231
6.36	Classifier decision boundaries after PCA	232
A.1	Synchronous Demodulation ($\phi = 0$)	241
A.2	Effect of filter order on Synchronous Demodulator	241
A.3	Synchronous Demodulation with variable phase-shift	242
A.4	Noisy signal and its frequency spectrum	243
A.5	Rectification of noisy signal	244
A.6	4-wires secondary LVDT coil signal and its frequency spectrum	245
A.7	Alignment of first harmonic of U_{sek} signal with U_p	246
B.1	Sweet spot identification	250
B.2	Source databases for the Historical one [31]	250
B.3	Simplified flowchart for data cleaning	252
B.4	NFF and Recertified units	255
B.5	IE sweet-spot range identification	258
B.6	IE sweet-spot range identification	258

Chapter 1

Introduction

In recent decades, advanced health monitoring systems and condition-based maintenance (CBM) have raised as disruptive technologies in aviation industry. These techniques may influence all the product lifecycle phases of complex systems, from the design to service stages. The concept is to evolve the maintenance protocol from passive to active, in order to predict in advance the right timing to perform preventive maintenance actions. Furthermore, the possibility to detect faults and degradations in early stages provide useful information to the concept and design phases, helping to develop more durable and fault-tolerant components. For this reason, advanced diagnostic methods are essential for complex and critical systems, especially in the aviation industry where any failure inconvenience may lead to delays, expensive on-ground operation or, in worst case scenario, safety issues. For this reason, the principal goal is to improve reliability of maintenance procedures and in the same time reducing cost with innovative autonomous techniques. Due to the increasing level of the technologies involved in each aircraft system, intelligent maintenance protocols have to take place instead of the old ones, in order to ensure constant reliability and serviceability of these systems. On this basis, advanced diagnostic techniques have to be designed in order to be autonomous and data-centric: the importance of collecting high-quality data from maintenance procedures enables further development in terms of failures prediction and prognostic.

This chapter introduces the historical evolution of aircraft maintenance procedures and the economic benefits in terms of advanced diagnostic. Afterwards, a particular focus in the aviation industry is presented, and in particular the health monitoring systems for safety critical components such as flight control systems. In this framework, the HyDiag research project, in collaboration with Lufthansa Technik AG, is presented. This thesis reports the results of four years long collaboration between Politecnico di Torino and Lufthansa Technik AG, in the framework of the HyDiag project on the development of innovative diagnostic procedures.

1.1 Historical Background

In aircraft industry, maintenance procedures ensure high standards of safety, operational reliability and passenger comfort. As reported in [1], the Air Transport Association (ATA) gathers together several professionals from aeronautic industries in order to create regulation for maintenance protocol in commercial aviation. In particular, this set of regulations has evolved during the years following the constant evolution of the aircraft components, and therefore their relative maintenance procedures. For instance, the first maintenance research project was promoted by United Airlines in 1965 in order to define common rules that could be applied for systematic review of the aircraft design [2]. Afterwards, in 1968, the "Maintenance Evaluation and Program Development" handbook was developed for the 747-100 and published by the ATA Maintenance Steering Group (MSG) , which gather together different stakeholders such as airframe manufacturers, airlines, suppliers and the U.S. Federal Aviation Administration . This document, also known as MSG-1, suggested to use decision logics to determine correct overhaul maintenance tasks, taking into account one particular component in a system and analyzing the most critical parts of this component. The possible maintenance actions were classified as:

- Hard Time (HT) : based on life limits or maximum intervals of time until a necessary maintenance task
- On Condition (OC) : the component can be subject to repetitive inspections during its lifetime in order to determine its conditions, leaving it on board until a failure occurs.

This set of regulations appeared for short time since already in the early 70', it was updated into the new "Airline/Manufacturer Maintenance Program Planning Document" or MSG-2, applicable to a wider range of aircraft. In this new document, a third possible action was introduced called Condition Monitored Maintenance (CM) : the component is not checked until a failure occurs. This control process could not be applied to all the aircraft components and in fact the safety critical ones were excluded by this kind of action. For instance, the engines could be On-Condition checked but not Condition Monitored while a single Pressurization Air Conditioning (PAC) valve can be treated with Condition Monitoring actions due to its less priority in terms of safety. The introduction of the CM action has not to be intended due to the high number of components in each aircraft component, and not all of them can be monitored with OC approach for both time and economical issues. From these two first set of regulations, the introduction of OC action guaranteed several advantages in terms of down-time reduction and maintenance cost decrease. The last upgrade MSG-3 is an analytical method based on the "Reliability-Centered Maintenance" (RCM) , published in the '80 from two United

Airlines executive engineers. During the years, this document has been revised and updated (until the current ATA MSG-3, 2015), but it still represents the core part of the actual maintenance rules. It proposed a task-oriented approach, that can identify the inherent reliability of the on-board systems and components, avoiding un-useful or late maintenance tasks. This document proposes three possible methods: preventive and predictive maintenance and components redesign-by-failure. In particular, the RCM method is based on the Failure Mode & Effects Analysis (FMEA) to identify failures and their relative roots. This method ranks the failures based on severity criteria, and it is widely used in a series of scientific applications [3–5]. This kind of analysis gives a better understanding of the anomalies roots and therefore reduce maintenance costs, giving a deeper insight of each connection between interconnected components in complex systems. In fact, especially in aeronautic components, a single degradation can induce a failure in many other connected systems and, if a systematic approach is not adopted, extracting each cause-effect relationship can be very demanding. The *ARINC 624* standard was developed in the early '90 as regulations for on-board maintenance systems, introducing directly failure-sensitive on board system that enables to collect more information about multiple systems to support further on-ground inspections. For this reason, several studies have been conducted on both on-board Advanced Diagnostic and Prognostic systems that can be adopted to evaluate the current health-status of monitored systems and their predicted remaining useful-life.

1.1.1 Prognostic and Advanced Diagnostic in Aeronautic Industry

In [2] Prognostic and Health Management (PHM) is defined as the engineering discipline that is focused on autonomous degradation detection and prediction of Remaining Useful Life (RUL) of the system under analysis. The RUL can be defined as the amount of probable time for which the component can keep working in its design condition before a detected degradation can evolve into failure. In Fig. 1.1 it is schematically described the approach to evaluate the RUL due to the evolution of a detected degradation into a failure. Firstly it is necessary, to implement an estimation algorithm which is able to predict the evolution of the fault dimension over time, and secondly it is necessary to set the operating limits of the test component. A suitable candidate for prediction algorithm is the Particle Filter, which has widely used also in the aerospace field [2, 6–9]. In the plot in Fig. 1.1, the initial point L_0 is identified as the "Initialize long-term prediction" and it used by the Particle filter algorithm to evaluate the mean value of the observed degradation and its relative uncertainty boundaries, with a probabilistic approach. The RUL is estimated as the time when the average predicted value encounters the "Hazard zone" of the prescribed acceptable limits. Mathematically, it is expressed

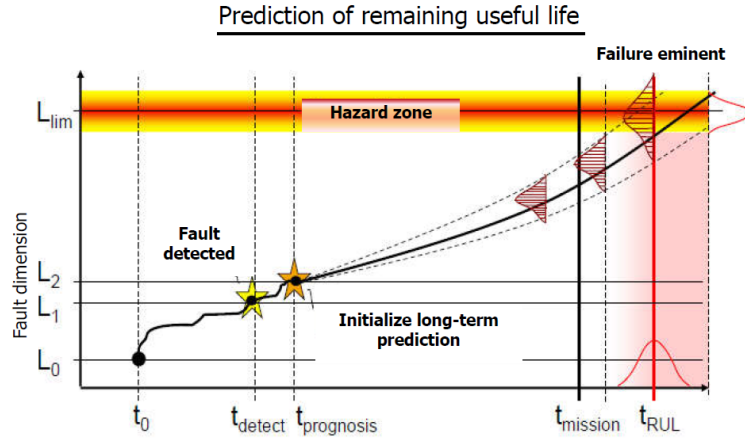


Figure 1.1: RUL prediction scheme [12]

by the median probability curve when "its average value is the most probable instant of rupture" [2]. This parameter can be evaluated in terms of operating hours, cycles or other time-metrics and it can be used to schedule the necessary maintenance tasks to avoid a failure, influencing both logistic and overhaul operations. In particular, the proposed method from the PHM experts can support the maintainers to schedule their tasks, and the logistics managers to support them during the warehouse management and spare parts supplies. As a consequence, these tasks may positively influence the work of the process or fleet managers that needs to plan the production schedules and mission requirements, but also it can give useful information during the design phase of more fault-tolerant systems [10, 11]. This kind of integrated approach has been proposed in [2] and described in the flowchart of Fig. 1.3, where the FMECA firstly and, almost at the end-life of a product, the PHM outputs constitute a useful source of information for further improvement of the next-generation product. The PHM approach enables an accurate data collection of faulty condition directly integrated during the operating life of the analysed component, keeping track of useful information that can complete the other set of data coming from the maintenance on-ground operations.

The International Standard Organization (ISO) has defined the ISO 13374-1 standard processing model, which is referred as the typical framework for processing data and provide basic PHM indications. Fig. 1.2 provides an overview of the framework blocks.

Implementing a reliable PHM system requires however to face a series of highly demanding technological challenges that are typically hard to solve:

- The fast analysis of different signal sources, coming from integrated sensors. A fundamental phase for fault identification is in fact the extraction of specific Health Features (HF) that can be used to identify the health status of the analysed component. These features needs to be highly correlated with the

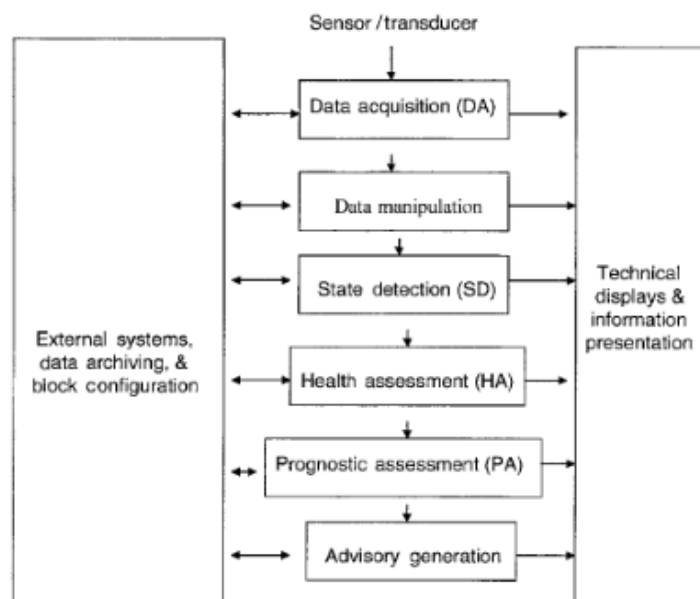


Figure 1.2: ISO 13374-1 processing framework

degradation to observe and as less as possible influenced by environmental conditions [13].

- Once the degradation is identified with specific set of HFs or their combination, it is necessary to maximize the RUL prediction with well tuned algorithms, such as the aforementioned Particle Filter.

For aeronautic components, these tasks needs to be implemented and integrated in the on-board monitoring systems, especially the stages regarding data collection from the signal and feature extraction. These processes would require dedicated avionic components with relatively high-computing power, that therefore needs to be certified and correctly installed in the aircraft avionic bay. In order to minimize these challenges, a possible solution would be to allocate the RUL estimation task on external on-ground facilities that are able to communicate in real-time with the on-board systems. Some big players of the Aerospace industry have already started this process: for instance, the Airbus *Skywise* platform is already able to extract the data from new-generation aircraft (such as the A350), fuse them with other important data from on-ground facility, and provide useful real-time information about the health status of each on-board component. More details about the economical benefits these solutions are bringing in the MRO market are reported in paragraph 1.1.2.

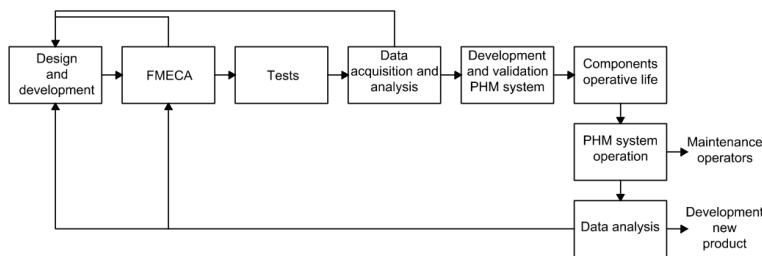


Figure 1.3: PHM technique for product development [2]

In this framework, data collection and manipulation and the definition of a reliable diagnostic algorithm for anomaly detection are two key-factors of any health-monitoring system, and even more if based on PHM algorithms. In the recent years, several research studies have been conducted over these two fundamental points. Most of these studies are focused on the development of effective fault detection on-board system, which are able to provide indications about the degradation entity and the impact over other connected systems [14–16]. In this framework, both model-based techniques, based on high-fidelity models [17, 18] or reduced-order ones [19], and data-driven techniques, based on historical data as in [20] and [21], have been widely tested to find the best methods to be developed in the on-board systems. However, especially in these latter cases of data-driven approach, several researches have highlighted the most challenging issue of collecting a reasonable quantity of high-quality data from the on-board systems and furthermore they are characterized by low-sampling features, therefore a considerable preprocessing is needed for extracting valuable time-series measurements. A important source of data is therefore represented by the great amount of results that can be extracted by the on-ground tests [22, 23]. Thanks to the modern technologies of Industry 4.0, a disruptive improvement of standard maintenance protocol needs to be pursued adopting innovative techniques such as automatic testing, structured data collection and pattern recognition techniques for anomaly identification through Machine Learning algorithms. All these topics constitutes the building blocks of the Advanced diagnostic for Flight Control System framework in this thesis.

1.1.2 Economic forecast of MRO market

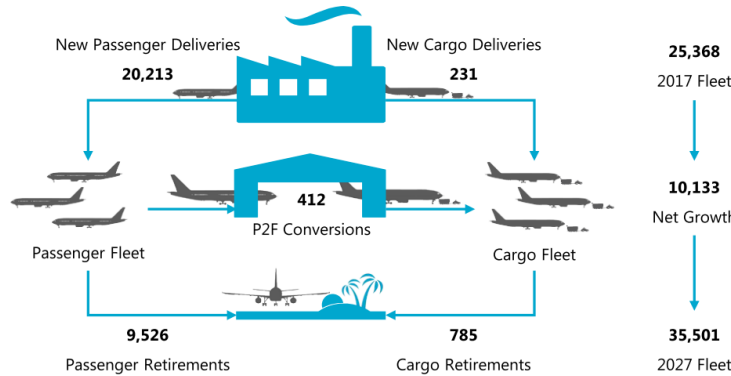
Over the last decades, advanced failure diagnosis has been a relevant topic in several engineering fields. Research groups have focused their activities over the the development of advanced and automatic technologies to detect failures or anomalous behaviours and furthermore implement decision-making algorithms that are able to react according to the failure potential impact over the system health

status. As previously mentioned, the interest in improving both on-board failure recognition algorithms and on-ground maintenance procedures have exponentially increased, due to the predicted increase in size of both passenger and cargo fleet, as in Fig. 1.4a. As reported in [24], the number of commercial aircraft will increase up to more than 35000. This raise includes both more than 20000 new-generation aircraft, that will induce the airlines companies and Maintenance Repair and Overhaul industries to face new technological challenges. In Fig. 1.4b the expected spend in terms of Compound Annual Growth Rate (CAGR) percentages are described per each MRO market segments in the next years. In particular, the total MRO spend will increase over the first five-year period of an equivalent 2.4% CAGR, and then up to 5.2% in the second half of the observed decade. Considering the full period, the global MRO market will grow 3.8% CAGR annually, hitting the maximum 109\$ billion of investment. This forecast influence all the segments, where the engines one represent the larger portion due to the new technological trends regarding high-performance materials or hybrid propulsion systems. For instance, the european Clean Sky project [25] targets to reduce up to 30% fuel consumption by 2035 adopting new radical hybrid propulsion systems for short-medium range aircraft. These kind of innovative technologies imply a necessary shift of methodologies and protocols at the base of maintenance procedures. Both OEM , MRO industries and all the market stakeholders needs to prepare and be trained for new technological challenges:

- The on-board failure detection systems will require more sophisticated hardware and software capabilities to perform advanced health-monitoring procedures, elaborating more inputs from several on-board sensors. For instance, in the framework of developing fully-digital and connected aircraft, the actual A350 includes more than 50000 sensors that are able to collect 2.5 TB of data per each day [26]. This huge amount of data needs to be processed by several new-generation control computers, whose serviceability and safety-critical have gained more importance.
- As consequence of the first point, the number of health-monitoring parameters from this new available quantity of data will considerably rise, therefore the MRO industries needs to define the methods to use them in the best way possible.
- As far as the tech-level of the on-board system grow, equivalently the on-ground testing infrastructures need to evolve in terms of equipments and personnel training.

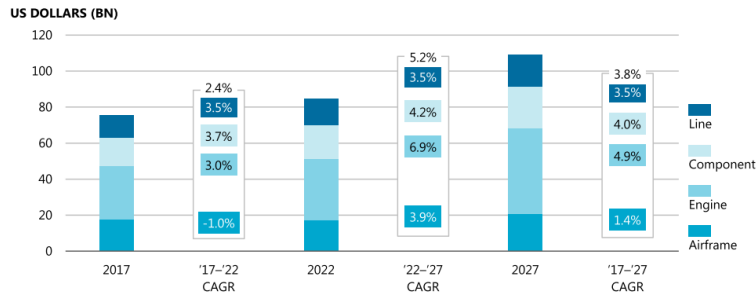
In the recent years, all the bigger market companies have introduced their health monitoring platforms, based on the increasing quantity of collected data. As reported in the ICF report in [27], in the last three years, several companies are

EXHIBIT 11: 2017-2027 GLOBAL AIRCRAFT DEMAND



(a) Global Aircraft Demand

EXHIBIT 32: 2017-2027 MRO MARKET FORECAST BY MRO SEGMENT



(b) MRO market segment forecast

Figure 1.4: Economic Forecast of MRO market [24] (Analysis performed before the COVID-19 pandemic)

already collecting the benefits of predictive maintenance approach: for instance, Delta airlines has achieved a 98% reduction of maintenance-related cancellations events, while easyJet has registered 31 correctly predicted failures events before they occurred, through the Airbus digital platform *Skywise* [28]. The graph in Fig. 1.5 reports the importance of on-board collected data in both the health monitoring and management phases. In particular, both Diagnostic and Prognostic disciplines represent the key areas of study which determine the necessary maintenance action while the aircraft is still operating or on-ground.

To sum up, effective health-monitoring and management systems can improve the aircraft operational reliability, avoiding on-ground immobilization time or delays, and they can drastically reduce the direct maintenance cost during on-ground operations, which are typically affected by two factors:

- Due to safety standards to be respected, typically the No Fault Found rate (NFF) is high especially for the component market segment. For instance,

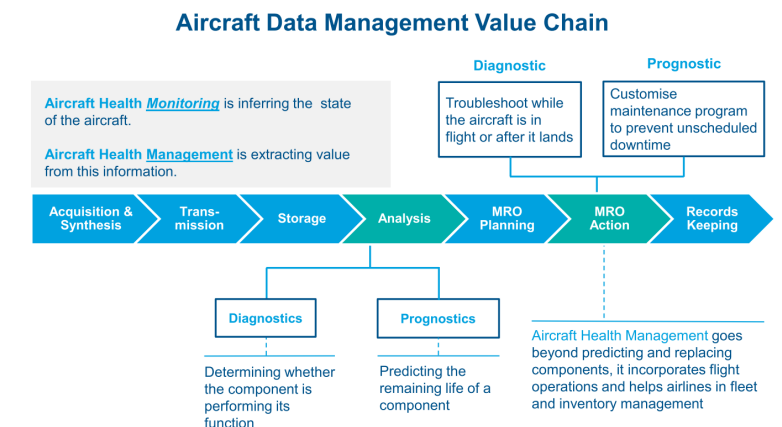


Figure 1.5: Aircraft Data Management Value Chain [27]

considering a couple of aileron units on a side of a wing, if during a flight an anomaly is detected from this group of actuators, both of them are dismantled and checked during normal on-ground maintenance operations. Even if this procedure ensures a high-level of safety, it is very likely that one of the tested units results fully serviceable, thus increasing the number of NFF units that are actually tested even without any anomaly. This phenomenon may lead to high testing cost, especially for this kind of units with an overall NFF rate above 25% [6].

- Each aircraft component is typically a complex assembly of parts whose working conditions are highly affected by all the others, therefore failure identification is typically a task that requires large efforts in terms of time and man-power [29].

In this scenario, the research project presented in this thesis aims to develop an innovative failure diagnostic framework for a specific class of aircraft components, in order to improve on-ground maintenance operations in terms of time, automation and reliability of anomaly detection. The target class of components includes all the Fly-by-Wire (FBW) flight control systems. Looking at the bar-chart in Fig. 1.4b, this class of component belongs to the "Component" segment: even if the predicted portion of investment is considerably smaller than other (such as the Engine sector), these units are still considered a key use-case in the MRO market, since due to their high NFF rate they represent an area of improvement of Maintenance procedures. Furthermore, the FBW hydraulic units will still be used as reliable actuation technology in the next decade especially for narrow and wide body aircraft.



Figure 1.6: Heavy Maintenance procedures in LHT facility [30]

1.2 Project Motivation

In the aviation MRO market, Lufthansa Technik AG (LHT) is a key international player with more than 20000 employees and providing maintenance services for the whole Lufthansa Group and other 800 customers worldwide. Considering the market shares in Fig. 1.4b, LHT facilities includes all the main segments. In particular, considering the Component market share, one of the company core business is the so called Total Material Operations: a complete tailored maintenance service that ensures a comprehensive aircraft overhaul and spare parts pooling. Due to the company in-house wide range of materials, LHT is able to guarantee short turnaround time and high material availability. In the Component market, these resources allows the company to guarantee overhaul services for more than 360000 Shop Load Events (SLE) each year. This considerable number of required SLE from the customer has driven the LHT Innovation departments to investigate and focus a lot of efforts over innovative maintenance procedures, which are able to keep high reliability and in the same time reducing time and cost.

In this framework, the HyDiag Project aims to develop an innovative testing protocol for FBW components, improving the standard test procedure in terms of automation, data collection, and failure recognition. These goals imply an overall update of the traditional standards in terms of testing capabilities. For instance, concerning the upgrades in terms of automation and data collection, the HyDiag Project needs to explore the possible improvements that needs to be fulfilled over the adopted test-benches, in terms of equipped sensors, testing software and hardware on-site data-collection capabilities. On the other side, the economical benefit of these investments can be already tangible in short-mid term in terms of time and man/hours reduction and overall component health-status evaluation, from the extracted results by the data-driven failure recognition algorithm. As consequence, an improved and faster failure detection can positively influence the overall repairing-shop management, in terms of work-schedule and logistic.

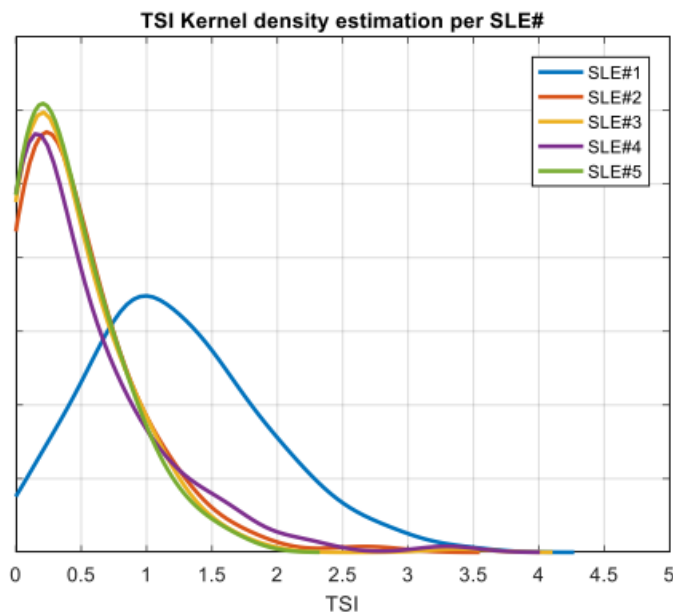


Figure 1.7: Normalized Elevator TSI distribution per SLE [31]

A detailed analysis of the project economical benefits have been in detail described in [31] and [32]. In both the previous research works, it has been reported an analysis based on the Time Since Installation (TSI), defined as the time-range that occurs between two consecutive SLE. This metric is particularly significant to evaluate the performance of the repairing tasks of failure identification during a SLE: the smaller it is, more likely is the chance to have conducted a poor failure recognition in the older SLE. For example, the plot in Fig. 1.7 reports the TSI distribution function per each SLE of a critical component: the A320 Elevator. Looking at these curves, it is possible to notice how the overall average TSI falls soon after the first repairing event. This result highlights for instance the economical benefit an upgrade in terms of failure recognition would bring if the average life of this use-case can be extended after the first SLE. Similar conclusion can be extended also to the other project use cases. In particular, from a technical point of view, the necessity of a more effective failure recognition is also justified by FBW actuators structural complexity, which makes more difficult the identification of one or more contemporary failure roots. Furthermore, from a technical point of view, the FBW actuators can be considered a perfect match for developing a more-automated testing procedure, since they can be electrically controlled. A detailed description of their physical structures are reported in Chapter 2.

From an economic perspective, four main reasons are listed in [17] that explain the motivations behind the choice of considering electro-hydraulic actuators as a key element for cost saving in the Component market sector. Beyond the already

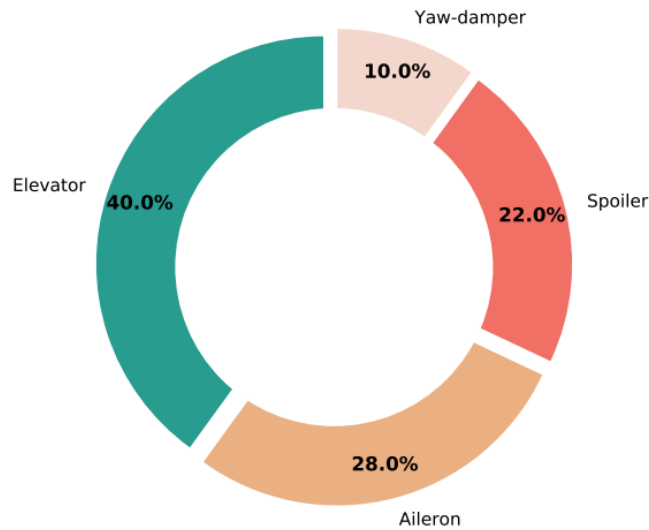


Figure 1.8: SLE percentages for A320 actuators in LHT shops during 2019 [33]

mentioned complex structure that contributes to increase the overall failure-rate of these components, it is important to mention that hydraulic systems are still considered the most common and reliable actuation solution for both flight-controls and landing gear systems, even in the next-generation and more-electric aircraft [6].

The first civil FBW aircraft was the Airbus A320, launched in 1988, and it still represents one of the most reliable systems and adopted by several airline companies around the world. This aircraft has always been in the top-ranking positions in the sector of medium-haul commercial aircraft, which is also the most profitable segment today. Furthermore, in 2012 it has been renewed and upgraded in the today A320neo version, which enables to reduce by 15% the fuel consumption and by 8% the operational cost. These improvements have increased the demand by the airline companies, ensuring the A320neo market share to remain stable in the next decades. For these reasons, the A320 family FBW system has been identified as the perfect match between technical and economical reasons and therefore pointed as HyDiag project use-cases. More details about the A320 FBW architecture are provided in 2.2. The piechart in Fig. 1.8 shows number of SLE in the LHT shops during 2019 of A320 FBW units. From this plot, it is evident how the aforementioned Elevator unit covers the biggest share of the whole registered repairing events, while still considerable percentages regard the Aileron and Spoiler actuators. Improving the maintenance procedures for these three particular units is of keen interest and in fact most of the efforts of this research work have been dedicated for these units.

The boxplots in Fig. 1.9 instead complete the TSI distribution analysis in Fig. 1.7 per each type of FBW unit. In particular, it is possible to observe how these distributions are particularly asymmetrical, with a median value (red lines) that

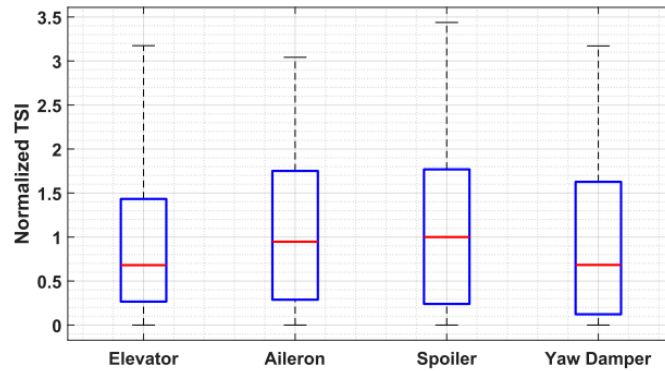


Figure 1.9: A320 FBW units TSI distribution after the repairing events in the LHT shops (from 2017 to 2020) [33]

tend to low TSI value. In particular, the median TSI value of the Elevator class is significantly lower than the others, giving another insight about the margin of improvement of its testing procedure.

Chapter 2

Flight Control Systems

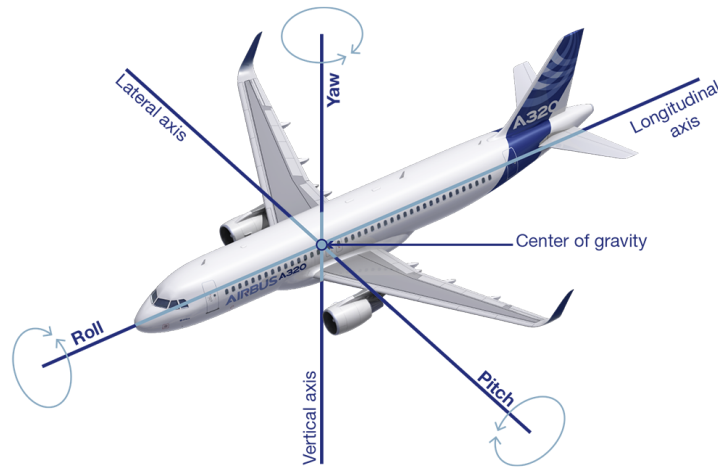
As previously described in Chapter 1.2, the A320 PFCS family has been chosen as reference group of units which deserve particular attention in terms of maintenance improvements. Their considerable number of repairing events is strictly linked to the great success of these short-medium range twin-jet narrow-body aircraft. Their extensive use in continental routes leads to a substantial number of manoeuvres, takeoffs and landing, compared to the long-range aircraft. The first part of the chapter introduces in details the A320 Flight Control Systems architecture. After this overview, each unit structure is described in terms of hydraulic scheme, focusing on the role of each main sub-component, and flight tasks.

2.1 General Overview

Considering the aircraft as a rigid body and locating an euclidean coordinate system in its centre of mass, it is possible to distinguish three principal rotational axes (Fig. 2.1) :

- **Longitudinal:** from tail to nose, and positive in the direction of the flight. The rotation over this axis is called Roll;
- **Vertical:** starting from the centre of gravity, it is perpendicular to the wings and oriented to the bottom part of the aircraft. Its relative rotation is named Yaw;
- **Lateral:** it runs from the pilot's left to right and parallel to the wings. The last principal rotation along this axis is called Pitch.

The translations along these three axes are referred with the same names. The pilot regulates the aircraft 6 degrees of freedom acting on the engines thrust and controlling the PFCS:



Reference Axis definitions

Figure 2.1: Aircraft Principal Axes

- **Roll Control:** the pilots command the deflection of both the Ailerons, at the tip of each wing. They are deflected in opposite directions: the one downwards increases the lift on the up-wing, while the one upward instead decreases the lift on the down-wing. In addition, Spoilers can also be slightly moved reducing Ailerons deflection, improving aircraft manoeuvrability;
- **Yaw Control:** rotations along the vertical axis are regulated by the Rudder PFCS on the vertical stabilizer;
- **Pitch Control:** Trimmer Horizontal Stabilizer (THS) (for long term regulation) and Elevators (for short term) on the horizontal tail are the PFCS used to control the aircraft inclination along the pitch axes.

Together with the PFCS, Secondary Flight Control Systems (SFCS) improve aircraft performances. Typically, they are intermittently used during the phases of takeoff, approaching and landing. For example, commercial aircraft are equipped with both Speedbrakes Spoilers, which deflect at full stroke while on ground during landing, Slats on the leading edge and Flaps on the trailing one, to modify the lift on each wing. Both PFCS and SFCS have evolved, starting from fully mechanical devices to fully digital FBW systems. The gradual introduction of FBW components on commercial aircraft started in the 80's, due the parallel evolution of electronic controlling logics. In 1988, the Airbus - A320 was the first commercial aircraft which adopted FBW technology for PFCS, combining the reliability of hydraulic actuators with electronic control logic directly linked to the Flight Control Computers FCCs (Fig. 2.2).

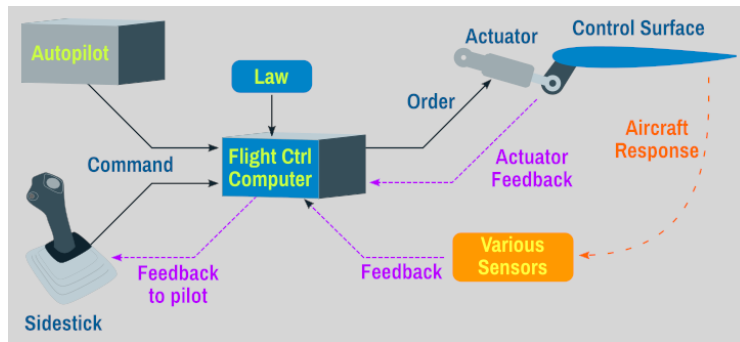


Figure 2.2: FBW Architecture

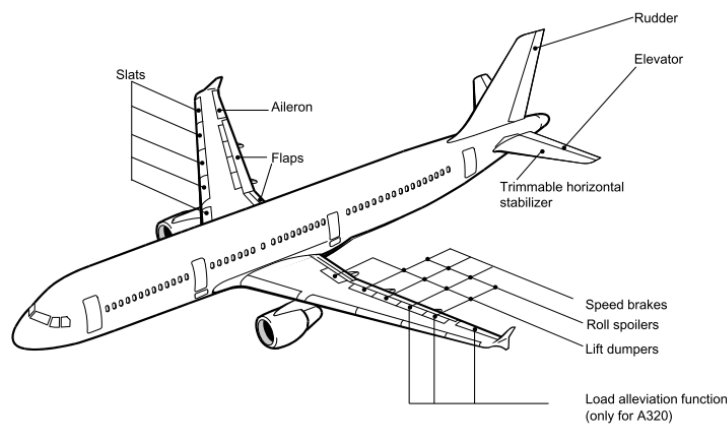


Figure 2.3: A320 Flight Control Systems [34]

The commands from the pilot cockpit are converted in electric signals and interpreted by the FCCs, where control-loop laws are implemented per each kind of actuator. The critical role of FCC are correctly commanding the unit position, monitoring its correct displacement and checking eventual failure of these unit.

Flight safety is guaranteed by the high level of redundancy of control, supply and actuators architectures. The next paragraph describes in detail both the complete redundant architectures of the A320 and highly-integrated structure of each use-case servo-command, which has been studied during this research.

2.2 A320 FBW Architectures

In analogy with Fig. 2.1, Fig. 2.3 illustrates where the PFCS and SFCS are located on the A320 body.

The first level of redundancy is guaranteed by the number and operating mode. Each servo-commands group, with at least two actuators, is installed on each surface

(Table 2.1). Specifically, two Ailerons are installed on each outboard portion of the wings, two Elevators are mounted on each side of the horizontal stabilizer, while three rudders actuators are positioned on the vertical tail. So, per each primary surface, at least two servo-commands are present: typically one works in the so-called Active Mode, while the other operates in Damping Mode, following the surface movements under its aerodynamic load. In case of failure, FCCs are ready to switch the working-mode of each couple of units and cut both the hydraulic and electric supply to the anomalous unit. Spoilers surfaces are the only which include only one servo-command: in fact, five Spoilers are mounted under each surface on the upper part of the wing. In case of failure, both the faulty unit and its relative on the opposite wing are hydraulically deactivated.

Table 2.1: A320 Servo-commands overview

Servo-command	n. x surf.	Command	Actuation type
Elevator	2	Electric	Hydraulic
Aileron	2	Electric	Hydraulic
Rudder	3	Electro-Mechanic	Hydraulic
Spoiler	1	Electric	Hydraulic
THS	1	Electro-Mechanic	Hydraulic

Due to its structural complexity, the entire A320 actuation system presents other two levels of safety-redundancy in both FCCs and Hydraulic supply architectures.

FCS Command Architecture

The A320 is equipped with seven FCCs which command and monitor the PFCS movements according to pilot's inputs. As shown in Fig. 2.4, the side-sticks orders are converted in electrical signals and directly sent to the relatives FCCs. At the same time, rudder pedals and trim hand-wheel are still mechanically coupled to both the rudder tail and horizontal stabilizer.

The aforementioned seven different kind of FCCs are:

- Two Elevator and Aileron Computers (ELACs), for monitoring active elevators and ailerons;
- Three Spoiler and Elevator Computers (SECs). They all achieve spoilers control, while two of them monitor the standby elevator;
- Two Flight Augmentation Computers (FACs). They check the rudder unit position and "*characteristics speeds calculation for displays*" [34].

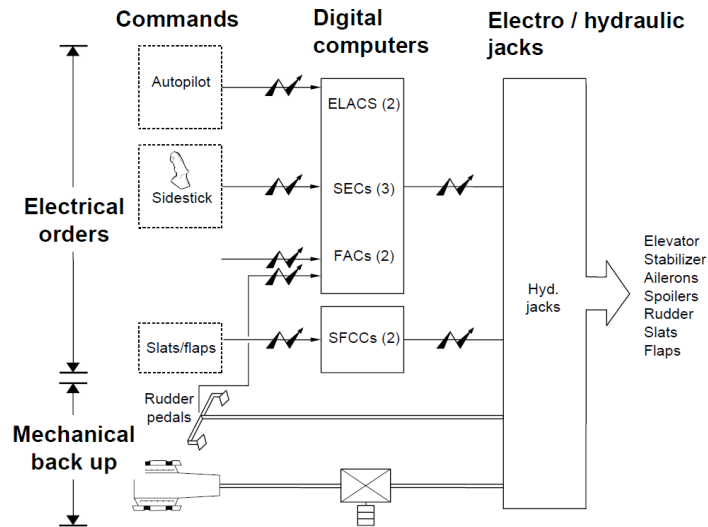


Figure 2.4: A320 Flight Control Computers Scheme [34]

Additionally, Slats and Flaps have two dedicated computers (SFCC). The data from the ELACs and SECs are collected by the Flight Control Data Concentrators (FCDC), and transmitted both to the Electronic Instrument System (EIS) and to the Centralized Fault Display System (CFDS). All these computers are powered by two separate supplies, each of them linked to an engine-driven generator. In case of power loss, an emergency generator, driven by the Auxiliary Power Unit (APU), ensures that all the FCCs can work safely. In this case, the Mechanical back-up of rudder and stabilizer control allows a stable control of the plane.

Hydraulic Supply Architecture

Three independent hydraulic lines supply the flight control systems (Fig. 2.5)

- **Green Line:** pressurized and driven by the left engine;
- **Yellow Line:** pressurized and driven by the right engine;
- **Blue Line:** emergency circuit, driven by the Ram Air Turbine (RAT) pump.

In case of one engine failure, the related supply pump is disconnected and the hydraulic line is then pressurized by the Power Transfer Unit (PTU). The Blue line is pressurized by the RAT if both the engines are faulty. In normal conditions, one ELAC monitors both Yellow and Green lines, while the second one is always ready to activate the emergency Blue one. The reservoirs in Fig. 2.5 are filled with Skydrol® type IV (or HyJet V), a specific phosphate-ester aeronautic fluid ([36], [37]). Using phosphate ester composites allows the supply lines to work in

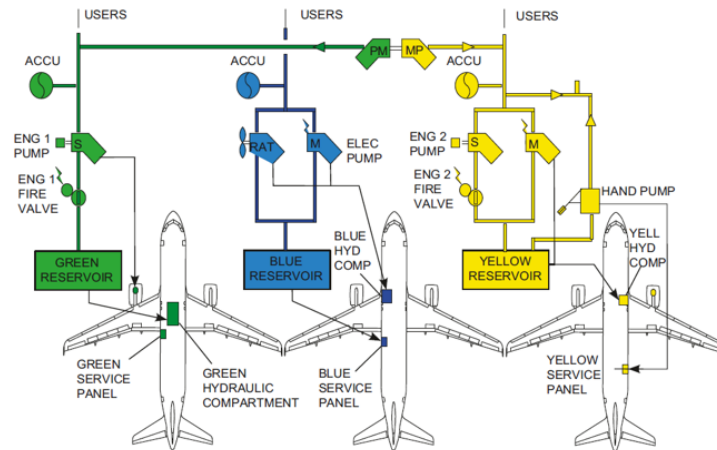


Figure 2.5: A320 Hydraulic Systems Layout [35]

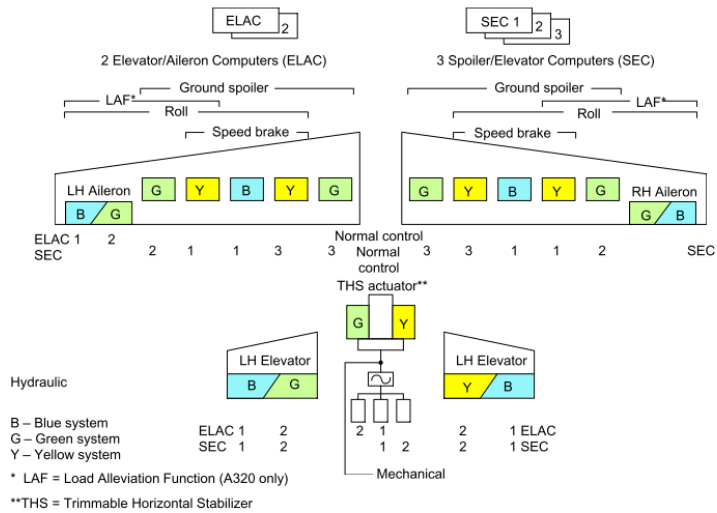


Figure 2.6: A320 Hydraulic Supply to FCS [34]

high pressure conditions (higher than 5000 psi), keeping the fluid efficiently fire-resistant, less effected by anti-erosion, antioxidants and rust-inhibitors additives. The way these lines supply the servo-command is described in Fig. 2.6.

This scheme clearly shows the positions of multiple actuators on each surface, and how each of them is supplied by a different hydraulic line. For both the Ailerons and Elevators groups, the active unit is driven by Yellow or Green line, while the one in damping mode receives the oil from the emergency Blue line. The THS ball screw is driven by two hydraulic motors, connected to both Yellow and Green lines. Since the Spoiler surfaces are actuated by a single servo-command, they are all connected to separated lines: in case of hydraulic failure on one of these lines,

the corresponding spoilers remain in fully retracted mode. The vertical stabilizer contains both three hydro-mechanical rudder actuators, all of them supplied by a different line, plus two additional Yaw Dampers actuators. These small units prevent the "Dutch roll" effect: an induced and adverse yaw during a roll manoeuvre. They are also FBW units, positioned on the bottom part of the vertical tail. They can be activated by the pilots and are automatically driven by the FACs, according to the data coming from the ELACs.

Flight safety is guaranteed not only by the redundant architectures of both command and hydraulic lines, but also by the complex and integrated structure of each servo-command.

2.3 A320 Elevator

Fig. 2.7 in detail represents the complete Pitch control system, including both THS and Elevators. In particular, this scheme highlights the differences between the THS mechanical command chain and the FBW Elevator one. The four Elevator units on the horizontal stabilizer are electrically commanded by the ELACs (or SECs in case of failure), while the THS is connected to the cockpit hand-wheel through the mechanical trim linkage. As highly integrated component, the Elevator includes in its structure 15 main sub-components, including electro-hydraulic valves, mechanical components and internal sensors. Table 2.2 collects all the main fifteen sub-components. The first column indexes mark these components in both Fig. 2.8 and 2.9.

This unit shares several components with the other servo-commands of the A320 family, but it is characterized by some peculiarities. The first characteristic component is the RD: a mechanical system which connects the Main Ram with the EHSV jet-deflector and the RVDT. It ensures that, in case of electrical failure, both the EHSV spool and the Main Ram are recentered in their null position. Its structure is explicitly described in paragraph 2.3. Another characteristic is the presence of

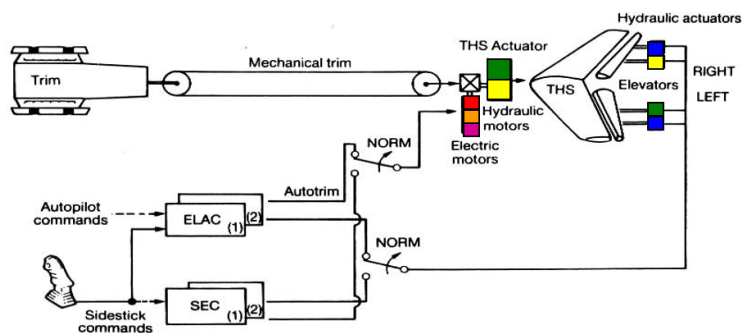


Figure 2.7: A320 Elevator Installation Scheme

Table 2.2: A320 Elevator Components

N.	Component	Abbreviation
1	High Pressure Filter	HP filter
2	HP-LP Poppets Valve assembly	HP-LP Poppets
3	Electro-Valve n.1	EV1
4	Electro-Valve n.2	EV2
5	Electro-Hydraulic Servovalve	EHSV
6	EHSV spool LVDT	EHSV LVDT
7	Mode Switching Valve	MSV
8	MSV sleeve LVDT	MSV LVDT
9	By-pass orifice	
10	Main Ram	
11	Rotary Variable Differential Transformer	RVDT
12	Accumulator	Acc
13	LP Maintaining Valve	
14	Feed Valve	FV
15	Recentering Device	RD

three integrated position sensors: a 6-wires RVDT for the Main Ram, which represents the feedback branch of the external position controller in the monitoring FCC, and two 4-wires monitoring LVDT for the second-stage EHSV spool and the MSV one. Three different integrated sensors facilitate failure identification during a maintenance procedure, without installing extra bench sensors. Last but not least, the elevator unit is the only A320 PFCS which can switch from Active to Damping Mode activating at least one of the two EVs in series. The illustrations in Fig. 2.8a and Fig. 2.8b indicates where the components in the previous list are positioned on a real unit. In particular, the photographed component is lacking the Main Ram movable eye-end, but it shows the integrated electrical sockets (with the red cap). In particular, the socket n.16 includes all the supply and measuring connections with the EHSV, while the n.17 is linked to the RVDT and n.18 with both EV1 and EV2.

This particular units has two main operating modes. During its Maintenance procedure, several tests have to be performed in order to check that each sub-component works correctly in each of them:

- **Active Mode** (Fig. 2.9a): typical normal operating condition. The unit is pressurized through the supply port A, while all the internal sensors (6, 8, 11) are electrically supplied. The fluid (Skydrol©IV or similar) is filtered in the HP Filter (1) and it pushes the HP-LP Poppets (2), connecting all the other subcomponents in Fig. 2.9a with the supply and return line. Since EV1 (3)

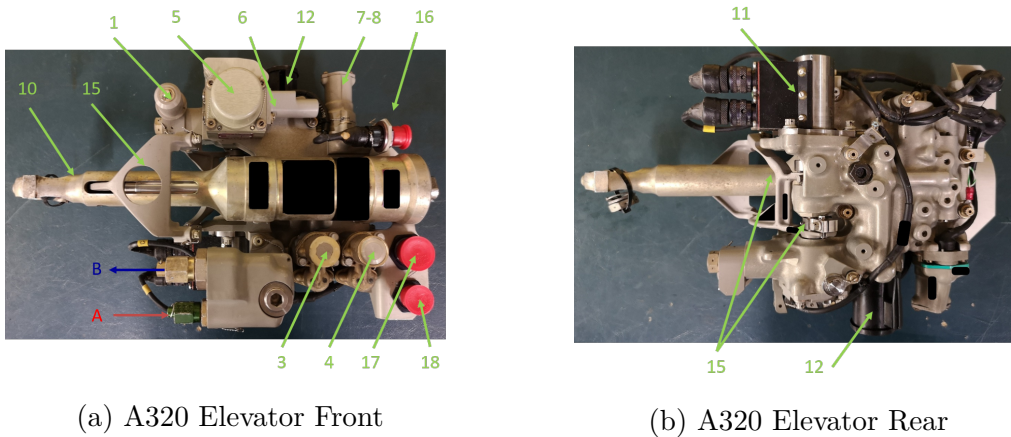


Figure 2.8: A320 Elevator Main Subcomponents

and EV2 (4) are normally open (both their solenoids are de-energized), the supply pressure pushes the MSV spool (7), compressing its spring and clearing the passages between EHSV (5) and Main Ram (10). The correct MSV position is detected by its LVDT (8). In this configuration, the movements of the Main Ram (10) are directly controlled by the flow provided by the EHSV (5), which receives as input the output command from the external Controller. The AC signal from the RVDT (11) is converted by an external demodulator, which estimates the movements of the actuator, closing the position-control loop on the external Controller. The EHSV LVDT (6) is then just used for monitoring purposes. On the return line, the Accumulator (12) keeps enough pressure to open the LP maintaining valve (13) and let the outlet flow to be discharged through the port B.

- **Damping Mode** (Fig. 2.9b): this operating condition makes the unit working as a damper, and it can be activated by the board ELAC in case of failure recognition. The Main Ram movements are inhibited by the MSV (8), which remains closed due to the reacting force of its spring. In this case, one or both the EVs solenoids (3)(4) are switched-on, preventing the pressurized oil to flow in the supply chamber of the MSV. Cylinder chambers (10) are then just connected through the By-pass orifice (9), while the EHSV ports are isolated. Eventual movements of the piston (10) (due for example to an external aerodynamic force) are guaranteed by the Accumulator (12), which provides enough flow to the cylinder, through the Feed-Valve (14) and the restrictor (9). Accumulator (12) also keeps a constant pressure on the return line, opening the LP Maintaining Valve (13), allowing any kind of internal leakage flow from the Main Ram and from the EHSV. In case of EVs electrical failure (3)(4), the hydraulic supply is interrupted, and both the HP-LP Poppets (2) and MSV switch their position, setting the unit in damping mode. The same

happens in case of pressure loss on the hydraulic line.

In addition to these two main operative modes, the Elevator presents a third particular **Recentering Mode**. In case of electrical failure, before switching the unit in Damping Mode, the Elevator needs to be recentered in its null position. In this case, the electric supply to the EHSV first stage is cut-off and, but the EHSV spool can be still recentered by its mechanical connection with the RD (15).

Elevator Recentering Device

The RD is a mechanical system which connects the servo-command piston, the EHSV and its position sensor RVDT. An extensive description of this device has been reported in [35], in particular its effect on the connected EHSV and its critical issues during maintenance procedures.

The Mechanical Lever (9) transform the linear movement of the main piston in a readable rotation by the RVDT sensor. This position transducer is connected, through a torsional spring (T1), with the RD Control Shaft (8). This main shaft,

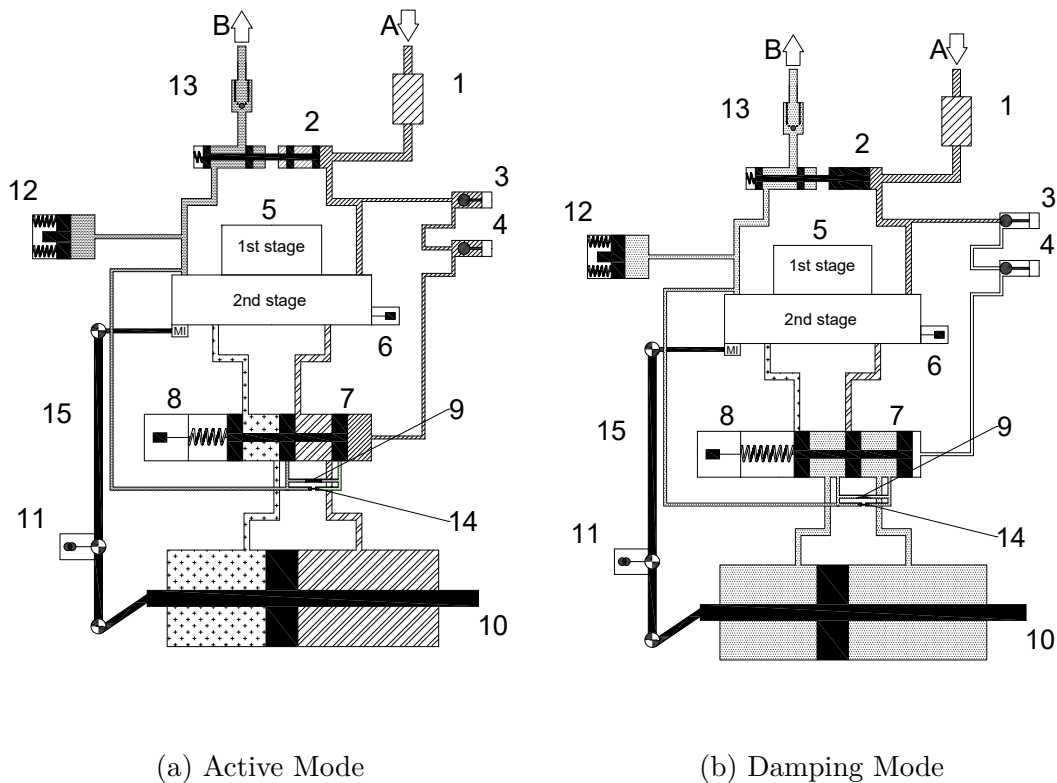


Figure 2.9: A320 Elevator Working Modes

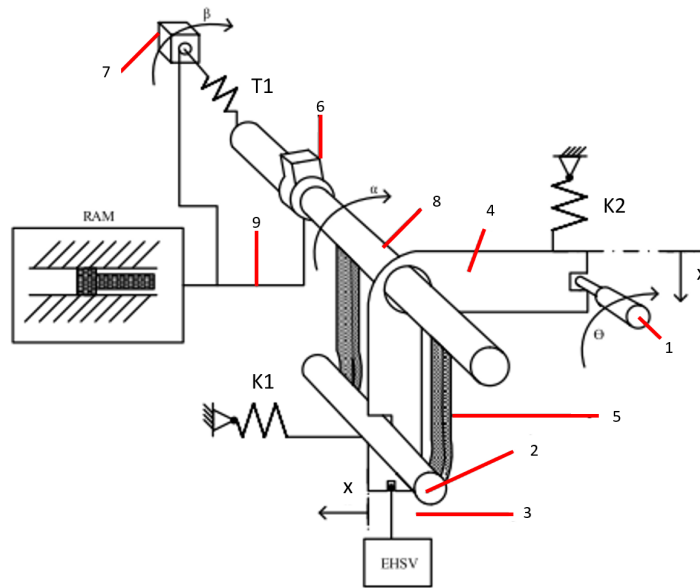


Figure 2.10: A320 RD Scheme [35]

visible in Fig. 2.8b, is actually connected to the Mechanical Lever as well through a particular eccentric (6), and internally to a second eccentric cam named Bell Crank (4). This component is directly linked to the Mechanical Input (3) of the EHSV, which can move directly the jet-deflector between first and second stage of the servo-valve. Bell crank rotation is limited by the Pin (2) and the Adjustable-stop (1): their positions influence the load of their respective Springs with different stiffness ($K1 > K2$). In particular, according to the Control Shaft rotation, the Pin (2) pushed by $K1$ can be in contact with the Hooks (5), or with the bottom part of the Bell crank, or with both. In this third case, the whole system behaves as a unique rigid body: a rotation of the Control shaft corresponds to a Bell crank oscillation, affecting the EHSV jet deflector position. This effect increases with Main Ram movements from its null position. When the Bell Crank reaches the Adjustable Stop (1), the counter-reaction of $K2$ stops any further rotation: in this condition, the control shaft can still coaxially rotate, but any input to the EHSV jet deflector remains constant.

Nominally, the RD force on the jet-deflector can be described with the green line in Fig. 2.11, depending of the Main ram position. The linear behaviour within the values x_{min} and x_{max} , correspond to the opposing force that the RD applies on the jet-deflector when the Main Ram starts moving away from its null position. When one of these positions has been reached, the Bell Crank has also finished its rotation on the RD pins (1) and (2). From these positions on, the RD effect is saturated and remains constant until the main ram end-strokes. If the RD force is within the

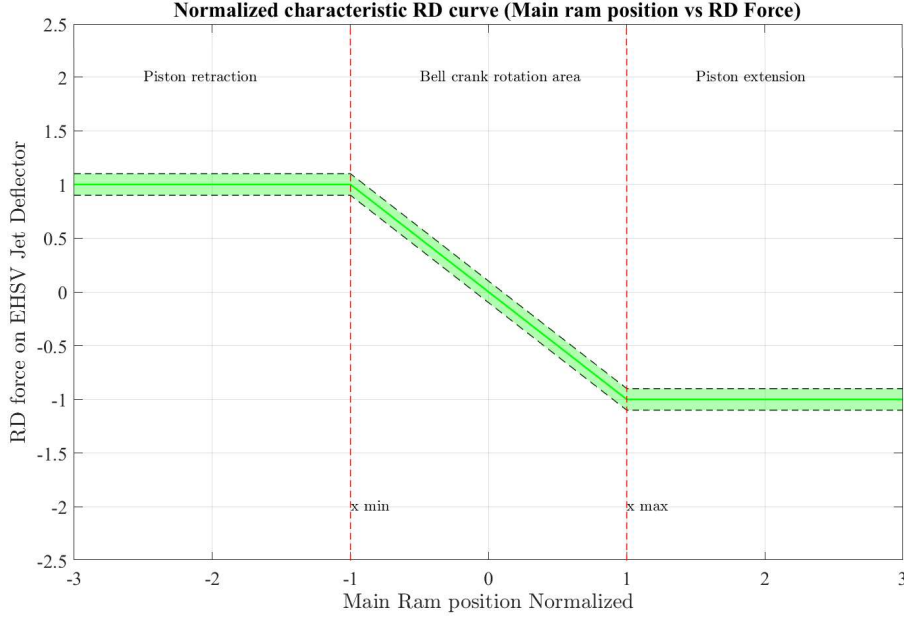


Figure 2.11: RD Force-Position characteristic curve

green limits band, it means that the mechanism is working properly.

During maintenance procedures, the RD force cannot be measured, but it is possible to adjust the Adjustable Pin (1) position and equivalently measuring the RD effect on the jet deflector, in terms of additional needed current by the EHSV Torque Motor to compensate this load on the jet-deflector (Fig. 2.12) The compensation current I_{RD} in [mA] can be calculated as function of the main position Pos in [mm] as the equation in 2.1. The value K_{RD} represents the ideal stiffness of the entire mechanism.

$$I_{RD} = \begin{cases} I_{min} & \text{if } Pos \leq Pos_{min} \\ K_{RD} \cdot Pos & \text{if } Pos_{min} < Pos < Pos_{max} \\ I_{max} & \text{if } Pos \geq Pos_{max} \end{cases} \quad (2.1)$$

Elevator EHSV Structure

In electro-hydraulic servo-systems EHSV represents the linking interface between control unit and hydraulic actuation system. It receives as input the electric signal from the external control loop, and proportionally produces a differential flow to the main cylinder chambers as output. All the A320 PFCS in this Chapter 2 are equipped with a jet-deflector two-stages EHSV. Similarly to the flapper-nozzle and

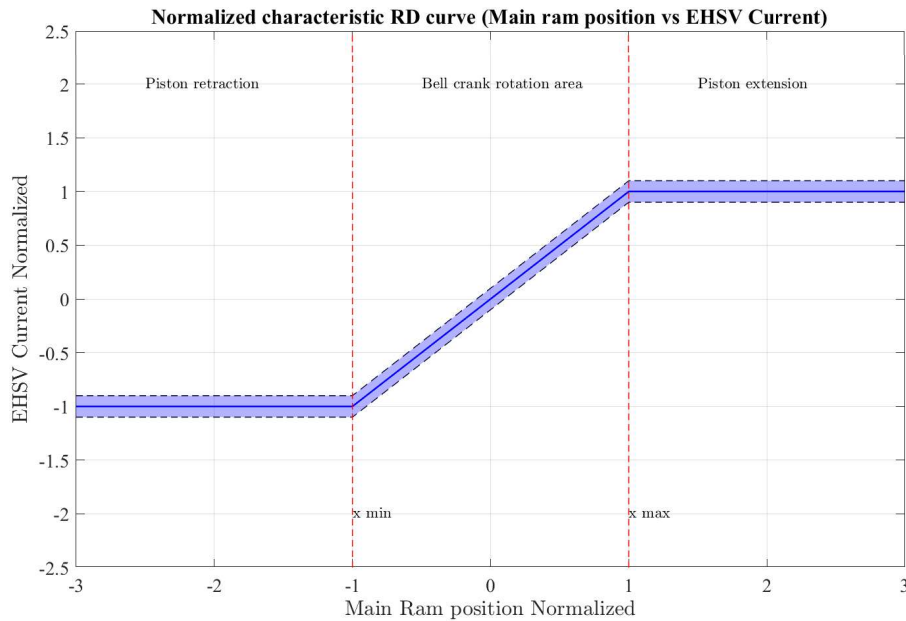


Figure 2.12: RD Current-Position characteristic curve

jet-pipe kinds, these valves are equipped with a first electric stage and a second hydraulic one. The main peculiarity is the presence of a fluidic amplifier between the two stages. Fig. 2.13 represents a simplified view of a typical jet-deflector EHSV. The upper part represents the electric first stage, containing a symmetric Torque-Motor TM. Two permanent magnets (1)(2) enclose an armature (3), where two symmetric coils (4) are wound. These coils receive the command current from the controller and develop a torque on the armature between the magnets [38]. To support and protect the first stage from fluid contamination, a flexure tube (5) separates and support the armature of the TM. The core part is represented by a small V-shaped fluidic amplifier (6) and its jet pan (7), which is connected to both supply pressure P (9) and return R. This small disc position (6) influence the fluid stream to the receivers (9). When magnetic force due to the coils current is disproportioned in the TM magnets air gaps, the armature swings to one side and moves the jet-deflector disc. This movement creates an imbalanced pressure difference between the spool chambers (10)(11). Under this pressure load, the spool (12) is free to move in its sleeve, connecting the ports C1 and C2 with supply and return pressure. The spool is directly linked, through a small ball-joint, to the jet-deflector with a feedback-spring (13). This elastic device applies a reacting force, proportional to the spool displacement, balancing the force applied by the armature (3). As already mentioned in Paragraph 2.3, the RD adds another force on the jet deflector, proportional to the piston position (Fig. 2.11). The mechanical

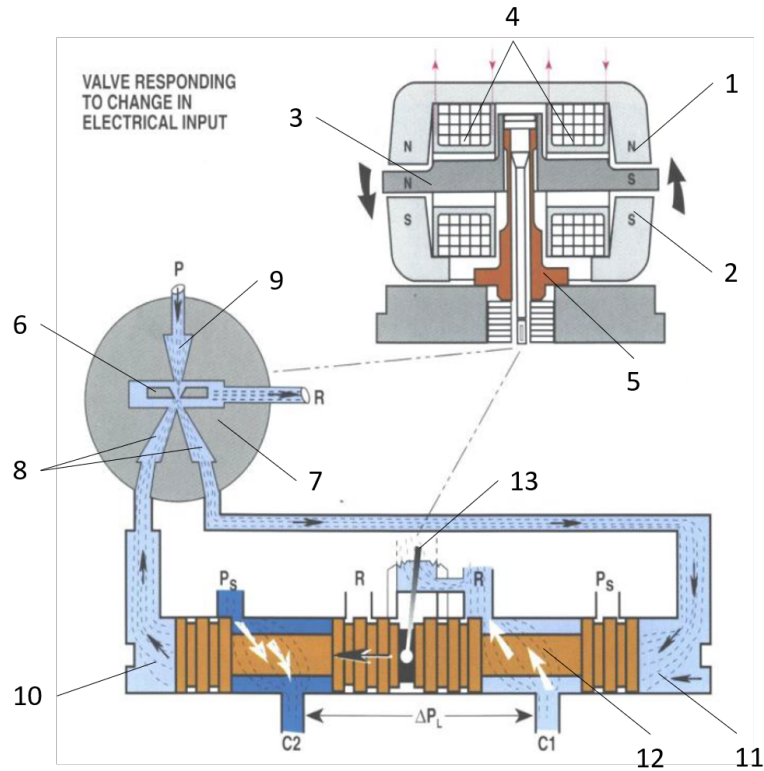


Figure 2.13: Scheme of a jet-deflector EHSV [rabie2009fluid]

input in Fig. 2.10 acts like a second feedback spring on the armature-jet deflector equilibrium.

2.4 A320 Aileron

On the tip of each wing, two Aileron PFCS are installed to control the roll rotation. In analogy with Paragraph. 2.3, Fig. 2.14 shows how these units are installed and supplied both hydraulically and electrically. In particular, both the ELACs share information with the SECs when spoilers are used for load alleviation issues during roll command.

Compared with the Elevator hydraulic scheme in Fig. 2.8, the Aileron structure in Fig. 2.15 is less complex and includes 11 main components (Table 2.3).

Comparing the hydraulic schemes of elevator and aileron, it is possible to observe how these units shares several components: both include a double-acting piston, a jet-deflector EHSV, a damping circuit with a MSV, feed valve and by-pass orifice, an Accumulator and a High-Low Pressure valves assy. By the way, they also differ in some particular aspects:

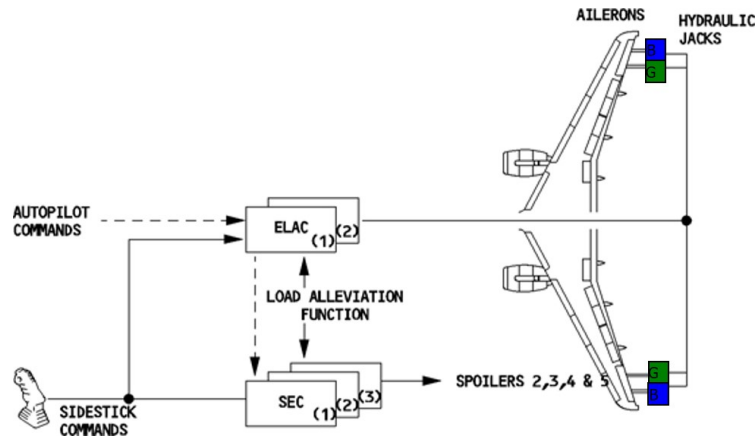


Figure 2.14: A320 Aileron Installation Scheme

Table 2.3: A320 Aileron Components

N.	Component	Abbreviation
1	High Pressure Filter	HP filter
2	HP-LP Poppets Valve assembly	HP-LP Poppets
3	Electro-Valve	EV1
4	Electro-Hydraulic Servovalve	EHSV
5	Mode Switching Valve	MSV
6	MSV sleeve LVDT	MSV Lvdt
7	By-pass orifice	
8	Main Ram	
9	Linear Variable Differential Transformer	LVDT
10	Accumulator	Acc
11	LP Maintaining Valve	
12	Feed Valve	FV

- The aileron include just one normally-opened EV. In contrast to the Elevator, when the EV is energized, it allows the supply flow to the MSV, keeping the unit in active mode;
- The main position sensor on the piston (8) is an internal 6 wires LVDT;
- This unit includes also a 4-wires LVDT (6) for checking the MSV position, but its EHSV (4) is not equipped with position sensor;
- The aileron does not include any mechanical RD. This means that in case of electrical supply failure, the Main ram is not mechanically recentered to its null position.

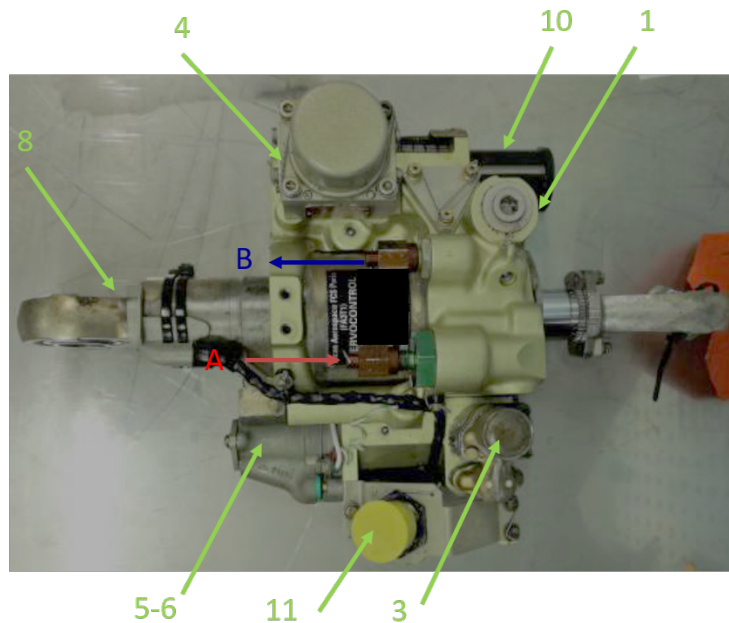


Figure 2.15: A320 Aileron Illustrated

The illustration in Fig. 2.15 points where all the visible components in the table 2.3 are located. In analogy with Fig. 2.8a and 2.8b, the socket 11 with yellow cap includes all the supply and measuring electric pin for both EV and MSV LvdT.

Two possible operating conditions of the aileron are:

- **Active Mode** Fig. 2.16a: as the elevator, the unit is normally supplied with pressurized Skydrol©IV through port A. Both the EV (3) and the EHSV (4) are electrically energized. Through the HP Filter (1), the flow pushes back the HP-LP poppet (2) block, and it is free to continue to the supply ports of the EHSV. Contemporary, since the solenoid of EV (3) is normally active, the MSV spool (5) compresses its reacting spring, opening the passages to the cylinder (8) ports. In this condition, the MSV LvdT (6) detects that the valve is opened. In this configuration, the main ram position is directly controlled by the flows from the EHSV outlet ports. The position-loop is then externally closed by its integrated LVDT (9). On the return line, the Accumulator (10) keeps enough pressure to open the LP valve (11) and discard the flow through B port.
- **Damping Mode** Fig. 2.16b: in the opposite way of the Elevator, the Aileron is switched to damping mode when the EV is de-energized. In this way, the supply pressure to the MSV is cut-off keeping it closed by its reacting spring. The actuators chambers are now connected through the damping restricto

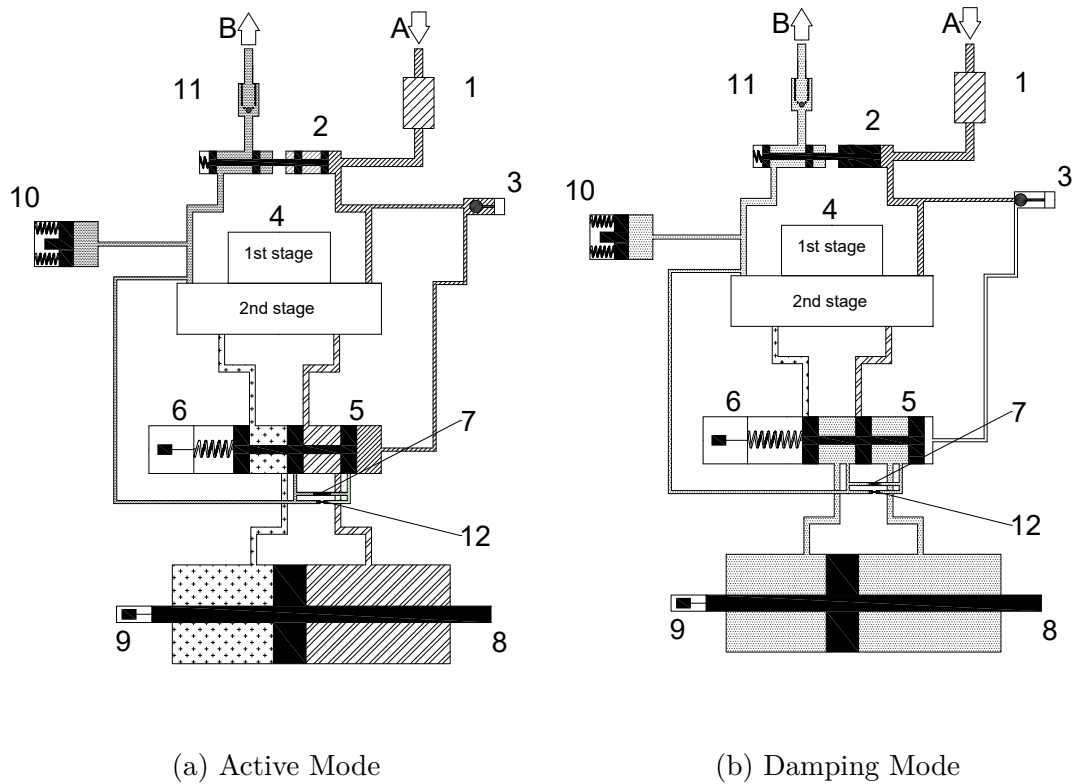


Figure 2.16: A320 Aileron Working Modes

(7), and the EHSV is hydraulically isolated. To compensate any leakage, the accumulator provides enough flow through the Feed valve (12).

An electrical failure in the supply line of the EHSV make the ELAC switching in damping mode, deactivating the EV. In case of electrical failure on EV instead, the unit can be set to damping mode hydraulically, shutting down its supply pressure.

2.5 A320 Spoiler

The upper surface of both the A320 wings includes five Spoiler actuators. They are used both to increase the descent rate, to improve the roll control with the ailerons and to increase drag force during landing manoeuvre. The scheme in Fig. 2.17 shows how these actuators are hydraulically and electrically supplied by their relative FCCs. In particular, since Spoilers are used for different purposes, they need to be activated accordingly during flight. For instance, Fig. 2.6 indicates how Spoilers n. 2, 3 and 4 may be used for both roll alleviation, speed-brakes and

ground-spoiler, while the fifth does not work as speed-brake while the first is used just during ground control manoeuvres. For this last particular application, the report [39] describes how the A320 adopts new SEC 120 standard to reduce the occurrences of hard-landing.

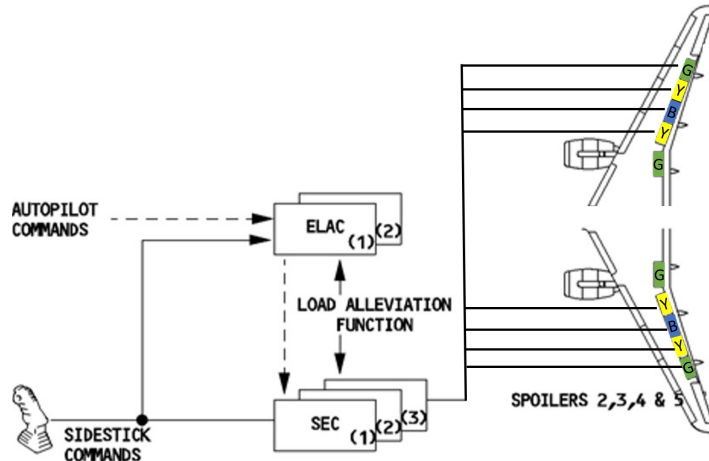


Figure 2.17: A320 Spoiler Installation Scheme

Compared with both Elevator and Aileron, the structure is less heterogeneous, but more hydraulically complex since its inner valves are highly integrated.

The list in Table 2.4 includes nine principal sub-components. The A320 Spoiler does not contain some of the most important parts of both Elevator and Aileron, such as an Accumulator on the return line, EVs for condition switching, a HP-LP valve assy between the supply and return lines, a feed valve for flow regulation during damping mode. The only included position sensor is the LVDT (9) on the main ram, which closes the feedback control branch. The working modes are

Table 2.4: A320 Spoiler Components

N.	Component	Abbreviation
1	High Pressure Filter	HP filter
2	Electro-Hydraulic Servovalve	EHSV
3	Mode Switching Valve	MSV
4	Inhibition Valve	IV
5	Integrated Valve Assy	
6	Main Ram	
7	LP Valve Check with Bypass	LPV
8	Maintenance Mode Lever	
9	Linear Variable Differential Transformer	LVDT

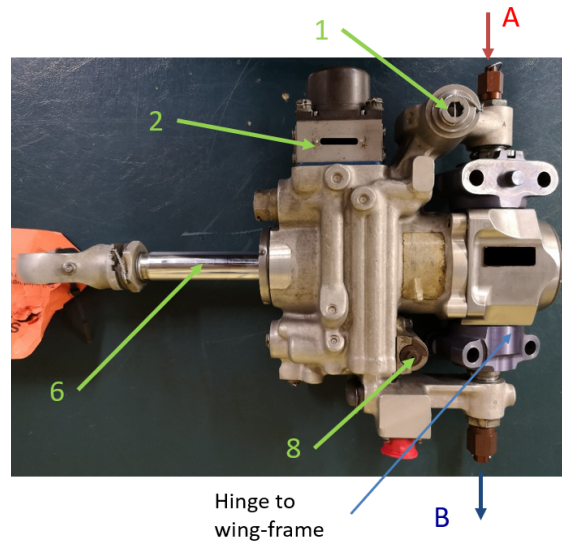


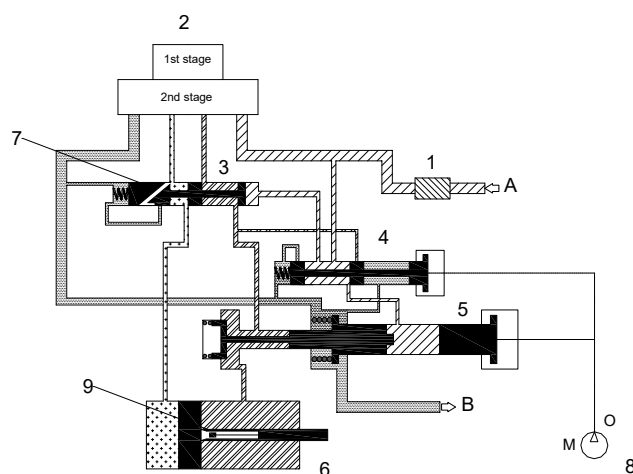
Figure 2.18: A320 Spoiler Illustrated

determined by the position of three main valves:

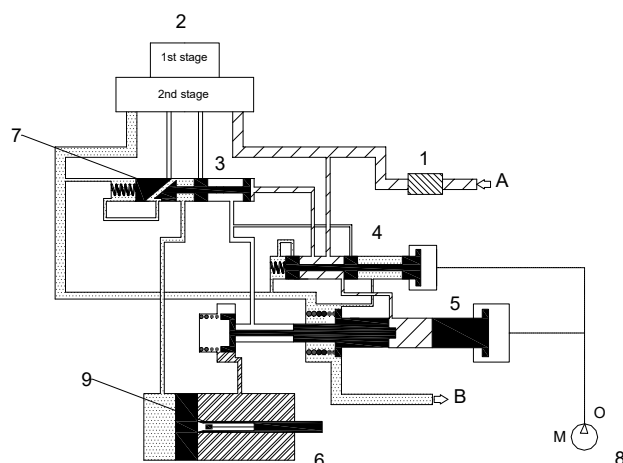
- MSV assy (3)-(7): compared to the Aileron and Elevator's one, this valve does not present any position sensor. By the way, it integrates the LP Valve Check and Bypass line (7), when its position is hydraulically switched during Maintenance and Locking mode.
- Inhibition Valve (4): it redirects the supply pressure to the MSV in Active mode. It can be manually activated by the Maintenance Mode Lever (8) during repairing tasks, short-cutting the high-pressure hydraulic circuit and inhibiting the EHSV command;
- Integrated Valve (5): in case of pressure loss, or maintenance operation, this complex set of valves isolates the cylinder chambers. In particular, its inner Anular Valve is closed in case of pressure loss (Locked Mode), avoiding any extension of the piston under aerodynamic load. In case of pressure spikes, its Overpressure tappet regulates the flow to cylinder chambers.

Fig. 2.18 indicates the few elements which are externally visible for this unit. The single effect piston (6) does not include a fixed-eye end to connect the unit to the wing-frame: Spoilers are hinged internally to the wings through two lateral attachments (in blue), which allow the unit structure to tilt while the piston is extending. As for the other units, Spoiler can work in two different modes:

- **Active Mode** (Fig. 2.19a): the unit is supplied through the port A and the return port B, since there is no accumulator. The Maintenance Lever



(a) Active Mode



(b) Locked Mode

Figure 2.19: A320 Aileron Working Modes

(8) is set to "O" (Operating). The supplied Skydrol©IV can flow through the HP filter (1) in all the HP circuit ducts. Both the IV (4) and Overpressure tappet - HP check in the (5) are supplied with high pressure flow, allowing both the MSV (3) to switch in its active position, and pushing the springs of the Anular valve in (5). This last valve opens the passage to the return line B. In this configuration, the command flows from the EHSV (2) are free to

move directly to the Main ram chambers (6). The position-loop of is then externally closed by its inner LVDT (9).

- **Locked - Maintenance Mode** (Fig. 2.19b): this configuration is triggered in case of pressure loss. In this working condition, the entire HP circuit is bypassed: both the IV position (4) and Integrated valve (5) change. In particular, the IV does not supply enough pressure to the MSV (3) which switches its position: the LP and bypass valve (7) connect directly the retraction chamber of the Main Ram (6) to the return line. Contemporary, the Anular valve springs in (5) close the way to the extension port of the cylinder. In this way, the piston movement are damped under the effect of the only aerodynamic forces on the surface and the unit cannot uncontrollably extend.

In case of electrical failure, the EHSV (2) is biased with enough current that allows the unit to retract completely, while the electric supply to the servovalve is disconnected. If the Maintenance Lever (8) is switched to "M" for overhaul procedures, the unit get completely depressurized since both the IV (4) and (5) are mechanically switched and the piston can be manually extended for inspection purposes.

2.6 Position Transducers

All the previously described units have one or more integrated Linear Variable Differential Transformers (LVDT) . These inductive position-to-electrical transducers return an output which is proportional to the position of its movable magnetic core [40]. Their reliability, dimensions and precision make the LVDTs the perfect candidates for several kind of applications, such as position monitoring of PFCS. The typical electric scheme in Fig. 2.20 shows three main components of these sensors. The magnetic core is free to linearly move within its cylindrical sleeve, which includes a central primary coil and two side secondary ones. As a typical transformer, the primary winding is connected to an AC voltage source. This principal coil induces an alternate voltage also on the secondary windings. The induction between the coils depends on the core position. Typically this component is threaded in order to connect it with the movable part whose position needs to be measured. According to the linked spool movement, the mutual inductance with one coil increases, while the opposite one decreases. This affects the difference of voltage amplitude on the secondaries. According to the module of this difference, the signal conditioning circuit (Demodulator) is able to extract both movement magnitude and direction.

Fig. 2.21 illustrates the nominal zeroth-order behaviour of an LVDT: the output voltage is linearly dependent to the input displacement. The secondary wirings are connected in series but in phase-opposition, so with 180° of phase shift between the

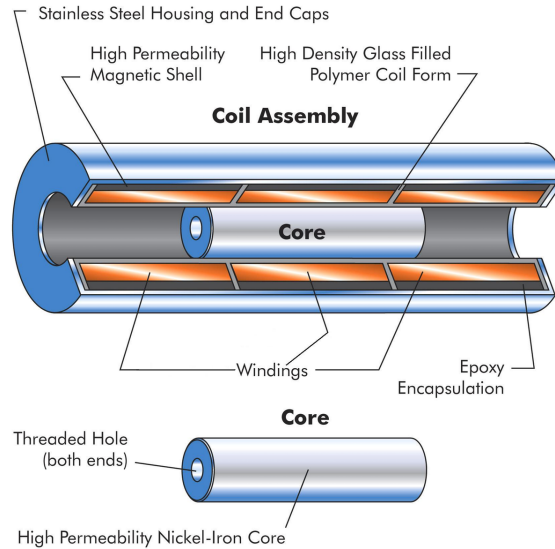


Figure 2.20: LVDT electric scheme [41]

secondary coils voltages. According to the core position, the Demodulator is able to extract both the direction according to the phase shift, and magnitude from the difference of amplitudes between the secondary voltages. When the core is its zero position, the induced voltage from the primary coil is equal on both the secondaries. This means that the LVDT output is zero, since the secondary coils compensate each other. A displacement in any direction determines a strong induction towards one of the two secondary windings, whose voltage amplitude increases accordingly, and an output signal which is in-phase or counter-phase with the primary. This coupling guarantees a linear behaviour and infinite resolution. However, when the core travels near its end-strokes, "less magnetic flux is coupled to the core from the primary" [42], reducing its linear behaviour.

As reported in [42], the main advantages of this transducer are its robustness, due to the absence of wear due to the lack of contact between core and windings, and its potential infinite resolution. In fact, even the smallest movement can be detected by a suitable signal conditioning circuit, which translates the voltage signal in position. Therefore, the LVDT resolution depends on the conditioner hardware characteristics. However, the LVDT presents its disadvantages and possible failures that need to be considered especially for diagnostic purposes, such as any biased-position of the core, temperature effect and magnetic distorted effects (in particular for Four-wires LVDT).

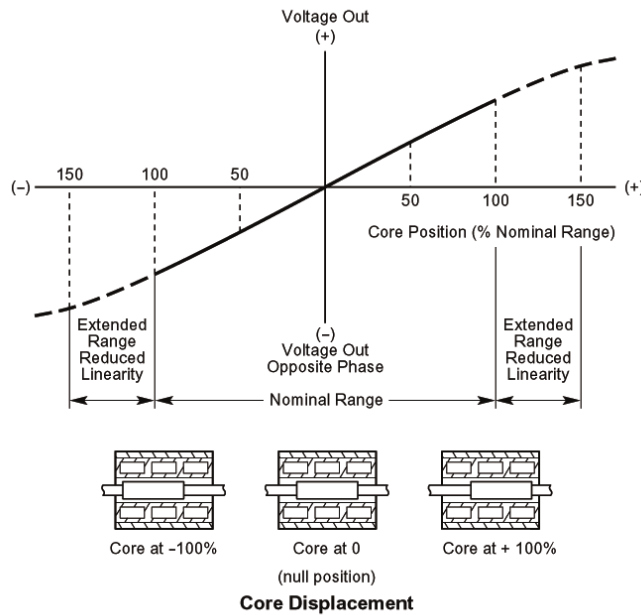


Figure 2.21: LVDT linear characteristic

2.6.1 Demodulation Techniques

The main working principle of the Demodulator conditioning circuit is to compare the phase and magnitude of the output LVDT signal with the voltage source connected to the primary. Typically, the carrier AC source frequency is at least 10 times higher than the expected maximum frequency of the core. In case of the PFCS, all the integrated LVDTs share the same primary voltage source at almost 2 KHz. The Demodulator configuration depends on the kind of connection used for the installed LVDT (paragraph 2.6.1). Typically, according to the secondary wirings connections, two possible configurations can be found (Fig. 2.22):

- Four-wires LVDT (Fig. 2.22a): the secondary coils are wired in order to output their voltage difference U_{ab} . On one hand, this configuration requires a simple signal conditioning circuit. On the other, these LVDTs are more affected by temperature and phase-lags between primary and secondaries.
- Six-wires LVDT (Fig. 2.22b): in this configuration, it is possible to read separately U_a and U_b . In terms of both temperature and phase-lag effects, this technique performs better if compared to the Four-wires one. However, both cost and complexity of the appropriate signal conditioning circuit are higher.

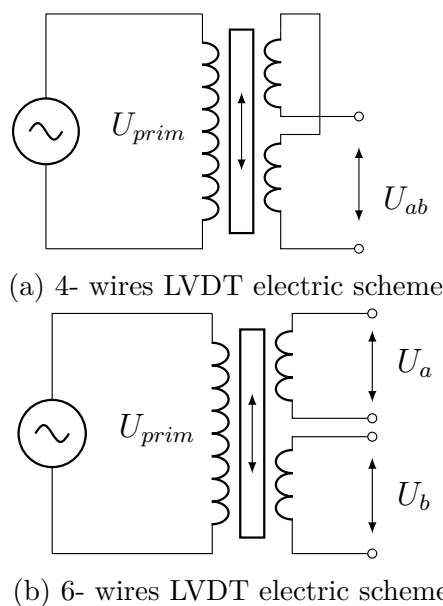


Figure 2.22: LVDT configurations

Both the demodulation techniques are in details described in the next paragraphs. The A320 PFCS are equipped with both Six-wires LVDT (or RVDT in case of the Elevator) as position sensor for their main piston, and Four-wires for the MSV (for both Aileorn and Elevator) and EHSV (just for the Elevator) spools.

Four-wires Demodulation

In order to extract both displacement and direction through a Four-wires LVDT, the output signal U_{ab} needs to be demodulated according to the primary excitation source. Several techniques have been designed for this purpose in order to reduce the aforementioned phase-lag and temperature effect ([43]-[44]). Some of these solutions regards novel circuit layouts such as in [45] and [46], or digital solutions in [47] or [48]. Two of the most popular techniques are:

- **Synchronous Demodulation**

Figure 2.23 illustrates the block diagram of a typical synchronous demodulator. Both the secondary coils output U_{ab} and primary voltage U_p are inputs of the signal conditioning circuit:

$$U_p = A_p \sin(2\pi f_c t) \quad (2.2)$$

$$U_s = A_s \sin(2\pi f_c t + \phi) \quad (2.3)$$

Two main steps characterize this technique: rectification and filtering. Firstly, the output from the secondary coils is rectified according to the sign of the

carrier signal $sign(U_p)$. In this way, the output signal U_{rect} is positive when two sinusoids are in phase, or negative if vice-versa. The rectified signal is sent through a low-pass filter (LPF), whose cut-off frequency needs to be tuned according to the displacement which need to be measured and the carrier frequency of the input signals. The LPF output represents the mean rectified curve which however is not equivalent to DC voltage. It can be easily demonstrated that the mean of the rectified signal is proportional to the amplitude of the sinus wave with a factor of $2/\pi$ (appendix A.1). By definition of rms value, if A_s is the amplitude of the secondary coil signal U_{ab} :

$$U_{rms} = \frac{A_s}{\sqrt{2}} = \frac{\pi}{2\sqrt{2}} U_{avg} \cos(\phi) \quad (2.4)$$

where in 2.4 A_{avg} is the amplitude of the filtered curve and ϕ is the phase shift between primary and secondary voltages. From the equivalent rms value, the demodulation gain K_{dem} returns the displacement. Many applications adopt this technique in both cases of analogue or digital demodulation for its simple implementation. For instance, both the test-benches in chapter 3 are equipped with analogue synchronous demodulation circuits for the Four-wires LVDTs. However, this technique is particularly sensitive to any phase-shift between primary and secondary coil, and any noise effect due to its odd-harmonics (appendix A.1).

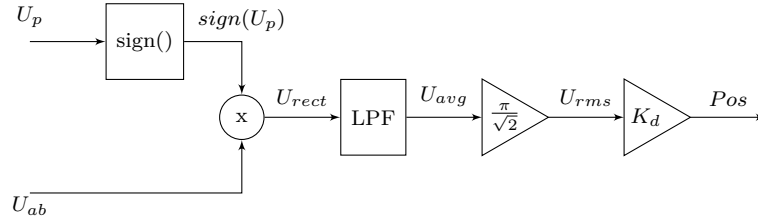


Figure 2.23: Synchronous Demodulator

- **Lock-in Amplifier**

As already reported in [43], a Lock-in Amplifier (LIA) is designed to extract a signal, with a known carrier frequency, from a noisy signal. Compared to the synchronous demodulator, on one hand it is more complex to implement both digitally or as hardware device, on the other it is less effected by any noise contribute while extracting amplitude and phase of the harmonic at the specified carrier frequency. As highlighted in Fig. 2.24, the input signal U_{ab} (eq. 2.3) is multiplied by an in-phase and quadrature sinusoids at the carrier frequency f_c to extract.

$$U_{ab} \sin(2\pi f_c t) = \frac{1}{2} A_s [\cos(\phi) - \cos(4\pi f_c + \phi)] \quad (2.5)$$

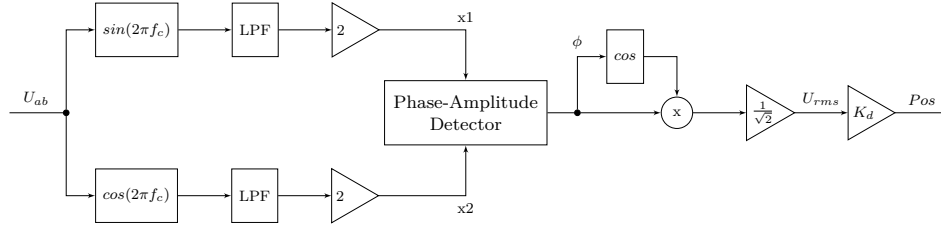


Figure 2.24: Lock In Amplifier Demodulator

$$U_{ab}\cos(2\pi f_c t) = \frac{1}{2}A_s[\sin(\phi) + \sin(4\pi f_c + \phi)] \quad (2.6)$$

Both the equations 2.5 and 2.6 includes two harmonics with twice the carrier frequency of the input signal in eq. 2.3. Both of them are removed adopting two LPF with order and cut-off frequency tuned according to the carrier frequency to extract. For instance, as suggested in [43], two second order LPF on both branches of Fig.2.24 would reduce the ripple to 1% when tuned to limit the bandwidth of the signal a decade below the reference carrier frequency. Once the filtered signals have been extracted and multiplied per 2, the results $x_1 = A_s\cos(\phi)$ and $x_2 = A_s\sin(\phi)$ are read by the Phase-Amplitude Detector module in order to evaluate:

$$\begin{cases} Ampl = \sqrt{x_1^2 + x_2^2} \\ \phi = atan\left(\frac{x_1}{x_2}\right) \end{cases} \quad (2.7)$$

Both $Ampl$ and $\cos(\phi)$ are then multiplied per $\frac{1}{\sqrt{2}}$ to obtain the rms value, as in eq. 2.4, and finally per the demodulation gain K_d to extract the final position. In [49] both the described techniques have been compared in case of nominal and noisy input signal U_{ab} , determining how the LIA technique is strongly less affected by noise level. However, due to the coherence with the hardware equipment of the testing consoles, the synchronous demodulation has been chosen to extract the signals from the Four-wires LVDT installed on the tested units. Furthermore, an anomalous level of noise may be used as health indicator for both these transducers and the connected spool.

Six-wires Demodulation

As previously anticipated in 2.6.1, the positions of all the main rod of the tested PFCS are measured by a Six-wires transducer. A particular mention is deserved by the Elevator, which is equipped with a Rotary Variable Differential Transformer (RVDT) (Fig. 2.25). Its working principle is analogous to a LVDT, with the only difference that the magnetic core rotates along the input shaft axis instead of move linearly.

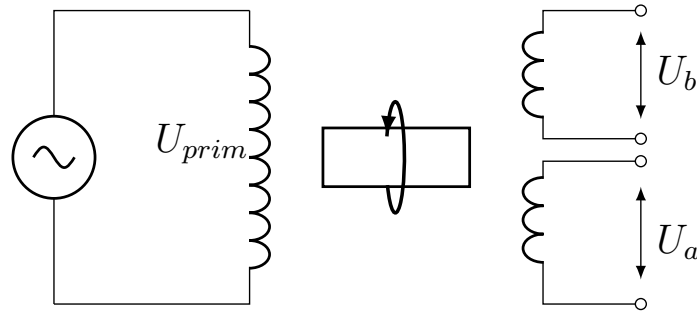


Figure 2.25: 6-wires RVDT electric scheme

The block diagram in Fig. 2.26 shows the steps which of a ratiometric demodulator. Both the secondary voltages U_a and U_b are firstly rectified, according to their own sign, then their rms values are extracted in the same way of the synchronous demodulator in Fig. 2.23. Both the values $U_{a_{rms}}$ and $U_{b_{rms}}$ have been calculated, the ratio R in eq. 2.8 is extracted:

$$R = \frac{U_{a_{rms}} - U_{b_{rms}}}{U_{a_{rms}} + U_{b_{rms}}} \quad (2.8)$$

In order to extract the correct position $Pos[mm]$, this value need to be multiplied per a fixed gain K_d . This value is reported in the sensor-spreadsheet and it is tuned according to the sensor nominal working condition. Through experimental results it is possible to check if this nominal value is still valid within a certain range: a percentage shift of K_d , evaluated in certain reference positions, can be a sign of a failure in the sensor.

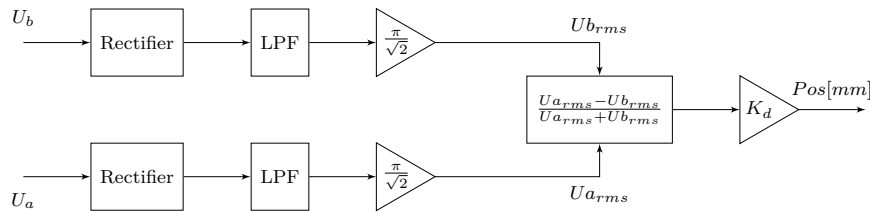


Figure 2.26: Ratiometric Demodulator

Chapter 3

Experimental Setup

The new maintenance protocol for the units in Chapter 2 has been firstly prototyped in hydraulic laboratory of the Innovation Components department (H-Lab), and already tested and validated in the test-consoles of the production Test-Field. These two different configurations present important differences both in their hydraulic schemes and testing consoles. However, the modular nature of the designed procedure makes it adaptable to different testing configurations. This chapter describes the main differences and common points of both the used test-consoles.

3.1 Prototyping Laboratory Test bench

The first prototyping and testing phase of the entire Hydiag Maintenance protocol has been performed in the hydraulic laboratory (H-Lab) of the Innovation Components department of the Lufthansa Technik facility in Hamburg. A schematic layout of the laboratory is presented in Fig. 3.1. Periodically, the Units Under Test (UUT) are delivered to the H-Lab from the Test-Field for research purposes or further investigation in case of hardly damaged units. Here the servocontrol is installed on a mechanical-fixture and connected both hydraulically to the Test Bench and electrically to the main Measurement System. This last component manages all the possible input to the UUT and collects the measured channels from the internal unit sensors. Since all the suitable control-laws are digitally implemented within the Measurement System, it is possible to command the UUT both in closed or opened loop, and eventually in active or damping mode giving the correct voltage to its electrovalves. Furthermore, a CAN bus module enables to command or collect measurements directly from the Test-Bench, such as supply oil pressure, return flow or temperature. All the input and output channels recorded by the Measurement Systems are available on the desk Workstation, where the user is able to manage the possible test through a

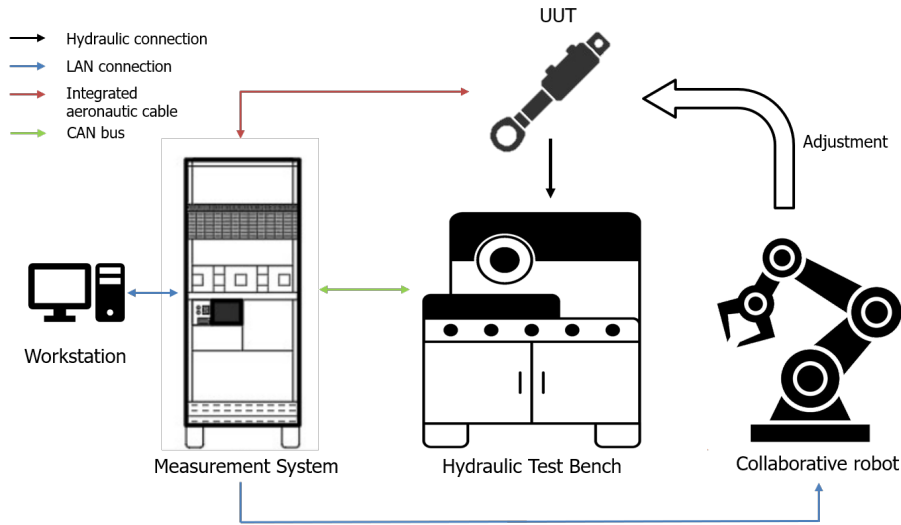


Figure 3.1: H-Lab configuration

dedicated GUI. Last but not least, the prototyping phase include also possible automatic adjustment performed by a Collaborative Robot. All the CoBot movements are implemented within its unit control but they can be coordinated by the user through the Measurement System with TCP/IP communication protocol. This last development aims to increase the safety level of testing procedure, since the robot may perform adjustment on the UUT during the testing phase itself, in a risky environment with high pressurized components.

3.1.1 Hydraulic Test Bench

The hydraulic scheme of the H-Lab Test-Bench is shown in Fig. 3.2. The oil tank on the bottom is filled with Skydrol HyJet IV at atmospheric atmosphere. It represents the only reservoir which both supplies the synthetic oil through the High-pressure circuit and collects the outlet flow from the return lines. The HP circuit starts with a fist Low Pressure pump (LP pump), with a maximum flow up to 140 l/min. When the test-bench is activated, the LP pump is directed through the main heat-exchanger, where it is cooled down by a counter-direction water flow. In order to prevent any thermal stress, an outlet thermometer is used to control that the cooled flow temperature is kept below 45°C. Afterwards, two High Pressure pumps (HP pump 1 and 2) are responsible to supply one of the two available supply ports (HP1 or HP2) with required pressure and flow rate to the connected UUT. Both the pumps are axial and pressure controlled up to 350 bar, while HP pump 1 can deliver flow rates up to 80 l/min, double the maximum achievable rate from HP pump 2. Furthermore, a small tank on HP pump 1 line is able to alleviate any flow oscillation. In case of A320 PFCS, both the ports HP1 and HP2 can be used.

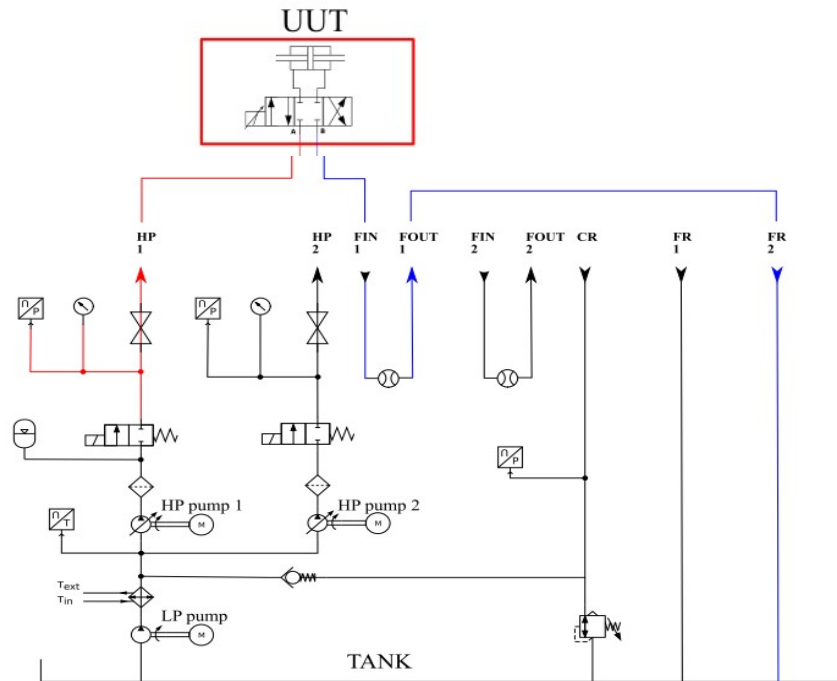


Figure 3.2: H-Lab Test Bench Hydraulic Scheme [32]

Both the high pressure lines can be activated or deactivated by two-way solenoid valves (V1 and V2).

On the return line, only one of the ports (CR) is pressurized up to 10 bar, while two Free Return ports (FR1 and FR2) may directly discharge the flow in the oil tank. If it is necessary to measure the supply or return flows, then the appropriate UUT hydraulic port need to be connected firstly to the flow-meter inlet hose (FIN 1 or 2) and then to its outlet port (FOUT 1 or 2).

All the test-bench working condition can be monitored from the Work Station, since both temperature, supply and return pressures and flows can be transferred through CAN bus communication to the Measurement system. This is an important feature in terms of test automation, since this makes the test-bench parameters controllable. For instance, it has been estimated that controlling the supply pressure via designed signal would shorten the Hydiag procedure of more than half. An higher level of automation means also increasing and improving the testing safety level. The testing chamber is fully closed by blast-proof glass doors, and the user controls all the testing variables not only from the desk workstation, but also from a touchscreen panel directly connected to the bench. Furthermore, using a CoBot for automatic adjustments during test prevents any human interaction with the UUT in highly pressurized environment.

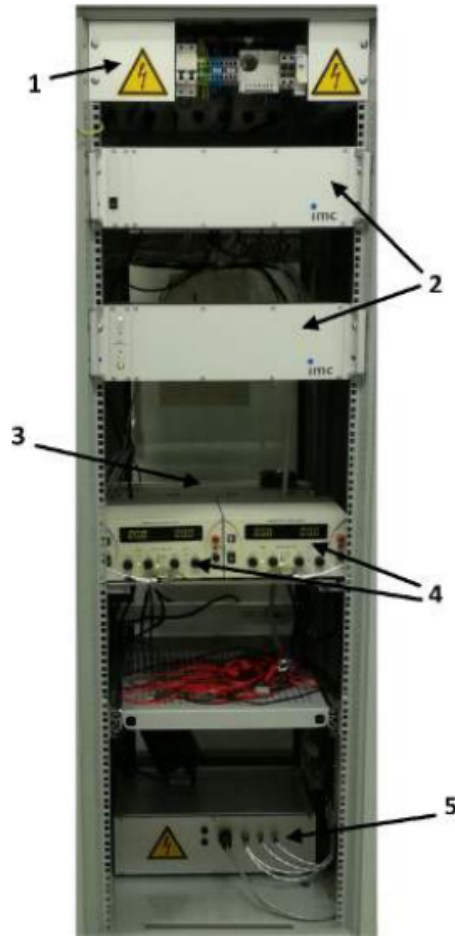


Figure 3.3: H-Lab Measurement System [32]

3.1.2 Measurement System

The H-Lab Measurement System in Fig. 3.3 provides all the necessary inputs to the UUT, acquires all the signals from both the Test Bench and the internal UUT sensors, and transmits them to the user Workstation in real-time.

The rack is composed by five principal components:

- Safety circuit and Thermostat (1): on the top of the rack, they protect the other connected units from eventual overheating and current spikes, through calibrated safety relays;
- Two Data Acquisition and Control (DAC) devices (2): they represent the core part of the real-time measurement system. They are responsible for the data-stream in all the analogic and digital channels which can be controlled during a test. Furthermore, the control loop logic to command the UUT are

digitally implemented within the *Online Famos* code, which is read in real time by the DAC that calculates and provides the correct electric command signal.

- U/I converter (3): this component converts the voltage signal from the DAC in direct command to the EHSV of the tested unit, in a settable range of ± 10 [mA], ± 50 [mA] or ± 80 [mA].
- Two additional Power Supplies (4): they are dedicated to the voltages of the present UUT electrovalves, in order to switch the unit in active or damping mode (used only in case of Elevator or Aileron servocontrol)
- Main electrical supply (5): power provider to all the components of the rack with available voltage ranges of $0/+12$ [VDC], ± 15 [VDC] and ± 24 [VDC].

As highlighted in Fig. 3.1 all the devices of the H-Lab are plugged to the Measurement system. The UUT integrated electrical plugs are connected via the same Skydrol resistant aeronautic cable which is used to connect onboard the servo-actuators with the dedicated FCC. As referred in section 3.1.1, a CAN bus communication is used between test-bench and measurement. A LAN connection allows the Measurement System to control the sequence of movements of the collaborative robot during an automatic adjustment, and to report the acquired data to the user Workstation. Here, the online code in *Online Famos* testing language is the core software part which establishes the steps of the complete Hydiag procedure.

3.1.3 Testing Software

The testing software provided by *IMC Studio* can be divided in two main parts:

1. GUI for channels setting and user-defined panel design;
2. Real time code in *Online Famos* language for channels acquisition.

The first part requires to set up all the channels, which can be classified in the following groups:

- Analog inputs: they include all the channels from the integrated UUT sensors, such as primary and secondary voltages from the LVDT or RVDT position sensors, the output command current from the U/I converter and activation EV voltages. These channels can be sampled up to maximum 50 kHz.
- CAN bus analog inputs: this class includes all the signals from the hydraulic bench described in paragraph 3.1.1;

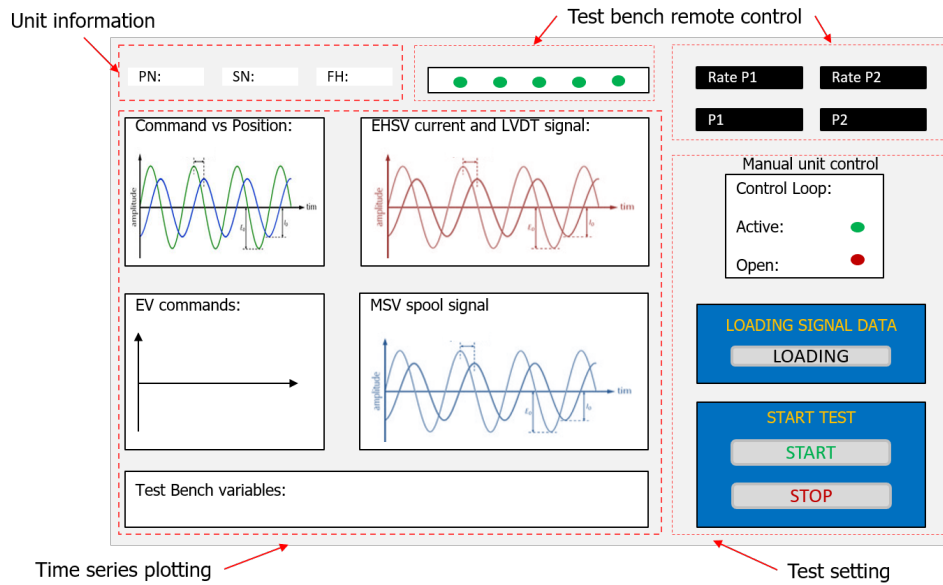


Figure 3.4: Workstation GUI panel

- Digital inputs: these channels may include eventual position sensors installed on the test-bench (such as external INC sensor) and the joints positions of the collaborative robots during the adjustment;
- Analog outputs: this group includes all the command signals which have to be provided to the UUT, such as command position and EVs voltages.
- Digital outputs: typically these channels are used to remotely control the Test bench parameters or manually command the unit if the Hydiag test is not running.

All these channels are managed by the real-time software in *Online Famos*. The DAC in Fig. 3.3 is able to elaborate all the information through the channels and assign them to a physical voltage output. The real-time code instead is able to assign values at a sample-rate of 100kHz to the defined channels using three different kind of variables:

- Process Variables PV : once a channel is defined, a PV is associated to the channel and it can be exported as collected measurement. If the variable is user-defined but not associated with any channel, then *Online Famos* code can elaborate its value but it cannot store and export.
- Display Variables DV : these user-defined quantities can be both set in the real-time code or through the panel GUI. Their scalar values can be exported or used to switch between different cases in the real-time code.

- Virtual Channels VC : these additional channels are created exploiting the acquisition characteristic of a specific physical one, and they are useful in case of visualization in the user GUI and they can be exported as the PV.

Fig. 3.4 presents a schematic view of the GUI interface in IMC Studio. The main panel presents four main sections:

- Unit information: it contains three main type-in boxes in which the UUT information can be reported and exported as DVs;
- Test Bench remote control: this section contains all the buttons to enable the remote control of the test bench and manually setting the nominal supply pressures on HP port 1 or 2.
- Test setting: these panels includes all the buttons which can be used to start a manual test (choosing the kind of input signal in active or open loop), or the complete HyDiag test. Before clicking the Start button, it is necessary to load the proper HyDiag signals.
- Time series plotting: this section includes all the possible plots from analogue or virtual channels that can be exported at when the test is over.

3.1.4 Collaborative Robot

In Fig. 3.1 the complete configuration of the H-Lab includes a Collaborative Robot (CoBot) to perform automatic adjustment during the Hydiag test. Due to its limited speed, payload, forces and joints torque, a UR10 from Universal Robot has been chosen to share the workspace with technician during the automatic test procedure. The aforementioned advantages are not the only ones [50]:

- Friendly programming user-interface for movement commands through a dedicated teach-panel;
- Installation flexibility over different surfaces;
- Compatibility with external Measurement System due to its TCP/IP or Modbus communication protocol;
- Good repeatability error of ± 0.1 mm. This value ensures enough precision during the adjustment.
- Due to its interactive nature, it is possible to set forces, torques and speed limits and modify them during each test;
- Protective stop trigger in case of collision;

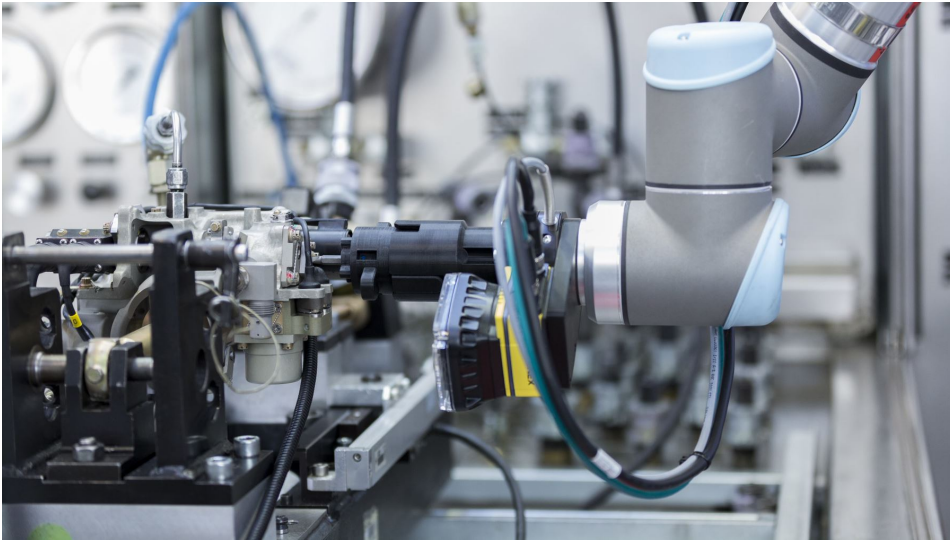


Figure 3.5: UR10 Robotic Arm for the 31075 Elevator adjustment. Courtesy of Lufthansa Technik: https://www.lufthansa-technik.com/press-releases/-/asset_publisher/Xix57wMv0mow/content/press-release-hydiag

- Possibility to install different devices on the hand-effector.

Fig. 3.5 shows a close-up picture of the last wrist of the UR10 during and adjustment on an Elevator servosystem. As it is possible to observe in the figure, the actual configuration of the UR10 is equipped with several three main devices:

- Camera adaptor: the robot is able to evaluate its trajectory waypoints through an industrial camera;
- Force and torque sensor: this device ensures to control the torquing force during the adjustment;
- Tool holder and Tool: both in 3D printed, the robot is able to select the necessary tool to perform the adjustment, magnetically coupling its fixed Tool holder with the adequate Tool.

The software which regulates the robotic arm movements is split in two main parts. The functions which define each joint movement (in terms of position and speed) are stored in its dedicated Control Unit, external to the Measurement System. The complete sequence of movements to perform is managed by the Measurement System via TCP/IP protocol. This ensures also to define collect some important measurements from the robot itself, such as the angular position and torque time-series during the automatic adjustment. This feature allows not only to automate the entire process, but also to analyse data which were not available otherwise and possibly detect new health-indexes of the UUT.

3.2 Test Field Test bench

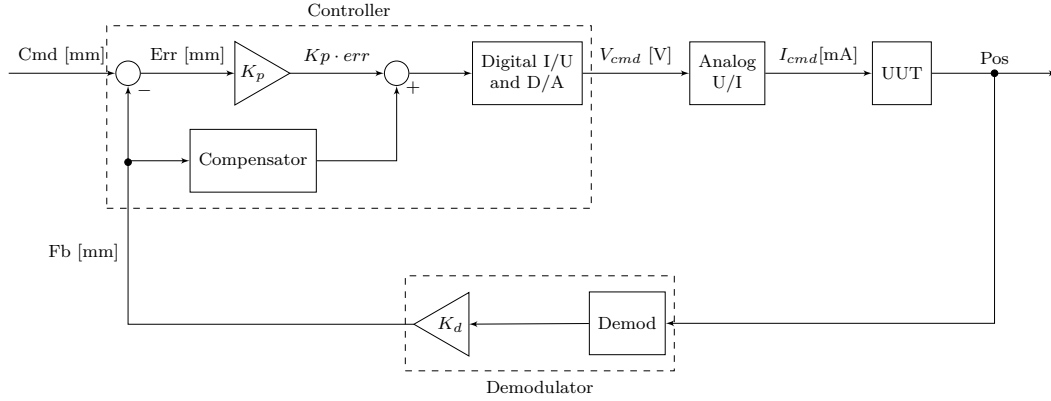
As referred in section 3.1, the first prototyping of the new Hydiag protocol has been developed in the H-Lab facility. During this phase, the complete schedule of operations to be performed on the UUT has been designed: from the automatic testing to the final robotic adjustments. In particular, the first automatic testing step has been deployed also in the production-site, in order to validate and adapt the prototyped signal on the Test Field benches. The technical differences between these testing-frameworks have lead to a consistent adaptation of the prototyped signal in order to be supported by both the Test-Field hydraulic bench and measurement system:

- Compared to H-Lab test bench, the Test-Field one does not allow to regulate remotely supply and return pressure. In addition, it does not digitally record both the inlet and outlet flows. These technical limitations have stretched the signal in time, since the technician has to manually regulate the supply pressure and report the flow values when required.
- The NED Test-Field measurement system has an integrated working station and a different test-software compared to the H-Lab one. The *LabView* interface of the NED needs different specification to run the complete Hydiag Sequence. These requirements are in detail explained in paragraph 4.4.

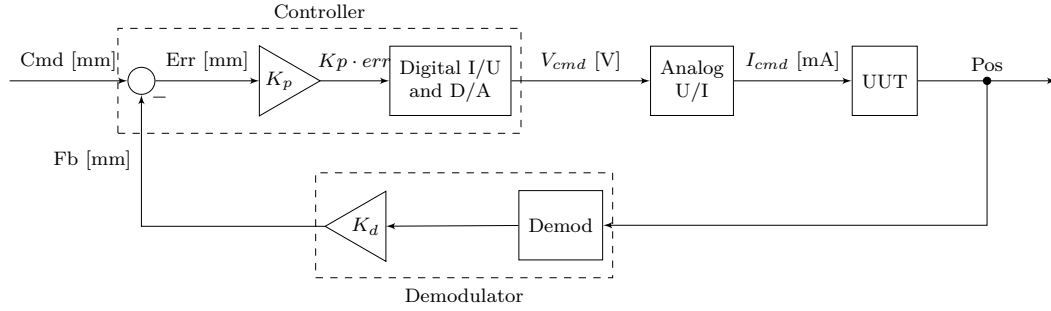
Beside the hardware and software differences, both the measurements systems adopt digital control loops which can be selected according to the UUT. This feature allows the user to select between closed and open loop configuration, and easily tune the controller parameters in case of any specification from the OEM. The following paragraph reports the proportional scheme for both A320 Elevator, Aileron and Spoiler.

3.2.1 Digital control loops

The block diagram of the Elevator control loop is reported in Fig.3.6a. The upper dashed area includes all the main logic blocks which are implemented within the digital proportional controller. In closed loop configuration, the input position command Cmd in millimetre is compared with the feedback Fb signal coming from the lower closing branch. Both the signals are compared in order to evaluate the position error signal Err which is then amplified according to the proportional gain K_P . The value of this gain is directly specified by the OEM and it is the same used by the FCC on board. The peculiarity of the Elevator proportional control loop is the presence of a Compensator, due to the RD action on the EHSV of the unit. As already anticipated in 2.3), the RD effect on the first-stage jet-deflector is linear close to the zero, while it saturates after a certain reference



(a) A320 elevator control loop



(b) A320 aileron and spoiler control loop

position, as in Fig.2.12. According to the piston position, the equivalent RD current need to be compensated by the Compensator. The equivalent current value is firstly converted in voltage and then the signal transformed in analogue, in order to be read by the external U/I converter. This block is tuned in order to counter-convert the voltage from the Digital I/U. The equivalent command signal I in milli-Ampere represents the input to the UUT servo-valve. According to the input signal, the EHSV regulates the flows in the UUT cylinder chambers, determining the piston position. The output signal Pos from the integrated RVDT needs to be interpreted and digitally converted by the feedback Demodulator. This component is not integrated in within the units, but it is analogically tuned in order to perform the ratiometric demodulation (as in 2.6.1), and digitally convert the signal in order to close the loop. It is interesting to notice that the nominal value of K_d needs to be compared with the actual one that can be calculated by the RVDT demodulation. This can be considered as an important health index to diagnose any failure in the piston position sensor.

For what concerns both the Aileron and Spoiler, they requires a simple proportional control loop that can be represented with the scheme in Fig. 3.6b. Beside the absence of the RD compensator, and different piston position sensors, all the other blocks of the digital control loop work in the same way of the Elevator one. This

remarks the advantage of adopting a digital control loop. Once the UUT is installed on the test-bench, the measurement software uploads the corresponding controller settings according to the unit PN. Each setting includes both the gain values and specific blocks to be activated, such as the Compensator in Fig. 3.6a.

Chapter 4

Automatic Entry Test

The first step of an Advanced Diagnostic procedure for all the PFCS in Chapter 2 consists in designing a testing procedure which is automatized, modular and scalable for all the units. In order to fulfil these requirements, new testing signals have been designed for all the Hydiag Project use-cases, which represent the first important stage of the Automatic Entry Test protocol. The first and second paragraphs of this chapter highlight the differences between the traditional maintenance procedure, proposed by the OEM, and the automatic one by the Hydiag Project. After this comparison, the structure of each signal will be presented, highlighting the designing concept.

4.1 Traditional OEM Maintenance Protocol

Maintenance procedures of PFCS are Reactive, based on the paradigm "repair on duty". In fact, if the real-time monitoring FCC detects an anomaly, it deactivates both the electric and hydraulic supply to the failed unit, which is disembarked when the aircraft is on ground for overhaul. This lead to high cost in terms of aircraft time-on-ground and maintenance procedures for MRO companies. The diagram in Fig. 4.1 explains the standard repairing process for testing and re-certifying a faulty PFCS in the LHT workshops.

When the aircraft is on the ground, line maintenance personnel performs a routine ground-checks: the Initiated Built-In Test (IBIT) . Typically, during these checks, the movements of each surface are checked by the monitoring part of each dedicated FCC, which checks both hydraulic and electrical nominal parameters. If all the IBIT tests are successful, the unit is still serviceable and remains on board. While on board, the real-time monitoring FCCs, which continually checks if the recorded feedback signals from each unit are still within the nominal working ranges. In case of anomaly, the faulty unit is immediately switched in damping

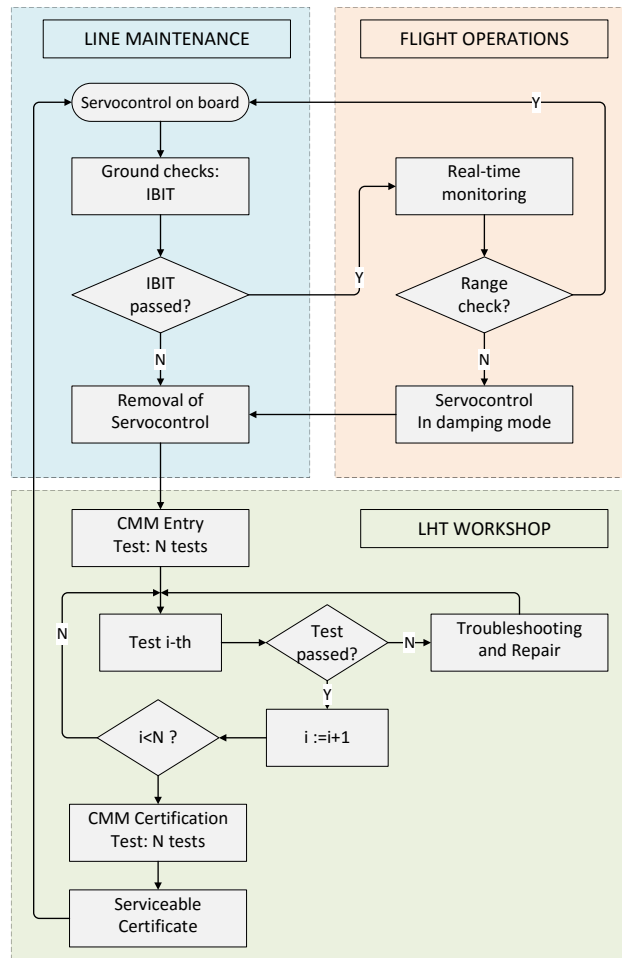


Figure 4.1: Standard Maintenance Protocol [32]

or locked mode, while the redundant hydraulic and electric architectures (Chapter 2) guarantee to safely proceed until the component get disembarked. Once in the LHT test-field, the faulty unit can be tested according to the Component Maintenance Manual (CMM) procedure, in order to identify the root of failure. This official document is released by the OEM and it contains a detailed description of each sub-component, its specifications and specific repairing instructions. Once installed on a certified test-rig, the standard CMM test is carried out by a specialized technician. Each CMM instruction targets a specific set of failure for each subcomponent and it needs to be manually performed. The operator supplies the Unit Under Test (UUT) with defined closed-loop position or open-loop current commands, and manual adjustment are in the meantime performed if needed. The

results from each measurement, the Health Features (HFs) , are then compared with the CMM limits: all the HFs out of these ranges can point to the failed UUT sub-component. Per each of these anomalous HFs, the CMM provides specific troubleshooting instructions which describe the sub-components parts to be repaired or replaced. Once all the supposedly faulty piece-parts have been exchanged, the repaired UUT needs to be re-tested, until all the HFs results within their ranges and no additional repairing is necessary. At this point, the unit can be labelled as Recertified, and it can be installed back on the aircraft.

Even if this entire process in Fig 4.1 is strongly reliable, especially in terms of unit recertification, it can be improved in some aspects:

- The complete procedure is resource and time expensive. Considering both the testing and repairing phases, a complete test-repair cycle can last more than two hours. Furthermore, the complete CMM testing phase needs to be repeated two or more times according to the failure isolation effectiveness during the initial Entry Test.
- Since measurements and adjustments are performed manually, the human factor has a high impact during the HFs extraction from each test. This can influence further comparison or analysis of similar test results.
- Each extracted HF is “isolated” from the others. This procedure does not allow the technician to detect correlations between the tests. Additionally, an accurate signal processing of the measurements can increase the number of condition indicators.
- The CMM test outputs are binary: a test can be Failed or Passed according to comparison of the extracted HF with its acceptable ranges. This can lead to a punctual description of the UUT health status, but usually does not guarantee the component’s enduring operative life. A unit may be sent back to the shop after few flight hours, since a previous unscreened degradation has not been detected. It would be more effective to identify an optimal range of HFs limits which can ensure a certain operative life to the repaired unit.

4.2 Automatic Procedure

In order to solve the standard procedure limitations in Paragraph 4.1, it is necessary to automate and scale the testing procedure for all the families of A320 PFCS. As already depicted in Paragraph 1.2, the first important block defined by the Hydiag Project is the implementation of an Automatic Entry Test for fast and reliable failure identification (Fig. 4.2). During this test, the UUT is stimulated by a structured and defined excitation signal, composed of different consecutive *Sequences*. Each of these signal portions includes all the necessary set of command

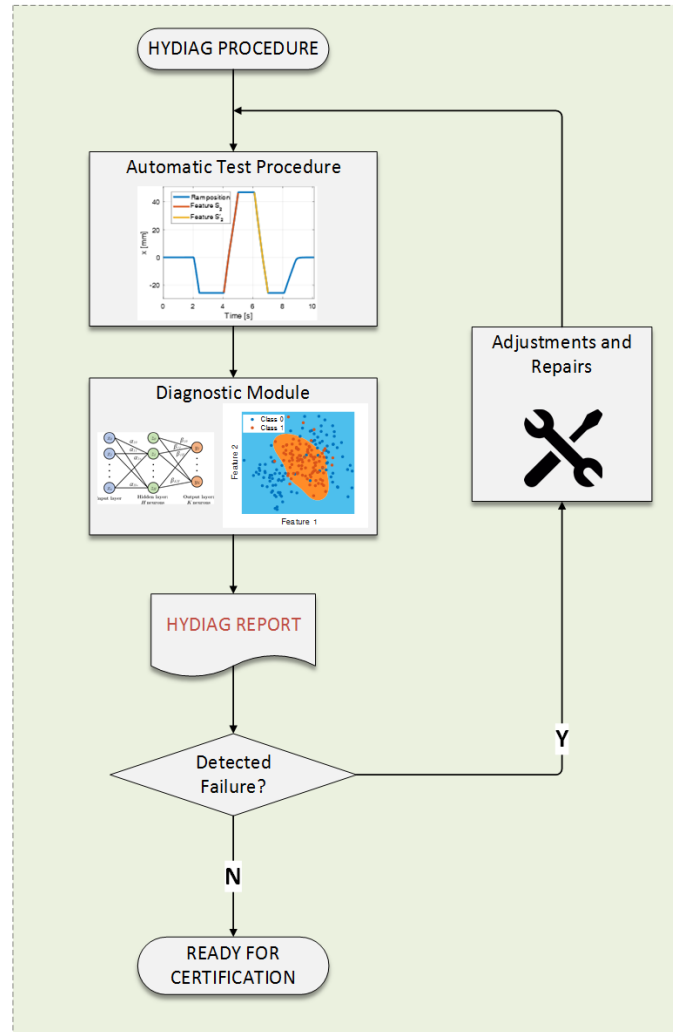


Figure 4.2: HyDiag Testing Protocol

to both the UUT - such as main ram position or EV voltages - and to the Test-rig - for example, fixing supply and return pressures. Each *Sequence* is characterized by a proper shape and it is fully parametrized according to the type of UUT. This property simplifies the comparison of different units responses from the same actuator category to the same signal.

All the collected measurements from each *Sequence* are post-processed in order to extract its distinguishing HFs. The complete set of HFs is composed of both traditional CMM results and additional parameters which improve eventual incipient degradations. The Diagnostic Module goal in Fig. 4.2 is to determine whether the UUT presents failures or incipient faults and in which sub-components present degradations. For all the traditional CMM HFs, each failure is diagnosed comparing the extracted results with their corresponding acceptable ranges. The Diagnostic

Module integrates this standard procedure with an innovative *Failure Classifier*, based on Machine Learning techniques, which exploit the complete set of CMM and additional HFs improving the failure detection accuracy. This method main advantage is to consider the HFs set as a global set of data, including correlations between the extracted parameters. In Fig. 4.2 the first two steps need to be repeated in loop until all the complete test is passed and all the degraded components have been repaired. To conclude, the automatic procedure improvements are:

- Time and resource savings: compared to standard CMM procedure, the first automatic testing drastically reduces the average measurement time of more than half, covering most of the critical results of the standard CMM procedure.
- Standardization: a fully scalable test and post-processing method improves reliability and repeatability of HFs extraction, simplifying further comparison between similar tested units.
- Reliable data-collection: the HyDiag protocol allows a complete collection and structured storing of both Diagnostic reports and time-series measurements from all the channels.
- Additional HFs: the automatic test provides a comprehensive overview of the UUT health status, combining both traditional with innovative HFs. Correlations between extracted results constitute a key-point for an improved failure and faults identification.

4.3 Modular Signal Generation

As already mentioned in Paragraph 4.2, the *Sequences* have the goal to extract in a unique and automated way the most critical HFs, according to the CMM of the relative unit, plus some additional Extra-CMM HFs for improved health check-up of the UUT. Since the servoactuators presented in Chapter 2 share several subcomponents, the *Sequences* shapes have to similar across the units. In this way, the same parametrized signal can be used to detect the same kind of failure for different types of PFCS. The strategy to build a modular Automatic Test for all the units can be summarized in the following points:

1. Analysis of the EHSA structure: identify the main sub-components which are shared among different actuators. For instance, since all the units can work in both Active and Damping mode, the MSV, EHSV and the Main Ram are three key components in the unit structures.
2. Identify the most critical CMM test. All the collected testing reports from the standard procedure in Fig. 4.1 are a precious source of information. In

fact, once structured in a unique database, they can be used to identify the most recurrent failed CMM tests for each type of unit. These tests need to be included in the Automatic procedure in order to keep high-reliability standards. A first estimator of test criticality is the ratio in Eq. 4.1:

$$w_i = 100 * \frac{Failed_i}{Tot_i} \quad (4.1)$$

with $i = 1, \dots, N_{test_{CMM}}$. The *weight* parameter w_i represents the percentage ration between number of recorded failed tests over the total number of records per each CMM test.

3. Check if the CMM tests are characterized by a proper signal. In order to keep the test automatic, it is necessary to take as reference the suggested signals from the manual. If the signal specifications are not complete, then a new reliable shape has to be defined which allows to achieve the same HF extraction as the standard one.
4. Try to integrate similar tests in one *Sequence*. Since the Automatic procedure needs to be reliable but also time-saving, if two or more tests are characterized by the same signal-shape, they can be collected in one *Sequence*. In this way, more than one HFs can be extracted from the same measurement.
5. Add specific *Sequence* according to the tested PFCS structure. During the procedure design process, it is important to identify the common parts of different PFCS, but also respect each structural peculiarity. For instance, the Elevator is characterized by its RD and it integrates more position sensors than the others, while the Spoiler is almost fully hydraulically controlled and its only integrated sensor is the main ram LVDT. Each of these characteristics can be a specific failure root that needs to be diagnosed with a tailored *Sequence*
6. Concatenate the *Sequences* with similar aspects: in particular, the complete excitation signal can be divided in four main sections. According to the test-rig characteristics these groups may need a manual regulation by the technician. As it will be in detail described in Chapter 4, the workshops test-benches differ from the prototyping H-Labor one. These differences influence the duration of each *Sequences* group but not their signal shapes.
 - (a) Preliminary;
 - (b) Pressure Adjustment: this section includes all the *Sequences* where supply and return pressures of the test-rig need to be regulated;
 - (c) Leakages *Sequences*: during these tests the return line flow needs to be recorded;

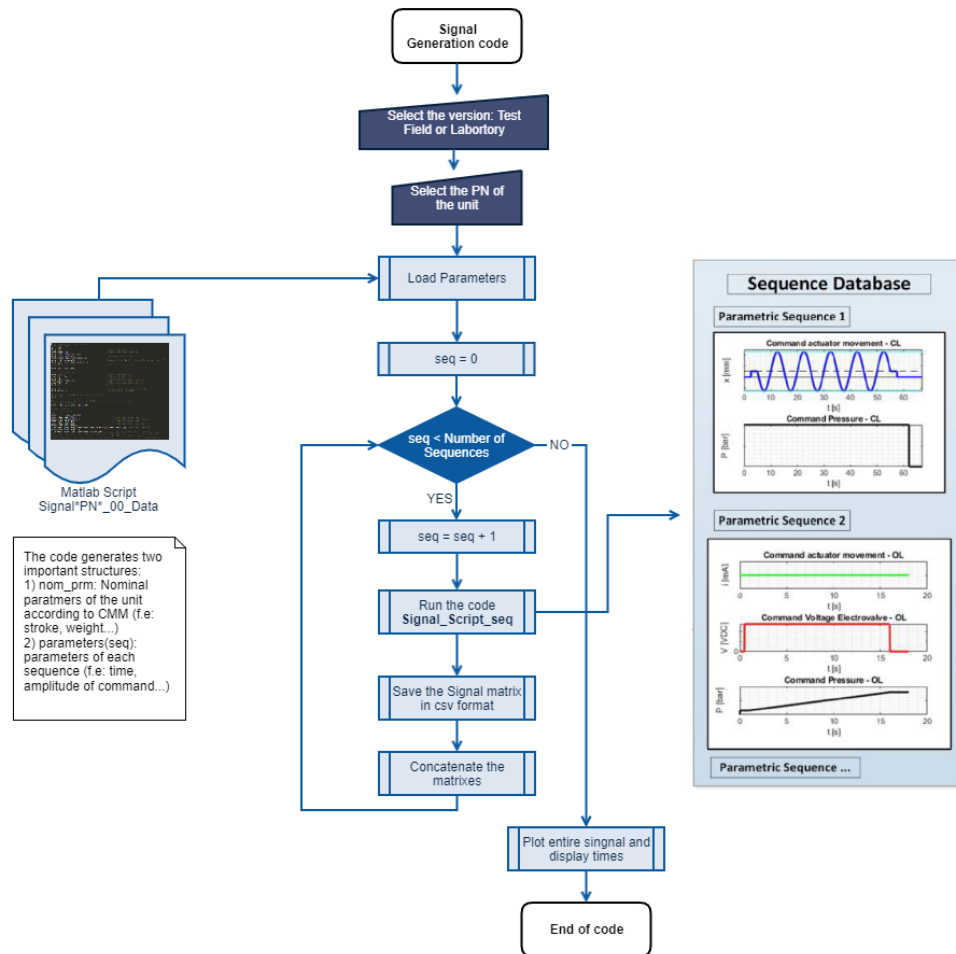


Figure 4.3: Signal generation code

(d) Fully automated *Sequences*

The complete excitation signal for each kind of UUT, *Matlab* script implementing the algorithm in Fig. 4.3.

Initially, the user needs to specify the target test-rig: Test Field or H-Lab one. Due to their constructive differences, they need slightly different formats of input signal files to run the complete Automatic Test. After choosing the characteristic PN for the kind of unit, the proper set of parameters needs to be loaded. This parameters-set contains both physical specifications of the unit, such as maximum stroke or nominal EHSV operative current, and also signal descriptive constraints such as time-ranges and amplitudes. Fundamental is the definition of the variable `seq_vec` which defines the *Sequences* order: each of them is identified with a number related to the chronological order of creation. Following the order defined by `seq_vec`, the code recalls each *Sequence* script and builds up a specific command

matrix which contains all the signal information to be read by the test-rig measurement software. This modular structure of signal generation allows to easily adapt, create or change the order of different *Sequences*. Per each kind of command, a specific colour has been selected:

- **Blue**: main-ram position command in closed loop;
- **Green**: servovalve current command in open loop;
- **Black**: supply pressure command;
- **Magenta**: EV1 supply voltage command;
- **Cyan-dashed**: EV2 supply voltage command;

All the plots in have been normalized according to each measurement channel reference value and their shapes are reported per each kind of unit, highlighting how each *Sequences* is parametrized and scalable. All the signals created by the code in Fig. 4.3 are saved as matrix and exported in *.csv* format. Each column represents one of the command channel which are listed before. These output files represents the raw input for the testing-software of both the test-benches in Chapter 3: *imc-Studio* of the H-Lab and *LabView* in the Test-Field console computer. In this way, it is possible to keep the modularity and coherence between the testing procedures.

4.3.1 A320 Elevator signal

The complete excitation signal of the Elevator is composed by twenty-six *Sequences*. The set of position, current, supply pressure or EVs voltages commands are plotted from Fig. 4.4. Compared to the other units, Elevator presents the longest signal (maximum 34 minutes in total), due to its structural complexity. As reported in Paragraph 4.3, each signal has been designed including the most critical CMM tests and it needs to evaluate the health status of each unit sub-component in both the working modes. The barplot in Fig. 4.5 displays the weights (Eq. 4.1) per each CMM test included in the Hydiag procedure, imposing a minimum threshold of *weight* > 1%.

In the first subplot of Fig. 4.4, both closed and opened loop commands alternates in order to extract traditional and additional HFs. In particular, this signal presents four completely extra-cmm *Sequences*, two of them with both the kinds of HFs, while all the others follow the CMM specifications. In total the set of collected HFs is composed by 44 traditional and 30 new-defined health indicators. Most of the *Sequences* are in active mode, where both the EVs are not supplied with voltage command. When the unit must be switched in damping mode, at least on of them is energized. At the beginning of the test, during the Preliminary and

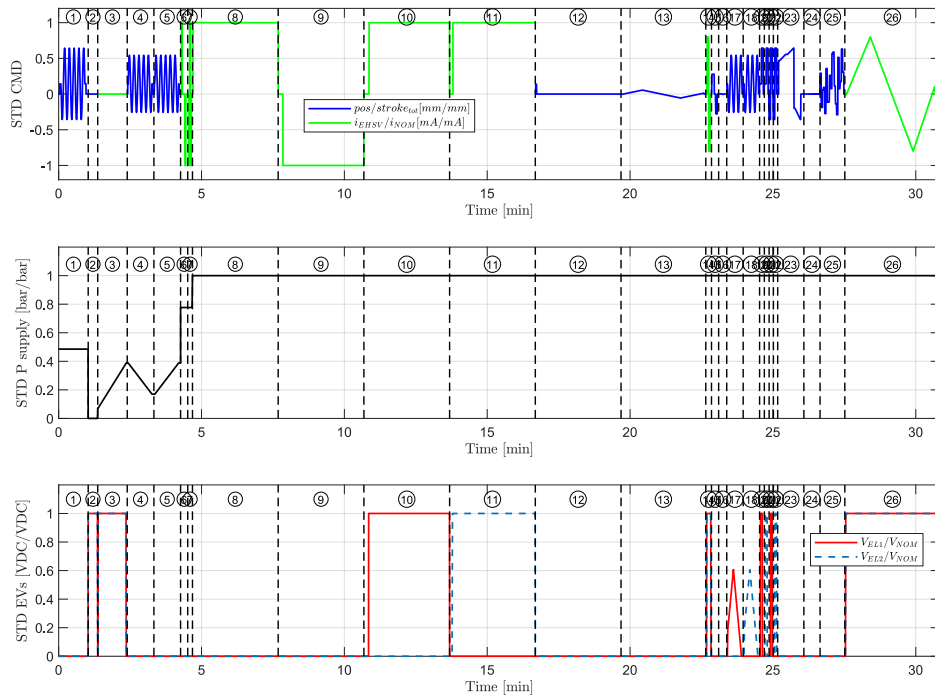


Figure 4.4: Elevator complete excitation signal

Pressure Adjustment *Sequences*, the supply pressure needs to be regulated while, from Seq.8 till the last one, this pressure remains constant to its nominal value. In the next paragraph, *Sequence* structures are presented, grouping them according to Paragraph 4.3.

Preliminary Sequences

The first part includes two *Sequences* with two required preliminary checks from the maintenance manual:

- **Sequence n.1**
 - Time: 62 seconds;
 - Entirely in Active mode;
 - Supply pressure constant (48.5% of the nominal value);
 - The main ram is moved to its middle position, then it starts sinusoidally moving at 0.1 Hz in order to complete 10 strokes, with an oscillation amplitude of 80% of the nominal stroke.
- **Sequence n.2**

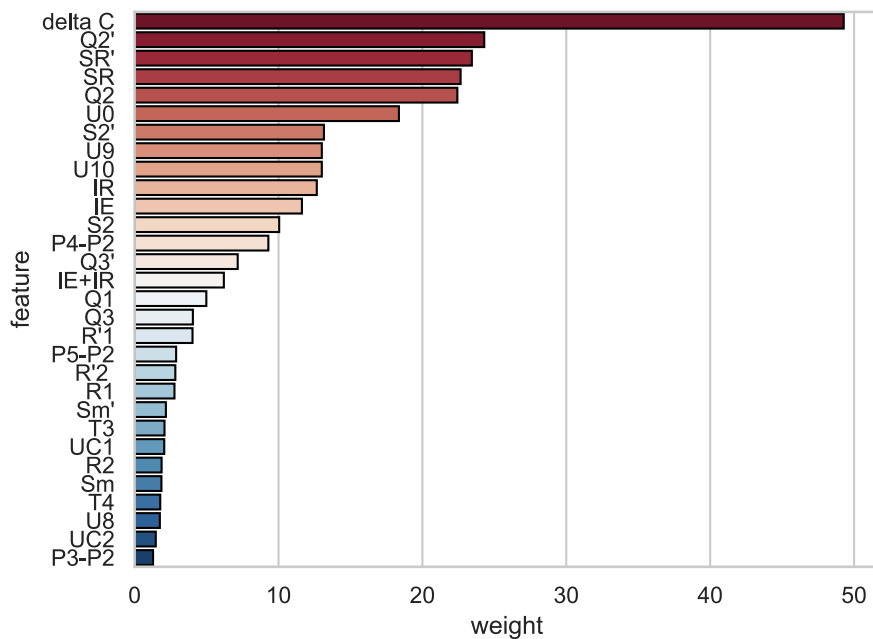


Figure 4.5: Elevator weight evaluation

- Time: 20 seconds;
- Entirely in Damping Mode: both EVs are energized with nominal voltage;
- Supply pressure null;
- Position command in Closed Loop at 0 mm. During this testing time, the technician needs to check the Accumulator status before completing the testing.

Table 4.1: Elevator preliminary sequences info

Seq. Name	Included Tests	HFs
1	Test n.2: Air Purging	<i>leakage</i>
2	Preliminary Check	<i>accu_drain</i>

Pressure Adjustment Sequences

From the third until the seventh *Sequence*, all the tests which require a supply pressure adjustment have been grouped. In particular, *Sequences* n.3 have been

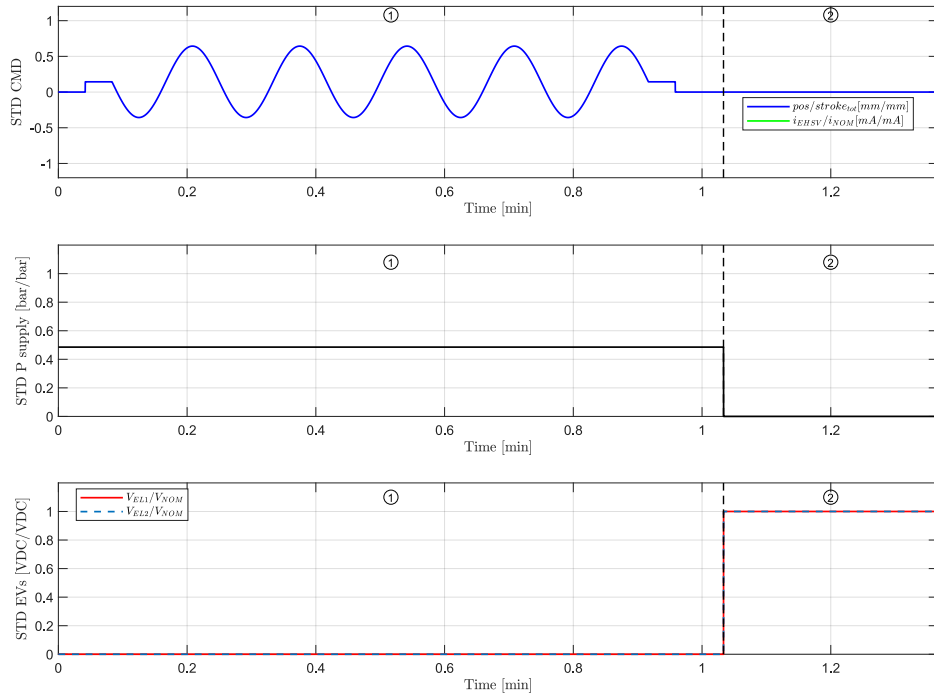


Figure 4.6: Elevator Preliminary Sequences

designed to check the correct working condition of the supply and return hydraulic lines. The length of these tests depend on the needed time by the technician to regulate the supply line from the analogue manifold. The measurements conducted in the H-Lab, described in Chapter 4, demonstrates that a digital control pressure channel through CAN-bus communication can shorten these *Sequences* of more than half.

The consecutive n.4 and 5 aim to check the opening and closing hydraulic conditions of the MSV, slowly increasing and decreasing the pressure on the supply line. *Sequence* n.6 is specific for the Elevator, since its goal is to check the unit performance in recentering mode, checking its speed to reach the null position in case of electrical failure (represented with 0 mA in open loop). This particular signal can be used to extract several features about the health status of the Elevator Recentering Device. As reported in Table 4.2, twelve HFs can be extracted from this Sequence: the first two are the traditional CMM recentering speed HFs, while the other ten are extracted from the EHSV LvdT recentering spool stroke, when the current commands are null. More details about these health-index are reported in Chapter 5.5.2. The last Sequence n.7 checks also the UUT maximum speeds during its complete strokes, commanding the unit in open loop from minimum to maximum position and vice-versa. All the consecutive *Sequences* keep the supply line pressure to the nominal value.

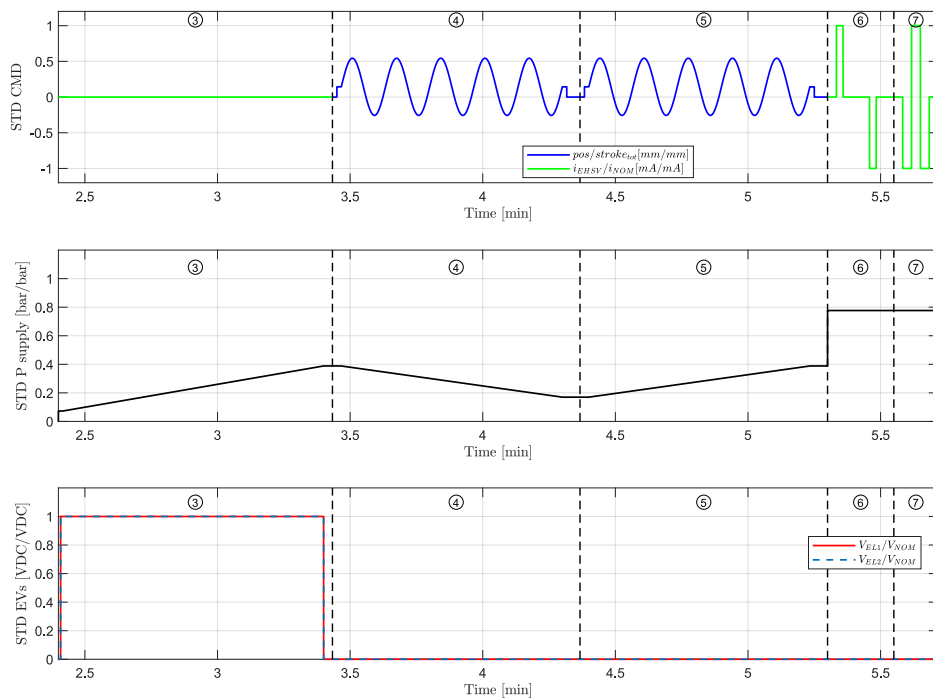


Figure 4.7: Elevator Pressure Adjustment Sequences

- **Sequence n.3**
 - Time: maximum 62 seconds;
 - Entirely in Damping mode: both EVs are energized with nominal voltage;
 - Slowly increase the supply line pressure;
 - Opened Loop current command to 0 mA
- **Sequence n.4 and 5**
 - Time: maximum 56 seconds;
 - Entirely Active Mode;
 - Two ramps of pressure: firstly decreasing and then an increasing one between two reference values;
 - Position command in Closed Loop: the unit moves to its middle position, then it starts to move sinusoidally at 0.1 Hz
- **Sequence n.6**
 - Time: 15 seconds;

- Entirely Active Mode;
 - Set the pressure at around 78% of the nominal value;
 - Current command in Opened Loop at 0 mm. Two steps in Open Loop from i_{max} to 0 mA and then from i_{min} to 0 mA, to simulate an electric failure.
- **Sequence n.7**
 - Time: 10 seconds;
 - Entirely Active Mode;
 - Set the pressure at around 78% of the nominal value;
 - Current command in Opened Loop at 0 mm. Two steps in Open Loop from i_{min} to i_{max} and back to i_{min} . The signal finishes at 0 mA.

Table 4.2: Elevator pressure adjustments sequences info

Seq. Name	Included Tests	HFs
3	High pressure valve test Low pressure valve test	$P2$ [bar] $\Delta P3$ [bar]
4	MSV Closing Pressure MSV LvdT in damping mode	$\Delta P5$ [bar] $U4$ [Vrms]
5	MSV Opening Pressure	$\Delta P4$ [bar]
6	Recentering speed from retracted stop Recentering speed from extended stop EHSV LvdT rising-time during main ram recentering stroke from extended stop EHSV LvdT step overshoot during main ram recentering stroke from extended stop EHSV LvdT falling-time during spool recentering stroke EHSV LvdT step downshoot during spool recentering stroke EHSV LvdT elbow time interval during spool recentering stroke EHSV LvdT falling-time during main ram recentering stroke from retracted stop EHSV LvdT step downshoot during main ram recentering stroke from retracted stop EHSV LvdT rising-time during spool recentering stroke EHSV LvdT step overshoot during spool recentering stroke EHSV LvdT elbow time interval during spool recentering stroke	SR [mm/s] SR' [mm/s] rt_{EN} [s] $ovsht_{EN}$ [%] ft_{EN} [s] $dwnsht_{EN}$ [%] dt_elbow_{EN} [s] ft_{RN} [s] $dwnsht_{RN}$ [%] rt_{RN} [s] $ovsht_{RN}$ [%] dt_elbow_{RN} [s]
7	Maximum speed with no load Maximum speed with no load	Sm [mm/s] Sm' [mm/s]

Leakages Sequences

Sequences from n.8 to n.12 aim to check the eventual inner leakages in the UUT, measuring the return line flow. Due to some test-bench limitations, these *Sequences* may last maximum three minutes, since the technician need to report the value from the return line analogue flow-meter. By the way, according to experimental campaign in the H-Lab (Chapter 4), these *Sequences* can be shorten up to 30 seconds with an accurate digital flow-meter.

The goal of the first two *Sequences* n.8 and n.9 is to detect an excessive return flow when the main-ram is in totally extended and retracted position. They need to be performed in Open Loop and Active mode. The consecutives n.10 and n.11 instead need to check the presence of inner leakages in the damping hydraulic circuit line, considering the MSV in its closed position. In contrast to the previous ones, these signals are still in Opened loop but in Damping Mode, alternatively energizing both the EVs. The last leakage *Sequence* is the only one in Closed Loop, at its goal is to detect an excessive leakage in UUT null position.

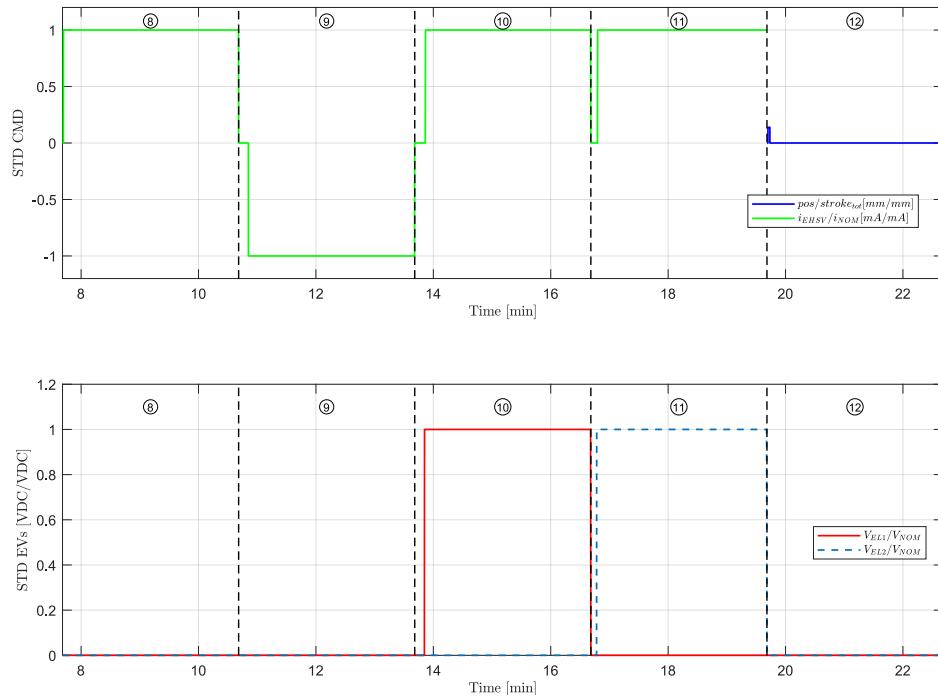


Figure 4.8: Elevator Leakages Sequences

- **Sequence n.8 and n.9**

- Time: maximum 180 seconds each;
- Entirely Active Mode;
- Constant supply pressure to its nominal value;
- Open Loop current command firstly from 0 to i_{max} (Seq. n.8), and then from 0 mA to its opposite i_{min} .

- **Sequence n.10 and 11**

- Time: maximum 180 seconds each;

- Both in Damping Mode. In particular, firstly EV1 and then EV2 are activated with V_{nom} voltage;
 - Constant supply pressure to its nominal value;
 - Open Loop current command from 0 to i_{max}
- **Sequence n.12**
 - Time: maximum 180 seconds each;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Initial small step in Closed Loop to and then recentering the unit at 0 mm.

Table 4.3: Elevator leakages sequences info

Seq. Name	Included Tests	HF's
8	Leakages in extended position Total extended stroke	$Q3$ [cm^3/min] $S1'$ [mm]
9	Total retracted stroke Recentering null position Limit cycle from extended position Leakages in retracted position	$S1''$ [mm] $S2'$ [mm] $2\delta C'$ [mm] $Q3'$ [cm^3/min]
10	Recentering null position Limit cycle from retracted position Leakages in damping mode (EV1)	$S2$ [mm] $2\delta C$ [mm] $Q2$ [cm^3/min]
11	Leakages in damping mode (EV2)	$Q2'$ [cm^3/min]
12	Leakages slaving mode MSV Lvdt in active mode	$Q1$ [cm^3/min] $U3$ [Vrms]

Automatic Sequences

From *Sequence* n.13 to 26, there is no need of interaction with the technician and measurements can run fully automatic for more than 10 minutes. In this larger group, *Sequences* can be classified according to their main target sub-components. Their disposition and duration have been optimized in order to stimulate a singular or a group of sub-components that usually affect each other. These sections can be gathered as:

- EHSV/RD/RVDT: *Sequences* n.13, 14, 15, 16, 24, 26;

- MSV/EVs: *Sequences* n.17, 18, 19, 20, 21, 22;
- RVDT/Main Ram: *Sequences* n.23, 25.

This classification leads to a better identification of each *Sequence*, although these categories boundaries are "blurred" due to the structure itself of the signal. The collected measurements from each *Sequences* contain all the possible information of its target sub-component, but it also includes other pattern that may be due to another related one, which need to be carefully extracted. For instance, *Sequence* n.23 has been designed to extract the correct demodulated ratio of the RVDT sensor, but it is also used to extract an additional extra-HF r_x , which represent the resolution command. This feature contributes to identify possible defect in the EHSV and in the main-ram sealings.

Similarly to the n.23, other *Sequences* are developed in order to extract from the same signal both traditional and extra-CMM HF (f.e Seq. n.15), while some sections are entirely dedicated to extract additional HF, like *Sequences* n.13, 16 and 26. All the traditional and extra-CMM HF collected in each *Sequence* are reported in Table 4.4.

The HF extraction methods per each *Sequence* are in detail described in Chapter 4.

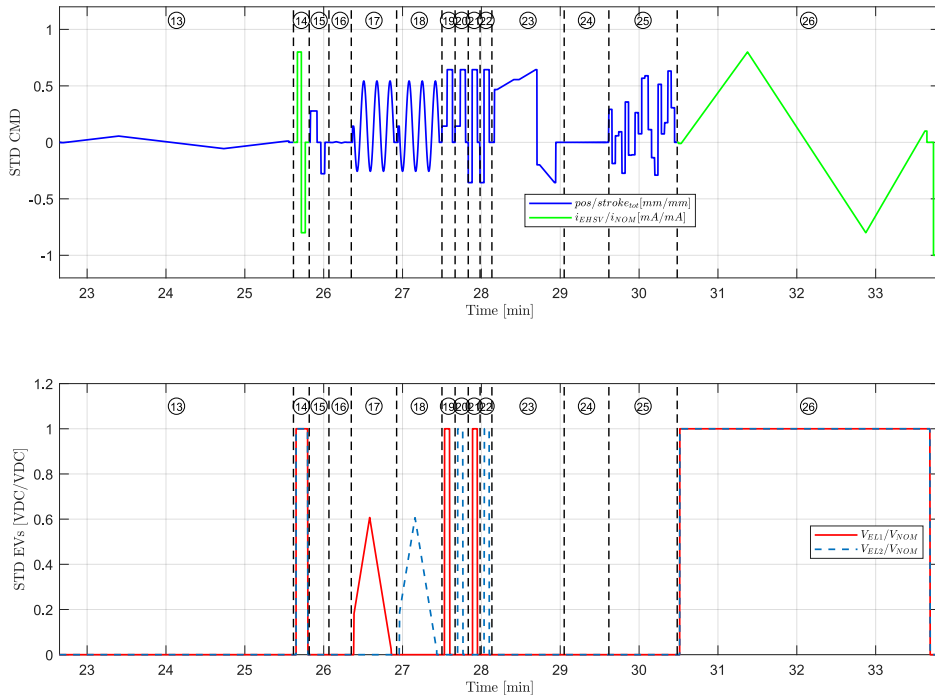


Figure 4.9: Elevator Automatic Sequences

- **Sequence n.13**
 - Time: 180 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Slow ramp in Closed Loop around the RD nominal range of action, in order to extract its characteristic curve (Fig. 2.11).

- **Sequence n.14**
 - Time: 12 seconds;
 - After few seconds, the Unit switches in Damping mode. Both the EVs are energized;
 - Constant supply pressure to its nominal value;
 - Two steps in Open Loop to 80% of both $*i_{max}$ and $*i_{min}$.

- **Sequence n.15**
 - Time: 15 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Two steps in Closed Loop interspersed with a command in null position.

- **Sequence n.16**
 - Time: 17 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Fast ramp command around its null position.

- **Sequence n.17 and 18**
 - Time: 34.5 seconds each;
 - Switching from Active to Damping mode. Rising and descending ramps voltage command to firstly EV1 and then to EV2 in their working ranges.
 - Constant supply pressure to its nominal value;
 - Command in Closed Loop: small step to mid position and then sinus wave at 0.1 Hz

- **Sequence n.19 and 20**
 - Time: 10 seconds each;

- Switching from Damping to Active mode. The EVs relay are switched off after few seconds the unit has been already commanded to its extended stop, simulating an electric failure on the EVs solenoids.
 - Constant supply pressure to its nominal value;
 - Closed Loop command with step to its extended position. At the end, the unit is slaved again to its null position.
- **Sequence n.21 and 22**
 - Time: 9 seconds each;
 - Switching from Active to Damping mode. EVs are switched ON at 1/3 of the commanded stroke of the unit.
 - Constant supply pressure to its nominal value;
 - Closed Loop command with step from retracted to extended stop. At the end, the unit is slaved again to its null position.
- **Sequence n.23**
 - Time: 55 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Closed Loop command. This signal can be divided in two parts:
 1. First step close to its target position of x_{goal} , stopping the unit 2 mm before. Then, a slow ramp drives to unit 2 mm above x_{goalex} , but keeping the unit at x_{goalex} for few seconds;
 2. Second step close to its target position of $x_{goalret}$ mm, stopping the unit at 2 mm before. Then, a slow ramp drives to unit 2 mm above $x_{goalret}$.
- **Sequence n.24**
 - Time: 34 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Closed Loop command with 1 Hz sinus wave, with increasing amplitude. The maximum commanded position needs to be close to 0.
- **Sequence n.25**
 - Time: 52 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;

- Random position steps in Closed Loop.
- **Sequence n.26**
 - Time: 200 seconds;
 - Entirely Damping Mode: both the EVs are energized;
 - Constant supply pressure to its nominal value;
 - Slow ramp in Open Loop around the nominal range of action of EHSV.

Table 4.4: Elevator automatic sequences info

Seq. Name	Included Tests	HFs
13	RD deadband amplitude RD current mean deadband RD position mean deadband RD stiffness RD current bias RD position bias	db_{rec} [mm] $I_{db_{rec}}$ [mA] $x_{db_{rec}}$ [mm] k_{rec} [mA/mm] I_{0rec} [mA] x_{0rec} [mA]
14	EHSV Lvdt in extended spool position EHSV Lvdt in retracted spool position	$U9$ [mVDC] $U10$ [mVDC]
15	EHSV Lvdt in null spool position EHSV nominal current in extended position EHSV nominal current in retracted position Signal to Noise Ratio of EHSV LVDT secondary coil (extended position) Signal to Noise Ratio of EHSV LVDT secondary coil (retracted position) Current noise amplitude in extended position Current noise peak frequency in retracted position	$U0$ [mVDC] IE [mA] IR [mA] $IE + IR$ [mA] $SNR(U_{sek})_{ext}$ [dB] $SNR(U_{sek})_{ret}$ [dB] $f_{U_{sek},ext}$ [Hz] $f_{U_{sek},ret}$ [Hz]
16	Maximum position bias Hysteresis amplitude	x_b [mm] h_x [mm]
17	EV 1 Maximum Non-Function Voltage EV 1 Minimum Operating Voltage	$U5$ [VDC] $U6$ [VDC]
18	EV 2 Maximum Non-Function Voltage EV 2 Minimum Operating Voltage	$U7$ [VDC] $U8$ [VDC]
19	EV 1 Breakdown Voltage EV 1 Pressurization time	$UC1$ [VDC] $T1$ [ms]
20	EV 2 Breakdown Voltage EV 2 Pressurization time	$UC2$ [VDC] $T2$ [ms]
21	EV 1 Depressurization Time	$T3$ [ms]
22	EV 2 Depressurization Time	$T4$ [ms]
23	RVDT ratio in extracted position RVDT ratio in retracted position Resolution test RVDT gain in extension RVDT gain in retraction	$R1$ [-] $R2$ [-] r_x [mm] K_{ext} [mm/(Vrms/Vrms)] K_{ret} [mm/(Vrms/Vrms)]
24	Threshold Test	$U11$ [mm]
25	Piston position sensor wiring Position offset in extended position Position offset in retracted position	ϕ_{UA} [°] ϕ_{UB} [°] ϵ_{ext} [mm] ϵ_{ret} [mm]
26	EHSV deadband EHSV deadband mean current EHSV deadband mean spool position EHSV sftiness EHSV bias current EHSV spool hysteresis	db_{ehsv} [mV] $I_{db_{ehsv}}$ [mA] $x_{db_{ehsv}}$ [mV] K_{ehsv} [mV/mA] I_{bias} [mA] $hyst_I$ [mA]

4.3.2 A320 Aileron signal

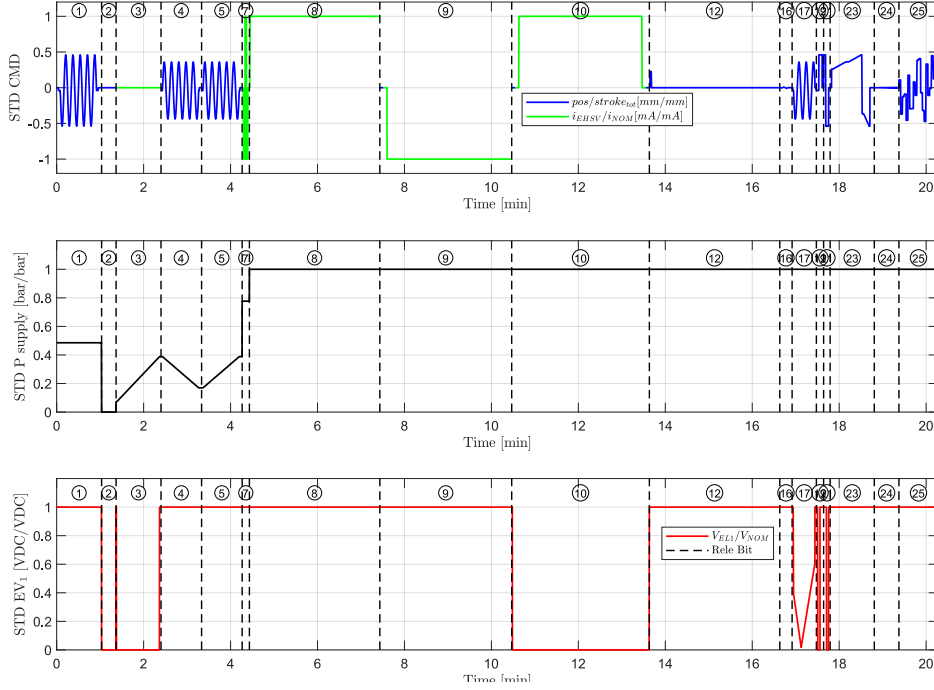


Figure 4.10: Aileron complete excitation signal

The full Aileron excitation signal contains seventeen *Sequences*, which are all in common with the Elevator one. Each signal shape has been re-adapted according to the Aileron nominal structural parameters. The major differences between the signal in Fig. 4.4 and 4.10 are:

- The RD and EHSV Lvdts absences in Aileron structure entail a shorter signal, since the previous two most extended *Sequences* n.13 and n.26 are now omitted. The total duration is around 20 minutes for this unit.
- The Aileron presents only one EV (Chapter 2.4) which switches the unit from Active to Damping mode when denegized. For this reason, the Aileron includes only the characterizing *Sequences* for one EV, and their signals are the complete opposite of the Elevator ones.

Despite these structural differences, the modular structure of the signal allows to create two similar testing procedure for these two different units. Similarly to the Elevator, barplot in Fig. 4.11 displays the weights (Eq. 4.1) per each CMM test in the Hydiag procedure.

On one hand, the simpler Aileron structure means a faster testing procedure, but on the other hand the lack of some key integrated sensors such as the EHSV

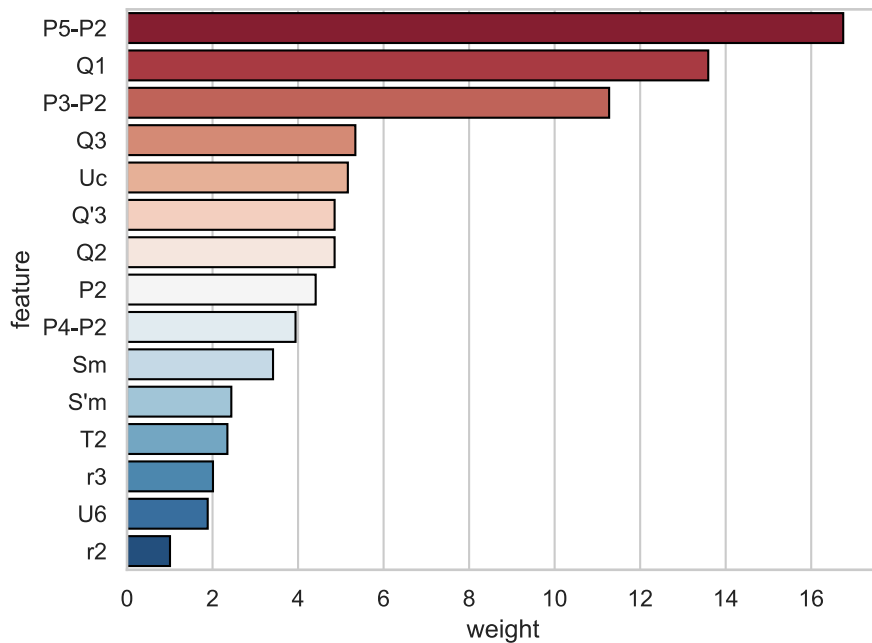


Figure 4.11: Aileron weight evaluation

Lvdt reduces the number of additional HFs. In this case, the set of collected results includes 30 traditional CMM and 12 new-defined health indicators.

The next paragraphs four different *Sequence* groups are presented according to Paragraph 4.3.

- **Preliminary Sequences:** the first two preliminary *Sequences* presents the same shape as the Elevator ones. Even in this case, during these sections the technician need to check the presence of leakage and the status of the drained accumulator.
- **Pressure Adjustment and Leakages Sequences:** compared to the Elevator ones, two main differences can be pointed. The first one regards the opposite shape of the EV signal, since the Aileron switches in Damping Mode when the EV solenoid is switched off. The second main difference is the lack of the sixth and eleventh *Sequence*, due to the RD and EV2 absence. Anyway, each *Sequence* goal does not change. As for the Elevator, the third section aims to check the correct working of the high and low pressure valves on the supply and return lines, the fourth and fifth target the MSV hydraulic opening and closing conditions, while the seventh is built to extract the maximum UUT speeds in Open loop. The leakages checks can be regularly performed both in active and damping mode.

- **Automatic Sequences:** the last seven *Sequences* can be performed automatically without any interaction with the technician. As for the Elevator, it is possible to group these tests according to their target sub-components. For example, the *Sequences* n.17, 19 and 21 have been designed to diagnose an eventual failure in the EV or MSV. All the others instead aim to check the behaviour and interaction between EHSV, LVDT and Main Ram.

4.3.3 A320 Spoiler signal

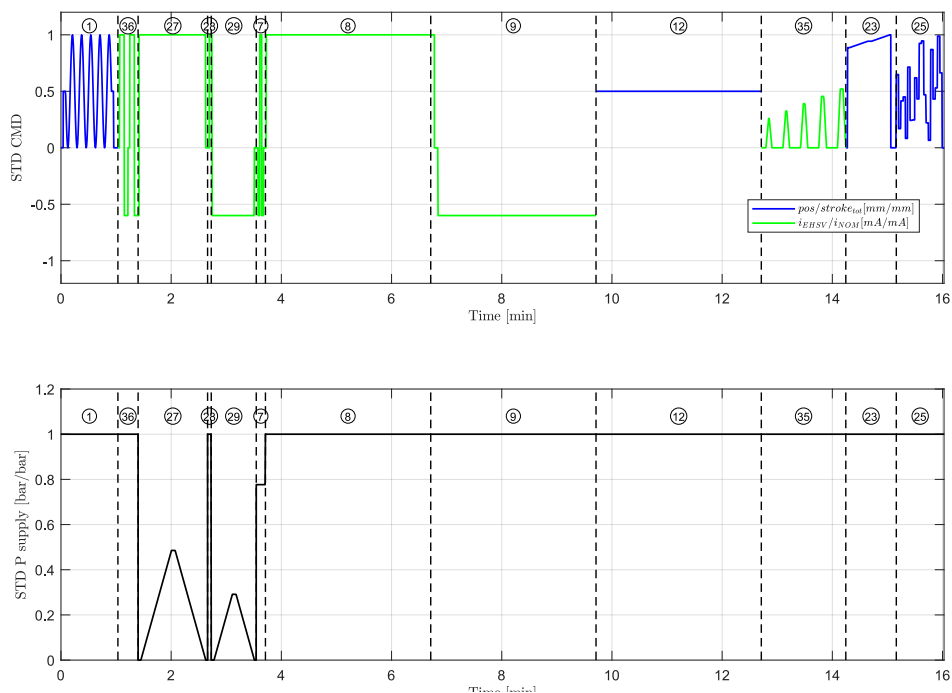


Figure 4.12: Spoiler complete excitation signal

Comparing the Spoiler structure in Paragraph 2.5 with Elevator and Aileron ones in Paragraph 2.4 and 2.5, several differences can be found even in the typical main sub-components:

- The single-effect main-ram with its minimum retracting stroke to zero. The connection between actuator and wing-frame is guaranteed by the blue linkages in Fig. 2.18.
- Two similarities with the Aileron structure is the presence of a jet-deflector EHSV without any spool sensor and a LVDT on the main ram. This last one is the only integrated sensor in the unit.

Table 4.5: Aileron automatic sequences info

Seq. Name	Included Tests	HF's
1	Air Purging test	Leakage
2	Preliminary Check	Accumulator retraction
3	High pressure valve test	P2 [bar]
	Low pressure valve test	P3 [bar]
4	MSV Closing Pressure	P5 [bar]
	MSV Lvd't in damping mode	U4 [Vrms]
5	MSV Opening Pressure	P4 [bar]
7	Maximum speed with no load	Sm [mm/s]
	Maximum speed with no load	Sm' [mm/s]
8	Leakages in extended position	Q3 [cm^3/min]
	Total extended stroke	F1' [mm]
9	Total retracted stroke	F1'' [mm]
	Leakages in retracted position	Q3' [cm^3/min]
10	Leakages in damping mode	Q2 [cm^3/min]
12	Leakages slaving mode	Q1 [cm^3/min]
	MSV Lvd't in active mode	U3 [Vrms]
16	Maximum distance to limits	d_{cr} [mm]
	Maximum position bias	x_b [mm]
	Hysteresis amplitude	h_x [mm]
17	EV 1 Maximum Non-Function Voltage	U5 [VDC]
	EV 1 Minimum Operating Voltage	U6 [VDC]
19	EV Pressurization time	T1 [ms]
21	EV Depressurization Time	T2 [ms]
	EV Breakdown Voltage	UC [VDC]
23	LVDT ratio in extracted position	r2 [-]
	LVDT ratio in retracted position	r3 [-]
	LVDT 0 setting	dD [mm]
	EHSV null bias	Ion [mA]
	Resolution test	r_x [mm]
	RVDT gain in extension	K_{ext} [mm/(Vrms/Vrms)]
	RVDT gain in retraction	K_{ret} [mm/(Vrms/Vrms)]
24	Threshold Test	U7 [mm]
25	Piston position sensor wiring	ϕ_{UA} [°]
		ϕ_{UB} [°]
	Position offset in extended position	ϵ_{ext} [mm]
	Position offset in retracted position	ϵ_{ret} [mm]

- As the elevator EHSV, the nominal measured current at commanded position is not null, but the controller provides an additional offset i_{nom} . This contribute does not compensate the presence of any RD, but a designed tilted position of the first stage TM. Like the Elevator, an electrical failure, which corresponds to null input current, makes the unit move to its null position which corresponds to its complete retraction.
- The spoiler MSV structure differs from the aileron and elevator ones due to the presence of an integrated by-pass orifice. Furthermore, as in Fig. 2.19a, its working condition is hydraulically regulated by Inhibition and LP Valve Check, instead of the aileron and elevator EVs. For this reason Fig. 4.12 does not present the third subplot of the solenoids command.

Furthermore, the presence of Inhibiting, LP Valve Check and Maintenance lock device requires the definition of new proper pressure-adjustment *Sequences*. For this reason, Fig. 4.12 presents only seven over twelve sections in common with the others. From the complete signal of about 16 minutes, 20 traditional CMM tests and 9 new-defined HFs are automatically extracted, drastically reducing testing time and reporting a more complete picture of the UUT health status.

Preliminary and Pressure Adjustment Sequences

Due to the absence of an Accumulator, the Preliminary group just include the first *Sequence* (red-dashed line in Fig. 4.13). By the way, within the Pressure-adjustment class, it is possible to label as "preliminary" sections *Sequences* n.36 and 28, since they need to be located before the n.27 and n.28 due to some necessary mechanical checks by the technician. In particular, *Sequence* n.36 signal needs to be repeated two times. The first time, the technician need to set the unit in Maintenance Mode and check that it does not move when supplied with nominal pressure and with maximum open loop current; vice-versa the second time it needs to run to its extended position when the maintenance devices is switched to Operating mode. This is first needed preliminary check of the Inhibiting Valve. Regarding the *Sequence* n.28, in operating mode, the technician need to check that the unit extends correctly when commanded with i_{max} current. This represents a preliminary check of the correct MSV switching position.

- **Sequence n.1**
 - Time: 62 seconds;
 - Entirely in Active mode;
 - Supply pressure constant (48.5% of the nominal value);
 - The main ram is moved to its middle position, then it starts sinusoidally moving at 0.1 Hz in order to complete 10 strokes, with an oscillation amplitude of 80% of the nominal stroke.

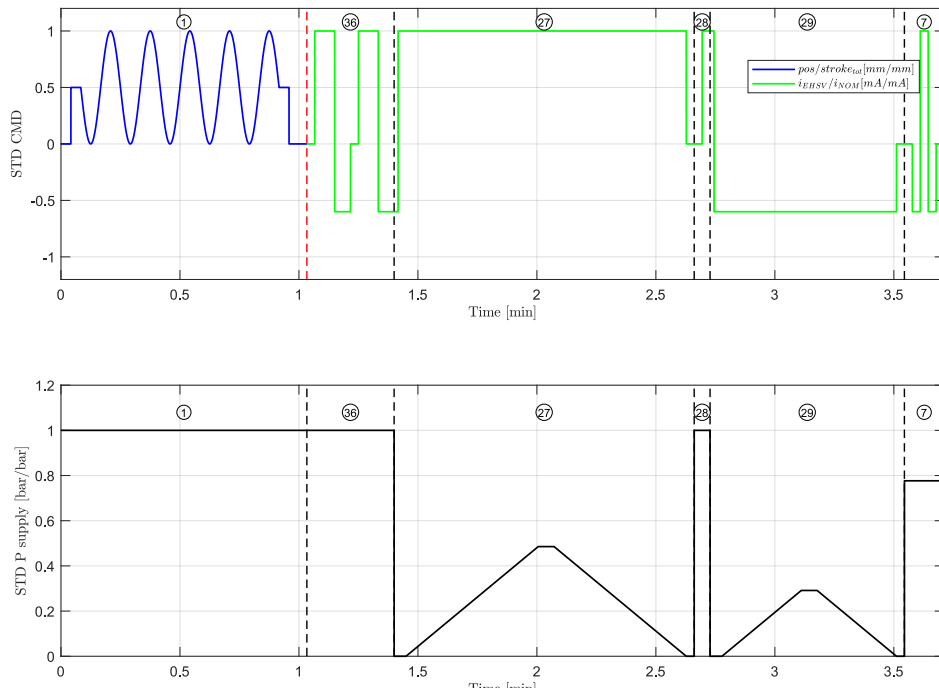


Figure 4.13: Spoiler Pressure Adjustment Sequences

- **Sequence n.36**
 - Time: 11 seconds;
 - Switch to Maintenance and to Operating Modes;
 - Constant supply pressure to its nominal value;
 - Opened Loop steps from i_{min} to i_{max} and back.
- **Sequence n.27**
 - Time: 75 seconds;
 - Entirely Active Mode;
 - Slowly increase the pressure from 0 bar until the unit starts moving, than back to 0 bar;
 - Opened Loop steps from i_{min} to i_{max} .
- **Sequence n.28**
 - Time: 5 seconds;
 - Entirely Active Mode;
 - Set the pressure to its nominal value;

- Step in Open Loop from 0 to i_{max} .
- **Sequence n.29**
 - Time: 75 seconds;
 - Entirely Active Mode;
 - Slowly increase the pressure from 0 bar until the unit starts moving, than back to 0 bar;
 - Step in Open Loop from i_{max} to null and then to i_{min} .
- **Sequence n.7**
 - Time: 10 seconds;
 - Entirely Active Mode;
 - Set the pressure at around 78% of the nominal value;
 - Current command in Opened Loop at 0 mm. Two steps in Open Loop from i_{min} to i_{max} and back to i_{min} . The signal finishes at 0 mA.

Table 4.6: Spoiler pressure adjustments sequences info

Seq. Name	Included Tests	HFs
1	Air Purging	<i>leakage</i>
36	Inhibiting valve check in M mode Inhibiting valve check in O mode	<i>IV1</i> <i>IV2</i>
27	Shut off valve closing pressure	<i>PS1</i> [bar]
28	Preliminary MSV check	–
29	MSV opening pressure MSV closing pressure	<i>PS2</i> [bar] <i>PS3</i> [bar]
7	Maximum speed with no load Maximum speed with no load	<i>Sm</i> [mm/s] <i>Sm'</i> [mm/s]

Leakages Sequences

Since the unit does not include any EV, this group of signals contains only the leakages tests in active mode of *Sequences* n.8, 9 and 12. Each signal shape has been readapted according to the unit structural parameters. For instance, leakages in slaving mode in *Sequence* n.12 need to be checked at the main ram middle-stroke, since the null position coincides with its retracted stop. Another

signal distinguishing feature is the intermediate step from i_{max} to null current in the second *Sequence* n.8. This step allows the extraction of the retraction speed in case of electrical failure.

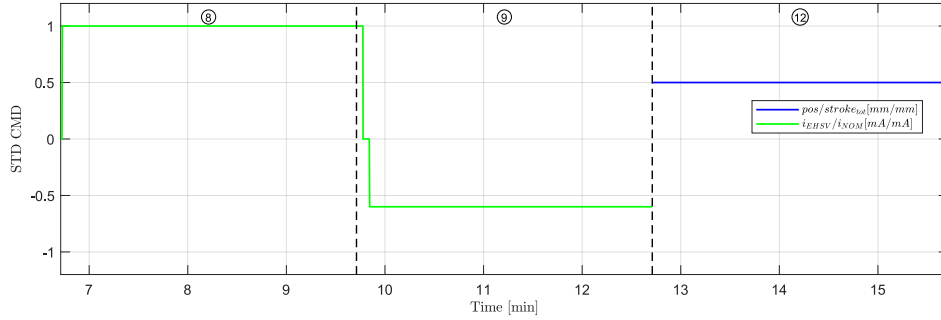


Figure 4.14: Spoiler Leakages Sequences

Table 4.7: Spoiler leakages sequences info

Seq. Name	Included Tests	HFs
8	Leakages in extended position Total extended stroke	$Q3$ [cm^3/min] $D1'$ [mm]
9	Leakages in retracted position Total retracted stroke Electrical failure	$Q2$ [cm^3/min] $D1''$ [mm] $V4$ [mm/s]
12	Leakages slaving mode	$Q1$ [cm^3/min]

Automatic Sequences

This group includes only three sequences that can be performed without any adjustment from the technician. *Sequence* n.35 in Fig. 4.15 is the only peculiar signal for this kind of unit. It is dedicated to the extraction of two important EHSV HFs: null bias and threshold currents. It consist of a series of open loop ramp signals which aim to obtain the minimum necessary current to make the unit moving and first stabilized current signal when the unit has reached a goal position.

- **Sequence n.35**

- Time: about 22 seconds;

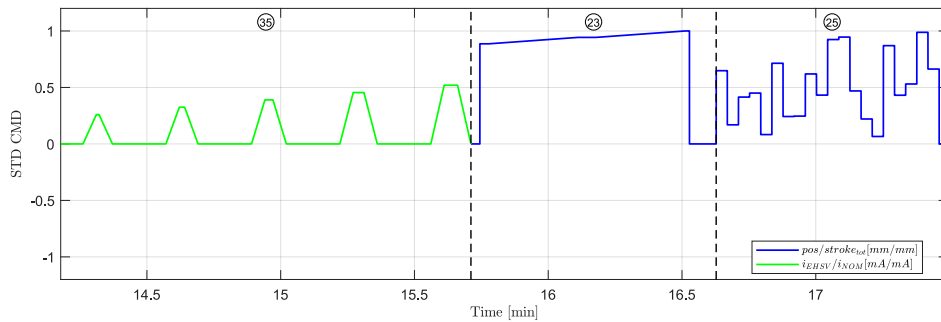


Figure 4.15: Spoiler Automatic Sequences

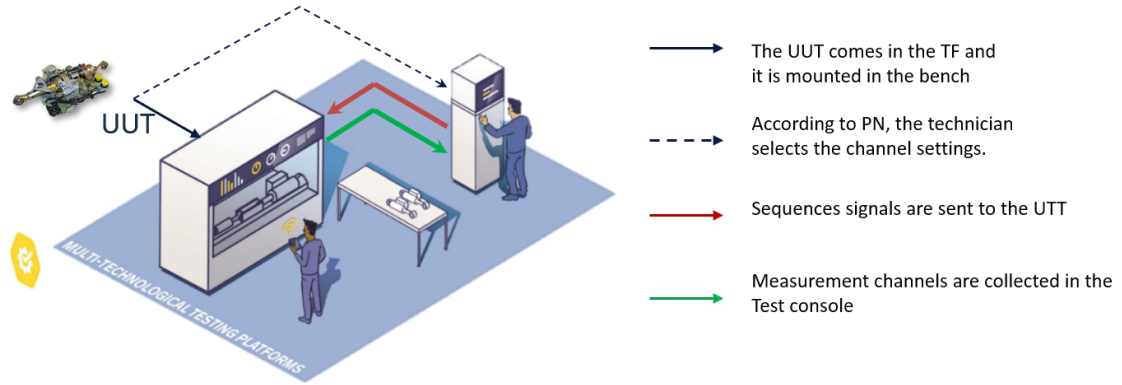
- Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Pyramidal periodic signal with increasing peak of current.
- **Sequence n.23**
 - Time: 55 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Closed Loop command. e to its target position of x_{goal} , stopping the unit 2 mm before. Then, a slow ramp drives to unit 2 mm above $x_{goalext}$, but keeping the unit at $x_{goalext}$ for few seconds;
 - **Sequence n.25**
 - Time: 52 seconds;
 - Entirely Active Mode;
 - Constant supply pressure to its nominal value;
 - Random position steps in Closed Loop.

4.4 Testing Framework

As already anticipated in Chapter 3, the automatic procedure has been tested firstly in the H-Lab prototyping bench and then translated according to the Test Field benches requirements. Figure 4.16 illustrates the flow of information from test-bench and measurement console during the Hydiag testing procedure. Firstly

Table 4.8: Spoiler automatic sequences info

Seq. Name	Included Tests	HF's
35	Servo valve null bias Servo valve threshold	I_n [mA] I_t [mA]
23	LVDT ratio in extracted position Resolution test LVDT gain in extension	V_r [-] r_x [mm] K_{ext} [mm/(V _{rms} /V _{rms})]
25	Piston position sensor wiring Position offset in extended position Position offset in retracted position Current noise amplitude in extended position Current noise peak frequency in extended position Current noise amplitude in retracted position Current noise peak frequency in retracted position	ϕ_{UA} [°] ϕ_{UB} [°] ϵ_{ext} [mm] ϵ_{ret} [mm] α_{ext} [mA] f_{ext} [Hz] α_{ret} [mA] f_{ret} [Hz]

Figure 4.16: Hydiag Test Framework (*LHTSmart Factory*)

the UUT is installed on the Test Bench, where it is hydraulically and electrically connected (blue line). According to the kind of unit, the technician is able to select the proper settings for the automatic Hydiag procedure (dashed-blue line). Even though the structure of automatic test is equal for both the test-rigs, due to the structural differences of the testing infrastructures, it has been necessary to develop a proper framework for both the test-consoles in order to perform correctly the automatic procedure. As already reported in paragraph 4.3, all the *Sequences* time-series are exported in dedicated .csv files which represent the starting point of the automatic test procedure. Figure 4.17 reports how the outputs of the Signal Generation code can be used for both the testing configurations (red line in Fig. 4.16).

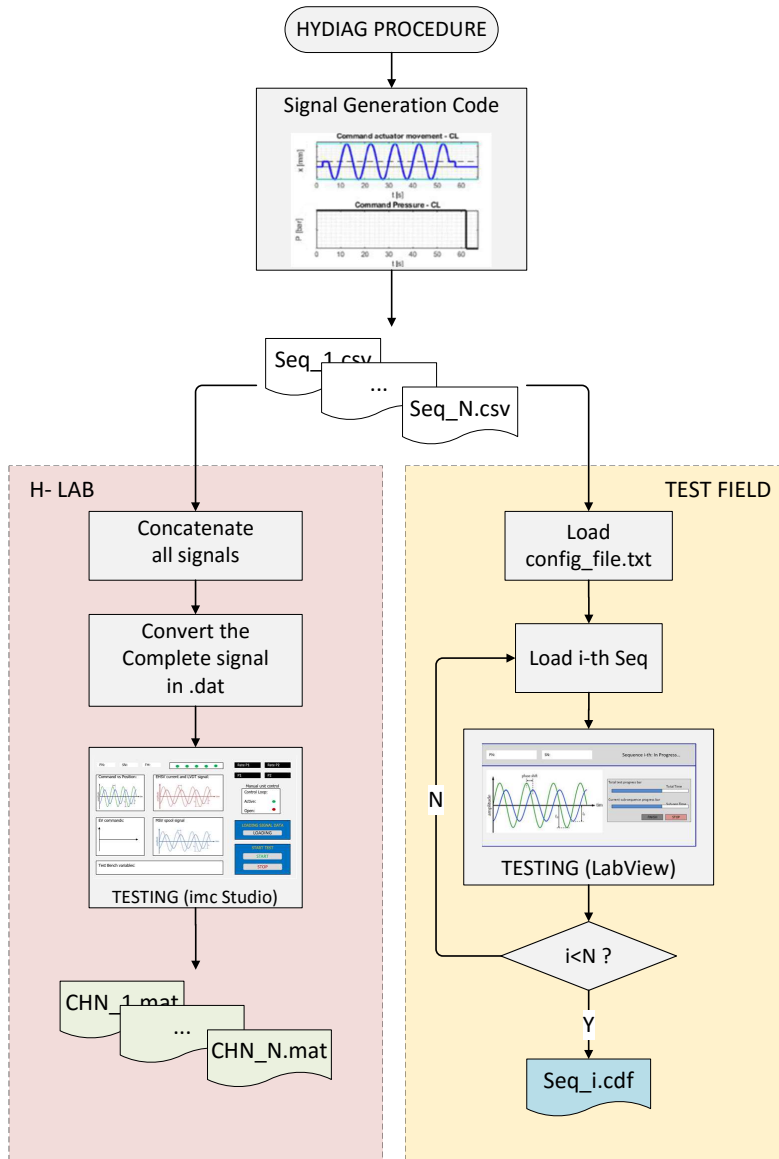


Figure 4.17: H-Lab vs Test Field Procedure

The H-Lab configuration requires firstly to concatenate all the output from the Signal Generation code and to convert them in proper .dat format. In this way, from the GUI interface in *imc Studio*, it is possible to control all the test-bench settings (such as supply pressure), and read the real-time evolution of the measurement channels. All the active channels have been previously set during the development of the GUI itself. From this side, the testing software allows the

user to customize each analogue or virtual channel configuration, such as applying a filter on a specific one or imposing different sample-rate to reduce the weight of the output measurements files. Once the entire test is over, the user is able to save the data in a proper folder both in `.dat` and `.mat` format, which can be directly processed by the Health-Features extracting code in Chapter 5. It is important to notice that, since the input loaded signal is obtained concatenating all the time-series *Sequences* from the `.csv` file, the H-Lab test outputs includes the complete time-series of all the active measurement channels. These files are represented in Fig. 4.17 as `CH_N.mat` which close the loop in Fig. 4.16 (green line).

The Test-Field configuration differs from the H-Lab one in different aspects. While in the prototyping phase the signals need to be concatenated, the *LabView* testing software loads sequentially the different `.csv` files according to a prior configuration file named `conf_PN.txt`. This document differs according to the UUT PN and the is selected directly by the technician trough an appropriate GUI. It contains all the information about the channels to be activated during each *Sequence*, their sampling rate and the working mode to set on the UUT (active or damping whether closed or open loop command). According to an internal sequencer, the LabView code loads the *i*-th stimulus-file in `.csv` format and performs the test accordingly. While testing, the technician is able to check the running time of the executed *Sequence* and eventually stop it in case of any anomaly. At the end of each *i*-th step, the LabView exports the complete time-series of the active channels in a unique file in `.cdf` format. This particular extension (Common Data File) allows both to store large vectors in relatively small memory space and also some important meta-data, such as test date and hour, sampling rate and name of the technician. These files are locally stored in a dedicated hard-disk and named according a particular format: `PN_SN_SAP_number_Sequence_i_YYYYMMDD_hhmmss_testtype.cdf`. This loop is repeated until all the *N* *Sequences* have been correctly performed and saved accordingly.

Storing the measurement files in separated folders and according to this format has improved a lot the traceability of each test. Significant is the last field of the format name `testtype`. It can assume two different string values: `E` if the Hydiag Procedure has been performed as Entry ("Eingang") Test or `A` if it is referred to an Exit ("Ausgang") Test after repairing the units. In fact, the actual deploying phase on the TF test benches expects to compare the effectiveness of HyDiag with the Standard Procedure according to the agenda in Fig. 4.18.

The collected data during this experimental campaign are referred to two different steps in the testing-scheduling of each UUT. This intermediate step before the complete implementation on the TF consoles has been necessary for several reasons. Firstly it has helped collecting feedbacks from the users to fix every signal of the Automatic procedure, until their final shapes in the previous paragraphs. Secondly, it was necessary to collect as much data as possible to develop, debug and test all the HFs algorithms of the Hydiag software in Chapter 5. Since both

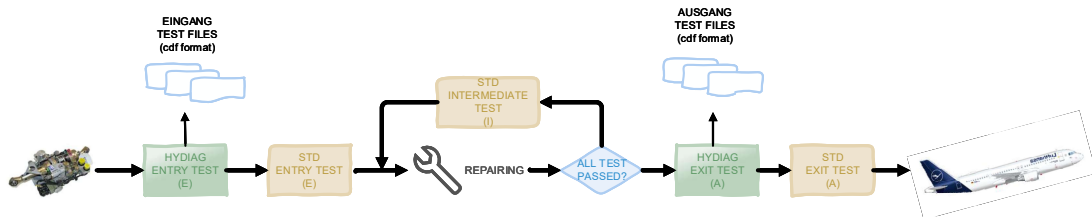


Figure 4.18: Hydiag Testing Scheme

unserviceable and repaired units were needed for diagnostic purposes, it has been necessary to collect both data from a prior Hydiag "Eingang" Test (E), to be performed as soon as an unserviceable unit is shipped to the shop, and the results after all the necessary repairs, testing and re-certifying the unit as serviceable in an "Ausgang" test. It is important to notice from Fig. 4.18 that currently the final Hydiag "Ausgang" test is used just for data-collection purposes while the final certification of a serviceable unit needs to be still performed via standard CMM procedure.

Chapter 5

Health Features Extraction and Collection

As already reported in [51]: "Data are rarely useful or usable in their raw form". This simple sentence summarizes one of the most important challenges for health-management researchers: how to extract significant information about the health status of the tested component from the collected measurements? How can the symptoms of a failure be clearly quantified by a unique and well-defined Feature? These questions are two cardinal points that need to be solved before building any prognostic and diagnostic module. However, both of them can be answered just through a deep knowledge of the system under testing. For this reason, it would be useless to use the raw data as complex structure of input to be used for a diagnostic module. Even if this kind of classifier would recognize any anomalous behaviour through different patterns, these might not have any physical meaning since the adopted machine-learning algorithm is not aware of the system complexity. The correct strategy to follow is based on the assumption that the researcher analysing the results knows where to find the information of a possible anomaly, and this can be numerically quantified by a defined parameter called Health Feature. The more complex the system is, the tougher the challenge results. Once collected all the measurements, before starting to identify any possible health indicator, it is important to know (and possibly register) the following points:

- Testing conditions such as time and location. These may influence both the system but also the test-bench working conditions;
- How the UUT has been stimulated in terms of test-signals and external factors (such as source of load or disturbance);
- The characteristics of the collected channels. For example, it is important to know if a channel is an input or an output coming from any integrated or external sensor of the UUT. Since most likely all the channels have been

sampled and digitalized, the parameters of the A/D converter need to be taken into account for any kind of signal manipulation.

Once all these initial points are solved, knowing how the system subcomponents works and how they are connected, it is important to define where to find the information. It can be a time or frequency domain indicator or a shape-characterizing parameter of the registered response: the important is to recognize which information are the most valuable according to the kind of recorded signal and component to analyse.

The following paragraphs described in details all the techniques that have been used to extract the features reported in Chapter 4, and how these are characterizing per each subcomponent of the described PFCS. The final goal of this delicate process is to extract and collect the most valuable information which are used to picture the health status of the tested component. As already anticipated in Chapter 4, the Health-Features group include both traditional ones from the CMM testing instructions and additional ones which have been designed to improve and complete the health check-up of the tested unit.

5.1 HyDiag software structures

In order to have a complete and structured HFs extraction process, a dedicated *Hydiag Software* has been developed to analyse the output of the adopted testing-console. As already anticipated in 4.4, depending on the testing equipment, the structure of the outputs can be different if they are collected from the H-Lab test-rig or the Test Field one. In the first case, at the end of each measurement, the testing-software in *imc Studio* exports the complete time-series of the active measurement channels in `.mat` format. These results have been widely used during the first prototyping phase to fix the *Sequences* shapes, refine the signal-processing techniques for features extraction and set all the requirement for the following production phase in the Test Field. The version in the following paragraphs is referred to this latter for production application, which is able to process directly the `.cdf` files from the NED console. Both the versions for H-Lab and Test Field results shares the same core-functions of HFs extraction and file exploring, while they differ regarding the pre-processing of the collected time-series. Since in the H-Lab the exported channels are differently sampled, they need to be up or down sampled accordingly in order to have a common unique length. This is a fundamental step that alleviate the tasks of the health-features extraction codes, as it is described in the next paragraphs.

The software has been developed entirely in *Matlab* and it is present in two different versions: a Product Version (PV) and a Debug (DBG) one. The main differences between these two versions are illustrated in Fig. 5.1. The first one in the light-green square is installed directly in an external computer (NUC), mounted in the

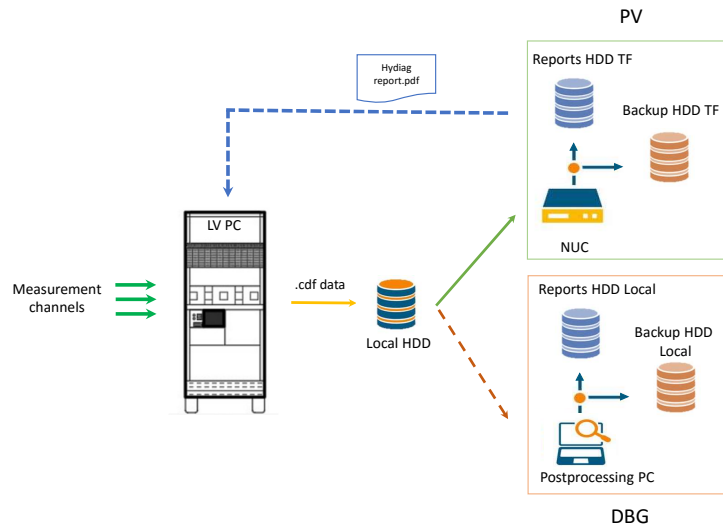


Figure 5.1: Hydiag SW versions flowchart

NED testing rack. Its working mode is coordinated by the LabView PC (LV PC) which runs the testing-software and is responsible for collecting the measurement channels. Through a specific hand-shake protocol between these two computers, once a *Sequence* has been performed according to the flow-chart in Fig. 4.17 the corresponding *Seq_i.cdf* file is exported in a local HDD.

The software versions in Fig. 5.1 share the core-functions of processing HF's extraction methods and reporting. However, they present significant differences in terms of locating the measurements files in the local NED HDD. These differences are graphically represented in the image with two different colours of the arrows connecting the PV and DBG versions with the data source, and they are mainly due to the way they actually work with these data. On the one hand, the PV version has been developed in order to communicate with the NED PC and elaborates the signal while the test is running. On the other hand, the DBG version has been designed to work "off-line" and to process the data when all the necessary datas have been already collected. More details about how the measurement datas are treated by both the versions are explained in the next paragraph.

5.2 Measurements data management

Fig. 5.2 shows how the PV version of the HyDiag software automatically manages the flow of measurement data which from the local NED HDD to the relative backup drives.

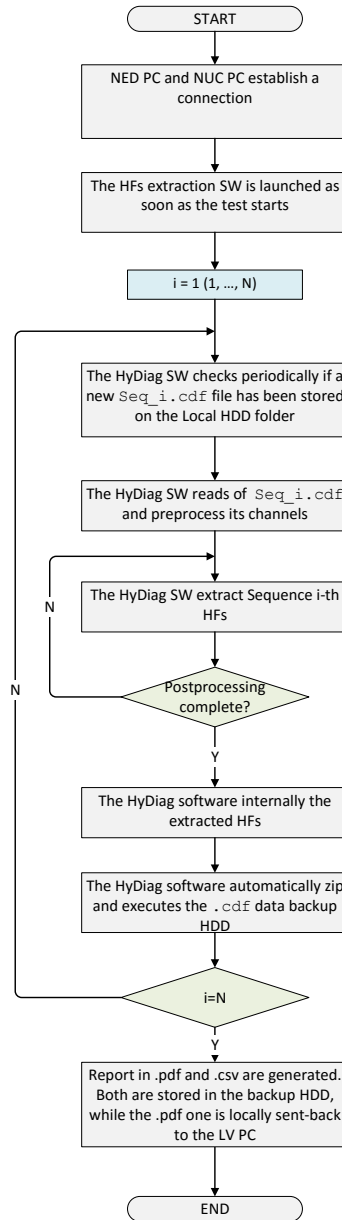


Figure 5.2: HyDiag software PV version data management flowchart

The flowchart in the plot is actually a simpler version of the "hand-shake protocol" which regulates the communication between LV PC in the test-console and the NUC where the *Matlab* HF's extraction software is installed. In particular, the PV software version periodically checks whether a new file has been added on the target folder of the Local HDD: if a new file is present, it processes the corresponding channels according to its proper HF's extraction script. The number of files exported by the LV PC and processed by the NUC depends on the PN of the tested unit, since each PN is characterized by its signal shape. Each processed file is firstly zipped and then transferred in a dedicated Backup HDD, catalogued in folders named according to the following label:

$$NED_n_PN_SN_YYYYMMDD_hhmmss_testtype$$

where:

- *n*: number of the Test-Field used for the test;
- *PN*: unit part-number, which identifies the kind of tested flight-control unit;
- *SN*: unit serial-number that is proper of the tested actuator;
- *YYYYMMDD_hhmmss*: test date and hour;
- *testtype*: differentiates the "Eingang" entry test (E) from the "Ausgang" (A) one, as reported in paragraph 4.4.

Once an automatic Hydiag Test is completed and all the HF's have been extracted, the software generates a complete report in *.pdf* for the technicians with all the extracted indicators, functional plots and troubleshooting instructions. This report and its equivalent form in *.csv* format are also stored in a dedicated folder which is named using the same label of the relative measurement files folder. More details about the complete structure of these reports are reported in paragraph 5.6. To sum up, the PV version in Fig. 5.1 has been designed in order to execute different tasks while the test is running on the measurement systems, such as not only extracting the HF's from each signal, but also managing the flow of data in input (the *.cdf* files) and in output (the printed reports). The DBG version shares with the PV one all the core-functions of preprocessing, health-features extraction methods and reporting. However, since it can be used "off-line" without any interaction with the LV Pc, the wrapper code does not need to manage the folder organization of the stored measurement files. At most, the DBG version has been designed in order to recognize the tests which have been previously analyzed from the new one, comparing the contents of both reference data and reports folders. The red-dashed line in Fig. 5.1 represents that the DBG software version need the data to be already collected in the specific folders, with the appropriate label as

name. This task is performed from an additional script, called `data_transfer`, which is dedicated to read the files in the NED local HDD, create the appropriate subfolders, compress and transfer the files appropriately and collect the test meta-data in a dedicate `INDEX.txt` file.

5.3 HyDiag software structure

The DBG version of the software has been designed to work on a single workstation after the testing phase has been completed. This facilitates a further analysis of the collected data or it can be used as benchmark to deploy new HFs extraction or fix bugs for the PV version. For this reason, it needs to be customizable and user-friendly. For example, it presents a dedicated GUI in Fig. 5.3 where it is possible:

- Select if the input files are from H-Lab or the Test Field. As already mentioned, the output files from these testing settings differ for both extension and structure. In case the option "Laboratory" in the drop-down menu is chosen, it is possible to check or not the presence of any Leakage during the test through the dedicated check-box (normally deactivated).
- Plot the raw time-series collected in the `.cdf` files. This option can be activated via the corresponding check-box and it has been particularly useful during the first debug phase of the code.
- Select the "Debug Mode" to print out dedicated plot per each HFs extraction method. All the plots in the following sections are obtained through this mode.
- Insert the Operator Name for reporting

The flowchart in Fig. 5.4 describes the scheme adopted to implement the DBG version of the HyDiag software. After pressing start from the GUI, the user need to select the folder containing the raw measurement files. In the actual configuration, this folder contains as many sub-folders as the tested units for all the PN. Besides the raw data from LabView in `.cdf` format, each dedicated sub-folder contains a specific log file in `.txt` format with all the information regarding the N measurement files in `.cdf` format. In particular, this file includes all the eventual typed-in inputs from the technician during the testing procedure. In fact, due to the technical limitation of the Test Field bench, some HFs have to be manually reported by the operator since it is not currently possible to digitally control or measure channels such as supply pressure (manually regulated) and return flow, while both of them are can be collected from the H-Lab prototyping bench. Due to these issues,

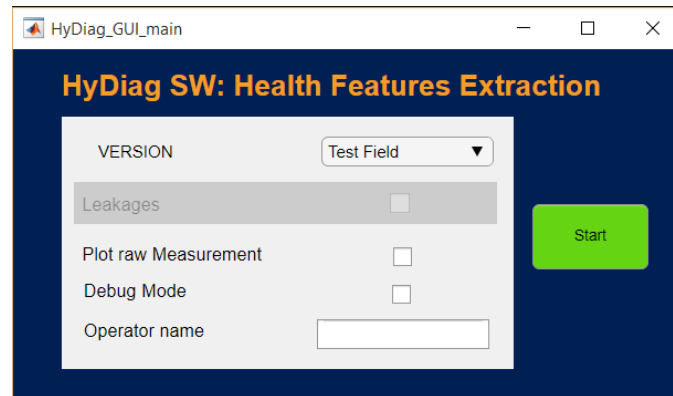


Figure 5.3: Hydiag SW GUI

during the transition phase between prototyping and production phase, it has been necessary to adapt the Pressure Adjustment and Leakages Sequences in Chapter 4.3 in order to allow the operator to manually set pressure values and eventually report any hydraulic HFs from the bench via dedicated "dialogue windows" in Lab-View.

Once the correct folder has been selected, the user need to choose the unit PN in order to load three sets of parameters files, represented by the light red blocks in Fig. 5.4. In order to make these files also usable for further Data-Analysis purposes, they are stored in tabular `.csv` files:

1. Design unit parameters such as nominal total stroke or demodulation gains for integrated position sensors.
2. Measurement settings from the dedicated `conf_file.txt`. As reported in paragraph 4.4, this file includes all the testing requirements, such as sample-rate or active channels during the Sequence execution.
3. All the HFs variable-names, unity of measure, upper or lower limits to be checked and eventually different reporting-names are stored in a dedicated files per each PN. If a new HF is defined in a specific Sequence, this file need to be modified accordingly or otherwise the report would not contains its information.

After all the parameters and settings have been defined, the user need to select the complete set of tests to be processed through a dedicated window, which reports the list of unique "ident" of the performed and collected tests. These ident-codes are built starting from the naming structure of the measurement files already mentioned in paragraph 4.4.

Once the user as selected M different unit tests from the GUI, their relative paths are saved and, starting from the first one, the code is can read each of

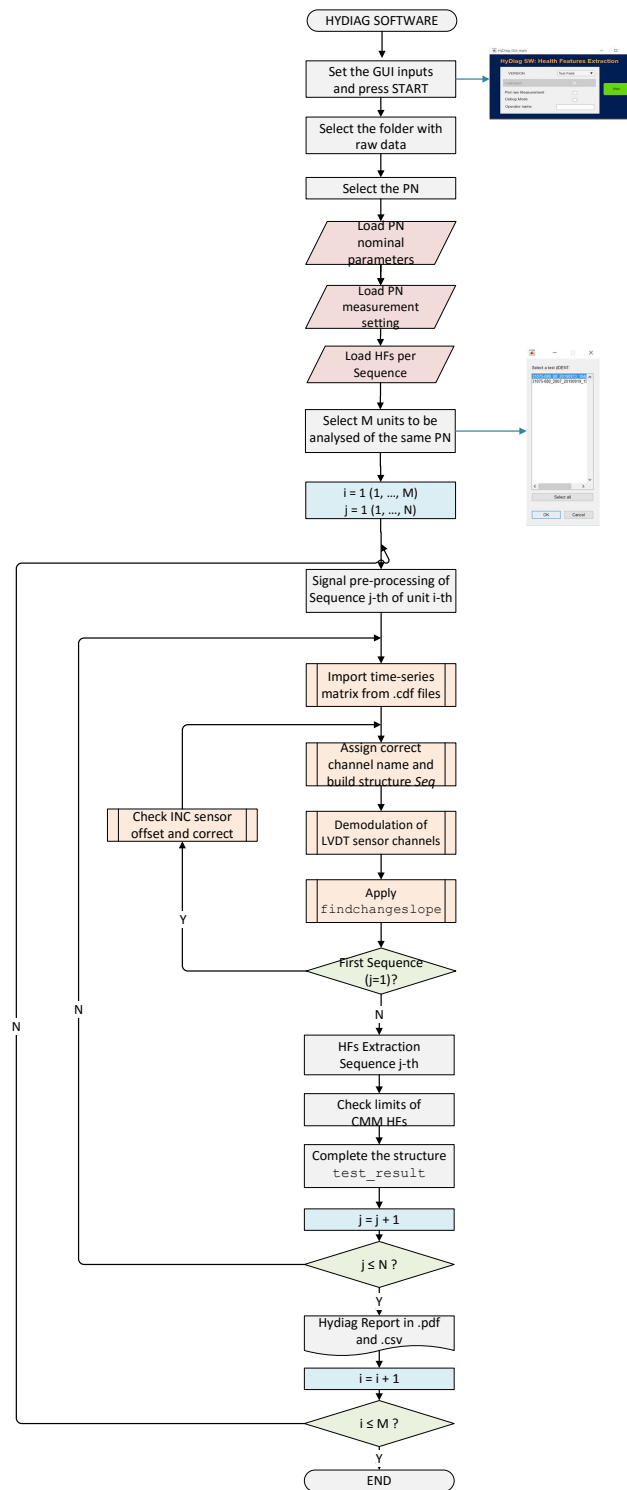


Figure 5.4: High-level DBG software flowchart

the N measurement files in each folder. Each i -th *Sequence* file needs to be opened in order to extract both its meta-data and the time-series vectors saved in a unique matrix, whose columns contains the measurement from each active channel. Since the `conf_file.txt` has been previously saved, a dedicated function (`extraction_channel.m`) assign to each column of the measurement matrix a proper name and store this results in the structure *Seq*. This variable is fundamental since it is used in every HFs extraction code. According to the measurement settings in the configuration file, the function builds the demodulated channels of the integrated LVDTs from their raw-channels. More details about this function are reported in the next paragraph. In this preliminary phase, from the initial *Sequence* it is possible to detect any anomaly in the testing procedure due for example to wrong cable connection or setting of any external sensor. For example, if the test-bench incremental sensor for main ram position has been wrongly set or wrongly connected, the Hydiag software reports this error both on the final report and it tries to correct any setting offset of the following *Sequences* which could affect the position measurement of the unit.

Once the channels time-series have been correctly read and pre-processed, the code proceed to extract from each of N *Sequence* the corresponding HFs and to check their prescribed CMM ranges for the traditional ones. Each HF with its limit and eventual test result (PASSED or FAILED) are saved in a dedicated Matlab table which is directly printed not only in the pdf report version for the technicians, but also exported as csv file. Both the reports are stored in dedicated folder which are named as the source data directory. In case of any error during the HFs extraction phase, an error-handling function would report the affected HF as "Not Extracted" in the pdf version and as NaN in the csv one.

Once the software has printed the reports for all the M selected tests, it can be stopped.

The flowchart in Fig. 5.4 represents the main steps for the correct working condition of the developed Hydiag software. The next paragraph reports some of the most important developed functions for both preprocessing the measurement channels and HFs extraction.

5.4 Sequences Loading and Pre-processing

This section explains in details how the time-series vectors from the measurement channels are loaded and preprocessed, as previously described in the first orange blocks in the flowchart in Fig. 5.4. These two important tasks are implemented within two dedicated scripts which are shared by both the DBG and PV software versions. As already anticipated in the previous paragraph, the first main output is the structure *Seq* which contains all the measurement channels from the opened `.cdf` data. The main developed function to create the variable *Seq* is named

`extraction_channel.m`, whose structure is presented in paragraph 5.4.1. As reported in the flowchart in Fig. 5.5, it recalls other important functions for LVDT channels demodulation and eventual noise filtering of secondary voltage. More details about these functions are reported in paragraphs 5.4.2, 5.4.3 and 5.4.4.

After the key variable *Seq* is complete, as reported in the subprocess blocks in Fig. 5.4, the Preprocessing loop requires an additional step: applying the function `findchangeslope`. This function has been built in order to identify any significant "change point" in the acquired signal. As already mentioned in [32], these points are statistically detected in correspondence of an abrupt variation of signal slope. The methods adopted by this function are described in paragraph 5.4.5. Typically not all the channels stored in *Seq* need to be analysed by `findchangeslope`, since some reference time-series can be taken into account to extract these particular points which delimit specific signal portions. For example, the Command channel, wheatear it is a position or current command (in closed or open loop respectively), is one of the channel which is always processed by `findchangeslope` since it does not depend by the unit response. The list of channels which need to be processed by this function are stored in the same configuration file which reports the names of the collected channels.

5.4.1 `extraction_channel.m` function

As reported in Listing 5.1, this function receives as input:

- `n_dwn`: downsampling factor;
- `fs`: sampling rate in Hz of the measured channels. This parameter is reported in measurement setting `conf_file.txt` and it is set to 100 kHz;
- `folder`: path of the measurement file;
- `chnls`: list of useful channels which need to be loaded and pre-processed per each *Sequence*. This list is included in an external configuration file of the selected PN;
- `meas_combo`: this particular parameter is reported in the `conf_file.txt`. Due to the setting of the NED test-console, it is possible to collect the channels from one additional LVDT, except the main-ram position one. For example, as reported in chapter 2, the Elevator unit includes two four-wires LVDT sensors for the EHSV and MSV spool.

Listing 5.1: Extraction channel function inputs

```
function Seq = extraction_channel(n_dwn,fs,folder,chnls,meas_combo)
```

The flowchart in Fig. 5.5 shows all the steps implemented within this function. Firstly, the code opens and read the measurement `cdf` following the path in `folder`. In order to open this file, the built-in Matlab function `cdfread` has been used. As reported in the official NASA website [52], this kind of file has been developed in order to store scalar and multidimensional large arrays (more than 2 GB), with their relative metadata for additional information storage. Both the measurement arrays and the metadata are saved in two dedicated structures:

1. `data`: cell array with N columns. Each column contains one measurement channel. This variable is converted in an equivalent matrix and then down-sampled according to the input factor `n_dwn`.
2. `var`: from the `cdf` metadata, the N measurement channels names are extracted and stored in this variable.

Once the measurement arrays have been loaded, the fundamental vector *Time* need to built knowing both the number of rows of the downsampled matrix `data` and the nominal sample-rate `fs`. This vector represent the first channel to be assigned at the output structur *Seq*.

Depending on the HFs to be extracted from these signals, not all of the columns stored in `data` need to be assigned in the output variable *Seq*. In fact, the input `chnl`s contains the necessary channels names to be exported and pre-processed. This selection improves the code performances in terms of execution time, since some of the collected channels are loaded in memory. However, the channels names order in the input `chnl`s may not coincide with the order in the extracted variable `var`. For this reason, each i – *th* channel name in `var` is compared with the content of the input `chnl`s. When the match between channel and input variable names is found, the corresponding field is created in *Seq*. This loop allows the user to write down in any order the names of the desired channels in the external file where `chnl`s is stored. Within this loop, an error handling function checks if the i – *th* column of the variable `data` is not empty or meaningful (f.e: made only by zeros), characteristic which denotes a correct measurement setting. An error would be printed in case of anomalous vector shape, which returns an empty field of the variable *Seq* and stops the code.

After defining all *Seq*, few more vectors needs to be added to this output variable: the demodulated signals from the integrated LVDT sensors of the unit. If the corresponding channels to primary and secondary voltages of these sensors are not empty, the main function recalls the appropriate demodulation Matlab scripts. As reported in the chapter 2.6, all the described PN includes a six-wires LVDT (or RVDT for the Elevator) for position monitoring of the main ram, which require a proper Ratiometric Demodulator to extract the demodulated signal from the raw primary and secondary coils voltage signals. A digital twin of this circuit has been implemented in the function `ratio_demod_6wires.m`. Its structure is presented in

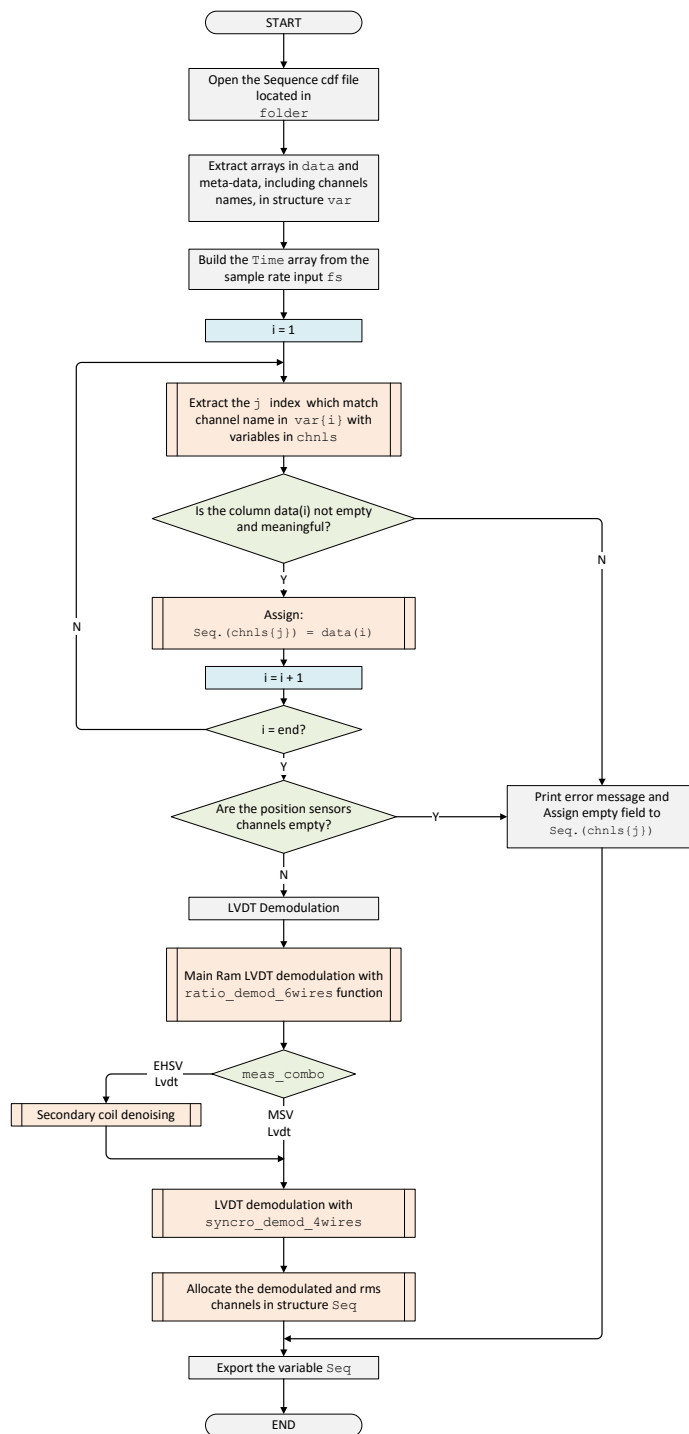


Figure 5.5: extraction_channel function flowchart

paragraph 5.4.2.

Except the Spoiler PFCS, both the Aileron and Elevator present at least one other four-wires LVDT: in the Aileron, it monitors the MSV spool position; while the Elevator is equipped with two of these sensors for both EHSV and MSV spool. In this case, the input variable `meas_combo` identifies if the collected channel secondary-wire measurement represents the position of one spool or the other. Therefore, its value depends on the *Sequence* goal, since this may be dedicated for detecting the health status of the EHSV or alternatively the MSV, and this can be easily deduced by the HFs list in the tables reported in chapter 2 per each PN signal. In both cases, the demodulation is obtained via the developed function `syncro_demod_4wires.m`; however, if `meas_combo` is referred to the EHSV LVDT, the secondary coil noise level needs to be filtered through an appropriate `denoising.m` function. The way the aforementioned scripts work are showed in the paragraph 5.4.3 and 5.4.4.

Once the sensors raw measurements have been demodulated, the output variable `Seq` is exported directly in the workspace and ready to be used by the corresponding HFs extraction script.

5.4.2 Ratiometric Demodulator function

Listing 5.2: Ratiometric Demodulator Function

```
function [U1_rms, U2_rms, U_demod] = ratio_demod(U_1, U_2, Kd)
```

This function follows the same steps of the block diagram in Fig. 2.25. It receives as input the raw secondary coils channels (`U_1` and `U_2`) and the demodulator factor `Kd`. The function gives as outputs both the rms time-series of the secondary coils, which have been obtained after rectifying and filtering the signals. A Low-pass second order Butterworth zero-phase lag Filter has been implemented within this function using the built-in Matlab functions `designfilt` and `filtfilt`. The final position in millimetre is represented by the output `U_demod`, multiplying the ratio R per the demodulator parameter `Kd`.

5.4.3 Signal Denoising function

Listing 5.3: Denoising Function

```
function chan_denoised = denoising(trgt, noise, fs, F_NOM, TRSHLD)
```

The main goal of this function is to eliminate any noise contribute of the channel `target` according to noise content of a reference one, represented by the input

noise. In this case, as already anticipated in paragraph 5.4, the channel to be cleaned is the EHSV LVDT secondary coil, while the reference noisy signal is the command current I_{cmd} (Fig. 3.6a).

This channel represents the link between control loop and EHSV first-stage. Any kind of external disturbance due to the testing condition may leave a trace on the spectral profile of this channel. For example, if the NED console is close to any source of external high frequency electro-magnetic noise (such as other consoles or particular devices), the traces of this interference may be visible in the main output of the test console, so the current command. In the same way, if during a test the NED console is too close to the hydraulic bench, any source of low frequency vibration from this latter component can affect the entire current command measurement. Furthermore, within the connection cable between measurement system and unit, the input current command and the output LVDT voltage are measured from different pins of the same integrated electrical socket, which increase the possibility of any interference between these two channels.

The only way to detect if any noise contribute of I_{cmd} affects the measurement of the LVDT secondary voltage U_{sek} is to compare their frequency spectrum, and identify any similarity between them around frequency ranges which are significant for the channels itself. This frequency spectrum comparison is the core concept of the function **denoising**. The four inputs in the Listing 5.3 are:

- **trgt:** target channels to be denoised, so the channel `Seq.U_sek`;
- **noise:** source of possible noise contributes, so the channel `Seq.I_EHSV`;
- **fs:** sample rate frequency of the input channels;
- **F_NOM:** significant frequency of the target channel.
- **TRSHLD:** minimum threshold which identifies any similarity in the channels frequency spectrum.

This latter input is significant if the target signal presents any fundamental harmonic which need to be taken into account and avoided to be filtered out. In this case, the LVDT secondary voltage, even if noisy, is surely characterized by a principal harmonic which has the same supply frequency of the primary wiring. For this reason, the function avoid to cancel this fundamental contribute. In this case, the function has been developed in a way to clean the signal from higher frequencies than **F_NOM**, and keep the others below this level which can also be a sign of anomalous spool oscillation or electrical LVDT fault.

The flowchart in Fig. 5.6 shows the function steps and the principle Matlab built in functions which are recalled, in particular **mscohere**, **findpeaks** and **bandstop** filter, which are all included in the *Signal Processing Toolbox*.

The plots in Fig. 5.7 shows the normalized I_{cmd} and U_{sek} channels and their relative

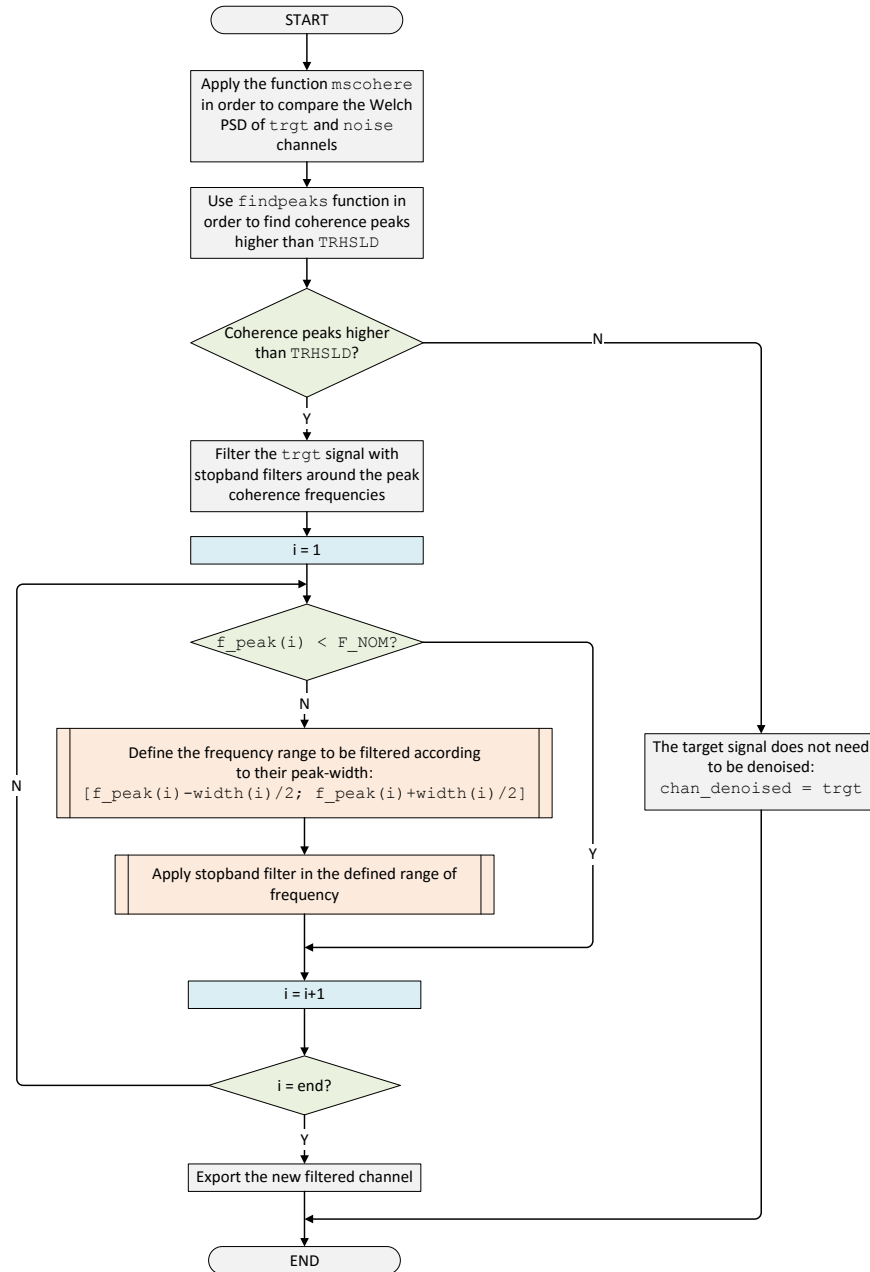


Figure 5.6: denoising function flowchart

Power Spectral Density (PSD) plots of the 15-th *Sequence* of a particular tested Elevator unit. In this case the **denoising** function has been effective, since both the channels have shown similarities in their PSD spectrum. Both the time-series have been Z-score normalized, centering and scaling the vectors in order to have null mean and unitary standard deviation:

$$I_{norm} = \frac{I_{cmd} - mean(I_{cmd})}{S_I} \quad (5.1)$$

$$U_{norm} = \frac{U_{sek} - mean(U_{sek})}{S_U} \quad (5.2)$$

where S_I and S_U are the respective standard deviation of the noisy sampled vectors. This normalization retains the shape properties of the original signals. Even after normalization, due to the noisy and time-varying nature of the input signals in Fig. 5.7, a PSD analysis for both the inputs allows a better comparison of the noise content of both the signals. Figures 5.7b and 5.7d show the Power Spectra of the corresponding channels, evaluated using the Welch method. Both the plots have been zoomed around 10 kHz and it is already possible to observe similar shapes in the power spectra around the nominal supply frequency $F_NOM = 1953Hz$ and its even harmonics. These similarities are detected using the built-in function `mscohere`.

This goal of this function is to evaluate the magnitude-squared coherence of the PSD estimation of the input channels, according to the Eq. 5.3:

$$C_{IU}(f) = \frac{|P_{IU}(f)|^2}{P_{II}(f)P_{UU}(f)} \quad (5.3)$$

where the subscripts I and U are referred to the target channel U_{sek} and the noisy reference one I_{cmd} . The ratio in Eq.5.3 represents the how correlated the cross Welch PSD of both the channels is with the single Welch PSD estimation of the input channels.

The Welch PSD estimator [53] is an approach for spectral density estimation, which is particularly effective since it starts from the same Periodogram evaluation as [54], but evaluated and averaged splitting the time-series vector in adjacent and usually overlapping segments. These portions of signals are multiplied by a window function which need to be accurately chosen. Fig 5.8 reports different kind of windows functions which are typically used in Welch method.

For example, the plots in Fig. 5.7b and 5.7d have been obtained splitting the time-series vectors on the left side with five Blackman windows with a 50% of overlapping between adjacent segments. Due to this 50% overlap, the tapered values at the beginning and end of the window in one segment are half-length window far from the adjacent segment limits. In this way, the loss of information

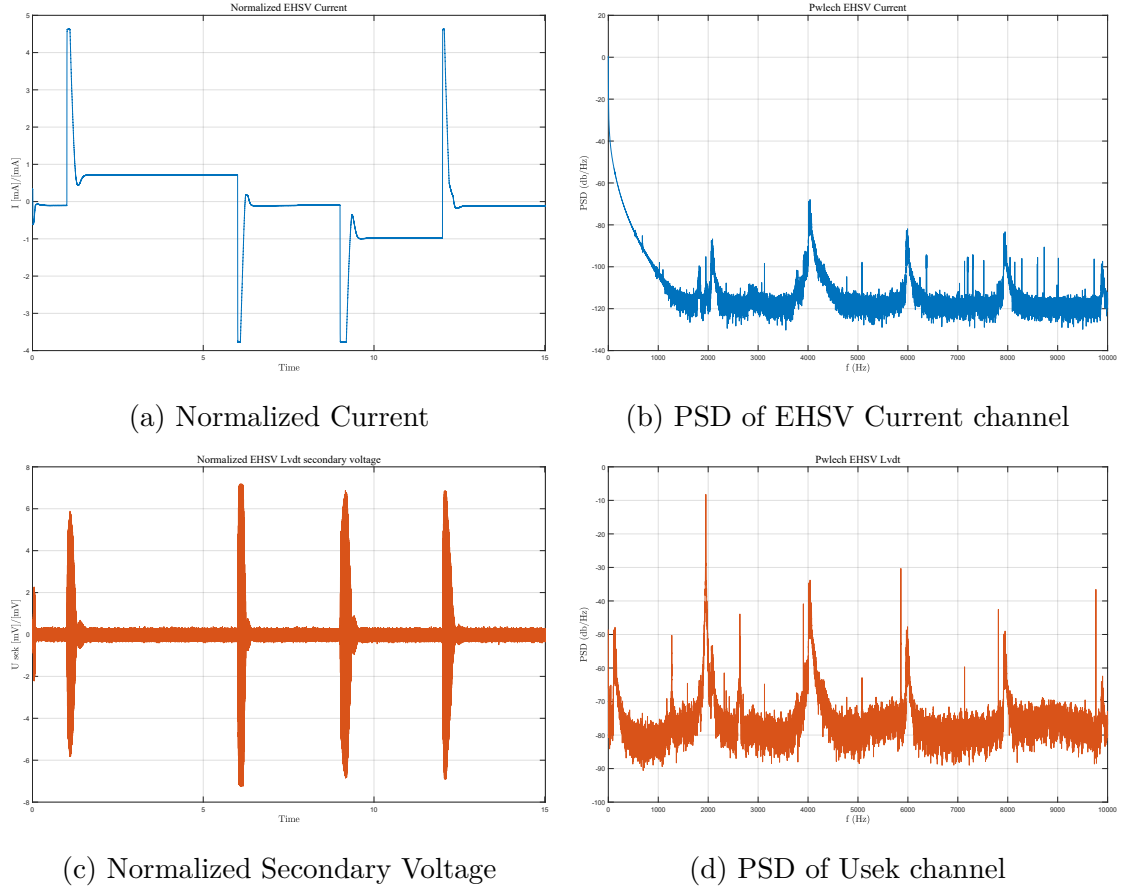


Figure 5.7: Time series inputs channels normalization and PSD analysis

due to the segmentation is avoided by the shape and overlap setting. As reported in [55] and [56], the Blackman presents different advantages, compared to other similar ones such as the Hamming and Hanning. Fig. 5.8 shows the shapes of five typical window function, imposing a number of points $N = 1024$. Compared to the Hamming and Hanning, the trade-off between main lobe width and side lobe attenuation makes the Blackman one a good choice for tapering discontinuities due to the segmentation.

The same kind of window has been used in the function `mscohere`, with $N = 1024$ (as in Fig. 5.8) and 50% of overlap, in order to calculate the PSD terms of Eq. 5.3. The plot in Fig. 5.9 shows the function results and the frequency ranges where the coherence coefficient C_{IU} overcome the accepted threshold `TRHSLD`, represented by the red-dashed horizontal line.

The black-vertical dashed line shows the reference frequency `F_NOM` position in the range of available frequencies for this analysis. It is important to notice that the maximum analysed frequency by this function is half of the effective sample-rate

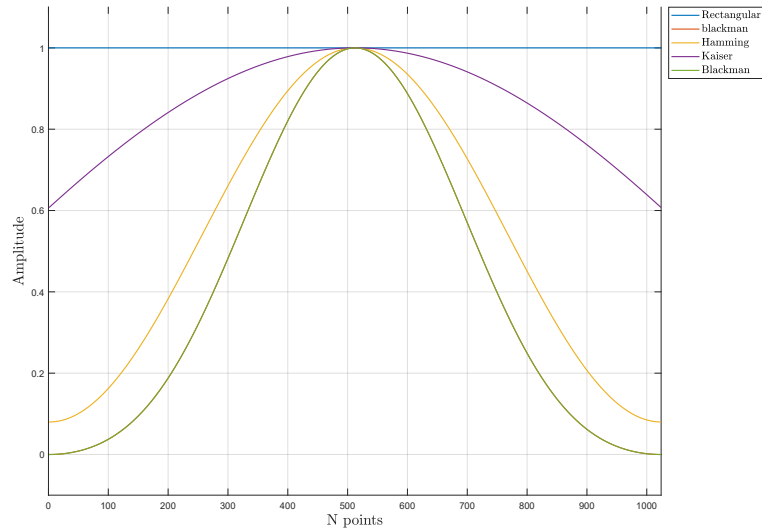


Figure 5.8: Windows Functions

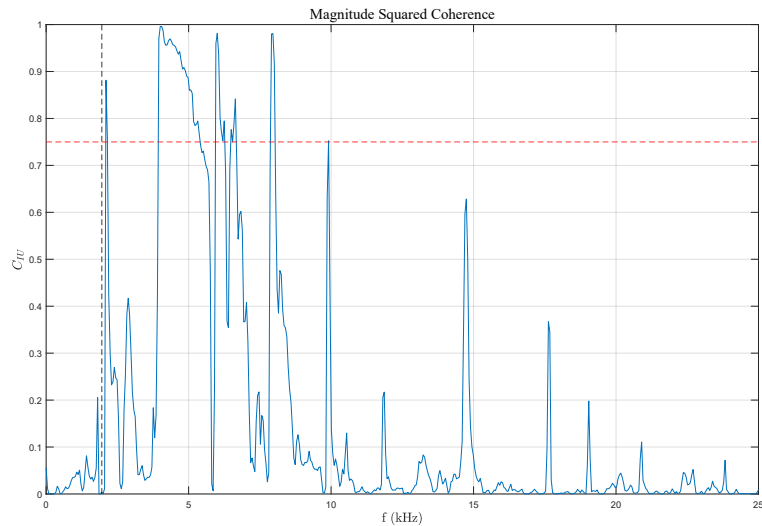


Figure 5.9: Magnitude Squared Coherence Coefficient

f_s input, which is 50 kHz due to the previous downsampling in the first step of `extraction_channel` function. The plot confirms the previous deduction obtained by looking the Welch PSD plots in Fig. 5.7b and 5.7d: both the channels present a similar contribute of noise close to the nominal frequency (black-dashed line) and in the range of its even harmonics between 4 and 10 kHz.

Due to the different nature of the input channels, the presence of similar noisy harmonics indicates that the I_{cmd} channel from the NED console affects the read

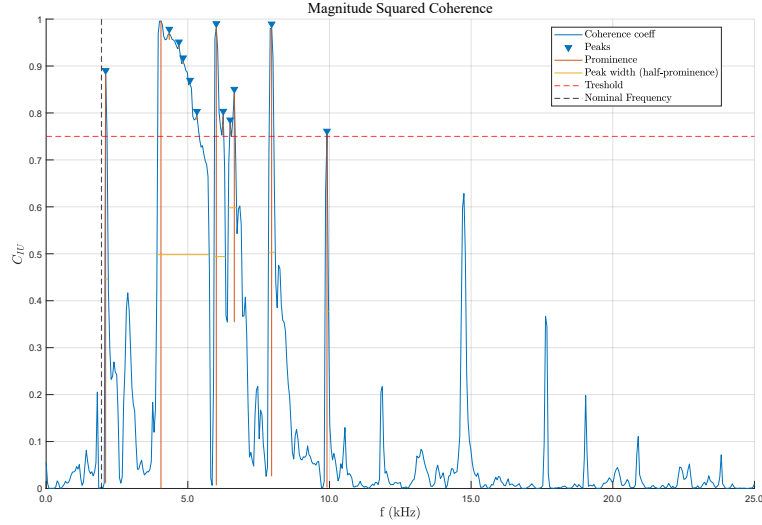


Figure 5.10: Magnitude Squared Coherence Peaks

channel U_{sek} from the internal LVDT sensor. The peaks above the red-dashed line need to be correctly identify and filtered in order to clean the `trgt` input.

The built-in function `findpeaks` has been tuned in order to extract the aforementioned peaks above the set threshold. As shown in Fig. 5.10, the function is able to find the peaks magnitude, its frequency coordinate, its prominence and width. These two latter features are particularly important to identify the range of frequencies that need to be filtered out by the `trgt` signal. The peak prominence measures how much the selected peak stands compared to the other close peaks, according to its height and location. In Fig. 5.10 each selected peak prominence is marked with a continuous vertical line. According to this value, the width of each peak is evaluated at half of each prominence. The `findpeaks` function outputs the width array as real number: once the position of each peak and its width are known, the extreme values $[f_{pk} - width/2; f_{pk} + width/2]$ identify the range of optimal frequencies which need to be filtered out by the stopband filter. These ranges are locally stored in an appropriate cell-array variable `stopband`.

As reported in the flowchart in Fig. 5.6, if there is not any peak which overcome the set threshold value, there is technically no need to filter the target U_{sek} channel, since it is not affected by the noise of the reference I_{cmd} channel. Therefore, the `stopband` variable would be empty and the `denoising` function outputs the original input channel.

Like in Fig. 5.10, both the peaks amplitude, coordinates and width are locally stored and the function continues to process the `trgt` input according to the loop in Fig. 5.6. Firstly, as already mentioned before, the peaks which are lower than

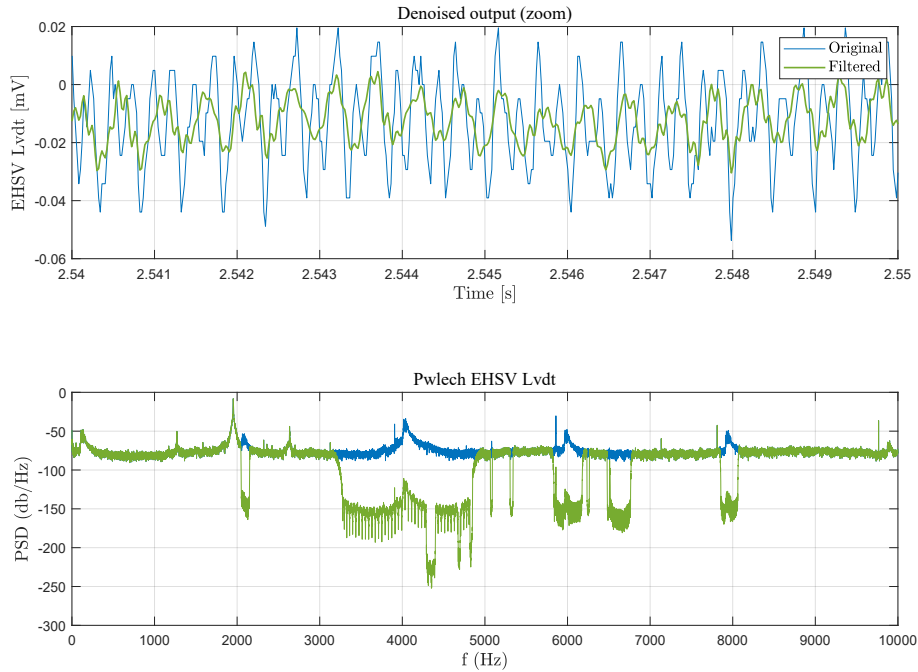


Figure 5.11: Filtered signal with denoising function

the input `F_NOM` are excluded from the filtering process, since they can be symptoms of mechanical or electric fault of the EHSV itself that need to be detected. Filtering out high-frequency noise contributes instead would allow a more precise demodulation of the LVDT channel, and therefore an accurate estimation of second stage spool position. For this reason, within the `stopband` variable, only the ranges which do not contain the reference frequency `F_NOM` constitutes the input of stopband filter, defined through the built-in function `bandstop`.

The filtering results are illustrated in Fig. 5.11. The first subplot reports a zoom which highlights the difference between the original noisy U_{sek} (in blue) and the final denoising output, after it has been filtered with consecutive stopband filters. The second subplot instead shows the difference between the original PSD spectrum and the final filtered one, highlighting how the ranges of frequency in Fig. 5.10 have been correctly identify and filtered by consecutive filtering process.

5.4.4 Synchronous Demodulator function

Listing 5.4: Synchronous Demodulator Function

```
function U_demod = syncro_demod(U_prim, U_Sek, K_d)
```

This function is used to extract the position measured by the four-wires LVDT which are present in both Elevator and Aileron units. This function reproduces the steps of a synchronous demodulator as in Fig. 2.23. As already reported in the chapter 2.6, these position sensors are more affected by external noises if compared with the six-wires ones. In Appendix A.1 it is demonstrated how the synchronous demodulator circuit is particularly affected by the odd-order harmonics effect of the carriage frequency, both in terms of amplitude and phase. Other demodulation techniques have been studied in order to reduce this effect, such as Lock-in Amplifier method [43] or DSP-Based approach [45]. However, for sake of coherence with the analogue circuits adopted in the test-consoles, the classic synchronous demodulation technique has been adopted. As already anticipated in the previous paragraph, in the particular case of the Elevator EHSV LVDT, this function is always preceded by the `denoising` in order to filter high-frequency noise.

The functions receives as inputs the primary `U_prim` and secondary wirings voltages `U_sek`. As in the aforementioned flowchart, this latter time-series is rectified according to the sign of the reference primary voltage, filtered using a Butterworth low-pass filter and multiplied per $\frac{\pi}{2\sqrt{2}}$ to obtain the final rms value. The final gain `K_d` translates the voltage channel into the final position. For example, in case of the Elevator, this function can be used for both EHSV and MSV spool. The value of `K_d = 1` in case of MSV LVDT channel, since the final value has to be in Volt, while for the EHSV spool the value depends on an additional CMM constant and need to be multiplied per 1000 to obtain the result in mV: $K_d = 1000 * K_{CMM}$

5.4.5 findchangeslope function

Listing 5.5: Findchangeslope Function

```
function [ipt, downsamp_chnl] = findchangeslope(channel,time,n,C)
```

The function presented in the listing 5.5 is characterized by the following inputs:

- `channel`: raw channel to be analysed;
- `time`: reference `Seq.Time` vector;
- `n`: optional downsampling factor to accelerate the function. Its default value is $n = 1$, which means that the signal is not additionally downsampled;
- `C`: optional threshold constant, with default value $C = 2$. Reducing this value increases the function detection sensibility.

The function returns as output the change-points indexes in the variable `ipt` and the eventual downsampled matrix `downsamp_chnl`, which includes as first column the downsampled time and as second column the corresponding channels. As reported in [32], [18] and [57], the function goal is to find the minimum number of point which determines local ranges where the signal shape can be linearly approximated with the minimum error.

For sake of clarity, it is better to explain the algorithm starting from supposing that it has identified N final change-points. This means that the input `channel` can be linearly approximated by $N + 1$ segments x_i :

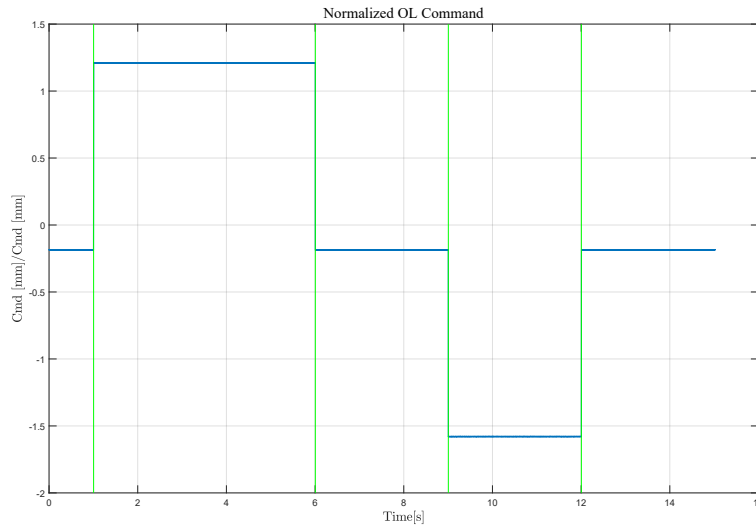
$$x_i = \alpha_j i + \beta_j + \epsilon_i \quad (5.4)$$

In Eq. 5.4, α_j and β_j are the linear regression coefficients in the interval between two consecutive change points $[ipt_j, ipt_{j+1}]$ and ϵ_i is the approximation error i -th segment. This latter value is used to define the cost function to minimize in order to find the optimal number of points ipt_j with $j = 1 \dots N$ in which each segment fit the original signal:

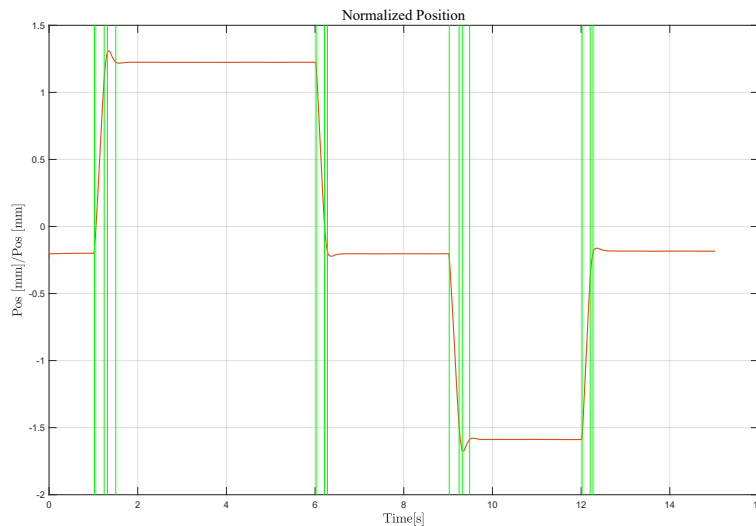
$$J_{opt} = \sum_{j=0}^N \epsilon_j^2 + C \cdot \sigma_x^2 \cdot N \quad (5.5)$$

In Eq. 5.5, the first term represents the squared error sum, while the second term is defined in order to avoid overfitting. Considering just the first sum, the optimal solution would be that every segments between two consecutive sampled points can be considered an optimal solution, therefore every point is included in the output `ipt`. The second term increases proportionally with the number N and with the squared variance σ_x^2 , so it depends on the statistical distribution of the sampled point around its linear approximation x_i . The coefficient C is the threshold constant which increases the cost of any additional change point, so it reduces the number of final change points. This value is set by default $C = 2$, but it can be set as input of the function in listing 5.5.

Fig. 5.12 illustrates how `findchangeslope` identifies the change points of two measured channels. The function has been applied both to the command signal and main ram displacement measured by the position sensor (Fig. 5.12b): each time instant corresponding to the change points is highlighted with vertical green lines. As expected in the first subplot in Fig. 5.12a, the Command is mathematically defined by the signal shape, so its change points are the ones at each step which are defined for the *Sequence* shape itself. This observation can be extended to all the Command signals both in open and closed loop. The second subplot in Fig. 5.12b instead shows how the function correctly identifies any significant change of slope in the measured position signal. For example, in this case, the tested unit is responding coherently to the command. However, it presents unexpected overshoots both during rising and falling steps which are correctly detected by the `findchangeslope`



(a) Normalized CL Command with its change points



(b) Normalized Position with its change points

Figure 5.12: Change point detection (example with Sequence n.15)

function, which identifies additional change points in correspondence of each overshoot peak. These informations may be used to extract important HFs which can help to detect anomalous behaviour.

This latter example shows how it is possible to extract two kind of signal features from the results of `findchangeslope`: those related to a sudden change and those calculated while the signal is stable on a certain level. These observations can be extended to all the collected channels. For this reason, two additional functions have

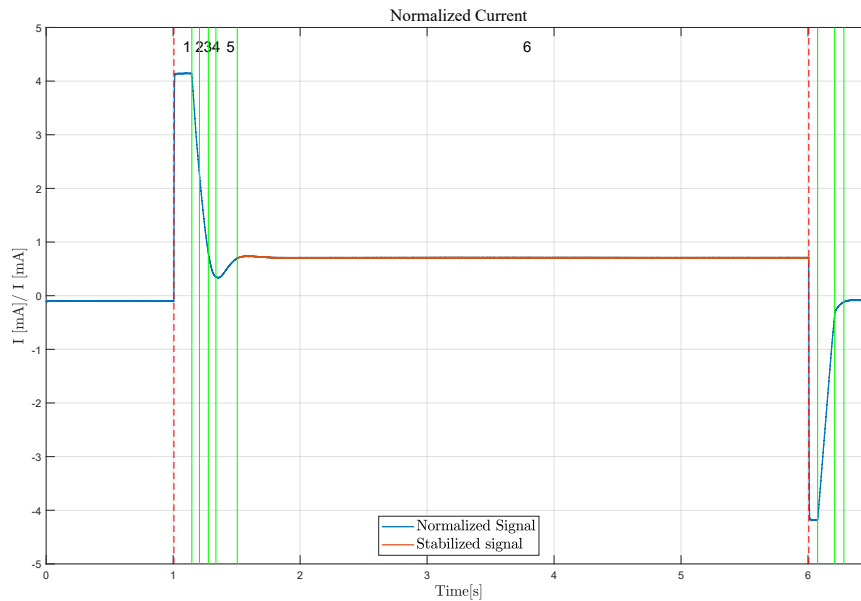


Figure 5.13: `stab_finder` example of Sequence n.15 Current signal (zoom)

been written for these purposes: `move_finder` and `stab_finder`. These functions are often used in each HF's Sequence extraction script in order to isolate specific portion of signals and extract the corresponding signal feature.

`stab_finder` function

Listing 5.6: Stable signal finder Function

```
function [t_stab_in,t_stab_end] = stab_finder(channel, ipt, dt, T_in,
      T_end, tol_slope, option)
```

The code presented in the listing 5.6 presents the complete set of inputs which can be specified for the function, including the optional ones:

- `channel`: channel to be analysed;
- `ipt`: vector of change points indexes, output of `findchangeslope` function;
- `dt`: minimum range of time where the signal needs to be stable;
- `T_in` and `T_end`: optional parameters which can be used to specify manually a time window. By default, these two values are the initial and final element of the time vector;

- `tol_slope`: optional input, which can be used to specify a maximum slope tolerance in absolute value;
- `option`: optional string input. If 'first' is selected as an option, the first stabilized time interval is selected, while if 'stable', it just pick last stabilized time longer than `dt`.

Fig. 5.13 shows how the function works, taking in account the EHSV command current in the same *Sequence* of the plots in Fig. 5.12. In this example, it has been considered a time window defined by the first two change points in Fig. 5.12a, considering just the first rising and falling positive position steps. These two instants are easy to extract as first two elements of the vector ipt_{cmd} , obtained from applying `findchangeslope` to the closed loop command channel. They have been highlighted with red dashed lines in Fig. 5.13 and they represents the optional inputs `T_in` and `T_end` of `stab_finder`.

Within this time window, the function identifies six time intervals which are identified in the plot with numbers from 1 to 6. All these portions of signals need to be scanned in order to identify those which last more than the input `dt`. If `tol_slope` is set as input, it represents an additional requirement in terms of minimum acceptable slope. The function checks which of the identified intervals respects the criteria of minimum length and eventual acceptable slope. In this case, the interval n. 6 is selected (continuous red line in Fig. 5.13). In this particular case, the average value of the interval n. 6 identified by `stab_finder` represents the biased nominal current in extended position *IE*, a fundamental HF which is used to identify any anomaly in the Elevator Recentering Device. The same analysis can be executed considering the time window defined by the last two elements of input ipt_{cmd} : in this case, the function `stab_finder` would isolate the current portion for the extraction of *IR* in retracted position.

`move_finder` function

Listing 5.7: Signal Variation Function

```
function [t_move_in,t_move_end,slope_move] = move_finder(channel,
    ipt, T_in, min_slope)
```

The function in the listing 5.7 is characterized by the following inputs:

- `channel`: channel to be analysed;
- `ipt`: vector of change points indexes, output of `findchangeslope` function;
- `T_in`: starting time;

- `min_slope`: minimum slope required in the time interval;

The function returns as output the corresponding initial and final instants of the time (`t_move_in`, `t_move_end`) selected window of time with a measured `slope_move`.

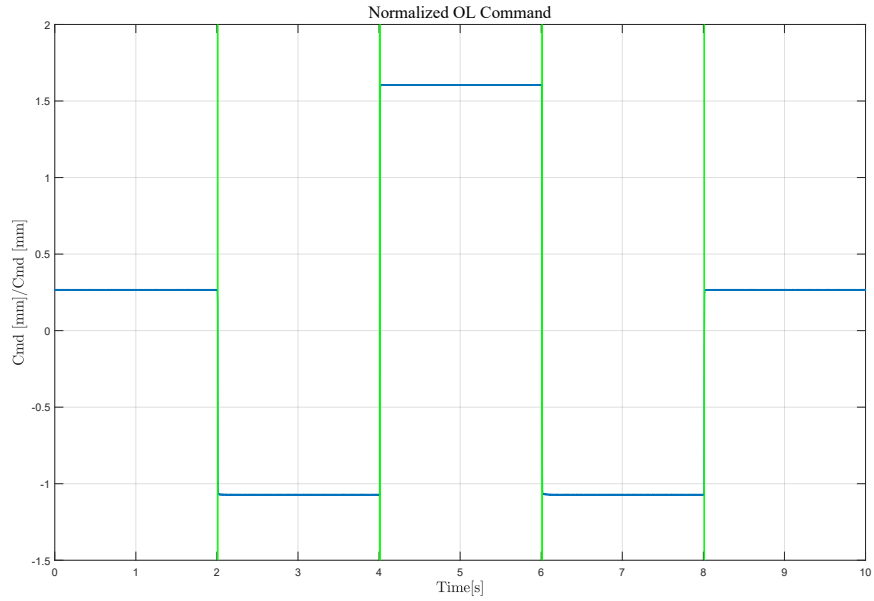
The plots in Fig. 5.14 represents the open loop command (Fig. 5.14a) and the corresponding main ram position (Fig. 5.14b) of *Sequence* n.7, which is used for all the PFCS. This signals results particularly useful for explaining how the function `movefinder` works to extract two important HFs: the maximum speed during complete open loop rising and falling command steps.

As usual, the green vertical lines represents the change points identified by `findchangeslope`. In the zoom window in Fig. 5.15, the extraction of the maximum speed in extension step depends on the `move_finder` setting. In this case, the red-dashed line represent the input `T_in` which can be easily extracted as the second `ipt_cmd` index in Fig. 5.14a. This value actually excludes from the analysis the intervals 1 and 2 in Fig. 5.15, so the sections 3, 4 and 5 are consecutively analysed. For these three remaining signal portions, the function checks which one is characterized by a slope higher than the input parameter `min_slope`. In this case, this value is easy to tune the minimum allowed speed is known as design parameter of the PFCS. The function is able to extract the red line in section 3, and it outputs both the initial and final instants, their corresponding indexes. In this range, `move_finder` linearly interpolate the respective portion of signal: the resulting linear slope is exported as output `signal_slope`.

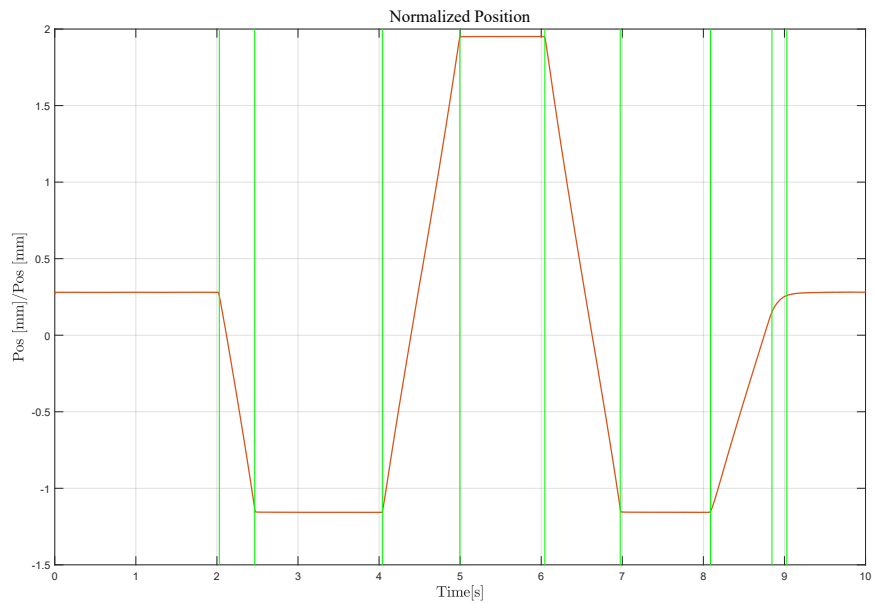
5.5 Sequences Signals HFs Extraction

In Chapter 4 it has been show how each PFCS Automatic Procedure is characterized by a well mathematical concatenation of signals called *Sequences*. Each of these sections has been designed in order to extract automatically a set of peculiar HFs which are significant to detect the UUT health status. Within the software structure, each *Sequence* presents its dedicated HFs extraction script which is recalled by the `main` (flowchart in Fig. 5.4). Due to the modular nature of the signal generation code, several *Sequences* are in common between Elevator, Aileron and Spoiler: this means that most of the HFs extraction codes can be used for all the PFCS with minor adjustments if needed. This improves both the complete code structure and readability.

Following the same line of Chapter 4, the HFs extraction codes would be divided in four main groups: Preliminary, Pressure Adjustment, Leakage and Automatic Sequences. It is also important to remember that the codes in the next paragraphs are shared by both DBG and PV HyDiag Software versions in Fig. 5.1, so they are



(a) Normalized OL Command with its change points



(b) Normalized Position with its change points

Figure 5.14: Change point detection (example with Sequence n.7)

referred the Test Field measurements. As already mentioned, due to some technical limitation of the Test Field benches, it is not possible to extract in a complete automatic way all the HFs from the collected signals. This means that several

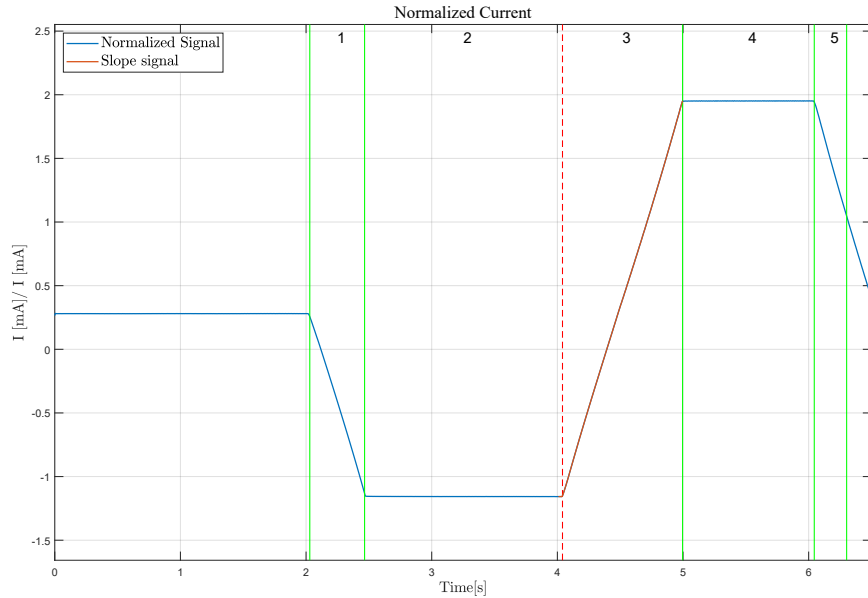


Figure 5.15: `move_finder` example of Sequence n.7 Position signal (zoom)

HFs, such as pressure or flow values, have to be manually reported by the operator through an appropriate testing GUI in LabView. These values are saved at the end of the testing session in a proper `INDEX.txt` file which is stored within the respective measurement data in the local backup HDD (Fig. 5.1). However, it is worthy to mention that this solution is just a temporary one in order to implement this innovative automatic procedure on systems which have been predominantly manual. For example, in the H-Lab test bench it is possible to run each PFCS Hy-Diag procedure in full automatic mode, since it is possible to remotely command the bench and all the measurement channels are digitalized.

The plots in the following sections have been properly normalized using the following criteria:

- The position command and measurement channels (from both external INC sensor and Main ram Sensor) have been scaled using as reference the nominal unit middle stroke $mid_stroke[mm]$;
- The current open loop command and measurement channel have been normalized using the reference maximum current $I_{max}[mA]$;
- The supply pressure channel represents the manual command from the operator. In particular, it changes during the Pressure Adjustment Sequences (as reported in Chapter 4). The plots shows its normalized behaviour to the maximum supply pressure $P_{max}[bar]$;

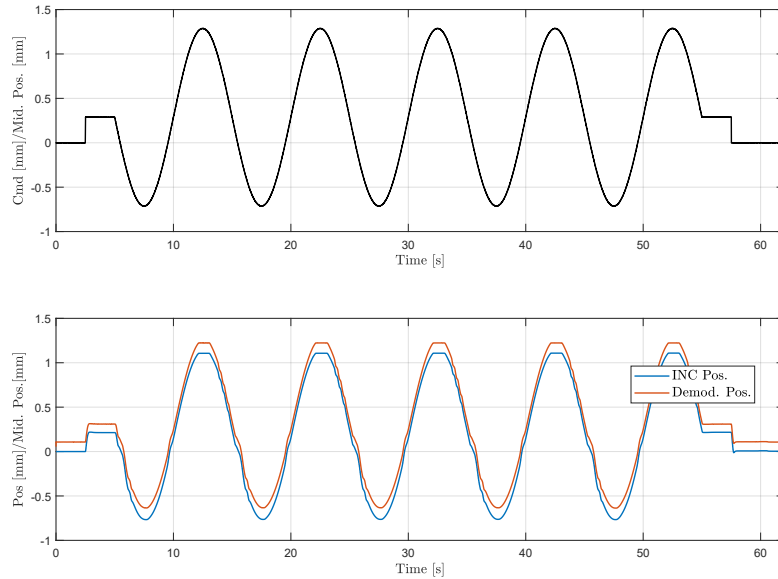
- The Electrovalve command channels of both Aileron and Elevator have been normalized according to their maximum supplied voltage $V_{max}[V]$;
- The MSV LVDT demodulated channel of both Aileron and Elevator represents the MSV spool position during the transition from active to damping phase and viceversa. Its time-series have been reported scaling the plots according to the nominal maximum position in Volts $MSVLVDT_{max}[V]$;
- The Elevator unit includes an additional position sensor for the EHSV second stage spool: also its position channel has been scaled using the maximum available position $EHSVLVDT_{max}[mV]$.

In addition, per each *Sequence* a recap table is reported which summarize the extracted HFs, the possible failure linked to each anomalous value and if they can be extracted per each tested PFCS. As already anticipated, both traditional and additional HFs have been designed in order to help the failure detection in a particular component of the UUT. In order to quantify how critical this anomaly can be, the extracted HFs values need to be compared with some reference ranges. Regarding the traditional CMM HFs, these limits are directly reported in this manual and they need to be respected in order certify a unit. The limits of these ranges will be reported as HF_{min} and HF_{max} per each of them. However, in Chapter 6 a further investigation about these limits is presented which identify risk areas within the suggested ranges. About the additional HFs, properly designed to improve the diagnose, these limits have been experimentally defined taking as reference a sample of recertified units, which have been completely repaired and ready to be installed on the aircraft.

5.5.1 Preliminary Sequences

The Preliminary *Sequences* signals descriptions are reported in Chapter 4. Both Elevator, Aileron and Spoiler include in this group the *Sequences* n.1 as reported in paragraphs 4.3.1 and 4.3.3. Apart from this difference, *Sequence* n.1 has been designed for all the PFCS to execute two important preliminary test:

1. The first one is a CMM HFs which need to be collected: the presence of any external Leakage while cycling the unit with sinusoidal command at a specific frequency. This is one of the HF the technician need to report manually during the when the unit has cycled for 10 complete strokes. Its value is reported in the aforementioned `INDEX.txt` file as boolean value: True if any leakage has occurred, or False if viceversa. The presence of any external leakage is a sign of worn out sealing, depending on where the leakage has been detected. The graphs in Fig. 5.16, 5.17 and 5.18 report the collected command in the first subplot, and the corresponding main ram position measured by the external INC sensor and the internal Position sensor in the second subplot.


 Figure 5.16: Elevator *Sequence 1*

2. A second important check regards the correct configuration of the external INC sensor for main ram position check, as reported in the flowchart of Fig. 5.4.

This latter check on the INC sensor setting consist in two main steps. Firstly, it is necessary to check if the technician has correctly connected the INC sensor to the measurement system. For example, in case of wrong cable connection, this channel would report by default a time-series with null values. The code needs to check if the INC sensor measurement vector is correctly following the shape of *Sequence n.1*, otherwise this channel would be substitute in the further sequence by the Position sensor demodulated channel. In this way, it would always be possible to extract the HFs regarding to unit position, even though the main ram sensor could not be reliable since it can be affected by a failure as well. For this reason, the HyDiag software would anyway execute the complete analysis of the signals but it would print an appropriate error message in the final report to alert the technician to repeat the measurement with the correct INC sensor to obtain more reliable results.

After checking the coherence between position and command, the script checks if the INC sensor initial offset has been correctly set within a tolerance of $\Delta inc_{max} = 0.01mm$. This threshold has been calibrated according to the minimum tolerance regarding the CMM HFs. This check is necessary since the operator has to correctly align the INC zero at the correct main ram null position before starting any test. If

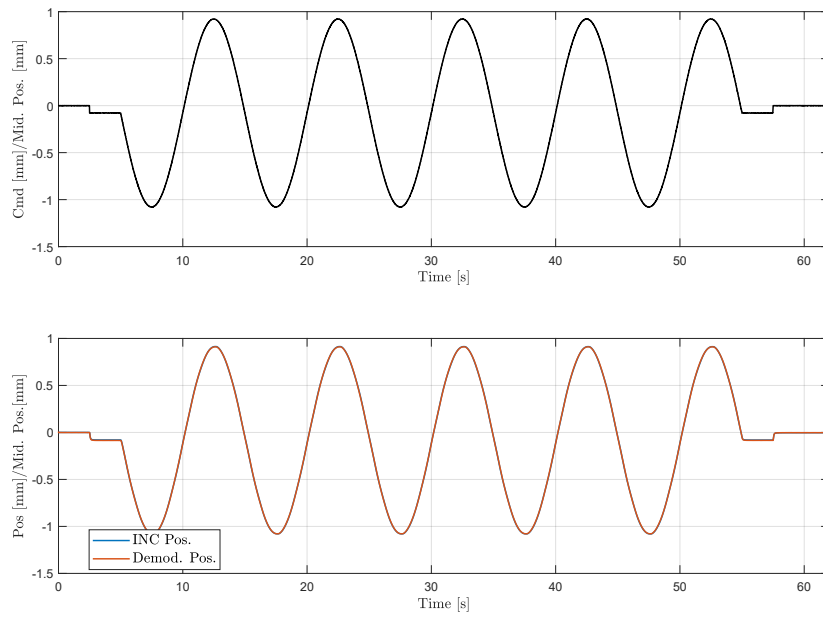


Figure 5.17: Aileron *Sequence n.1*

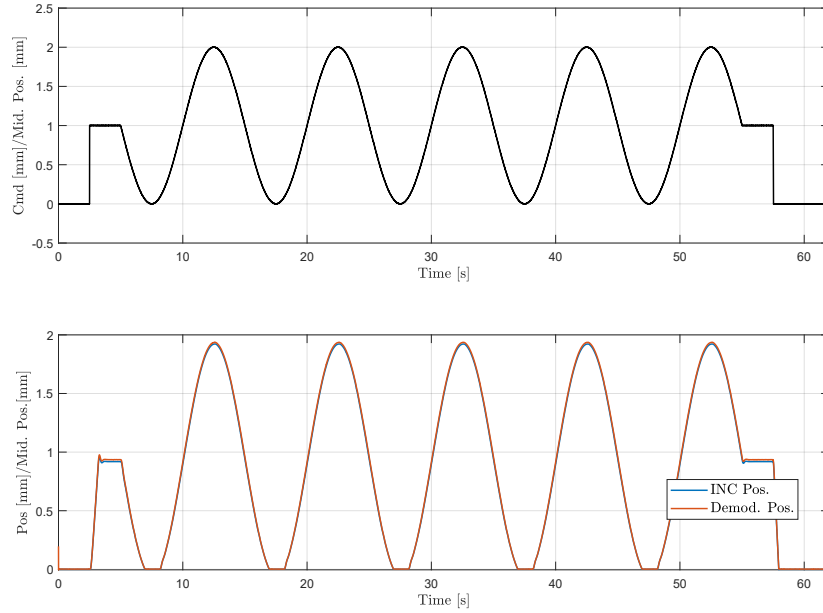


Figure 5.18: Spoiler *Sequence n.1*

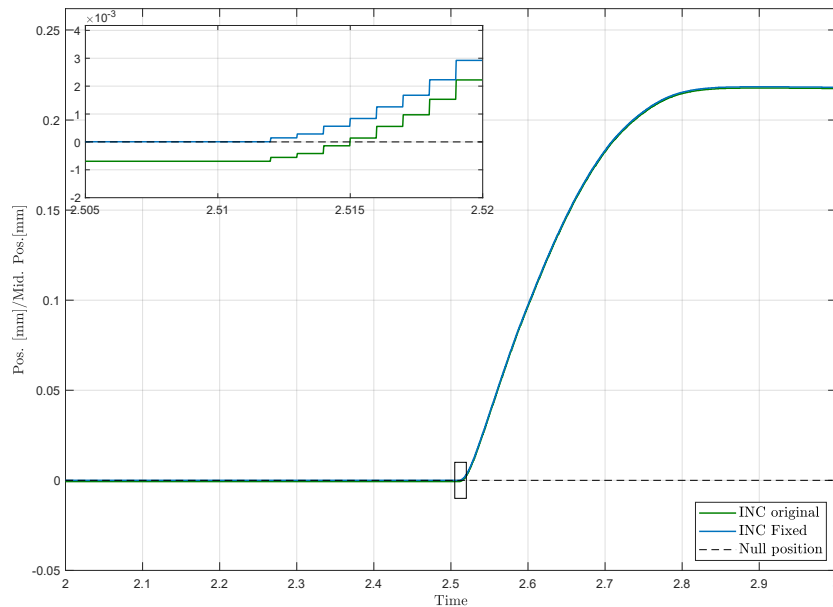


Figure 5.19: INC sensor fixing

this procedure has not been correctly executed before starting, the postprocessing *Sequence* n.1 code is able to correct this offset, knowing the nominal zero position of the unit.

For example, Fig. 5.19 shows how the code fixes the small offset (around 0.03 mm) between the original signal (green line) and reference 0 position, shifting the time-series as the blue line. This latter channel becomes then the reference in order to evaluate the main ram position sensor offset, as shown in the second subplot of Fig. 5.16. The other two plots in Fig. 5.17 and 5.18 illustrate the same collected measurement for *Sequence* n.1 for both Aileron and Spoiler.

Within this Preliminary group, the Elevator and Aileron include also the *Sequence* n.2 which aims to check if the accumulator is empty, component which is not installed on the Spoiler. If the accumulator does not completely discharge during this test, the collected HF would be $accu_{drain} = False$, which can be sign of a problem within the accumulator piston or on the component draining port. This latter unit presents in this preliminary group the tailored *Sequence* n. 36, which aims to manually check if the Inhibiting Valve switch works properly. For both these sequence, the extracted HFs have to be manually reported by the operator through a designed dialogue-window in LabView where the appropriate HF checkbox is reported. The corresponding values are reported in the aforementioned `INDEX.txt` file as boolean values: True if manual test has been successful, or False if viceversa. In case of False result, it means the operator was not able to manually switch

correctly to Maintenance "M" mode, so there can be a mechanical problem of the switch itself or a clogging of the inhibiting valve. In Table 5.1 it is possible to read a recap of the Preliminary Sequences per each PFCS, the extracted HFs with relative check to perform and eventual troubleshooting extraction in case of anomaly according to the result of Check column.

Table 5.1: Preliminary Sequences HFs Recap table

Seq	HFs	Check	Troubleshoot	E	A	S
1	<i>leakage</i>	<i>leakage = True</i>	External sealing degradation	✓	✓	✓
	Δinc	$\Delta inc \geq \Delta inc_{max}$	Incorrect setting of the INC sensor	✓	✓	✓
2	<i>accu_drain</i>	<i>accu_drain = False</i>	Problem in accumulator or in draining port	✓	✓	
36	<i>IV1</i>	<i>IV1 = False</i>	Mechanical check of manual switch in "M" mode or clogging of the inhibiting valve			✓
	<i>IV2</i>	<i>IV2 = False</i>				✓

5.5.2 Pressure Adjustment Sequences

This paragraphs describes the adopted methods for HFs extraction of Pressure Adjustment *Sequences* reported in Chapter 4. These sections are all characterized by a regulation of supply or return pressure. From each *Sequence* set of signals, several HFs can be extracted by each of them that are not strictly pressure values, but influenced by the UUT supply and return conditions.

Table 5.2 summarizes the *Sequences* which are included in this group. As it is possible to observe first three are in common between Elevator and Aileron units, due to their similar structure. The sixth *Sequence* has been tailored to identify any anomaly in the Elevator Recentering Device component. The *Sequence* n.7 is instead adaptable for all project uses-cases since its main goal is to determine the main piston maximum speeds during two complete strokes of the unit. The last two (*Sequences* n.27 and n.29) are instead characterizing for the Spoiler unit, and they both are preceded by two appropriate adjustments, which have to be manually executed by the technician to switch the working mode from Maintenance to Operating Mode and back to Active (Paragraph 4.3.3). Similarly to Table 5.1, both the Checks to be performed and relative Troubleshooting indications are reported. The next subsections shows in detail how each of the listed HFs has been extracted.

Table 5.2: Pressure Sequences HFs Recap table

Seq	HFs	Check	Troubleshoot	E	A	S
3	$P2$	$P2 < P2_{min}$	Check the HP valve spring	✓	✓	
		$P2 > P2_{max}$	Check the HP valve spring or Filter Clogged			
	$P3$	$\Delta P3 < \Delta P3_{min}$	Check the LP valve spring	✓	✓	
		$\Delta P3 > \Delta P3_{max}$	Check the LP valve spring or Filter Clogged			
4	$P5$	$\Delta P5 < \Delta P5_{min}$ $\Delta P5 > \Delta P5_{max}$	Check MSV spring	✓	✓	
	$U4$	$U4 < U4_{min}$ $U4 > U4_{max}$	Check MSV LVDT	✓	✓	
5	$P4$	$\Delta P4 < \Delta P4_{min}$ $\Delta P4 > \Delta P4_{max}$	Check MSV spring	✓	✓	
6	SR and SR'	$SR < SR_{min}$ or $SR' < SR'_{min}$	Adjust the RD mechanical input or reduced speed due to EHSV flow or piston sealing jammed	✓		
		$SR > SR_{max}$ or $SR' > SR'_{max}$	Excessive EHSV flow			
	rt_{EN}	$rt_{EN} < rt_{EN_{ref}}$	Check the RD bell crank and control shaft connection	✓		
	$ovsht_{EN}$	$ovsht_{EN} > ovsht_{EN_{ref}}$	Check the RD springs or EHSV flow	✓		
	ft_{EN}	$ft_{EN} > ft_{EN_{ref}}$	Check the RD bell crank and EHSV mechanical input connection	✓		
	$dwnsht_{EN}$	$dwnsht_{EN} > dwnsht_{EN_{ref}}$	Possible fault in EHSV feedback spring	✓		
	dt_elbow_{EN}	$dt_elbow_{EN} > 0$	Free-play between bell crank and EHSV mechanical input	✓		

	ft_{RN}	$ft_{RN} < ft_{RN_{ref}}$	Check the RD bell crank and control shaft connection	✓
	$dwnsht_{RN}$	$dwnsht_{RN} > dwnsht_{RN_{ref}}$	Check the RD springs or EHSV flow	✓
	rt_{RN}	$rt_{RN} > rt_{RN_{ref}}$	Check the RD bell crank and EHSV mechanical input connection	✓
	$ovsht_{RN}$	$ovsht_{RN} > ovsht_{RN_{ref}}$	Possible fault in EHSV feedback spring	✓
	dt_elbow_{RN}	$dt_elbow_{RN} > 0$	Free-play between bell crank and EHSV mechanical input	✓
7	Sm and Sm'	$Sm < Sm_{min}$ or $Sm' < Sm'_{min}$	Check the EHSV flow and Filter if clogged	✓ ✓ ✓
		$Sm > Sm_{min}$ or $Sm' > Sm'_{min}$	Check the EHSV flow	✓ ✓ ✓
27	$PS1$	$PS1 < PS1_{min}$	Check MSV sleeve and spring	✓
29	$PS2$	$PS2 < PS2_{min}$	Check MSV sleeve and spring	✓

Sequence n.3

The plots in Fig. 5.22 show two examples of the collected channels of *Sequence* n.3 for both Elevator and Aileron. Both the units are equipped with an Accumulator on the return line and two check valves on both supply and return lines, called High Pressure and Low Pressure Valves, as shown in the hydraulic schemes of paragraphs 2.3 and 2.4. This *Sequence* has been designed in order to extract both the minimum opening pressures of both these valves: $P2$ and $P3$. In order to extract both the HFs in a unique *Sequence*, the unit need to be set in Damping mode and a specific Supply pressure command needs to be apply. In the TF configuration, this command is manually regulated by the technician according to the specification which are reported on an appropriate Dialogue window. These instructions basically follow the designed command it is possible to observe in Fig. 4.7. The green line in the upper subplots of Fig. 5.20 and 5.21 illustrates the manual pressure adjustment and the values of $P2$ and $P3$ which have been noted by the technician in the appropriate dialogue window.

In fact, these two values need to be manually reported by the technician while

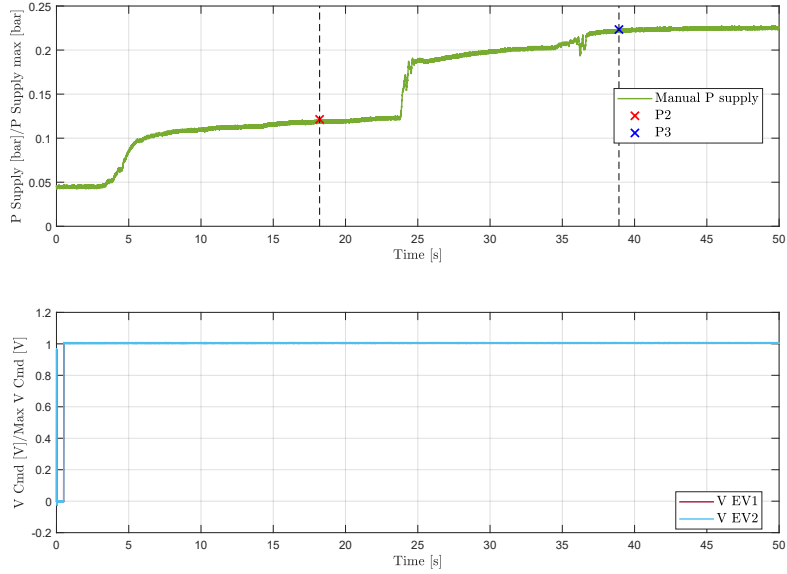


Figure 5.20: Elevator *Sequence n.3*

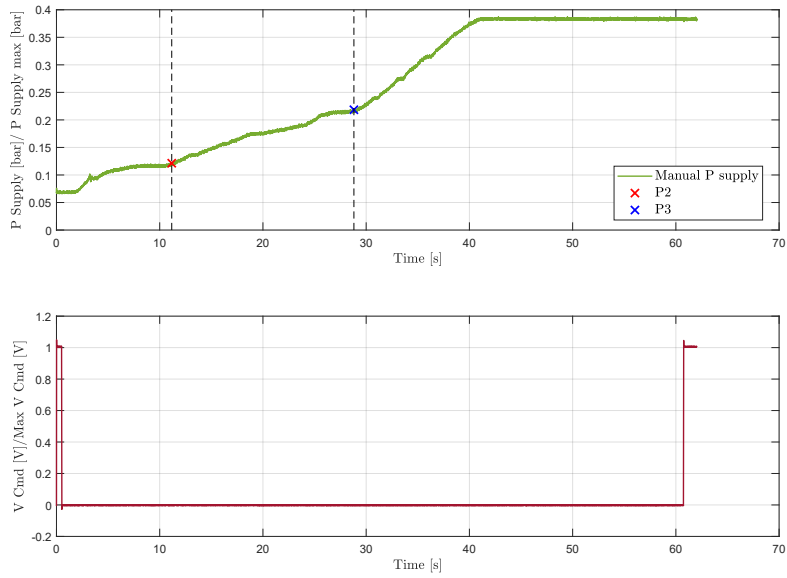


Figure 5.21: Aileron *Sequence n.3*

Figure 5.22: *Sequences n.3* for both Elevator and Aileron

regulating the pressure:

- $P2$ needs to be reported when the accumulator starts to be filled. During the previous preliminary *Sequence* n.2, the unit is forced in its null position and the supply pressure is set to zero, in order to drain completely the accumulator through the appropriate draining port. Once it is completely empty, the technician needs to check what is the minimum supply pressure that is necessary to open the High Pressure Valve allowing the flow to fill start filling the hydraulic circuit until the accumulator. At the minimum pressure $P2$, the flow is able to react the contrasting accumulator piston force, filling its compartment. If this feature does not respect the check in Table 5.2, it can be a sign of a possible failure in the HP valve, which does not allow the flow to enter the unit with the correct pressure, or in the draining circuit itself.
- $P3$ is instead the minimum pressure which is necessary to open the Low Pressure Valve on the return line, and it needs to be measured when the test-bench flow meter start to register a certain flow on this line. At current state, the flow-meter channel cannot be digitally collected so even this index need to manually reported, after reading the value on this analog sensor. In Table 5.2 it is reported to check if the difference $\Delta P3 = P3 - P2$ respects the limits ranges: if its value is too low, it can be caused by the LP valve seat or spring; otherwise if too high it can be due to a overpressurization of the hydraulic circuit due to filter clogging or worn out LP sealings.

Even if these values are not fully automatic extracted, the code is able to plot both the HFs values on the supply line time-series, as in the first subplots in Fig. 5.22. The other two lower graphs shows instead how the Damping mode is set in the opposite way in these two units: the Elevator requires both the electro-valves to be energized, while the Aileron needs its electro-valve to be switched off.

Sequence n.4 and 5

These signals have been designed to extract three traditional HFs of the MSV of both Elevator and Aileron. Figures 5.25 and 5.28 shows the equivalence of the *Sequences* signals for both Elevator and Aileron. In both the cases, for both the units, the actuator is commanded in closed loop with an initial step to its middle position, then it is commanded to move sinusoidally with an amplitude of 80% its full stroke. The main difference between these two *Sequences* is the supply pressure command. In *Sequence* n.4 in Fig. 5.23 and 5.26, the supply pressure is manually reduced following the same procedure of *Sequence* n.3: the operator reads a series instructions over an dedicated dialogue window, trying to reproduce the decreasing ramp signal. In *Sequence* n.5 the pressure command is instead a symmetric rising ramp.

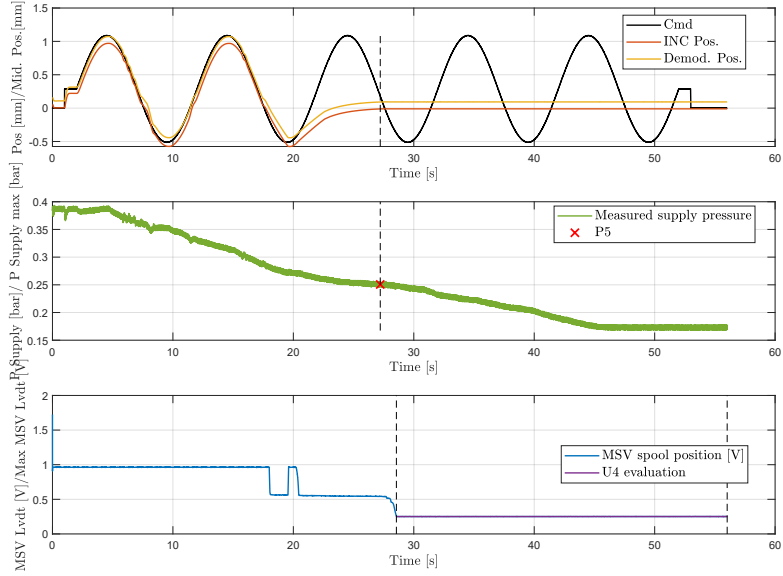


Figure 5.23: Elevator *Sequence* n.4

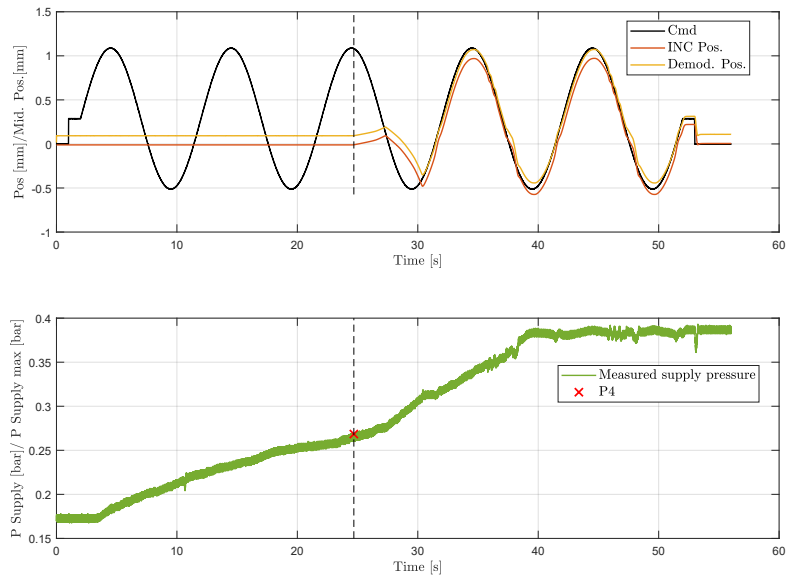


Figure 5.24: Elevator *Sequence* n.5

Figure 5.25: Elevator *Sequences* n.4 and 5

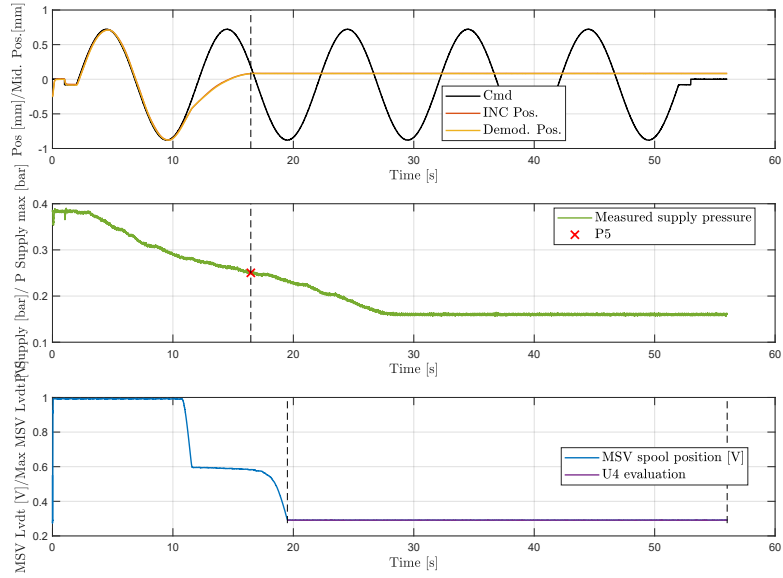


Figure 5.26: Aileron *Sequence* n.4

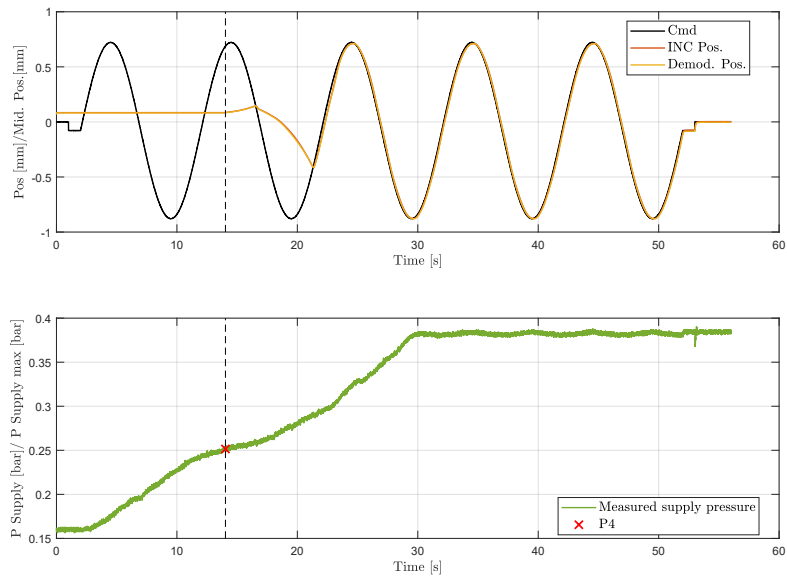


Figure 5.27: Aileron *Sequence* n.5

Figure 5.28: Aileron *Sequences* n.4 and 5

During *Sequence n.4*, the slowly decrease of pressure makes the MSV switch its position, closing any link between supply and return line with the main piston chamber and setting the entire system in damping mode. This critical value of $P5$ represents the minimum pressure value needed to counteract the MSV spring force. Therefore, when the unit supply pressure reaches the $P5$ value, the main ram stops follow the sinusoidal command, stabilizing its position. The first moment when the position signal is stable can be easily identify using `stab_finder` function, and it can be used to both identify the $P5$ value in the pressure time-series and the consequently the region of MSV LVDT signal where to evaluate $U4$. This value represents in fact the reference position in Damping mode the MSV spool can reach. The exact opposite situation is instead described by the plots in Fig. 5.24 and 5.27 where the value $P4$ can be easily identify using the function `move_finder`. The CMM suggest to report the differences $\Delta P5 = P5 - P2$ and $\Delta P4 = P4 - P2$ as failure indicators. Any out-of-limit HFs can be correlated to failure of the MSV spring or an excessive friction between MSV spool and sleeve due to any debris. The same kind of failures can be also adducted in case of $U4$ outside the indicated limits, with the addition of a possible electrical failure of the MSV LVDT itself.

Sequence n.6

This *Sequence* has been designed in order to some of the most characterizing HFs of the Elevator Recentering Device (RD), setting a constant supply pressure. As shown in the first subplot of Fig. 5.29, the unit is commanded in open-loop with in two steps. The first one from 0 mA to its maximum current I_{max} and back to its null position in order to fully extend main ram and slave it back in its recentered position. The second one is symmetric to the previous one, totally retracting the unit with its minimum current I_{min} command. These steps back to null current command intend to test an electric failure with a sudden lack of power supply. The normalized command channel is coloured in black, while the vertical green line represents the ipt_{cmd} change points selected by the function `findchangeslope`. These indexes represents the starting point to evaluate the recentering strokes, shown in the second subplot of the measured positions from both the external INC sensor and the integrated main ram RVDT. The function `move_finder` takes as input the second and fourth ipt_{cmd} as starting points, and it is able to extract the slopes of the red and yellow areas. These two linear slopes represents the recentering speed HFs SR and SR' which are listed in the CMM. The signal sections identified by `move_finder` have been highlighted in the second subplot in Fig. 5.29 with dashed vertical black lines. If these two indexes does not fall within the suggested ranges, the manual reports a list of possible failure which can both point to an anomaly in the RD or in the EHSV. For example, on one hand, an higher speed than the suggested upper limit (SR_{max} and SR'_{max}) can be a symptom of an excessive flow from the EHSV. On the other hand, a lower

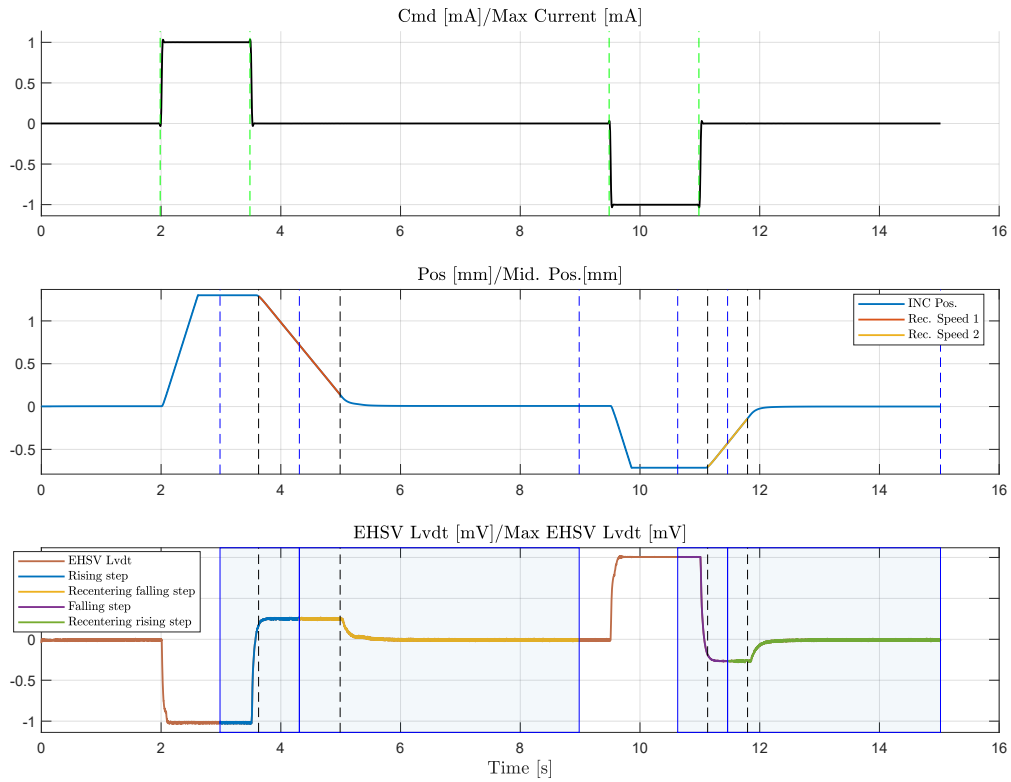


Figure 5.29: Elevator Sequence n.6

recentering speed than the lower limit (SR_{min} and SR'_{min}) can be both due to a flow problem in the EHSV, which can present an excessive leakage, but also a wrong regulation or a defective component of the RD mechanical chain. This latter possibility create an ambiguity of failure detection, since it points out two possible roots (EHSV and RD) but also does not really distinguish if the RD need to be simply regulated or some of its parts need to be replaced. Distinguishing between a simple regulation or a component replacement deeply affects the maintenance cost, and it is an issue that need to be clarified.

In the light of this, new additional HFs have been defined, investigating the behaviour of the EHSV LVDT. This channel (third subplot in Fig. 5.29) illustrates the position of EHSV second stage spool, and it has been extracted using the function `syncro_demod` in paragraph 5.4.4. From this plot it is possible to observe how the EHSV spool moves coherently with the command and the RD action in case of instantaneous step back to null current. For example, the first rising step to I_{max} causes a complete rotation of the EHSV first stage torque-motor which immediately unbalances the command flows over the second stage spool. Moved by

a considerable difference of pressure, the EHSV spool moves almost instantaneously to one of its end stroke, with negative sign in Fig. 5.29. When the command signal is set back to 0 mA in open loop, the main ram does not stuck in its maximum extracted stroke but it starts moving back to its null position thanks to the RD. The absence of current in the torque motor unbalances the force equilibrium over the jet deflector, which is now just affected by the RD action through the EHSV mechanical input. This force moves rapidly the spool in the opposite direction, overcoming its null position according to the force the RD is able to apply, according to its characteristic force curve in Fig. 2.11. This step has been highlighted in the first blue squared area in the third subplot in Fig. 5.29. This position allows the flow to move in the cylinder chambers and its piston is able to retract at the extracted SR speed. When the piston reaches the linear area in Fig. 2.11, the RD tends to reach its stable configuration, pushing the spool to its null position and closing any flow connection with the cylinder chambers. This action is highlighted in the second blue rectangle in Fig. 5.29, and it is particularly useful to detect any RD failure. The same considerations are of course true also during the second step part of the command signal, in the symmetric stroke from retracted stop to zero position. The four blue areas in Fig. 5.29 isolate four EHSV spool steps: the Rising step during the recentering piston stroke from Extended to Null position (EN) (in blue) and its consecutive Falling one (in yellow); and symmetrically the Falling step during the recentering piston stroke from Retracted to Null (RN) (in violet) and its consecutive Rising small step (in green). Per each of these feature, the following HFs have been defined:

- rt_{EN} [s] and $ovsht_{EN}$ [%]: Rising Time of the first aforementioned piston stroke from Extended to Null, and its overshoot;
- ft_{EN} [s] and $dwnsht_{EN}$ [%]: Falling Time of the second recentering step and its relative downshoot;
- dt_elbow_{EN} [s]: short time lapse of possible stable position of EHSV spool during its recentering stroke. This HF is a very promising one since the presence of a small "elbow" during the recentering spool stroke may identify an excessive free-play between Bell Crank and EHSV Mechanical Input, due to a worn out of the Bell Crank contact face;
- ft_{RN} [s] and $dwnsht_{RN}$ [%]: Falling Time of the piston stroke from Retracted to Null, and its downshoot;
- rt_{RN} [s] and $ovsht_{RN}$ [%]: Rising Time of the last recentering step and its relative overshoot;
- dt_elbow_{RN} [s]: equivalent to dt_elbow_{EN} but for the last step.

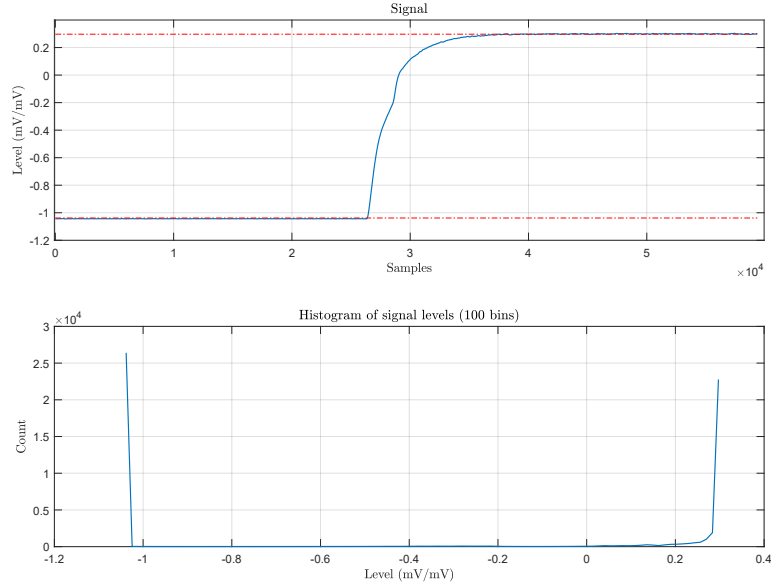


Figure 5.30: Step State Level Analysis

The analysis of the aforementioned four steps has been conducted using the statistical State-Level approach. As reported in [33], this method is based over a statistical analysis over the points distribution in the signal. It starts determining the maximum and minimum amplitudes of the signal in order to define a range, where to set a certain number of bins (100 by default) each with a defined width that depends on the range amplitude. Once the bins have been defined, the algorithm counts how many signal points can be assigned to each bin. Typically, in a step signal, two bins are filled with the more points than all the others, and these are the ones related to the higher and lower levels. In Fig. 5.30, the lower level contains almost 25000 points in the bin of the lower limit (around -1), while the highest one includes almost the same number of points but at the upper level (around 0.2). All the others bins are neglected during the State Levels evaluation.

Once the State Levels have been identified, it is possible to extract the characteristic of each step response. The plots in Fig. 5.31 illustrates the main features of each of the highlighted EHSV LVDT steps in Fig. 5.29, using the built-in Matlab functions `risetime`, `falltime` and `overshoot`. For example, from the rising steps in Fig. 5.31a and 5.31d it is possible to extract the two characterizing rise-time rt_{EN} and rt_{RN} , taking as reference the cross from the 10% to 90% of the State Levels, highlighted by a green and red cross in the plots. The same also happens for the falling steps in Fig. 5.31b and 5.31c for the evaluation of ft_{EN} and ft_{RN} .

In addition, each State Levels is surrounded by specific boundaries, with a

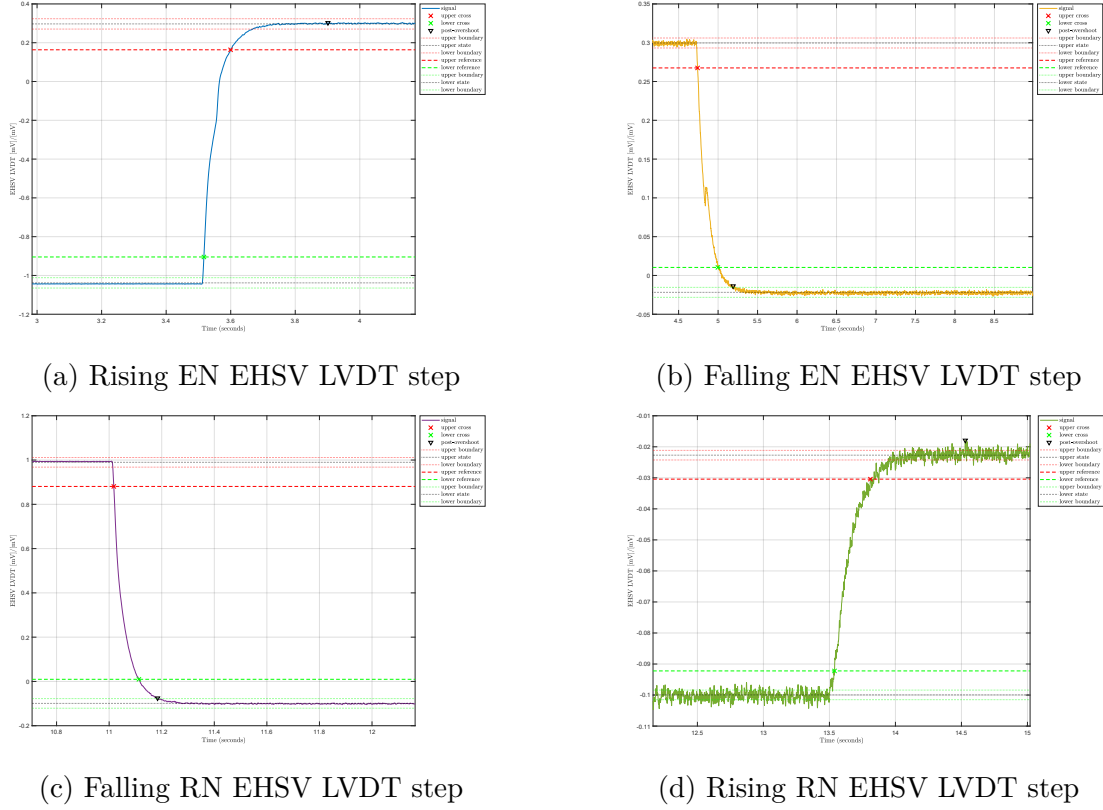


Figure 5.31: EHSV Steps Analysis

predefined tolerance of $\alpha = 2\%$:

$$\begin{cases} S_{low} = S_n - \alpha(\Delta S) \\ S_{up} = S_n + \alpha(\Delta S) \end{cases} \quad (5.6)$$

where S_n can be both the upper or lower State Level and ΔS is their difference. These boundaries are significant in order to determine any positive or negative overshoot of the step response. The built in Matlab function `overshoot` takes as input the identified State Levels and evaluates the maximum deviation from the reference State Level according to the following:

$$\begin{cases} \text{oversht}[\%] = \frac{100(O-S_2)}{\Delta S} \\ \text{dwnsht}[\%] = \frac{100(O-S_1)}{\Delta S} \end{cases} \quad (5.7)$$

where S_2 is the maximum while S_1 the minimum State Level and O is the identified peak point. Using this definition, the HFs oversht_{EN} , dwnsht_{EN} and symmetrically dwnsht_{RN} and oversht_{RN} have been extracted.

The last two HFs to describe in *Sequence* n.6 section in 5.2 regards two particular

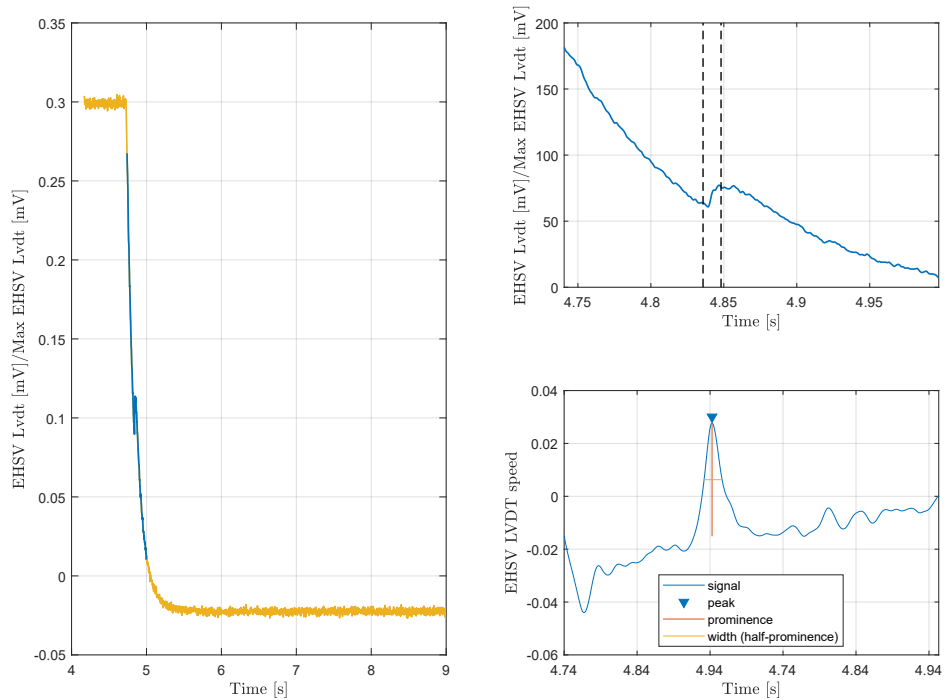


Figure 5.32: EHSV LVDT Elbow Detection

time-ranges which can be extracted during the recentering strokes in Fig. 5.31b and 5.31d. The extraction of these two values has been performed building a tailored function called `elbow_detection`. An example of how this script works is shown in Fig. 5.32 where the extraction of dt_elbow_{EN} is performed.

As reported in this figure, the function recalls `move_finder` in order to extract the blue slope of the step. This slope presents the final part of the recentering process where the RD Bell Crank pushes the EHSV spool to its null position through its connection with the Mechanical Input. Any anomalous change of slope in this section is very likely to be caused by a worn contact surface between Bell Crank and Mechanical Input. In order to precisely detect any elbow in the nominal descending (or ascending for dt_elbow_{RN}) stroke, the function `elbow_detection` evaluate the speed curve within the isolated portion of signal, using the command `diff`. Due to the nature of the signal, the general behaviour of this curve should be known: in case of falling step (like the one in the example plots), its sign should be always negative; while it need to be positive in case of rising step. However, an eventual elbow in this curve should determine a sudden change of the expected sign, as reported in the lateral subplots in Fig. 5.32. In this case, using the `findpeaks` function in the speed curve, it is possible to extract any peak in this curve and if

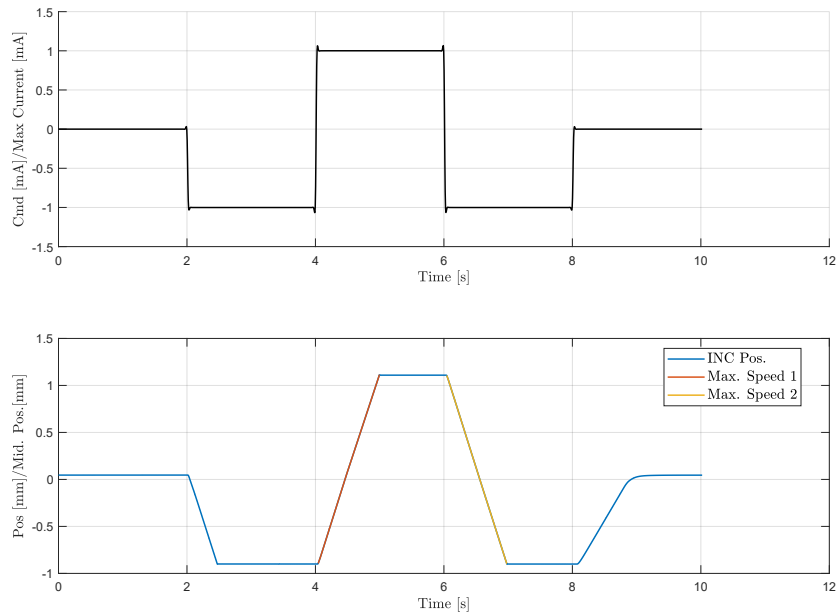
its sign is coherent with the expected one. If the local peak sign has an opposite sign respects to all the other points, the function detects it and it is able to identify its width. This peak width is taken as reference to extract the appropriate dt_elbow HF. Observing the two subplots on the right of Fig. 5.32, the peak width limits are reported in the original signal time-series with two dashed-black lines. If the function `findpeaks` does not identify any anomalous countersign peak, the function reports as output a default null value.

This method is used also to extract the same kind of HF also from the MSV LVDT channel during the transition from Active to Damping mode for the Elevator (or viceversa for the Aileron).

Sequence n.7

This *Sequence* has been designed for all the PFCS in order to extract the maximum piston speed from totally retracted stop to extracted Sm , and viceversa Sm' . As reported in the CMM, these HFs need to be extracted with a supply pressure that is less than 80% of the nominal one. The plots in Fig. 5.33 illustrates how these HFs are identified for the Elevator PFCS. In particular, the second subplot shows the main ram response to the upper open loop command. It is important to notice how the main ram returns to its null position during the last step command from I_{min} to zero, due to RD effect. This behaviour is typical of the Elevator PFCS: both the Aileron and Spoiler remains in their fully extracted position since the open-loop command does not consider any feedback in position and they do not include any mechanical RD system.

In the upper subplot, the green vertical lines represents the change points identified during the initial preprocessing phase by `findchangeslope`. Similarly to *Sequence* n.6, these points are uniquely identified since the signal shape is a priori defined and they can be used as reference points to extract the highlighted red and yellow slopes using `move_finder`. Low values of Sm or Sm' may be caused by a defective EHSV, which can present external leakages, or by worn piston sealings or clogged filter. On the other hand, an excessive speed can be due to a high EHSV flow gain. Among these troubleshooting instructions, the most significant are the EHSV related ones, since its repairing cost or replacement are way higher than any of the other aforementioned components. Typically, any piston sealings failure may be difficult to detect since these parts are by default replaced when the unit is dismounted. The same happens for any Filter clogging issue, since it is a typical routine procedure to clean this component if a unit needs to be repaired.

Figure 5.33: Elevator *Sequence* n.7

5.5.3 Leakages Sequences

In this section, the HFs methods for the Leakage *Sequences* group are described. As already reported in Chapter 4, these signals have been defined in order to perform the suggested CMM leakages tests, which are typically longer than the others. In fact, even in the HyDiag procedure, these *Sequences* are characterized by a maximum length of 3 minutes, covering most of the total duration of the Automatic Test. These longer testing times are due to the fact that the Test Field hydraulic benches are equipped with analogue flow-meters which do not allow to digitally collect neither the supply or return flow measurement channels. As consequence, the collected measurement files do not contain any information about the regulated flow. In order to overcome this problem, the leakages HFs are manually reported by the technician, using some specific LabView dialogue windows, as soon as the measured flow by the flowmeter stabilizes. Typically, this procedure may require couple of minutes according also to the installed UUT. For this reason, a maximum testing time of 3 minutes is set per each of these *Sequences*, even if the operator is able to stop the testing signal as soon as the registered flow value looks stable, reducing considerably the testing time. To conclude, the length of these signal strongly depend on the testing condition, and in particular on the used test bench. For example, during the prototyping phase in the H-Lab, these *Sequences* length is reduced to less than 1 minute, since the H-Lab test-bench is equipped with more

performing digital flow-meters. This sensors upgrade affects of course also the leakage HFs extraction, which can be performed in a fully automatic way. As for the Pressure Adjustment *Sequences*, table 5.3 summarizes all the HFs that are extracted by the *Sequences* in this group. From this table, it is possible to observe how the first two *Sequences* n.8 and 9 are designed for all the PFCS while the last three are tailored for both the Elevator and Aileron, due to their switch to Damping mode through Electro-valves.

Table 5.3: Leakages Sequences HFs Recap table

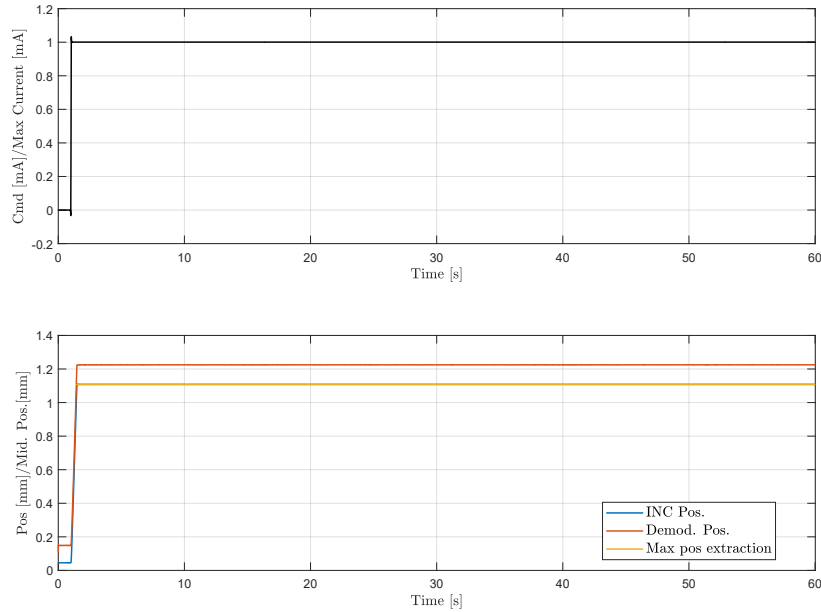
Seq	HFs	Check	Troubleshoot	E	A	S
8	$Q3$	$Q3 > Q3_{max}$	Check the EHSV flow and that the unit does not present an external leakage	✓	✓	✓
	Pos_{max}					
9	$Q3'$	$Q3' > Q3'_{max}$	Check the EHSV flow and that the unit does not present an external leakage	✓	✓	✓
	Pos_{min}	$Pos_{max} - Pos_{min} < S1_{min}$ $Pos_{max} - Pos_{min} > S1_{max}$	Check the outer piston sealings	✓	✓	✓
	$S2'$	$S2' < S2'_{min}$ $S2' > S2'_{max}$	Check the RD Bell Crank and Control Shaft input, and its connection to the EHSV	✓		
	$2\delta C'$			✓		
	$V4$	$V4 < V4_{min}$ $V4 > V4_{max}$	Check the EHSV flow			✓
10	$Q2$	$Q2 > Q2_{max}$	Check the EHSV flow and EV1 sealings	✓	✓	
	$S2$	$S2 < S2_{min}$ $S2 > S2_{max}$	Check the RD Bell Crank and Control Shaft input, and its connection to the EHSV	✓		
	$2\delta C''$	$max(2\delta C) > 2\delta C_{max}$	Replace the EHSV	✓		

11	$Q2'$	$Q2' > Q2'_{max}$	Check the EHSV flow and EV1 sealings	✓	
	$Q1$	$Q1 > Q1_{max}$	Check the EHSV flow	✓	✓
12	$U3$	$U3 < U3_{min}$	Check the MSV spring, the presence of any debris and the LVDT wirings	✓	✓
		$U3 > U3_{max}$	Check the MSV LVDT wirings		
	ϕ_{msv}	$\phi_{msv} < \phi_{min}$	Check the MSV LVDT wirings	✓	✓
		$\phi_{msv} > \phi_{max}$			

Sequence n.8

As reported in the previous paragraph, this *Sequence* has been designed in order to extract the stalled leakage HF $Q3$ in extended position. For this purpose, the unit is commanded in open-loop with a unique step from 0 to I_{max} in order to slave the piston position at its end stroke, setting the maximum supply pressure P_{max} . This setting would not change for all the consecutive *Sequences*. The plots in Fig. 5.34 report two typical measurements for this *Sequence*. It is worthy to notice how in this example the *Sequence* has been stopped after 1 minute. This means that the technician at the test-bench has decided to stop the measurement since the return line flow-meter was reporting a stable and unique value, which is exactly $Q3$. This parameter is manually reported by the operator and collected in the measurement `INDEX.txt` file. This particular HF is characterized by a maximum limit $Q3_{max}$: if the extracted value is higher than this latter reference one, it can be a sign of an excessive EHSV flow gain due to an inner anomaly or any external or internal undetected leakage.

This *Sequence* has been also used to extract part of an important HF, which is the maximum unit stroke $S1 = Pos_{max} - Pos_{min}$ that need to be extracted per all the PFCS. Due to the shape of the designed signal, the maximum position Pos_{max} can be easily extracted as the mean value of the maximum stable position, identified by `move_finder`. This signal portion is highlighted in yellow in the second subplot of Fig. 5.34. It is important to notice the offset between the INC position sensor channel and the Demodulated channel from the inner sensor. Since any uncorrect INC sensor setting is fixed during the preprocessing of the *Sequence*, this measurement has been chosen as reference one to extract Pos_{max} . The excessive offset between the yellow and red lines may be an indicator of a biased RVDT or LVDT sensor on the main ram.


 Figure 5.34: *Sequences n.8* for all PFCS

Sequence n.9

Table 5.3 reports the HFs that can be extracted per each PFCS from this *Sequence*. All the units share the extraction of both $Q3'$ and Pos_{min} for the final evaluation of the total stroke $S1$. These two indexes have been extracted using the same approach of previous *Sequence* n.8. Following this list, the next two consecutive HFs $S2$ and $2\delta C'$ are characteristic for the Elevator PFCS. The adopted methods to extract them are illustrated in Fig. 5.35.

The first subplot shows the black open-loop command line, while the second illustrates the measured position channels. Using the same approach of *Sequence* n.8, the yellow portion of the INC sensor channel indicates the minimum effective end-stop Pos_{min} . This parameter can be easily extracted as the mean value of the highlighted signal, identified with `stab_finder`. In this case, the function needs as input `T_in` the step time of command from null to I_{min} . Viceversa, if this instant of time is set as `T_end`, the function `stab_finder` would isolate a second section which has been coloured in black in the second subplot. This portion of signal is used to evaluate the Elevator recentering position from extended stroke $S2'$. As it is possible to observe from the plot, the main ram is still stacked in its extended position at the end of the previous *Sequence* n.8. For this reason, the initial 10 seconds of command at 0 mA drives the unit back to its null position, pushed by

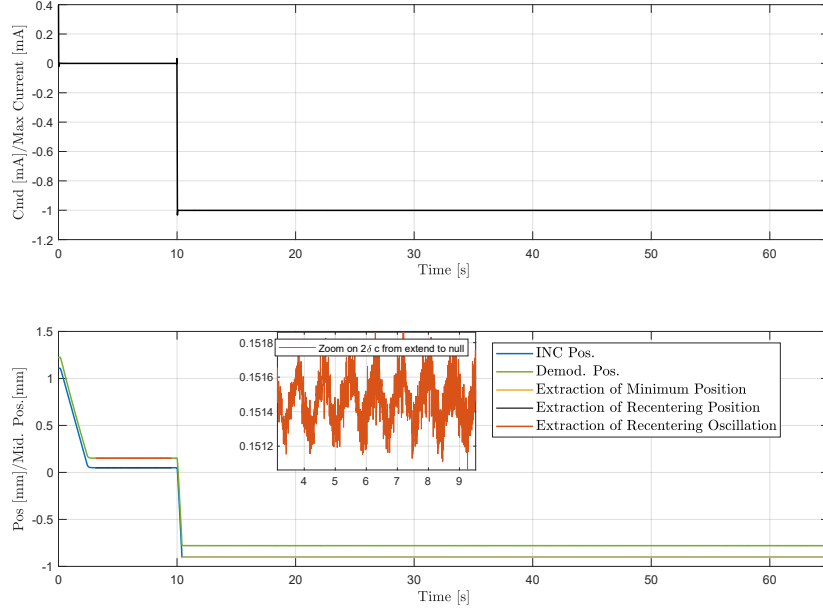


Figure 5.35: Elevator Sequence n.9

the mechanical RD. As reported in the CMM, this HF need to be within a specific range defined by $S2'_{min}$ and $S2'_{max}$, measured taking as reference the totally extracted end-stroke. If this HF is outside the limits, it can be a clear sign of a defective RD which can be manually adjusted or it need a further investigation and be replaced. As already anticipated in paragraph 5.5.2, a failure in the RD can be misclassified if also the EHSV present an anomalous behaviour. For this reason, this *Sequence* has been used to extract another important HF for a correct diagnosis of the EHSV: the peak-to-peak recentering oscillation $2\delta C$.

As reported in the CMM, this index needs to be extracted during the recentering stroke of the unit with the nominal supply pressure P_{max} . Both these conditions are respected in *Sequence* n.9, an in fact the same time range for the extraction of $S2'$ can be used to evaluate $2\delta C'$ during the recentering movement from extended to null position. As illustrated in Fig. 5.35, this oscillation is extracted from the highlighted red portion of signal from the RVDT demodulated position channel. This channel has been preferred to the INC sensor one because of its better resolution and since the peak-to-peak extraction do not take into account any position offset. The second subplot in Fig. 5.35 reports also a dedicated zoom where the oscillating behaviour of the unit is clearly shown.

The oscillation peak-to-peak amplitude is evaluated performing a local Frequency Spectrum analysis through Fourier Transform on the isolated area. The plot in Fig. 5.36 shows the single-sided amplitude spectrum, cutting the frequency range

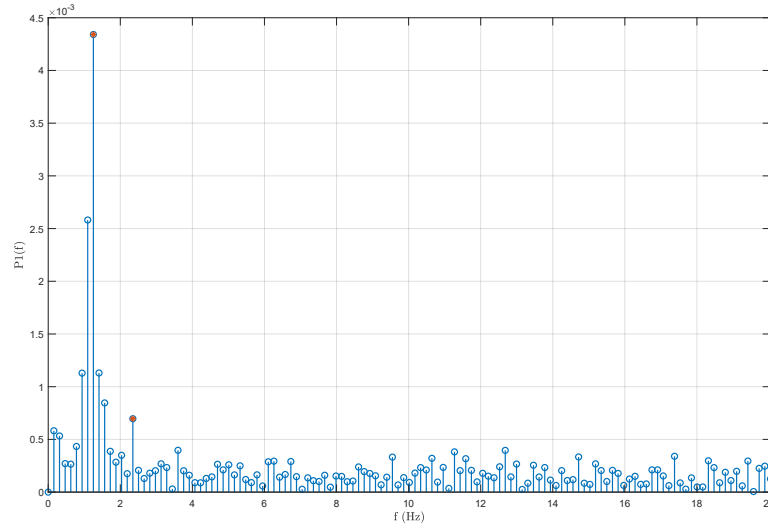


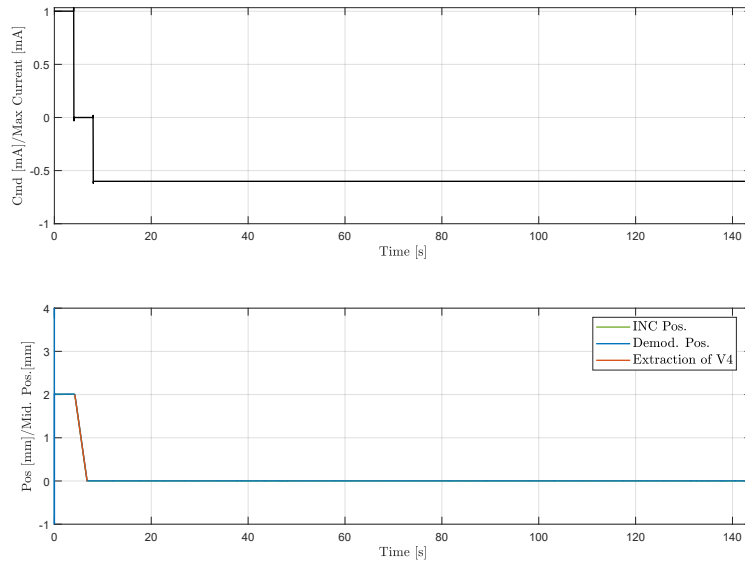
Figure 5.36: *Sequences n.9* Oscillation Frequency Spectrum

at maximum 20 Hz. This values has been set considering the dynamic response of the Elevator, which nominally presents an attenuation of -3 dB at 5 Hz, considering the main ram piston position. In this example, the most significant harmonic presents a characterizing frequency of about 1.2 Hz and a normalized peak amplitude which represents half of the needed peak-to-peak $2\delta C'$.

Excluding the extraction of $2\delta C'$ and $S2'$, the same methods for the definition of both $Q3'$ and Pos_{min} are used also for both the Aileron and Spoiler PFCS. For this latter one, this *Sequence* can be used to extract the retracting speed $V4$. The EHSV of this particular unit need a specific threshold of current I_{trshld} to fully extend the unit main ram. Below this threshold, the actuator does not move from its fully retracted position, which also coincide with its null one. In case of electrical failure, if the EHSV supply current is set to zero, the Spoiler is in its totally retracted position due to the structural EHSV current bias of I_{trshld} . The HF $V4$ represents the nominal retraction speed in case of electrical failure. Fig. 5.37 illustrates the extraction process of this HF. Using `move_finder`, it is possible to isolate the red area indiscriminately from the INC sensor or main ram LVDT demodulated channel, and extract its linear slope which represents $V4$.

Sequence n.10 and n.11

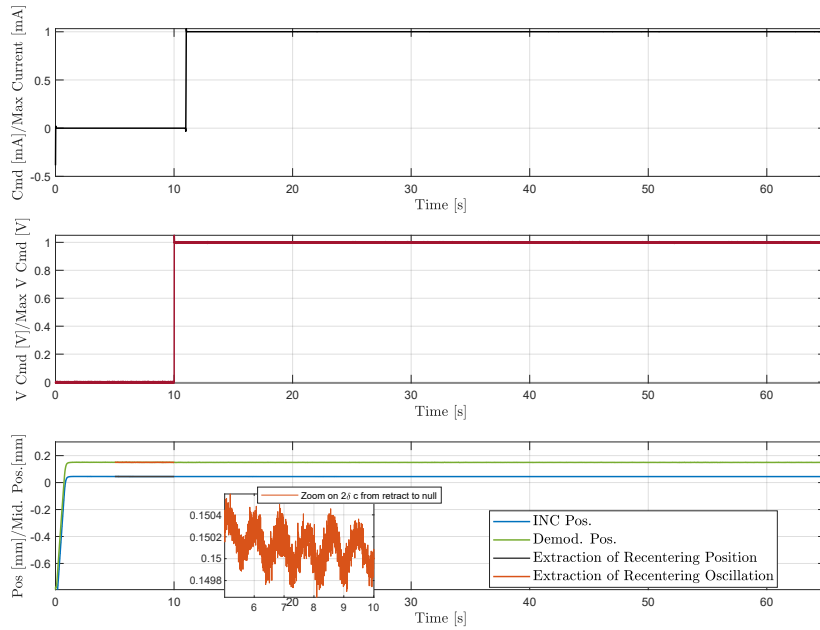
Sequence n.10 has been designed with the main purpose to extract the leakage HFs $Q2$ in damping mode for both Elevator and Aileron. The plots in Fig. 5.38 reports and an example of this *Sequence* applied on the Elevator PFCS. The first subplot represents as usual the designed command, which present an open-loop

Figure 5.37: Spoiler *Sequences* n.9

step to I_{max} at soon after 10 seconds. This instant of time is in fact the activation moment of the first Electrovalve, which is switched on with a step to V_{max} . In this way, from this moment until the end of the test, the technician can manually report the stabilized return flow $Q2$ in damping mode. This same procedure is adopted to extract the same HF from the Aileron, with the only difference regarding the Electrovalve command which needs to be switched off with a specular signal from V_{max} to null voltage.

About the Elevator PFCS, this *Sequence* is also used similarly to the previous one to extract both $S2$ and $2\delta C$ peak-to-peak amplitude during the recentering stroke from totally retracted position to null. Comparing the measured position in Fig. 5.35 with the one in Fig. 5.38, the two main analogies regards both the extraction of $S2$ from the measured position by the INC sensor, and the evaluation of the oscillation $2\delta C$ from the RVDT demodulated channel. In particular, in order to respect the CMM requirements, both the highlighted sections in black and red have been extracted using `move_finder` specifying as time range the entire initial part part of the signal when the unit is still in Active mode, before its switch in damping at around 10 s.

Once both the peak-to-peak oscillation have been extracted in *Sequence* n.9 and 10, the HyDiag SW checks if the maximum HF between these two values is below an accepted threshold $2\delta C_{max}$. This limit value is reported in the CMM and it depends also on the unit Flight Hour. Any out of range value would be a symptom


 Figure 5.38: Elevator *Sequences* n.10

of a defective EHSV, which is not able to provide a regular flow to the piston chambers due to an oscillating second-stage spool.

The *Sequence* position and Electorvalve command shapes are also used in the consecutive *Sequence* n.11. The main goal of this test is to extract the equivalent leakage HF $Q2'$ obtained when the units is commanded in damping mode only activating the second Electrovalve, using the same command step from 0 to V_{max} .

Sequence n.12

Both the Elevator and Aileron PFCS share *Sequence* n.12 in order to extract the leakage HF $Q1$ and the maximum MSV LVDT spool position $U3$ and eventual phase shift between primary and secondary wirings ϕ_{msv} . The plots in Fig. 5.39 report the closed-loop command, which slaves the unit in its null position after a rapid and small movement at around 25% of the reference stroke. After this small step, the operator can manually read the stable value of the return flow $Q1$ when the unit is recentered to zero. During this test, the supply pressure is always constant and set to P_{max} and the unit is in active mode. In this condition, the MSV spool is fully opened and its LVDT register the reference maximum value $U3$.

The yellow section in the third subplot defines the area identified by `stab_finder` and then averaged in order to extract a unique value of $U3$. An out of range HF

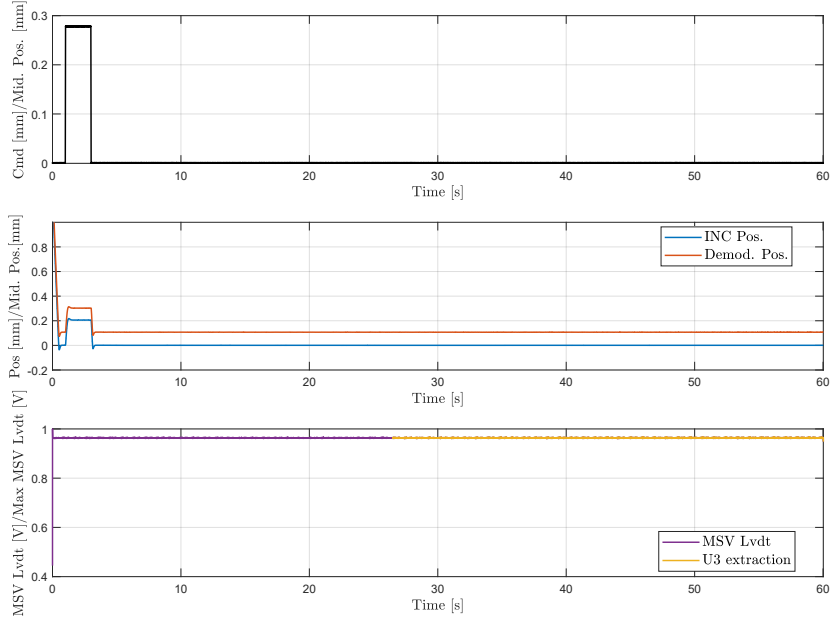


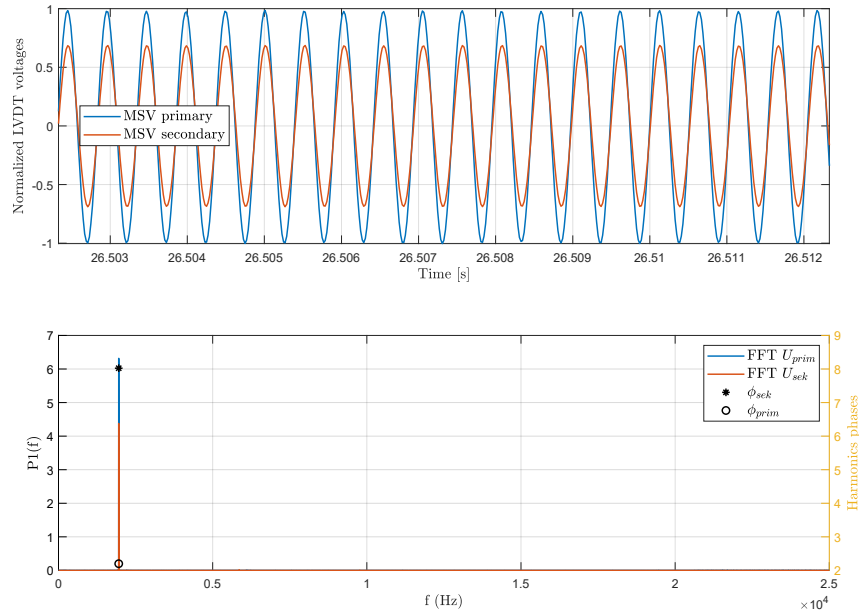
Figure 5.39: Elevator *Sequences* n.12

would be a good insight of a defective MSV spring, presence of debris in the component sleeve or even a defective LVDT sensor. In order to check this sensor integrity, an additional check about possible phase shift ϕ_{msv} between primary and secondary wirings is performed.

The results of this analysis are reported in Fig. 5.40. The first subplot shows a particular zoom where both the sinusoidal primary and secondary waves are visible. Both of them are supplied at the same reference frequency of 1953 Hz and the secondary coil is considerably less noisy if compared with the EHSV Lvdtd one. For this reason, it is not preprocessed with the `denoising` function in paragraph 5.4.3 and it present a clear Frequency Spectrum as the one in the second subplot. As reported in this plot, both the fourier harmonics decomposition show as expected a unique harmonic at the nominal frequency. The distance between the two dots \circ and $*$ represents the phase shift ϕ_{msv} in degree between primary and secondary coils.

5.5.4 Automatic Sequences

The *Sequences* grouped in this paragraph constitutes the full automatic part of the HyDiag testing procedure. During these tests, the operator needs to set a constant supply pressure of P_{max} and wait the complete test to be finished without any further interaction. As already mentioned in 4.3.1, the *Sequences* order


 Figure 5.40: Elevator *Sequences* n.12 MSV spectrum analysis

and duration have been designed in order to stimulate a singular or a group of sub-components that affects each other. For example, from *Sequence* n.17 to 22 all the possible tests over Electrovalves and MSV have been grouped for both Elevator and Aileron. Table 5.4 reports all the extracted HFs and the corresponding troubleshooting instructions.

Table 5.4: Fully Automatic Sequences HFs Recap table

Seq	HFs	Check	Troubleshoot	E	A	S
13	db_{rec}	$db_{rec} > db_{rec_{max}}$	Adjust the RD and check its clearances	✓		
	$I db_{rec}$	$I db_{rec} > I db_{rec_{max}}$				
	$x db_{rec}$	$x db_{rec} > x db_{rec_{max}}$				
	k_{rec}	$k_{rec} > k_{rec_{max}}$ $k_{rec} < k_{rec_{max}}$	Check the EHSV flow Check the RD springs stiffness			
	$I O_{rec}$	$I O_{rec} > I O_{rec_{max}}$	Possible high EHSV offset current			
	$x 0_{rec}$	$x 0_{rec} > x 0_{rec_{max}}$				

14	$U9$	$U9 < U9_{min}$ $U9 > U9_{max}$	Check the correct movement of EHSV spool in its sleeve	✓
	$U10$	$U10 < U10_{min}$ $U10 > U10_{max}$	Check the correct movement of EHSV spool in its sleeve	✓
15	$U0$	$U0 < U0_{min}$ $U0 > U0_{max}$	Check the EHSV mechanical input	✓
	IE	$IE < IE_{min}$ $IE > IE_{max}$	Fix the RD position and check its Bell Crank	✓
	IR	$IR < IR_{min}$ $IR > IR_{max}$	Fix the RD position and check its Bell Crank	✓
	$SNR(U_{sek})_{ext}$	$SNR(U_{sek})_{ext} < 0$	High vibration level of EHSV spool	✓
	$SNR(U_{sek})_{ret}$ $f_{U_{sek},ext}$	$SNR(U_{sek})_{ret} < 0$ $f_{U_{sek},ext} < f_{supply}$	Check the EHSV first stage	
	$f_{U_{sek},ret}$	$f_{U_{sek},ret} < f_{supply}$		
16	x_b	$x_b > x_{b_{max}}$	Check the RD bell crank setting	✓
	h_x	$h_x > h_{x_{max}}$		
17	$U5$	$U5 < U5_{min}$ $U5 > U5_{max}$	Replace the Electrovalve 1	✓ ✓
	$U6$	$U6 < U6_{min}$ $U6 > U6_{max}$	Replace the Electrovalve 1	✓
18	$U7$	$U7 < U7_{min}$ $U7 > U7_{max}$	Replace the Electrovalve 2	✓
	$U8$	$U8 < U8_{min}$ $U8 > U8_{max}$	Replace the Electrovalve 2	✓

19	$UC1$	$UC1 < UC1_{min}$ $UC1 > UC1_{max}$	Replace the Electrovalve 1	✓
	$T1$	$T1 < T1_{max}$	Check the Damping Circuit lines (from Electrovalve 1 to MSV). In addition check the EHSV flow and the piston dynamic sealings.	
20	$UC2$	$UC2 < UC2_{min}$ $UC2 > UC2_{max}$	Replace the Electrovalve 2	✓
	$T2$	$T2 < T2_{max}$	Check the Damping Circuit lines (from Electrovalve 2 to MSV). In addition check the EHSV flow and the piston dynamic sealings.	
21	UC	$UC < UC_{min}$ $UC > UC_{max}$	Replace the Electrovalve	✓
	$T3$	$T3 > T3_{max}$	Check the Damping Circuit lines (from Electrovalve 1 to MSV)	✓ ✓
22	$T4$	$T4 > T4_{max}$	Check the Damping Circuit lines (from Electrovalve 2 to MSV)	✓ ✓
23	$R1$	$R1 < R1_{min}$	Check that the unit has reached the goal position, otherwise check the RVDT 0 setting	✓ ✓ ✓
		$R1 > R1_{max}$		

R2	$R2$	$R2 < R2_{min}$	Check that the unit has reached the goal position, otherwise check the RVDT 0 setting	✓	✓	✓
		$R2 > R2_{max}$				
	r_x	$r_x > r_{x_{max}}$	Check the EHSV flow and the piston dynamic sealings	✓	✓	✓
	ΔU_{rms}	$\Delta U_{rms} > \Delta U_{rms_{max}}$	Check the main ram LVD setting		✓	
	dD	$dD > dD_{max}$	Replace the ESHV			
	Ion	$Ion > Ion_{max}$	Replace the ESHV			
24	$U11$	$U11 > U11_{max}$	Replace the ESHV	✓	✓	
25	ϕ_{UA}	$\phi_{UA} < \phi_{UA_{min}}$	Check the main ram sensor (LVDT or RVDT) wirings	✓	✓	✓
		$\phi_{UA} > \phi_{UA_{max}}$				
	ϕ_{UB}	$\phi_{UB} < \phi_{UB_{min}}$	Check the main ram sensor (LVDT or RVDT) wirings	✓	✓	✓
		$\phi_{UB} > \phi_{UB_{max}}$				
	ϵ_{ext}	$\epsilon_{ext} > \epsilon_{ext_{max}}$	Check the Dynamic Sealings	✓	✓	✓
	ϵ_{ret}	$\epsilon_{ret} > \epsilon_{ret_{max}}$				
26	db_{sv}	$db_{sv} > db_{sv_{max}}$	Excessive hysteresis in the EHSV first stage			
	$I db_{sv}$	$I db_{sv} > I db_{sv_{max}}$				
	$x db_{sv}$	$x db_{sv} > x db_{sv_{max}}$		✓		
	k_{sv}	$k_{sv} > k_{sv_{max}}$	Check the feedback spring stiffness			
		$k_{sv} < k_{sv_{max}}$				
	I_{bias}	$I_{bias} > I_{bias_{max}}$	Excessive EHSV null bias			
	$hyst_I$	$hyst_I > hyst_{I_{max}}$	Excessive torque motor magnetic hysteresis			
35	In	$In < In_{min}$ $In > In_{max}$	Replace the EHSV			✓
	It	$It < It_{min}$	Replace the EHSV			✓

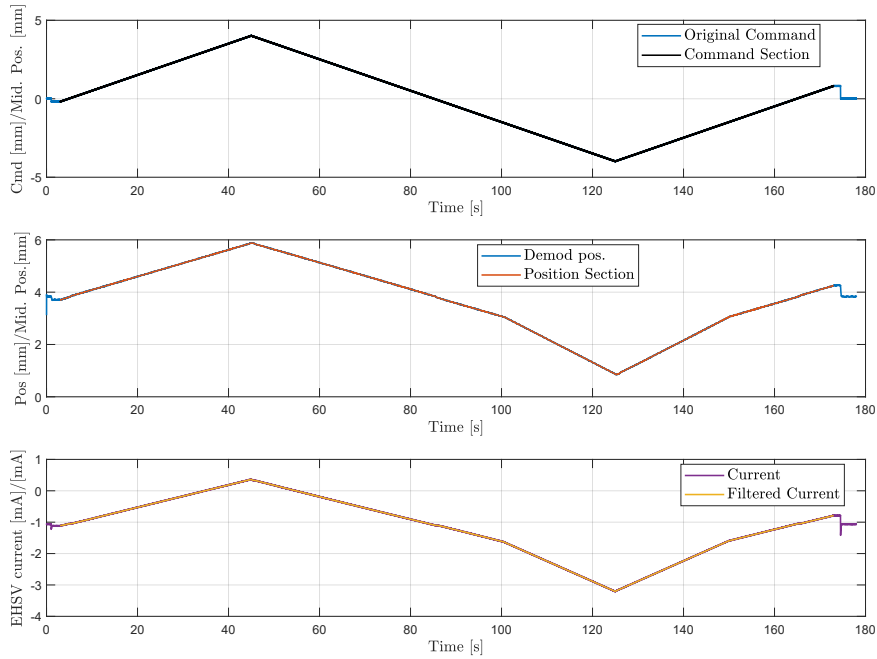


Figure 5.41: Elevator *Sequences* n.13 command

$$| I_t > I_{t_{max}} |$$

Sequence n.13

Sequence n.13 has the main goal to extract the Elevator RD characteristic curve in steady state condition, plotting the input current as function of main ram position. Due to the RD action over the EHSV spool, the ideal way to perform this test would be by slowly commanding the unit in open loop within the nominal RD range of action, and measuring the corresponding main ram position. However, this kind of test would not be effective in case of degraded RD: a current command above the actual RD saturation, would move indefinitely in open loop the main ram, which would continue until its end-stop and excluding any steady state analysis. In order to overcome this problem, the *Sequence* has been designed in closed loop, with a slow triangular ramp signal with a slope of 0.1 [mm/s], spanning the position range 2 [mm] above and below the nominal position RD ranges.

The plots in Fig. 5.41 illustrates both the closed loop command, the relative main ram position and EHSV current from the control loop circuit. In particular, the first subplot shows the complete command in blue and the extracted part in black which need to be taken in account for the RD characteristic extraction. From

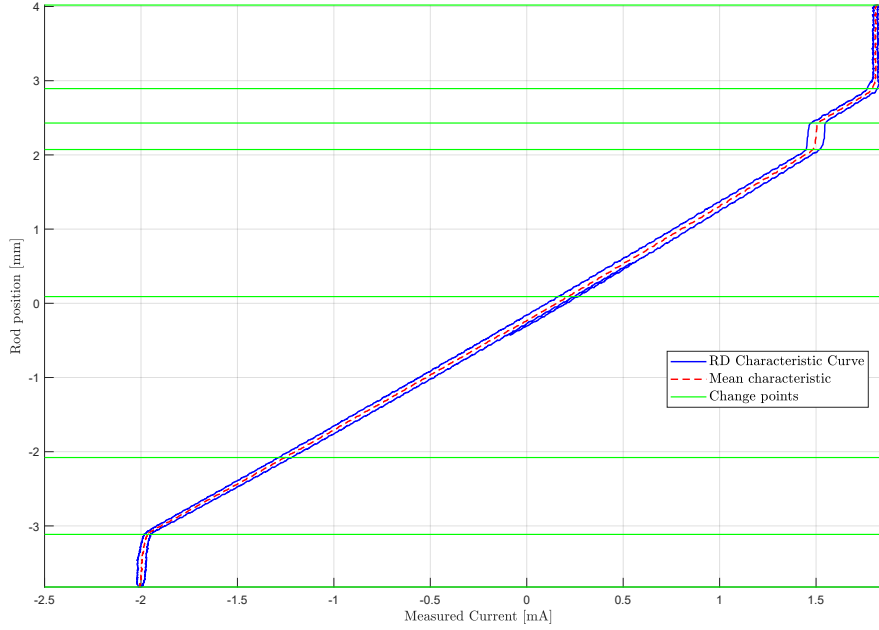


Figure 5.42: Elevator *Sequences* n.13 RD characteristic

the complete signal, the initial and final steps to its reference null position need to be excluded in order to consider just the slow ramp signal in both ascending and descending sections. The RD characteristic curve can be plotted as the red part in the second subplot as function of the filtered current in the third plot. The obtained results are shown in Fig. 5.42 with blue line and it represents the flipped curve in Fig. 2.12

This curve represents the starting point for the HFs extraction of this *Sequence*. Looking at this graph, it is possible to notice an hysteresis due to the different positions the main ram reaches during the rising and falling ramps. From the obtained curve, the mean slope can be calculated interpolating, for each unique ram position, the mean current between the ascending and falling curve portions. The obtained results are plotted with red dashed line. From this line, `findchangeslope` identifies any significant change point, which are marked with green horizontal lines. For each interval between two consecutive change points, the code linearly interpolates the portion of mean curve in order to extract each local slope. Averaging these mean values, the code evaluates the global RD stiffness k_{rec} , which need to be compared with the nominal K_{RD} in eq. 2.1. Furthermore, the first and last interval between change points defines the saturation limits of the characteristic curve.

Fig. 5.43 shows the extraction of the first two HFs: the RD bias position x_0 and bias current I_0 . These indexes are extracted taking as reference the mean curve distances from the origin coordinates. Since this test is performed in active mode,

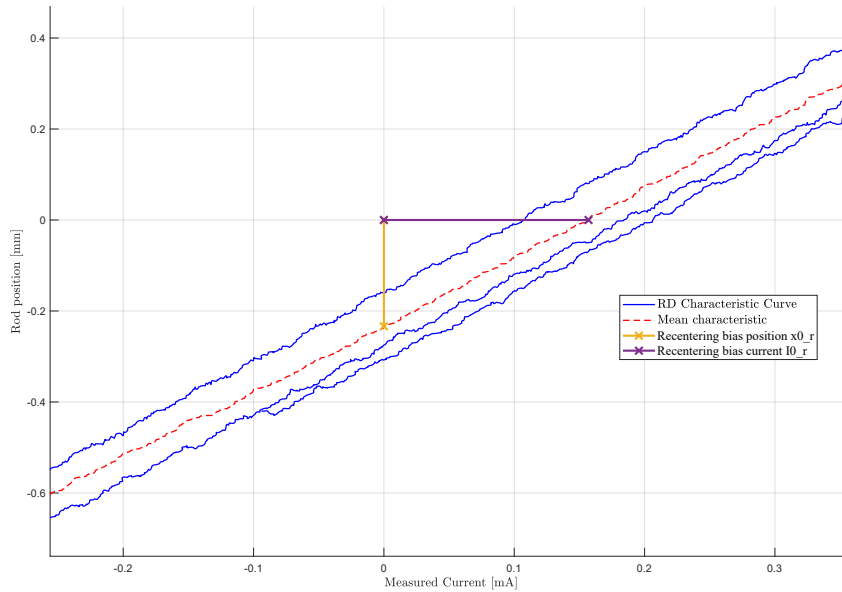


Figure 5.43: Elevator *Sequences* n.13 RD characteristic: x_0 and I_0 evaluation

these values are actually influenced also by the eventual EHSV null bias current. In particular, this latter parameter can be considered as an additional HF to be extracted when the unit is in damping mode, and considering just the command and measured EHSV spool position as in *Sequence* n.26.

Another important HF to be extracted from the red dashed mean curve is a possible deadband area, defined as a region where the mean curve presents a sudden change of slope due to a significant ram position variation caused by a small current change [32]. This deadband area can be detected by checking which change points interval is characterized by a mean curve with slope that differs more than $\pm 30\%$ of the nominal K_{RD} . From this evaluation, the first and last saturation intervals are excluded.

Fig. 5.44 shows the results of this analysis. The black dashed lines mark the curve saturation limits, while the yellow dashed ones identifies the deadband coordinates $x_{db_{rec}}$ and $I_{db_{rec}}$. A more detailed view about these HFs is shown in Fig. 5.45. The vertical green line represents the deadband amplitude db_{rec} , which exceeds the threshold of $\pm 30\%$ of the nominal stiffness K_{RD} . In order to exactly extract this amplitude, the linear mean curve sections of the previous change points intervals are prolonged in order to intersect the vertical line $I_{db_{rec}}$. The presence of a deadband area can be a good sign of an excessive clearance in the RD assembly. For example, a backlash between Control Shaft and Bell Crank may affect the final main ram position due to a specific current command. The case of Fig. 5.45 shows how the position suddenly changes even if the measured command current

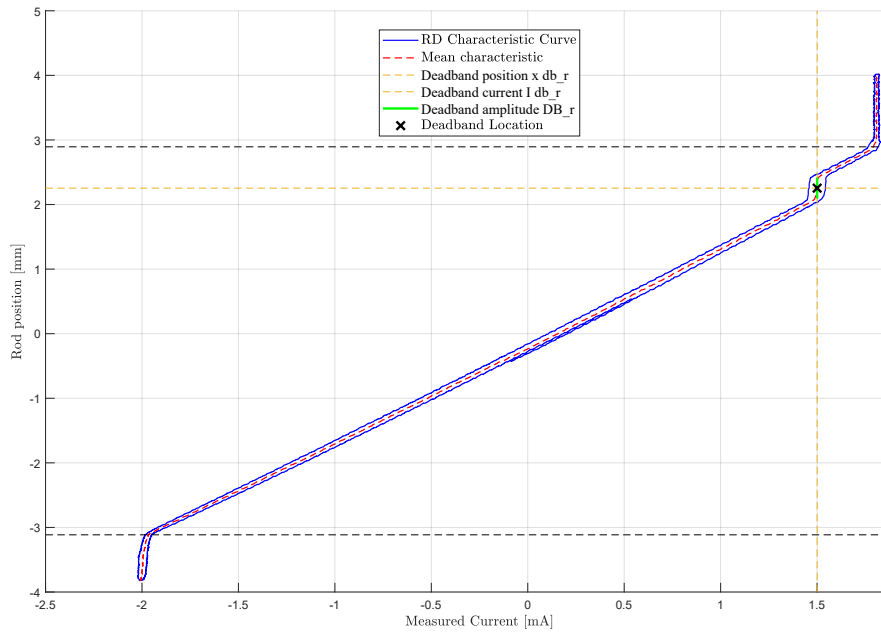


Figure 5.44: Elevator *Sequences* n.13 RD characteristic: deadband identification

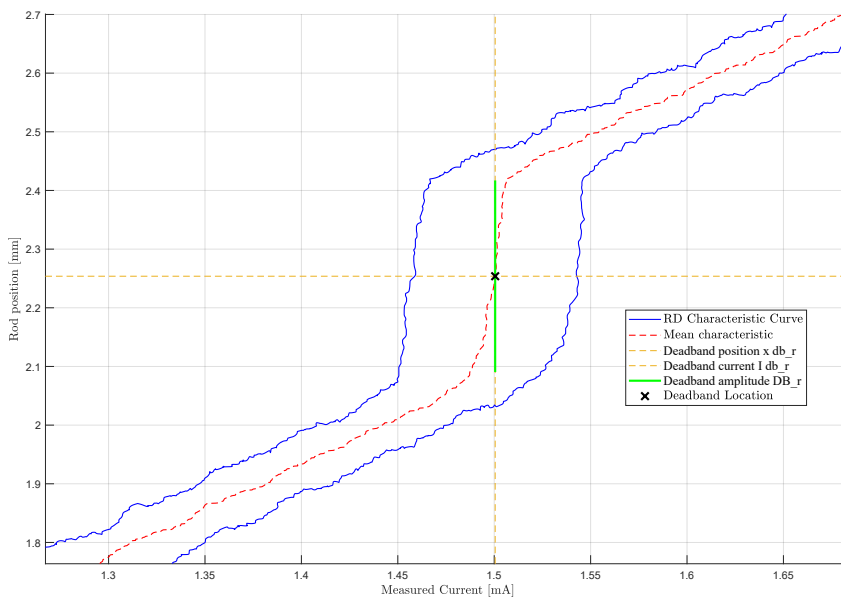


Figure 5.45: Elevator *Sequences* n.13 RD characteristic: deadband identification (zoom)

remains constant in this interval, which is sign of a missing Bell Crank rotation,

with corresponding lack of Mechanical input movement. Due to the Bell Crank backlash, the RD action on the EHSV spool is ineffective, in fact the control loop does not change its compensating output. Once this installation free-play between Control Shaft and Bell Crank is recovered, the RD is able to act correctly again over the EHSV spool and the control loop needs to rapidly change its output command current to minimize the error between command and expected lower position.

Sequence n.14

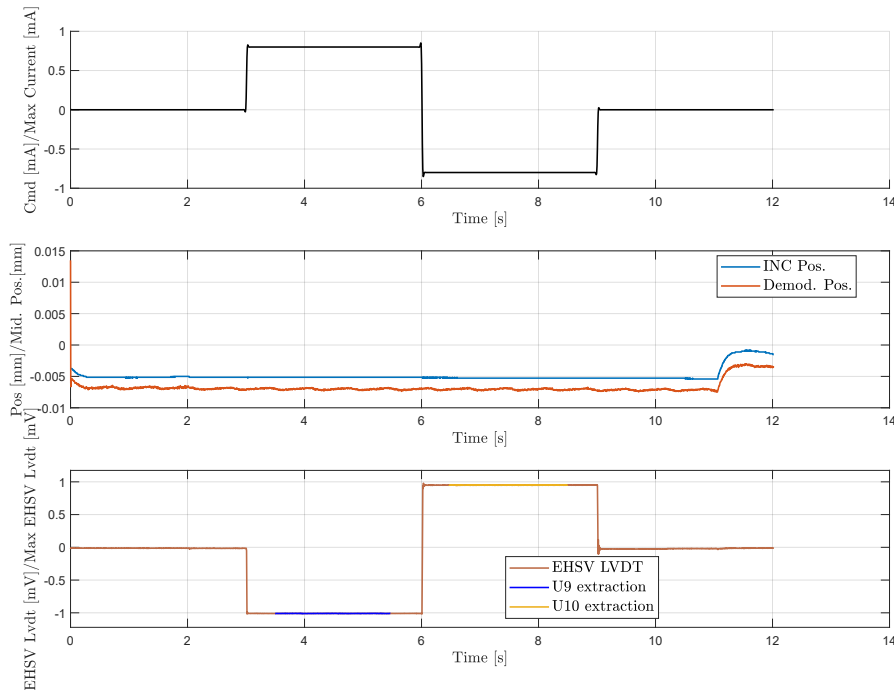
Sequence n.14 has been designed in order to extract two important Elevator EHSV HFs: the minimum $U9$ and maximum $U10$ EHSV spool end-strokes. These values needs to be reported in [mVrms] and they are extracted from the demodulated EHSV LVDT channel. In order to isolate the EHSV behaviour, the unit is set in damping mode activating both the electrovalves and it is commanded in open loop with two fast current steps to I_{max} and I_{min} . The first subplot in Fig. 5.46 illustrates this command in black line, and the main ram response in damping mode is highlighted by the second subplot where basically the piston is slaved in its null position. The third graphs shows instead the signal portions that are isolated using `stab_finder` in order to evaluate $U9$ and $U10$. These HFs are calculated as the mean values of the violet and yellow portions in the third subplot. Out of ranges values of these HFs can be a significant sign of an anomalous movement of the EHSV spool in its sleeve due for example to an incorrect command flow, an excessive rigid feedback spring or presence of debris in the EHSV second stage.

Sequence n.15

Sequence n.15 includes six crucial tests about the health status of both EHSV and RD. As reported in Fig. 5.47, the *Sequence* is in active mode, and the unit is slaved with two position steps firstly in extracted and then in retracted positions. The steps ensures that the RD apply a constant load over the EHSV jet-deflector, reaching its saturation.

During these steps in closed loop, the unit reaches the goal positions and the EHSV spool in its null position $U0$. This HF is evaluated as the mean value of the averages signals extracted by the EHSV LVDT, and highlighted in blue in the third subplot of Fig. 5.47. An out of range value of $U0$ may be caused by two possible irregularities:

1. Incorrect calibration of the RD. If the characteristic force curve in Fig. 2.11 is positively or negatively biased, its saturation limits may be shifted and therefore an excessive or low load is applied over the EHSV jet-deflector when the unit is slaved in any position outside the linear range.
2. Uncorrect movement of the second-stage spool in its sleeve. This possible failure may be due to debris or a wearing of the EHSV feedback spring.

Figure 5.46: Elevator *Sequences* n.14

Both the listed failures highlights how this HFs is not able to determines uniquely whether there is a failure in the EHSV or the unit just need a recalibration of the RD. For this reason, four other HFs have been extracted from this *Sequence* to help separating these effects, as reported in Table 5.4

IE and IR extraction

Focusing on the RD, *IE* and *IR* represent the nominal currents in extracted and retracted stop the EHSV need to counteract the RD action. These two values are evaluated as the mean value of the current signal isolated by `stab_finder` during the position steps. These areas are highlighted in green and yellow in the second subplot of Fig. 5.47. These two features are significant to evaluate the RD health status, which can includes damaged components or it may just need an external adjustment. In particular, the CMM indicates as troubleshooting for these HFs to recalibrate turning the eccentric screw in Fig. 5.48, which is directly connected to the bell-crank eccentric part. Turning the eccentric (1), it is possible to regulate the springs pre-load, therefore tuning the maximum forces the RD can apply on the EHSV jet deflector. Looking at the plot in Fig. 2.11 in chapter 2, this regulation process fixes the saturation limits the nominal linear RD force characteristic curve. The only way to estimate these loads is to extract the values of *IE* and *IR*, which

represent the current the EHSV needs to counteract the RD action. If these two HFs are out of range, it is very likely that the RD needs to be recalibrated but does not include any damaged component. It is important to notice that the RD is the only Elevator part which may need a recalibration to work properly, without any parts exchange. Correctly identifying if the RD is damaged or needs to be adjusted is a good improvement in the diagnostic procedure in terms of components cost and repairing time. Several previous works have investigated how to automate the adjustment process [50], to evaluate if the RD is adjustable or not [35] and how it is possible to extract additional HFs while adjusting the RD with a collaborative robot [49].

$SNR(U_{sek})$ and $f_{U_{sek}}$ extraction

In order to check the EHSV spool health status, two additional HFs have been defined from the secondary coil LVDT signal analysis. As already reported in Chapter 2.6.1, the Elevator EHSV is equipped with a four-wires LVDT which is more affected by noise if compared with the six-wires configuration. The Appendix A.1 illustrates different demodulation techniques for this kind of sensor, and how a noisy secondary signal influences the final demodulated one, according to the adopted technique to process it. The HyDiag SW determines the EHSV spool

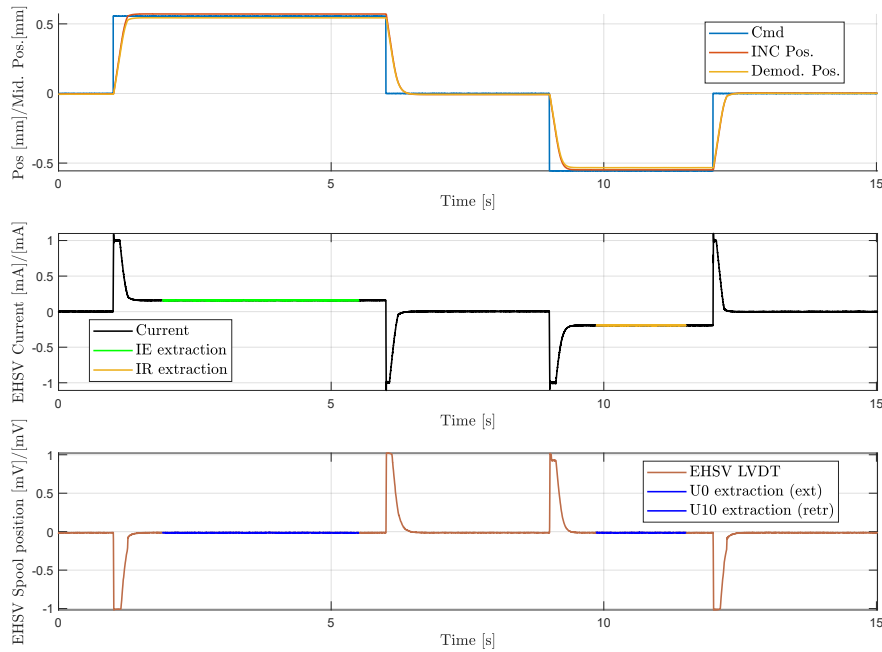


Figure 5.47: Elevator Sequences n.15

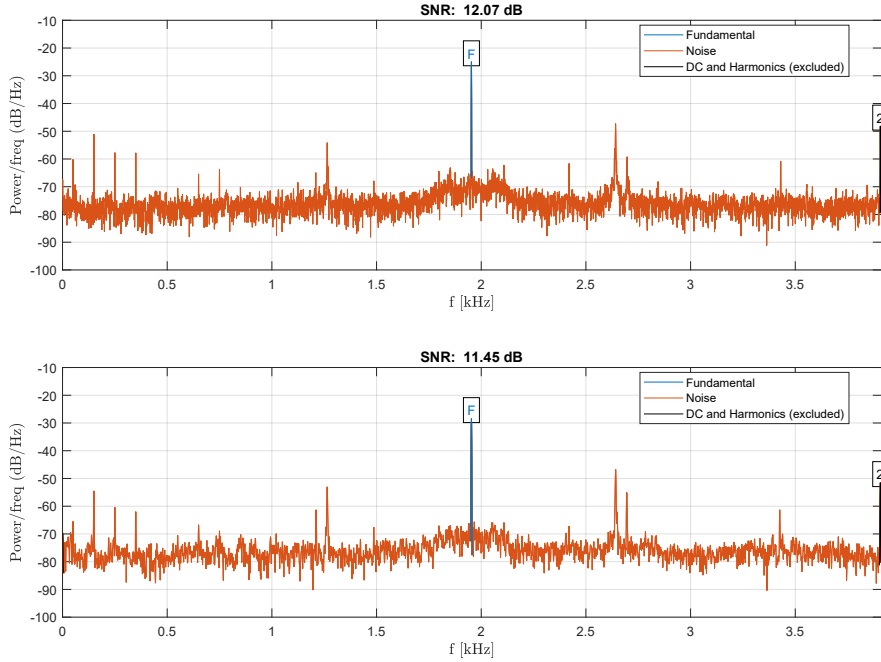


Figure 5.49: Elevator *Sequences* n.15: SNR and f_{peak} extraction from PSD analysis

lower frequencies than the reference f_{supply} . For example, if the signal power spectrum presents unexpected peaks at higher frequencies than the reference one, it can be a symptom of electrical failure of the LVDT channel. It is important to remember that any possible external interference during the measurement is eliminated through the `denoising` function during preprocessing. If the power spectrum shows any anomalous low-frequency peak, it can be a symptom of an unexpected spool oscillation, due to a mechanical or hydraulic instability. For this reason, together with the $SNR(U_{sek})$ ratio, the $f_{U_{sek}}$ peak oscillation is extracted from the power spectrum analysis of the secondary coil. The plots in Fig. 5.49 illustrates the results of this analysis over the portions of raw secondary coil signal in the same time-interval of evaluation of U_0 .

In both these plots, it is possible to observe how the code can easily extract the fundamental harmonic (in blue) from the remaining part of the signal, simplifying the SNR evaluation. The final HF to report is an average value of both the SNR ratios evaluated while the unit is in extended (first subplot) and retracted positions (second subplot). The same is valid also for the relative f_{peak} .

Sequence n.16

Sequence n.16 has been designed in order to evaluate a fast hysteresis around the null position. Using the same approach of *Sequence* n.13, the code needs firstly to extract the useful portion of both command and position signals, which correspond to the ramps in Fig. 5.50. Once these sections have been extracted, the code extracts and plot the position hysteresis around the null position, as in Fig. 5.51. From this curve, it is possible to extract both the maximum hysteresis h_x and the maximum distance from null command and reached position x_b .

Sequence n.17 and 18

Sequence n.17 has been designed in order to extract two traditional HFs of the first EV of both Elevator and Aileron: $U5$ and $U6$. Comparing the plots in Fig. 5.52 and 5.53, it is possible to notice how the command shape is the same for both the units, while the EV voltage signal looks mirrored. This is due to the opposite ways of switching from active to damping mode for these units. For both the cases, $U5$ represents the necessary command the EV needs to switch the unit from active to damping mode. The switch between these two working modes is evident from the first subplot of both the figures: in fact, the unit stops following the sinus wave command as soon as it reaches a specific value which is exactly $U5$. The opposite considerations are true for $U6$, which is the command the EV needs to start the switch back to active mode. The extraction of these values has been eased by the function `stab_finder`, which directly extract the instants of time when the unit is slaved in damping mode (highlighted with black dashed lines).

Sequences n.18 has the same shape of its previous, but its just dedicated to the Elevator unit, in order to perform the same test on the EV2 and extract the equivalent $U7$ and $U8$ HFs.

Sequence n.19 and 20

Sequences n.19 and n.20 have been shaped in order to retrieve important information about the switching conditions from damping to active modes for both Elevator and Aileron. It is important to notice that, similarly to the previous *Sequences* n.17 and n.18, the latter 20th is characteristic only for the Elevator, since it needs to evaluate the same behaviour of *Sequence* n.19 but for the second EV. For this reason, the plots in Fig. 5.55 and Fig. 5.57 are identical, except for the second subplot command for the EVs. However, it is clear from all the plots in this section how both the position and voltage EV command have been modularly built in order to extract the same HFs, beyond the type of tested unit.

Focusing on *Sequence* n.19, the plots in Fig. 5.55 and 5.56 describe the methods used to extract the switching response time $T1$ for both Elevator and Aileron. $T1$ has been defined as the time interval the main ram takes to start moving, when the

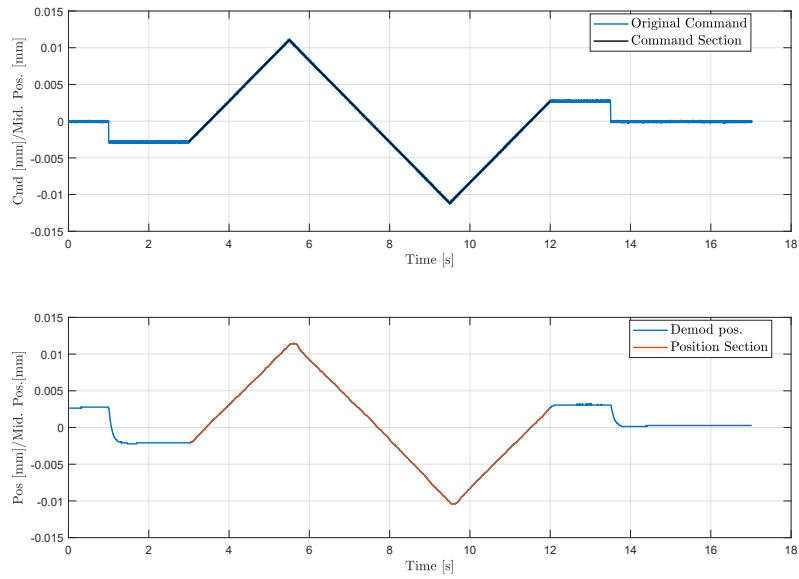


Figure 5.50: Elevator *Sequence n.16*

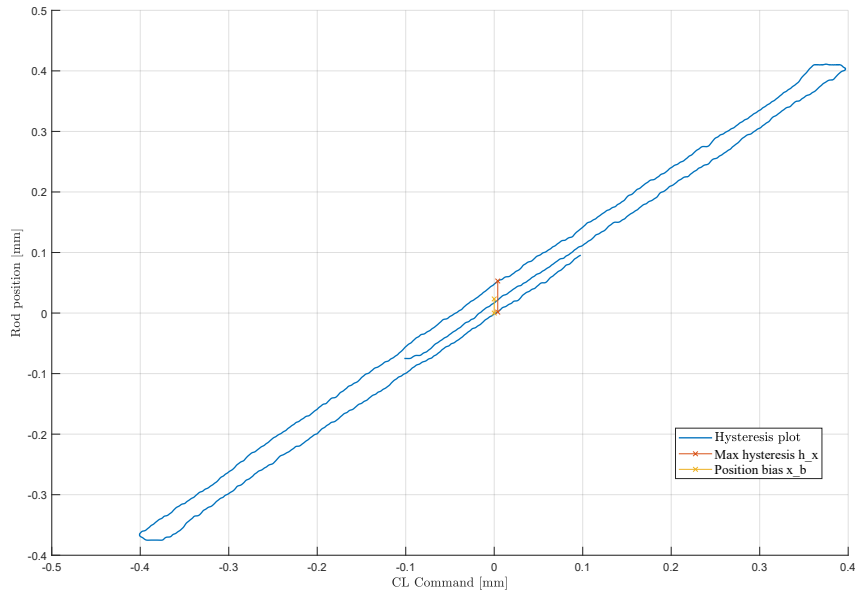


Figure 5.51: Elevator *Sequence n.16* Hysteresis Plot

unit switches from damping to active mode. For both Elevator and Aileron, after two seconds, the unit is set to damping mode switching on or off the EV1 respectively for the Elevator and Aileron. The switch is clearly evident in the second and third subplots of both the Figures: as soon as the EV1 is triggered, the MSV spool

steeply moves to its totally closed position, shutting down any connection between the main ram chambers and the EHSV.

For both the units, after the 5th second, the EV is triggered again in order to set the unit back in active mode. In particular, this switch in Fig. 5.55 appears a little bit later (at around the 6th second). This delay is actually related to the possibility to extract the breakdown voltage $VEV1$, which represents the minimum signal peak due to an instantaneous discharge of the EV solenoid. The same strategy has been used in *Sequence* n.20 for the evaluation of $VEV2$ of the second electrovalve. The time range $T1$ is evaluated from the EV switch command until the first measured main ram movement, which determines the switch effectiveness. This latter value can be easily extracted using the function `move_finder` over the main ram position signal (in the first subplot). In all the plots, a small zoomed window is dedicated to the extraction of $T1$ (or $T2$ for the *Sequence* n.20).

In analogy with Fig. 5.32, the same analysis over the MSV LVDT step signal can be conducted in order to find any anomalous "elbow" during the spool stroke. As shown in Fig. 5.58, the function `elbow_detection` is able to detect any slight change of slope in the signal step, which can be a symptom of an incorrect spool movement.

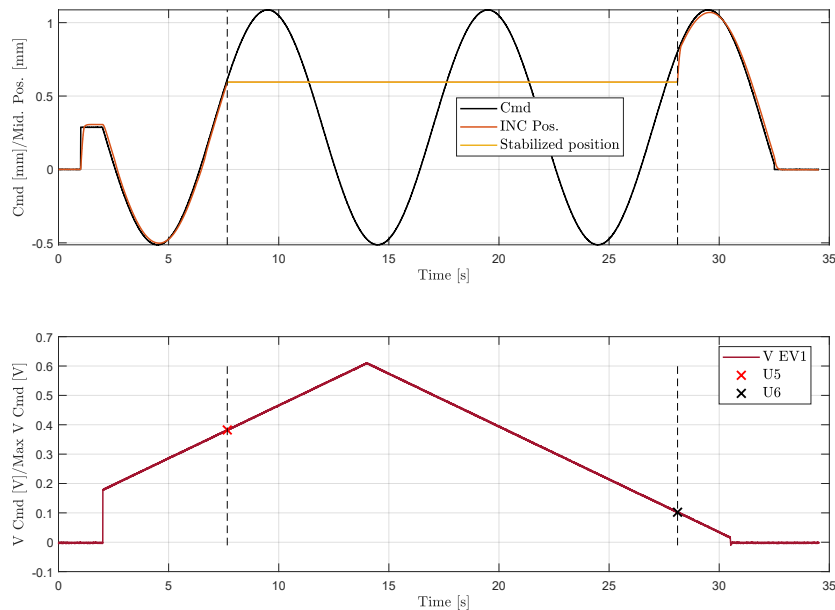


Figure 5.52: Elevator *Sequence* n.17

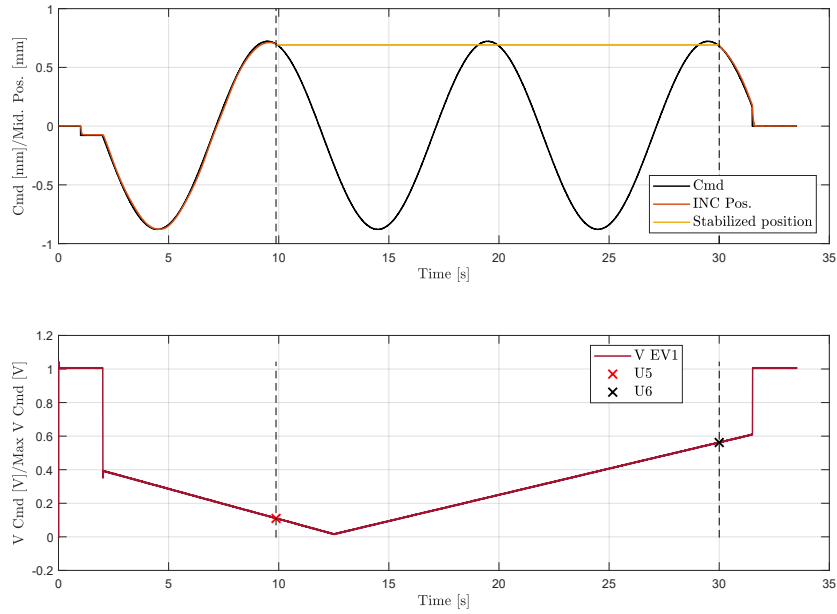


Figure 5.53: Aileron *Sequence* n.17

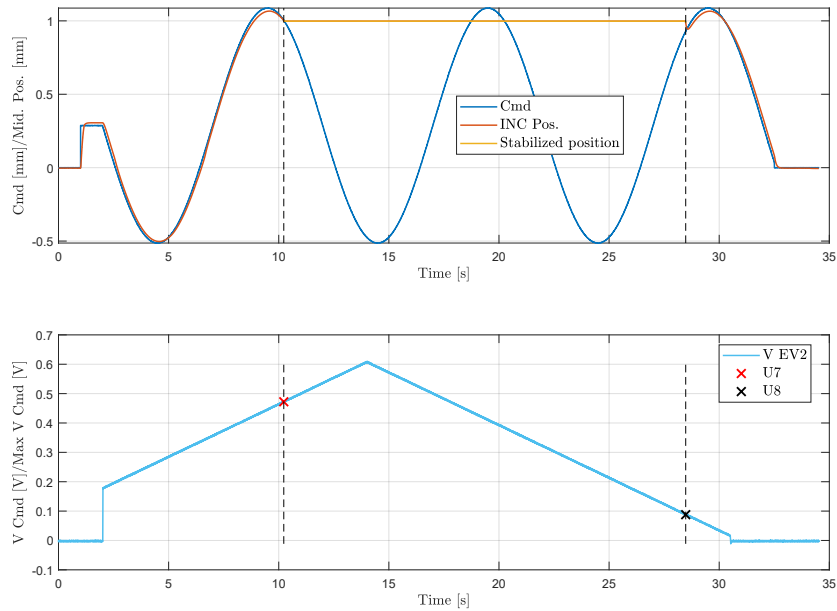


Figure 5.54: Elevator *Sequence* n.18

Sequence n.21 and 22

Sequences n.21 and n.22 are shaped with the goal to simulate an abrupt switch from active to damping mode while the main ram moves. Analogously to the previous couple of *Sequences*, the 21th has been used for both Elevator and Aileron, while the 22th can be applied only over the Elevator since it replicates the previous *Sequence* behaviour with the other EV2.

The first plot in Fig. 5.59 shows how the unit is commanded after 3 seconds in a step from retracted to extended position. In order to simulate a sudden switch to damping mode while the main ram is still moving, the EV1 is activated at one third of the main ram stroke, determining the MSV spool movement which return in its closed position. This step response is analysed in order to extract the falling time $T3$ (and $T4$ for *Sequence* n.20). The dedicated window in the third subplot shows the instants which are used to evaluate this time range, highlighted with blue and black lines. The red one indicates the moment when the EV is switched on: the distance between red and dashed lines in the window represents the time needed by the oil to flow from the EV to the MSV command chamber.

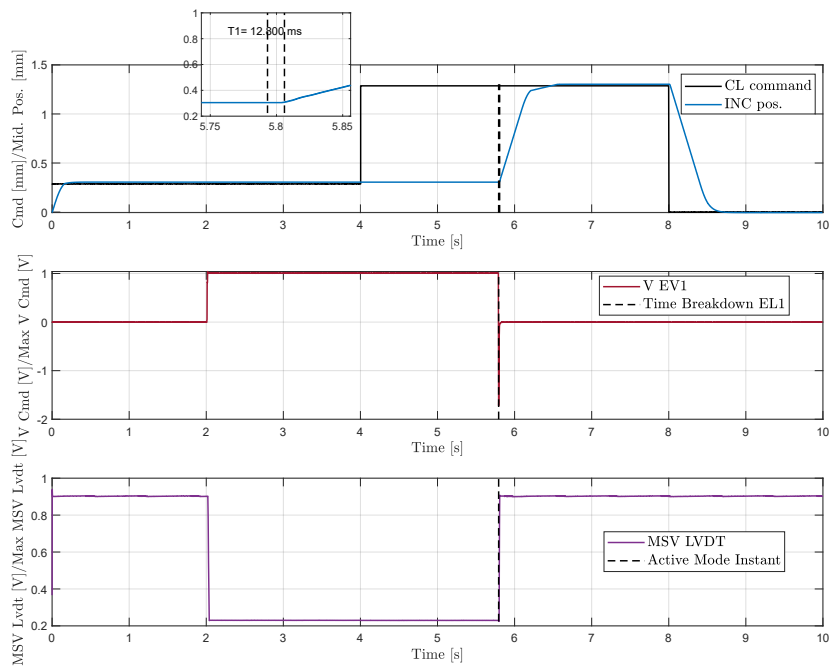


Figure 5.55: Elevator *Sequence* n.19

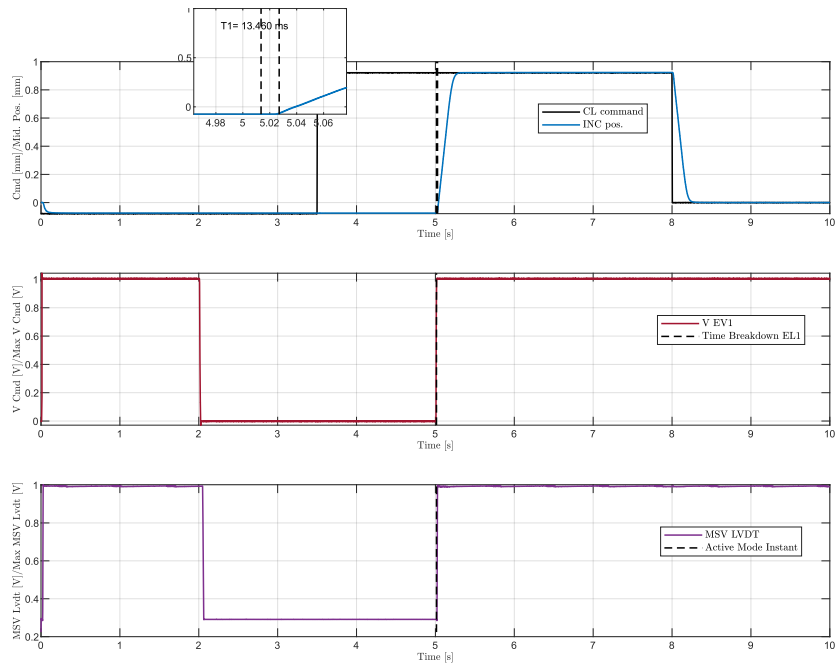


Figure 5.56: Aileron *Sequence n.19*

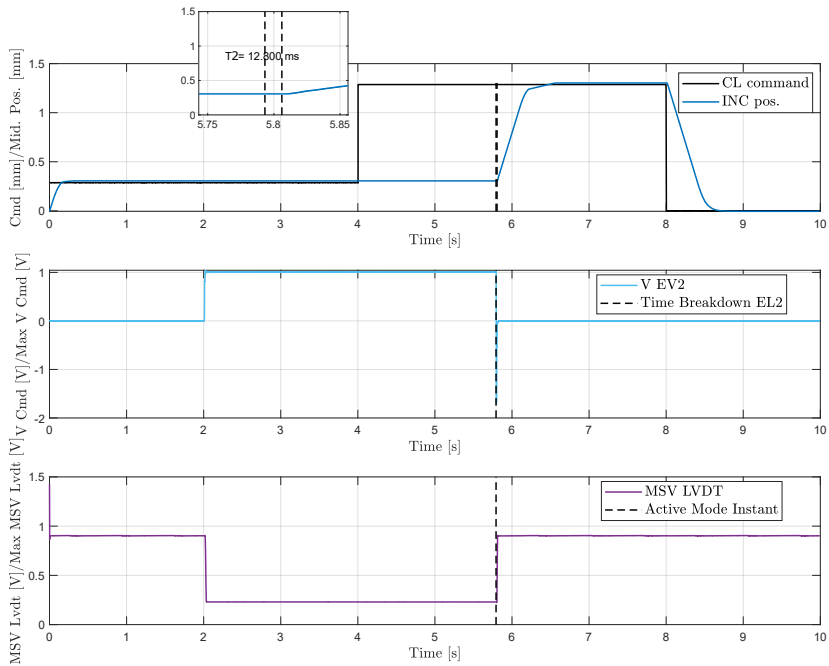


Figure 5.57: Elevator *Sequence n.20*

Sequence n.23

The main goal of *Sequence n.23* is to extract the HFs of the main ram position sensor. It is important to remember that the Elevator differentiates from the other

two use-cases since it is equipped with an RVDT, while both Aileron and Spoiler include a linear LVDT. All these sensors are configured according to the six-wires scheme (see 2.6.1), so it is possible to extract both the rms voltages from the secondary coils $U1_{rms}$ and $U2_{rms}$. From these signals, it is possible to evaluate the ratio R in eq. 2.8 in order to check if the position sensor is correctly set. The CMM indicates to slave the main ram in defined retracted and extended positions (respectively ret_{goal} and ext_{goal}) and to evaluate if the ratio R is included in an acceptable range. The way this test is implemented within the Hydiag procedure is described in Fig. 5.61, 5.62 and 5.63.

Observing these plots, it is possible to notice how the signals of both Elevator and Aileron are equivalent, firstly commanding the unit to extended position and then in retraction. The Spoiler signal in Fig. 5.63 presents just extended portion, since the null position correspond to totally retraction. Anyway, in all these plots, the first subplot shows the position command and relative measurements from both the external incremental sensor and the demodulated demodulated signal from the position sensor. The second subplot illustrates the $U1_{rms}$ and $U2_{rms}$ signals which are extracted from the raw sinusoidal signals of the secondary coils with the function `ratio_demod` (5.2). The command signal has been designed in order to make the unit slowly moving within a range of ± 2 mm around the goal position. In this way, the ratio $R1$ in extension and $R2$ in retraction, according to Eq. 2.8, extracting

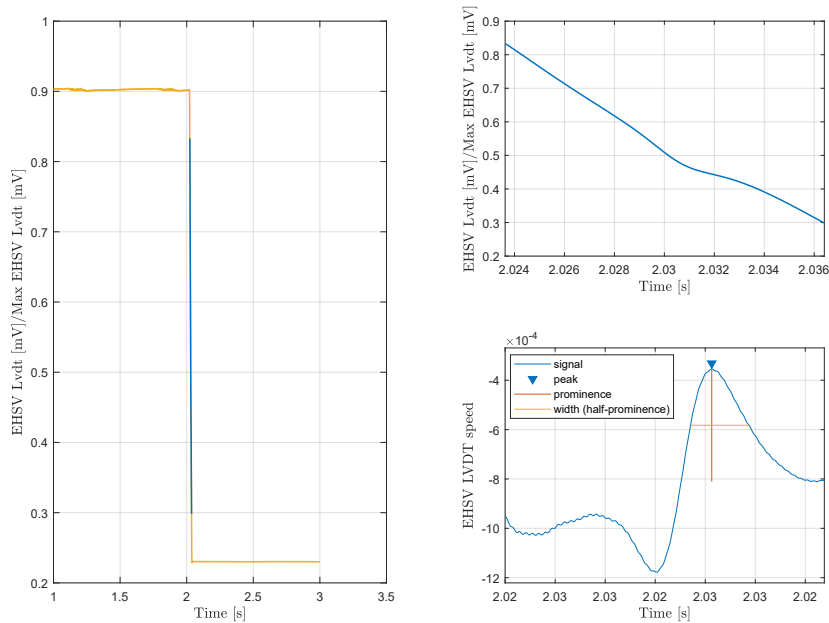


Figure 5.58: MSV LVDT Elbow Detection

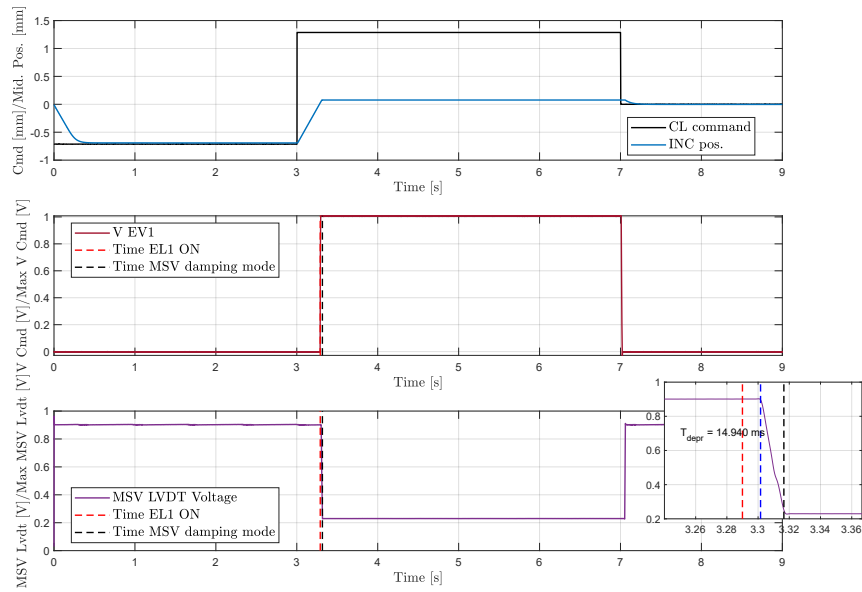


Figure 5.59: Elevator *Sequence n.21*

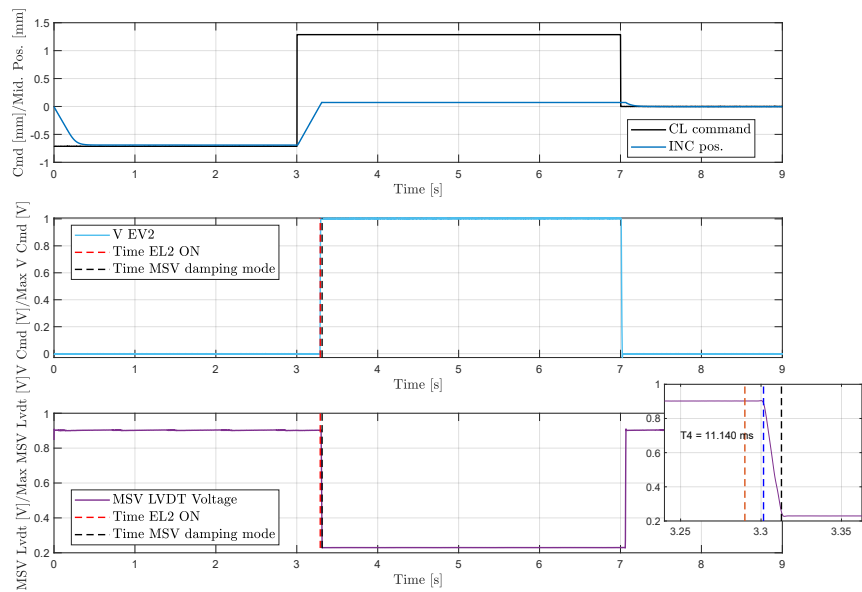
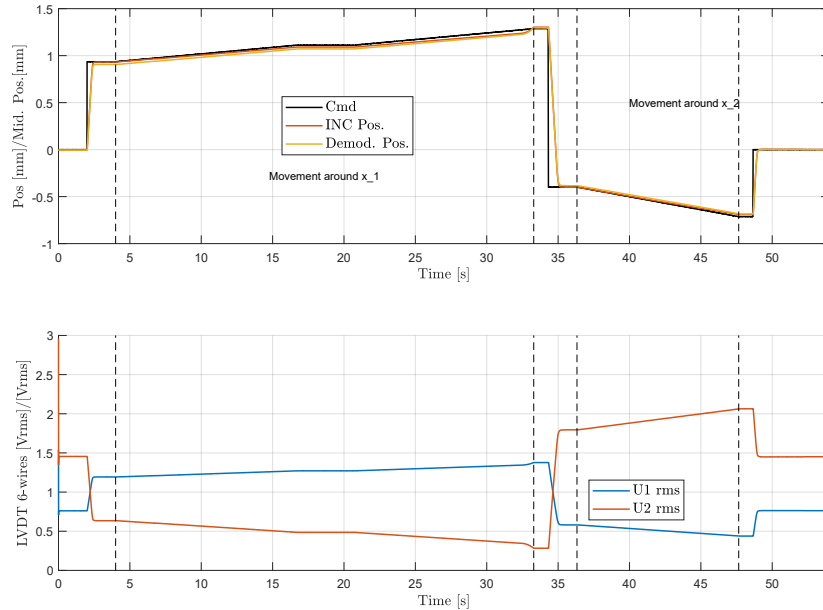


Figure 5.60: Elevator *Sequence n.22*

the mean values of $U1_{rms}$ and $U2_{rms}$ in these position ranges.

Before evaluating $R1$ in extension and $R2$ in retraction, the code checks if the

Figure 5.61: Elevator *Sequence* n.23

unit has reached the effective goal position from the incremental sensor signal, which is considered as reference since the *Sequence* goal is to evaluate any fault in the integrated position sensor. In case the main ram has not reached the goal position, the code prints out an warning message and it linearly extrapolates $R1$ or $R2$ according to the collected measures. In case of the Elevator unit, an anomalous value of one of these features may be fixed simply adjusting the RVDT zero-setting, since this position sensor is also mechanically connected to the RD which may need a consequent adjustment. For both Aileron and Spoiler, the only solution would be to dismount the main ram LVDT and check the ferromagnetic core position and wirings connections. This *Sequence* is also designed to extract the EHSV resolution r_x for all the tested units. Looking carefully the position command signal in all the plots, it is possible that the rising linear command within the acceptable position range presents a short steady part, which has been positioned around the nominal goal position. As reported in [32], this signal portion has been designed in order to extract the delay between command and main ram position response. Fig. 5.64 is obtained by plotting the measured channel from the incremental sensor against the position command, during the movement around the extended goal position. The steady part of this curve represents exactly the command variation the main ram needs to change its position. This HF can be easily extracted using `stab_finder` and it represents the resolution r_x . Too high values of this HFs can be due to an incorrect movement of the EHSV spool or a high degradation of the main ram

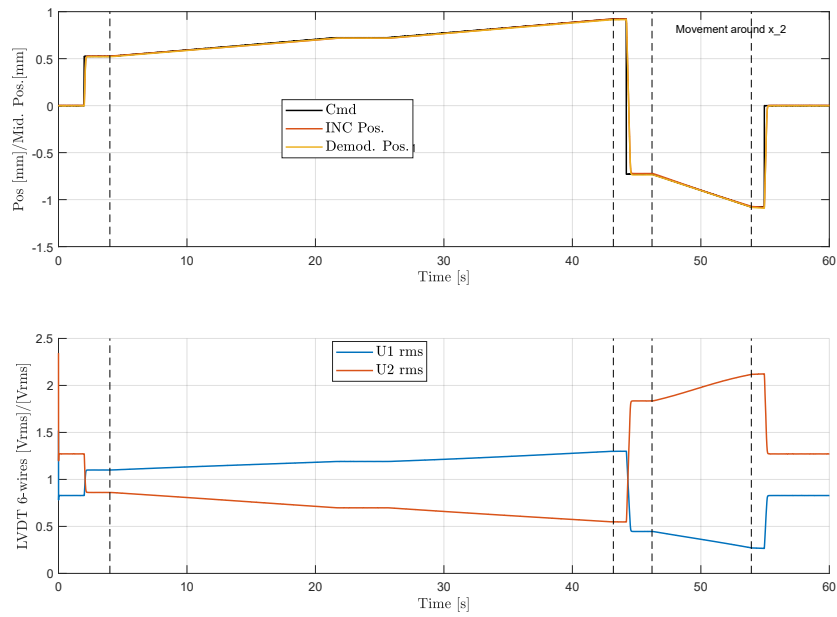


Figure 5.62: Aileron *Sequence n.23*

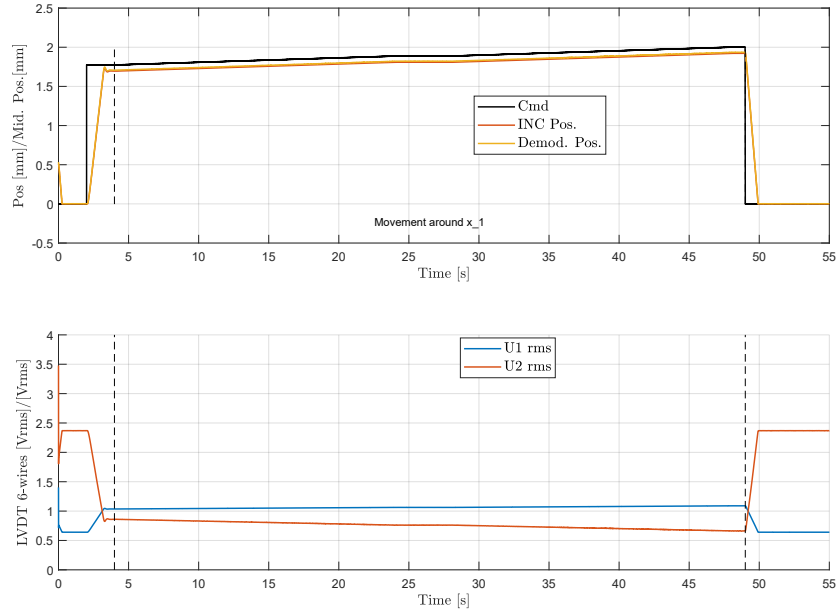


Figure 5.63: Spoiler *Sequence n.23*

dynamic sealings.

Sequence n.24

This *Sequence* has been integrated in the HyDiag testing procedure for Elevator and Aileron, with the goal to perform oscillation threshold test, prescribed in their respective maintenance manual. The standard procedure proposes to perform this test using a series of sinusoidal position command signals, with increasing amplitude and constant frequency. The HyDiag procedure proposes to shorten this method with a unique amplitude-swiping sinus wave, as reported in the first subplot in Fig. 5.65. These plots illustrate the Elevator behaviour, but the same considerations apply also for the Aileron.

The goal of this signal is to extract, in less than 35 seconds, the minimum required command $2\delta c$ to measure a reference peak-to-peak oscillation $2\Delta C$. This parameter is reported in the unit CMM and depends on the unit flight-hours. The $2\delta c$ HF is particularly significant to evaluate any EHSV degradation. The HF extraction code firstly filters the command input signal from any electrical noise with a simple second-order butterworth filter ($f_{cut} = 25Hz$). The second step is to correct any offset of the reference INC sensor, and then extract the sinus waves peaks of both command and position signals using the built-in Matlab function `findpeaks`. The extraction of $2\delta c$ is guaranteed by a while loop which stops when the requirements $2\delta c \leq 2\Delta C$ is fulfilled. The third subplot in Fig. 5.65 shows exactly when the code stops its iterations in order to find the dedicated HF. It is important to notice actually how the command signal has been overestimated

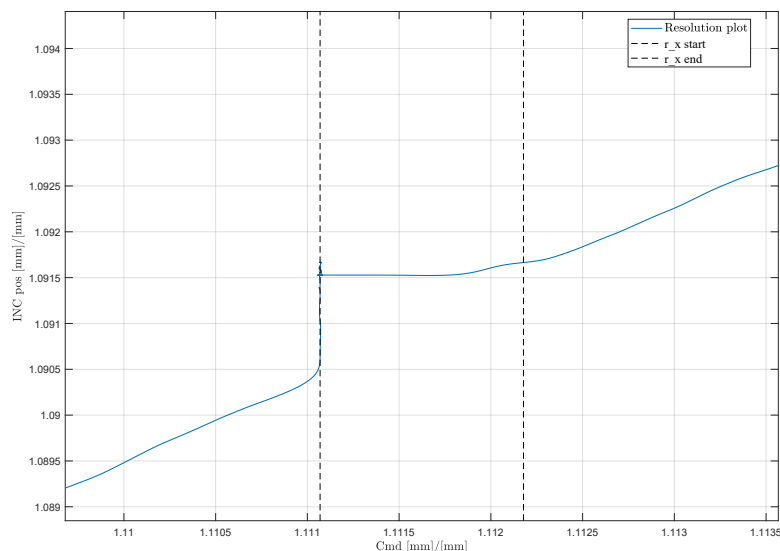


Figure 5.64: Elevator *Sequence* n.23 resolution plot

in order to consider even the possibility to have a totally defective EHSV. In the extreme case the command is not sufficient to extract $2\delta c$, the code prints out an error message to the user and in the final report which states the highly degraded EHSV health status.

Sequence n.25

Sequence n.25 has been designed for all the project use-cases, and it is based on a series of random position steps within the minimum retracted and maximum extended positions. In order to unify the testing conditions, these position steps have been actually defined during the signal design and fixed as command parameters. From the main response channels, it is possible to extract the average extended ϵ_{ext} and retracted ϵ_{ret} offsets. The HF extraction code is in fact able to distinguish between rising and falling steps, according to the sign of the recognized slopes (highlighted with vertical dashed-black lines). An excessive values of these HFs can be due to an inefficient EHSV or more likely to degraded dynamic sealings on the main ram.

This *Sequence* has been particularly useful during the prototyping phase, since it has been used to extract particular dynamic features to design a base-line physical digital twin of the tested unit. As reported in [18] and [17], this model has been used to test how the extracted HFs by the HyDiag protocol, when combined and

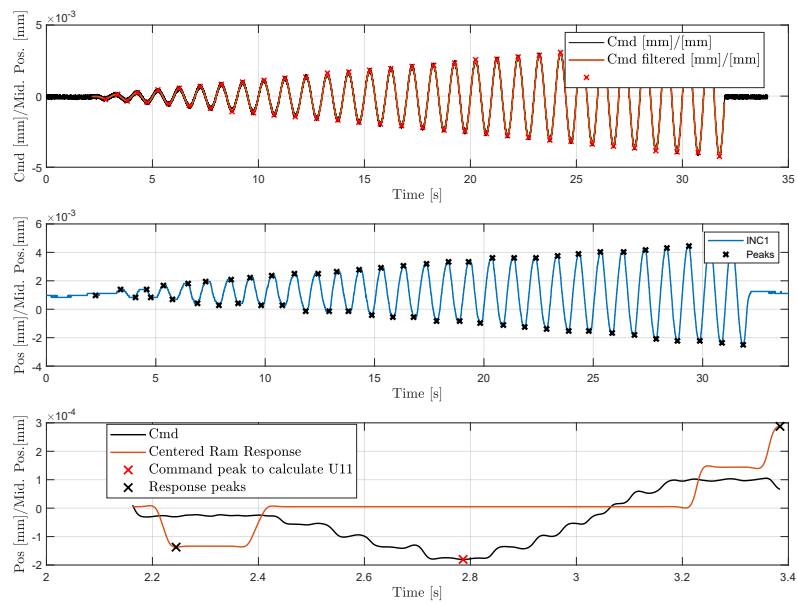
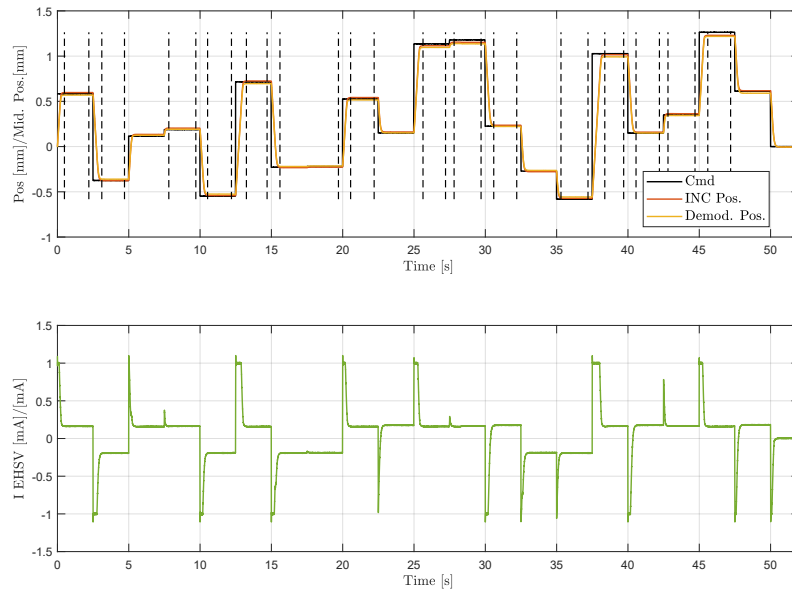


Figure 5.65: Elevator *Sequence* n.24

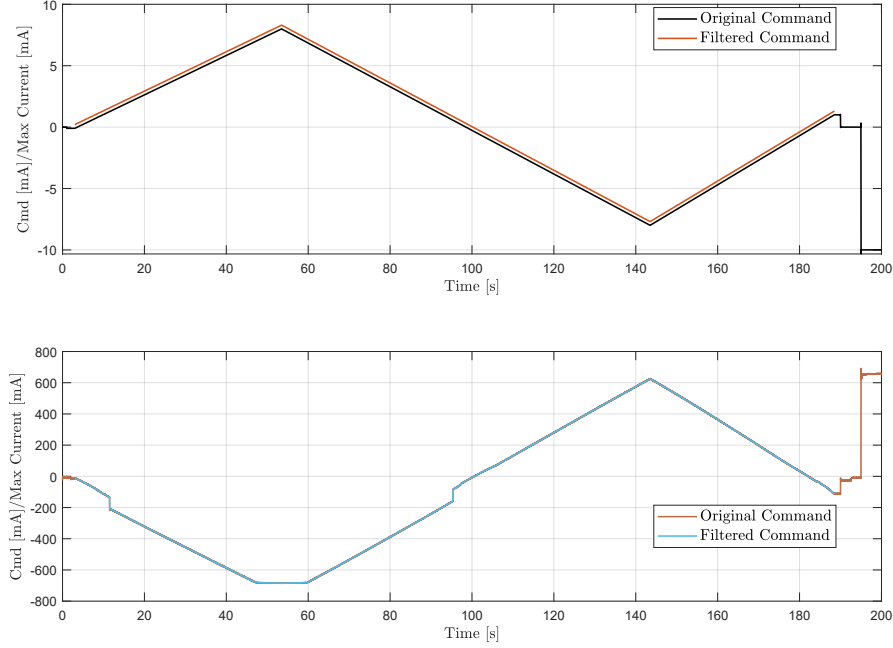
Figure 5.66: Elevator *Sequence* n.25

divided per target components, can be used to detect a degradation of a specific part of the tested unit.

Sequence n.26

Comparing Fig. 5.41 and 5.67, it is possible to notice some similarities regarding the command signal in the first subplot. The strategy used to design both the *Sequence* is indeed the same: apply a slow ramp signal (of about 3 minutes) in order to analyse the steady-state behaviour of the target unit and extract its characteristic curve. The main difference between the testing condition regard the unit working mode: while in *Sequence* 13 the unit is in active mode and receives a slow position ramp command, in the 26th on both the EV are turned off, setting the unit in damping mode. The first subplot in Fig. 5.67 shows the input current command to the Elevator EHSV: since the unit is set in damping mode, the open-loop command is intended to move the EHSV spool in order to extract its position. Plotting together the highlighted portion of command and the relative EHSV spool position, it is possible to extract the EHSV characteristic curve in damping mode. It is important to notice how the current command variates from I_{min} to I_{max} in order to cover the entire EHSV spool positions range.

The EHSV characteristic curve is exposed in Fig. 5.68. Using the same approach of Fig. 5.42, the red-dashed curve represents the mean slope, calculated as series of mean value between each unique ascending and descending spool position. The


 Figure 5.67: Elevator *Sequences* n.26 command

horizontal green lines are evaluated using the `findchangeslope` function.

Using the same approach for the evaluation of k_{rec} in *Sequence* n.13, the global EHSV stiffness k_{sv} is extracted averaging the mean-curve values in each portion defined by two consecutive green lines. As the RD characteristic curve, also from the EHSV one is possible extract any deadband area and its amplitude db_{sv} and consequently its mean coordinates xdb_{sv} and Idb_{sv} . All these HFs are shown in Fig. 5.69. As reported in [32], an excessive deadband can be caused by an excessive backlash between spool and EHSV feedback spring. This anomaly can be detected in this *Sequence* due to the signal shape and to the damping mode condition, which exclude any dynamic influence of the RD. However, it has been observed how a defective RD which needs a recalibration may influence the presence of an EHSV deadband, due to its connection with the EHSV jet-deflector: the RD initial setting influence also the null position of the EHSV jet-deflector through the EHSV mechanical input. Other two HFS which can be extracted by the EHSV curve are the null bias current I_{bias} , as the distance the axes origin and the current the spool needs to reach its null position, and the curve hysteresis value $hyst_I$.

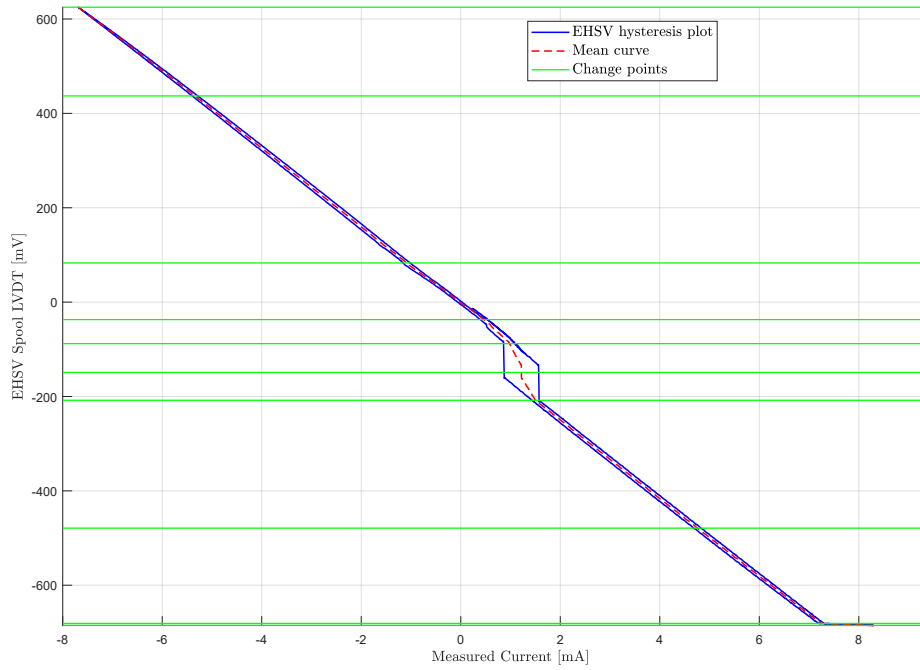


Figure 5.68: Elevator *Sequences* n.26 EHSV characteristic

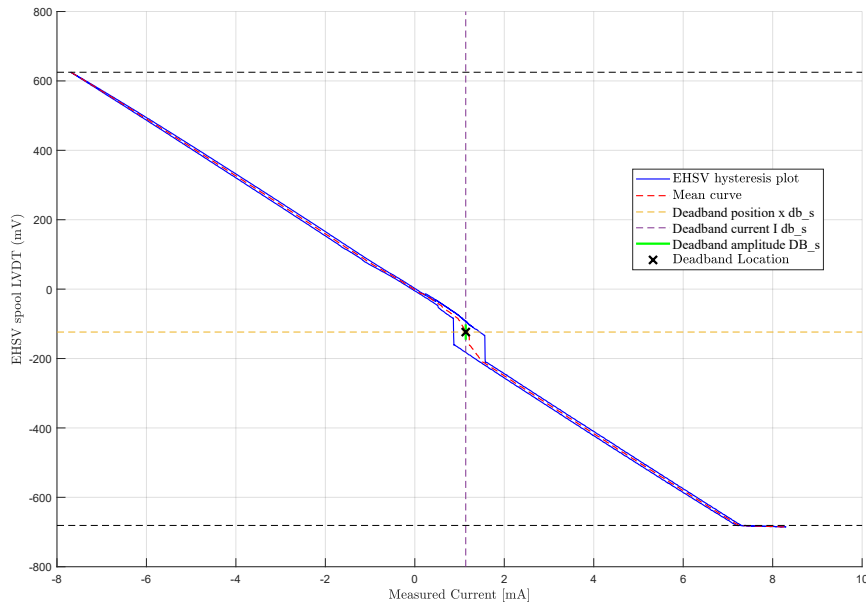


Figure 5.69: Elevator *Sequences* n.26 EHSV HFS evaluation

5.6 Hydiag Report

As already anticipated in the previous paragraphs, the final output of the Hy-Diag HFs extraction software is a comprehensive report which collects all the information extracted by the collected signals. The flowchart in Fig. 5.70 reports the steps the script `postpro_report_main` follows in order to print the extracted HFs in both the reports formats: `.csv` for further data-collection, and `.pdf` for the technician. As already mentioned in Fig. 5.1, both the software versions store both the measurements data and reports in back-up local drive. Fig. 5.71 shows how both the measurement and reports folders are named according to the label already anticipated in paragraph 5.2

Each subfolder contains the main outputs of the HyDiag HFs extraction software: the `.pdf` report, the relative `.csv` formats reports and eventual plots from the analysed *Sequences* (f.e: Hysteresis plots of both *Sequences* n.13 and n.26). Before building both the kinds of reports, the software creates the appropriate report folder according to the aforementioned ident-code. After this preliminary step, the code `postpro_report_main` principal tasks are to collect all the extracted HFs in two main structures:

- `feat_struct`: this variable includes all the traditional CMM HFs stored the previously populated structure `test_result`. At the end of each *Sequence* processing code, the extracted CMM HFs are collected in `test_result`, where also the relative CMM test limits are stored. Once this variable has been completely filled, the `postpro_report_main` code reads its values, stores the HFs according to the test number and, according to the HFs limits, stores the relative troubleshooting instructions if the HFs value is outside the prescribed limits. All these information are stored within the `feat_struct` variable.
- `feat_struct_notCMM`: all the additional HFs which are not reported in the CMM are collected in this variable, using the same approach of the previous one. In this case, both the limits check and the relative troubleshooting instructions are currently omitted. These additional information will be directly integrated in the HyDiag report when the *Diagnostic Module* will be ready to give additional instructions according to the health-status of each unit component. Currently, the *Diagnostic Module* is still in a prototyping phase, but its promising results are in details described in Chapter 6.

These variables constitutes the core of both the kind of reports. Firstly, the script concatenates `feat_struct` with `feat_struct_notCMM` and converts the final structure in a unique table with the format in Table 5.5:

This table is directly printed in `.csv` file and stored in the proper folder. In order to build the `.pdf` version, the *Report Generator Matlab* toolbox has been

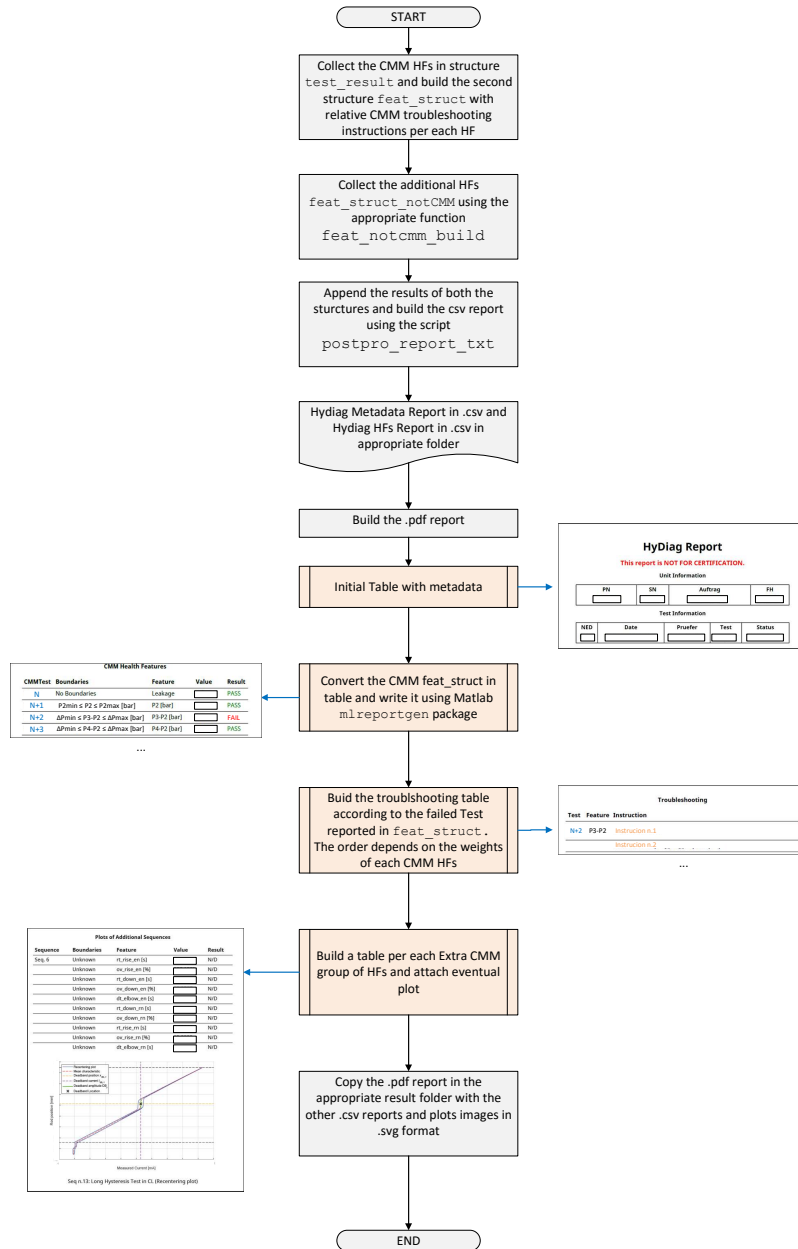


Figure 5.70: HyDiag Report generation flowchart

used. The graphic side of this report has been appropriately designed in order to meet the technician requirements, who can use it to read the collected test results

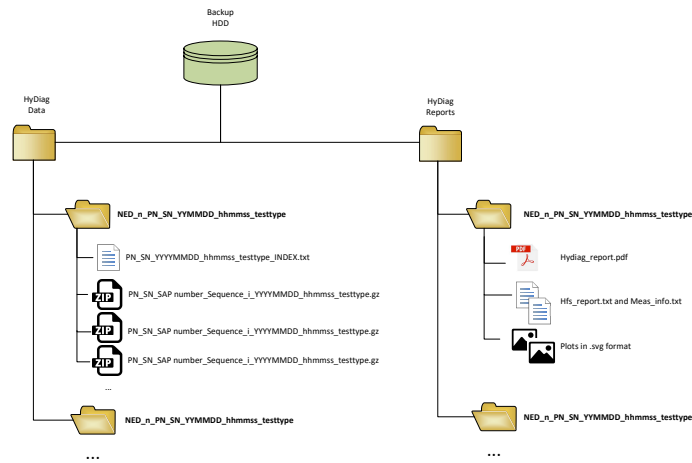


Figure 5.71: Folder organization for both data and reports

and eventual troubleshooting instructions. As reported in Fig. 5.70, the complete file is built following consequent steps:

- Step 1: build and print two tables with all the unit and test information, such as date and operator name;
- Step 2: convert and print the information in the `feat_struct` variable for all the CMM HFs. In this case, the last column is filled with PASS string if the HFs is within the prescribed limits, or with FAILED if vice versa;
- Step 3: All the troubleshooting instructions of the FAILED tests are collected and reported in a separated table. The order of these lines actually follows the HFs *weights* evaluation already mentioned in Chapter 4. Once all the failed tests have been identified, they are sorted according to their *weight* in descending order.
- Step 4: The Extra-CMM HFs are reported in separates tables, typically followed by the relative explanatory plot.

In conclusion, this *.pdf* report represents a comprehensive document which give to the final user a complete overview of the performed tests during the HyDiag

test_num	hf_name	hf_value	test_result
----------	---------	----------	-------------

Table 5.5: HyDiag Report structure in *.csv* format

procedure. It is important to assert that this final product of the HyDiag software has been already designed to include the results coming from Data-Analytic studies and the Diagnostic Module Classifier. For example, Appendix B reports an interesting study, based on the analysis of the standard reports collection (referred as Historical Database), which has identified some sweet-spots of fixing within the suggested CMM limits. This analysis can already improve the simple binary distinction between PASS and FAILED tests, including a "grey-zone" where the CMM test is technically passed but not in an optimal range. This result based on "historical" data may contribute to improve the global results, together with the output from the HFs Classifier in the Diagnostic Module. About this latter topic, the final goal is to integrate the information of the CMM Troubleshooting section with the results coming from the Diagnostic Module Classifier. The idea is to create an *Advanced Troubleshooting* which merges the traditional repairing indications of each single test, with the classification results from the Diagnostic Module which considers the combination of more HFs. This improvement would transform the normal test-by-test repairing indications in a global indication of components to exchange or to repair, considering how the HFs can be interdependently connected. The final report would therefore represents the connection between HyDiag HFs software and Diagnostic Module, which can then interacts in a unique framework.

Chapter 6

Data Collection and Diagnostic Module

The flowchart in Fig. 4.2 shows the designed protocol during the HyDiag project to obtain an automatic and intelligent diagnostic process for PFCS. The Automatic Test Procedure has been described in its phases in both Chapter 4, regarding in its testing signals, and in Chapter 5, where the collected measurement channels analysis and HFs extraction methods have been described. In the paragraph 5.6, methods and functions to build the HyDiag Report at the end of the HFs extraction process have been described. It has also been mentioned how this report has been designed to include also information coming from the Diagnostic Module. The goal of this last block is to enrich the set of information in the test-report with indications about the health status of each sub-component. These evidences have to be extracted from information regarding both the Standard procedure, so called A-Priori Knowledge (as in Appendix B), and the HyDiag Testing procedure itself. Fig. 6.1 illustrates the way the A-Priori Knowledge results and HFs Classifier need to confluence in a third Reasoner, which needs to collect all the data and output the correct information in the HyDiag Report.

This chapter will in particular focus over the Classifier that is responsible to detect the health status of each sub-component from the extracted HyDiag HFs. This module core part includes a pattern recognition algorithm, based on a Supervised Machine Learning (SML) . This class of algorithms needs to be trained with a suitable dataset where both features and target labels have to be reported [58]. The first paragraph 6.1 describes the methods used to gather the HyDiag reports results in a unique Diagnostic Database, exploiting robust techniques of data-mining and collection. The second 6.2 paragraph describes the components failures and how they have been assigned per each tested unit. These labels are essential for training and validating the SML candidates. In paragraph 6.3 feature-selection methods are described for performance improvement of the tested SML algorithms. This step is particularly important to identify the HFs that are more representative and useful

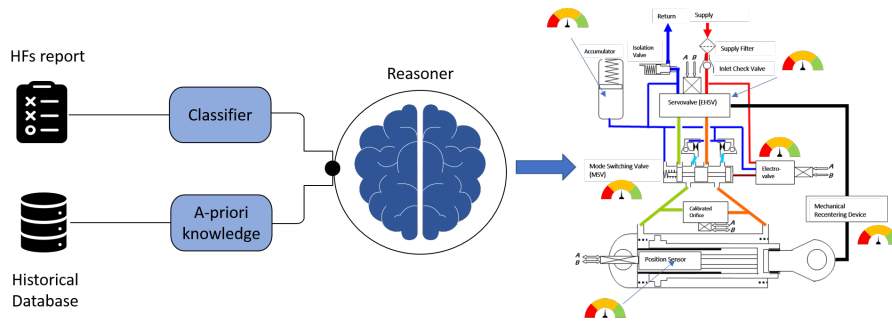


Figure 6.1: Diagnostic Module Parts

for a correct supervised failure identification. The last two paragraphs 6.4 and 6.5 are focused on the SML candidate algorithms: the first one describes each considered algorithm, while the last one shows a classification comparison between these algorithms.

6.1 HyDiag Reports Database Building

The Classifier module in Fig. 6.1 is responsible to extract a complete overview of the health status of each unit sub-component, elaborating as input the HF's extracted during the HyDiag testing procedure. Its core is based on SML algorithm that needs to be trained in order to assign the correct health-status label over the tested component. The most characterizing feature of such algorithm is indeed the necessity to be trained over a significant set of input data, where both data and labels needs to be well-structured and correctly preprocessed. Therefore, the HyDiag reports results need to be collected and structured in a unique standardized way. Several studies have highlighted the importance of data-collection, typically considered as a challenging bottleneck, especially if regarding historical and wrongly-collected data [59–62]. For this reason, a solid data collection framework has been built in Python with the main target to build a unique HyDiag Database. All the scripts and functions for this purpose are included in a unique folder `db_building` and they exploit many functions of the *Pandas* library to output a complete Dataframe which collects both information from the HF's reports and from the SLE exchanged components sheets. Fig. 6.2 graphically explains the developed process to build the final HyDiag Database.

The Python script `db_build_hydiag_reports.py` has been written in order to explore the results in the *HyDiag Report* folder (Fig. 5.71) and collect all the information in the `.csv` format files. The final result is the Reports Dataframe, exported in a local folder in `.csv` format as well. This final table includes basically all the metadata about tested units (e.g: PN, SN, FH, test date and hour) and, most importantly, the same columns of Table 5.5 from the HF's reports. This source

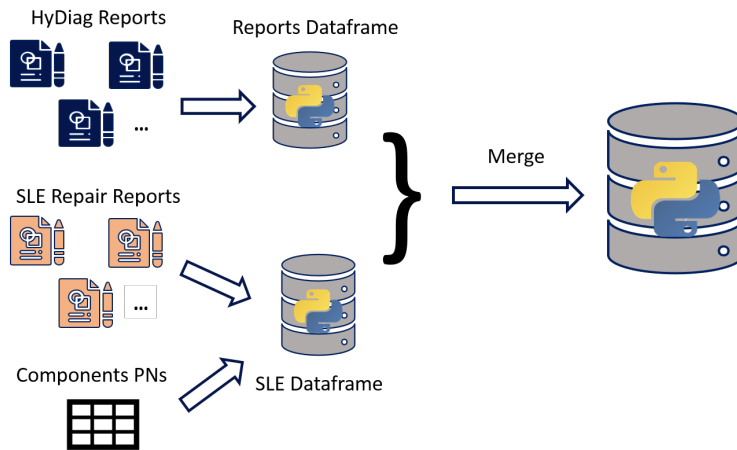


Figure 6.2: HyDiag Reports building process

of information constitutes the first half of the final HyDiag Database.

The second data source is represented by the SLE Dataframe, which collects all the information from the SLE repair reports of the tested units. These latter reports are completed once the units have been re-certified after a complete repair. In fact, the SLE reports includes all PN of the exchanged parts of the anomalous components and, additionally, some comments from the technicians about the observed problems during the repairing tasks. The biggest difficulty while reading these reports is to associate a specific component PN to the exchanged or repaired part. For example, each sealing is labelled with a specific PN and typically a specific kind of sealing can be used in different parts of the PFCS. For example, the Elevator PFC is composed by over 600 parts labelled with different PNs but, as reported in 2.3, twelve main functional sub-components can be identified. The script `db_buil_sle_sbss_merge.py` aims to assign per each component PN in the SLE Repair reports the correct sub-component. In this way, it is possible to retrieve information about the exchanged or repaired parts in the tested component. In order to fulfil this task, the script needs to load another fundamental file which is the Component PNs list. This file has been compiled manually starting from the technical exploded-view drawings of each PFCS and assigning per each component the correct sub-component where it belongs. However, only the constituting parts of each sub-component have been considered: for instance, only the sleeve and spring PNs have been assigned to the MSV sub-component, while the relative sealings have been assigned to a another group of components named "Mechanical spare parts". This group includes all the components that have been considered not necessary for the Advanced Diagnostic purposes, such as small sealings, plates, rosettes and screws. The EHSV represents a particular case in this classification since it is considered as a Linear Replaceable Unit (LRU): if a failure occurs, it

PN	SN	DATE	HOUR	EHSV	RD	MSV	ROD	ACCUMULATOR	...
ELVTR	XXX	01/01/20	15:00:00	1	0	1	0	0	
ELVTR	XXX	01/01/20	15:00:00	1	0	0	1	0	
ELVTR	XXX	01/01/20	15:00:00	0	0	0	1	0	
ELVTR	XXX	01/01/20	15:00:00	0	1	0	0	1	
...							

Figure 6.3: SLE Dataframe in pivot form

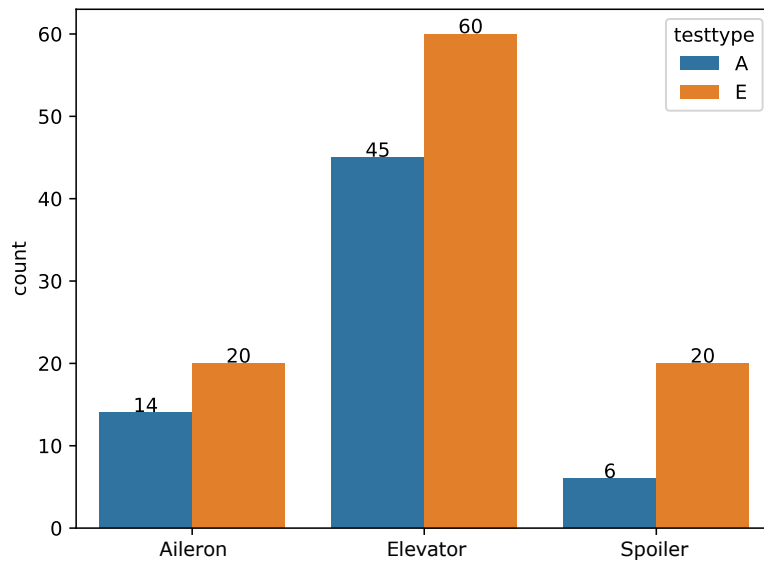
is directly exchanged with a new one without any further repairing or piece-parts replacement.

Once both the Reports and SLE Dataframes have been exported in *.csv* format, they need to be merged to obtain the final HyDiag Database. This process is developed in the `db_buil_rep_clst.py` script, exploiting the function `merge` of the Pandas Python Package. Due to the different structure of these two dataframes, the SLE one is converted as a pivot table grouping the table rows according test information, such as PN, SN, date and hour of the tested units. The pivot-table shape is illustrated in the scheme of Table 6.3:

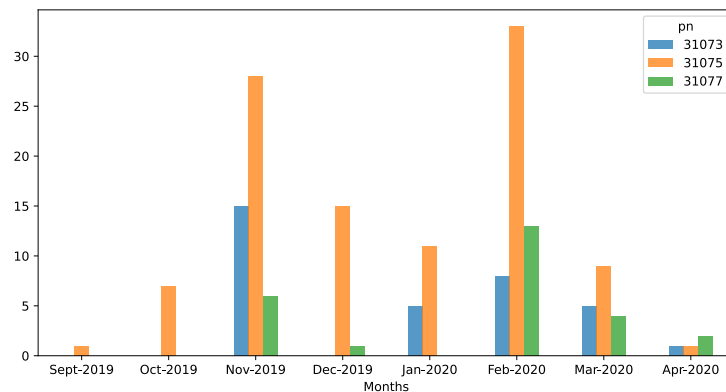
The components columns are filled with 1 or 0 according to the exchanged piece-parts for the reference component. For instance, if the MSV reacting spring has been replaced and its PN is reported in the SLE report, the corresponding MSV column value is 1. In this way, according to the values of these cells, it is possible to evaluate all the main sub-component that have been fixed or replaced before re-certifying the unit. In order to keep also the information about the single exchanged parts, the corresponding portion of SLE Dataframe per each unit is extracted and exported in *.csv* format in the corresponding report folder. This task is performed running the script `db_build_sle_export.py` in the *db_building* root folder.

Fig. 6.4a and 6.4b illustrates the distribution of the collected results in the HyDiag Database both in terms of test-type and date. Observing the plots, 165 HyDiag tests have been collected for both Entry (E) and Exit (A) tests from September 2019 to April 2020. Around 64% of the overall collected results regards the Elevator PFCS, which is technically the most complex unit in terms of installed components. The discrepancy between E and A tests can be explained considering all the units which have been tested during the deployment phase of the HyDiag procedure in the LHT shops. More than 50% of the overall Elevator tests have been collected during the month of February.

Due to the higher number of results and its structural complexity, the Elevator PFCS has been chosen as reference unit to tune the Classifier algorithm. For this reason, its portion of HyDiag Dataframe has been extracted with proper information about the extracted HFs and exchanged components.



(a) Collected results grouped per test-type



(b) Collected results grouped per date

Figure 6.4: HyDiag Database analysis

6.2 Failure labels assignment

The previous paragraph has described the methods to collect and organize in a unique standard all the results from the HyDiag testing procedure, in terms of extracted HFs per each unit component. As previously reported, the built HyDiag Database has been used as benchmark to test different supervised machine-learning algorithms in order to build the Classifier part within the complete Diagnostic

Module. This class of algorithm is designed to extract patterns and similarities in a given dataset, which includes both characterizing features and proper labels [13, 58, 63, 64].

In this thesis, a data-driven approach has been adopted to tune and test all the candidates algorithms for the Classifier module. This kind of approach differs from the model-based one since it is founded over test-bench measurements obtained during an experimental campaign. Both pros and cons of such approach have to be considered. On the one hand, the data-driven approach ensures a direct link with the tested systems, while the model-based one requires a long phase of models development, tuning and validation. On the other hand, the data-driven approach requires a considerable amount of data which is usually difficult to collect due to many experimental factors while, in the model-based approach, the simulations can potentially ensures any quantity of necessary data. Furthermore, especially for diagnostic purposes, the model-based approach is typically used since it is possible in the simulation environment to inject any failure with the desired level of degradation, especially if the model is a physical high-fidelity one [17, 18]. For a data-driven approach it would be necessary to collect these experimental results per each possible failure within the tested units. Looking at the structural complexity of the target Elevator unit, this task would require a long-lasting experimental campaign and a detailed reporting of the observed failures. The graphs in Fig. 6.4a and 6.4b illustrate how the experimental campaign in the LHT shops has not ensured the forecasted amount of data, due to the abrupt stop after February 2020. However, the collected measurements have been valued through a detailed failure-detection campaign over three fundamental units components: the EHSV and its connected RD and the MSV. For instance, for both RD and MSV it has been possible to completely dismount each component to check the health status of each piece-parts. Thanks to the support of the LHT workshop technicians, it has been possible to experimentally connect any anomalous pattern of the collected experimental results with an observed component mechanical degradation. An extensive description of the classified failures through the extracted HFs is described in the following paragraphs per each component.

6.2.1 Servovalve Failures

As extensively described in Chapter 2.3, the EHSV represents the interface component between control and actuation parts of the PFCS. Its complex structure and crucial function makes this component particularly interesting in Diagnostic field. Several publications in literature reports intensive studies over possible root of failures using a model-based approach [65–70]. Also within the HyDiag Project, an extensive research has been conducted over the effectiveness of the extracted HFs from the HyDiag testing protocol, exploiting a developed high-fidelity model of the Elevator PFCS. The structure and validation of this model are both reported

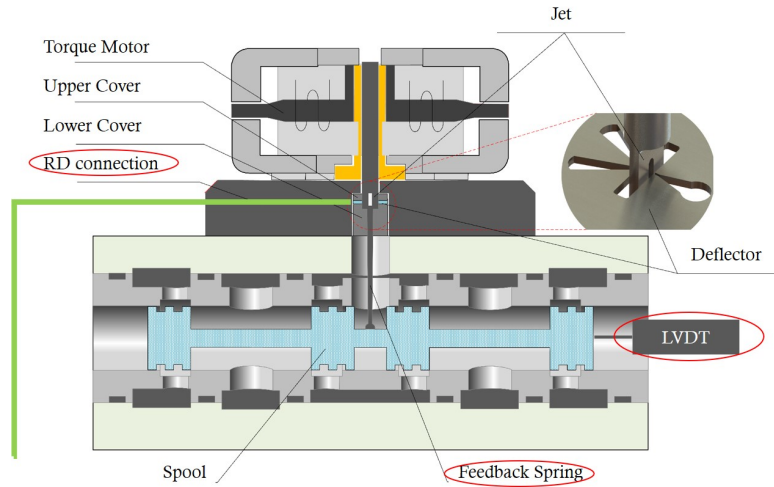


Figure 6.5: EHSV structure

in [17] and [18]. Both these works are focused on the failure identification of the EHSV, injecting different levels of degradations within this component: in [17] a preliminary binary classification analysis of the EHSV health status is reported, while in [18] a more complex feature analysis per class of detected failure have been conducted over simulation campaign.

The results in this latter work have laid the foundation for the correct failure labelling of the EHSV component from real experimental data. However, it has not been possible to correctly verify if the simulated outcomes or failure hypothesis, since it has not been possible to disassemble the EHSV and check the health status of each sub-component. Therefore, according to the standard procedure, the EHSV failure classification can be considered as binary (Serviceable vs Unserviceable), without a deeper analysis of the inner status of each sub-component shown in 6.5.

However, through the collected HFs in the HyDiag Reports, three recurrent anomalous behaviours have been observed that can point one of the two highlighted parts (red circles) in Fig. 6.5 as failure root:

1. **Failure 1:** the raw signal from the EHSV LVDT presents high level of noise compared with the nominal carrier frequency f_{supply} . As previously anticipated, this anomalous behaviour can be detected from the spectral density analysis of the raw channel U_{sek} in *Sequence* n.15 (Fig. 5.49). It has been experimentally observed from the collected results that typically the noise peaks are located at frequencies higher than the nominal one, leading a possible electrical insulation failure of the LVDT itself, or in its connection. However, it cannot be excluded to find anomalous noise peaks at on the left side of the typical PSD plot in Fig. 5.49, which would be a symptom of "low-frequency" spool oscillation. At current status, such anomaly has not been measured in

the data sample.

2. **Failure 2:** the second recurrent anomalous behaviour has been detected comparing the *Sequence* n.26 Hysteresis plot (Fig. 5.68). In particular, it has been observed that, comparing the base-line plots of re-certified units with first time tested ones, some of them were affected by a significant decrease of gain K_{sv} , measured as average slope of the current-spool position curve. This trend testifies a progressive insensitivity of the EHSV to supply current variation. This anomalous behaviour can be due to a stiff jet deflector which does not move freely as it should, due to an strong constant action of the RD or an unbalanced position of the deflector plate itself. Unfortunately, as previously mentioned, it has not been possible to dismount the EHSV stages to check the effectiveness of these hypothesis. Therefore, in the current work, this failure would be labelled but it may need an additional experimental verification.
3. **Failure 3:** the EHSV response to a current step-command presents an anomalous overshoot. During the HyDiag Test, several steps in open or closed loop are used to extract the dynamic response of the unit. In particular, in *Sequence* n.6 it is possible to extract both the EHSV LVDT position during the recentering strokes from totally extracted and retracted positions. These open loop signals are particularly important to extract information of both EHSV and RD health status. Regarding the first component, an under-damped EHSV spool position overshoot may be a symptom of excessive stiffness of the EHSV feedback spring or a progressive wearing of the contact surface between spool and spring.

6.2.2 Recentering-Device Failures

An extensive analysis of the RD health status has been reported in [49], where two recurrent degradations have been observed:

1. **Failure 1:** wearing in the contact surface between RD Bell-Crank and EHSV mechanical input. In normal operating conditions, a small free-play between these two contact surfaces needs to be guaranteed in order to allow the Bell-Crank to apply the correct load to recenter the EHSV jet-deflector position. However, a large degradation can bring a sensible reduction of this nominal load. In [49], a high-fidelity model of the RD is presented and a dedicated analysis about load reduction at increasing free-play is reported. Fig. 6.6 shows a schematic CAD drawings of the degraded contact surface and two pictures from exchanged components.
2. **Failure 2:** wearing in the contact surface of the main-body actuator housing

and the RD drive-shaft. This latter component rotation is mechanically connected with the main ram position (through the external leverage, visible in Fig. 2.8). This rotation is not supported by any bearing, therefore it causes a gradual degradation of the relative housing hole where it needs to be located. An excessive degradation may generate a misalignment of the drive-shaft influencing the complete RD behaviour. In the first figure in Fig. 6.7, the typical degraded area is highlighted in yellow, photographed in the other two figures.

3. **Failure 3:** the RD is not properly connected to the EHSV, therefore it does not recenter the main-rod in case of null supply current. This problem may be due to a loss of contact between two components of the RD mechanical chain. In this case, the RD needs to be completely disassembled in all its parts and properly re-connected to the EHSV mechanical inputs.

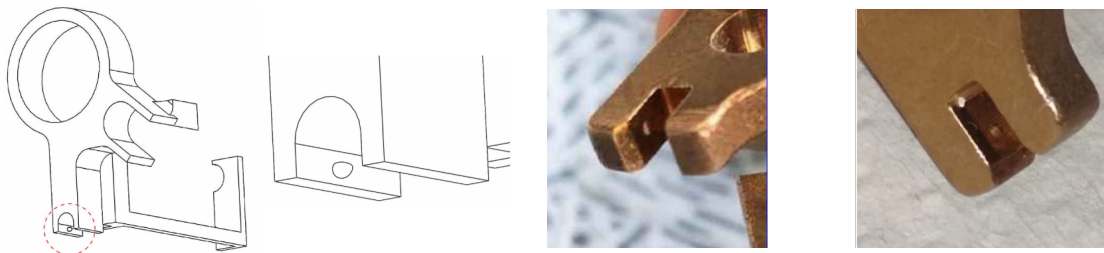


Figure 6.6: RD Bell-Crank degraded surface [49]



Figure 6.7: RD Control Shaft degraded surface [49]

These recurrent failures have been catalogued observing the wearing status of substituted components, coming directly from the LHT repairing shops. Beside these two recurrent failure, it is important to remember that the RD is the only Elevator component that can be adjusted and recalibrated during the test procedure. Nowadays, this procedure is manually performed by the LHT technicians. As previously described in Chapter 3, the HyDiag protocol purposes an innovative adjustment procedure with collaborative robots. More details about this topic are

reported in [35, 49, 50]. This particular characteristic make the RD failure classification more difficult than other components since it is necessary to discern between degraded RD, affected by the listed wearing conditions, and RD which simply need a recalibration. The chosen criteria to distinguish these cases and correct failure label assignment are reported in 6.2.4.

6.2.3 Mode Switching Valve Failures

A comprehensive study about the MSV health status has been described in [33]. Compared to the RD mechanical linkage, the MSV presents a simpler structure as shown in the left figure in Fig. 6.8. Two possible recurrent failures have been observed:

1. **Failure 1:** MSV spring degradation, which can lead to out-of-limit end stroke positions or slower dynamic response during the switching from active to damping mode (and vice-versa).
2. **Failure 2:** corroded damping-circuit, due to an excessive wear of MSV spool (for instance in the right photo in Fig. 6.8) or from the upper-stream damping circuit.

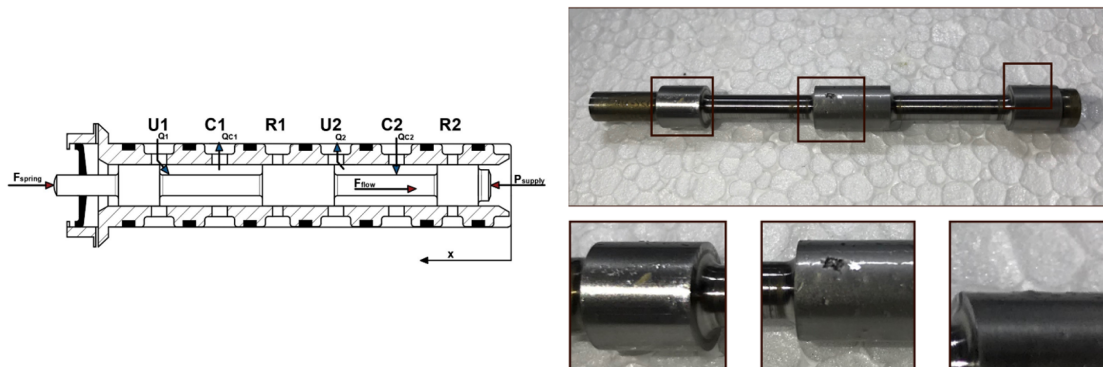


Figure 6.8: RD Bell-Crank degraded surface [33]

6.2.4 Failure Labels assignment criteria through HFs

The described failures in paragraphs 6.2.1, 6.2.2 and 6.2.3 need to be correctly assigned to the observed measurements in the HyDiag Database. This task represents one of the essential blocks of the workflow implemented in the dedicated Python script `adv_diagnose_classification`. This code represents the current development status of the Classifier module in Fig. 6.1, starting from

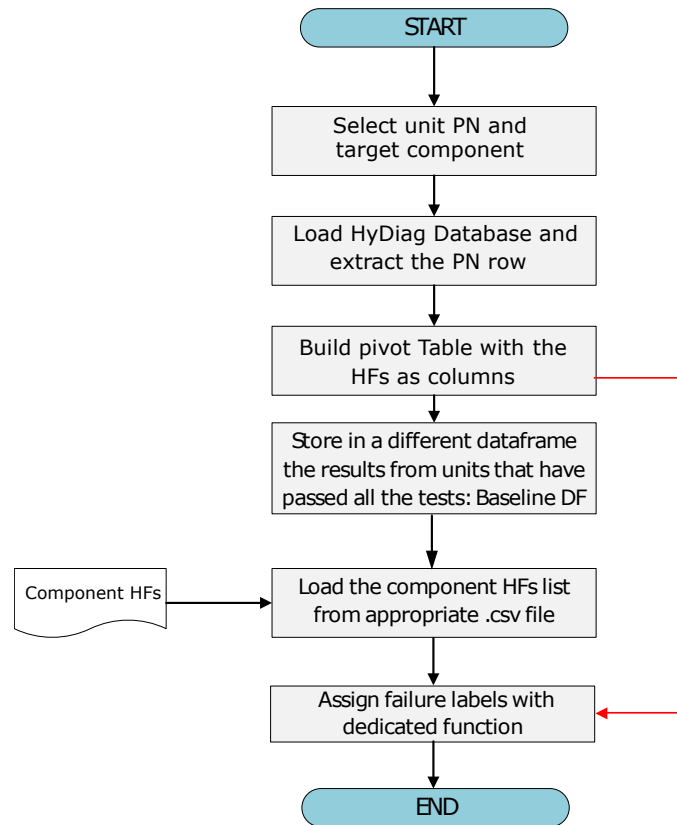


Figure 6.9: Failure labels assignment flowchart

the Database preprocessing to the tuning of four ML candidates. In this paragraph, the data processing and failure assignment criteria through the corresponding HFs are described. The flowchart in Fig. 6.9 explains the steps implemented in `adv_diagnose_classification` for automatic failure labels:

It is important to notice that the flowchart steps have been designed with the idea to extend this procedure per all the project use-cases and their specific sub-components, as soon as a considerable amount of units are tested with the HyDiag procedure. In order to keep the code scalable and re-usable for the other project use-cases, the first flowchart step allows the user to choose the unit PN and target component. For instance, as already mentioned, the analysis is focused on the Elevator results and in particular on three main sub-components, which are considered as sensitive root of failure: EHSV, RD and MSV.

Once unit and target component have been chosen, the HyDiag Database is loaded and filtered according to the unit PN. Afterwards, its shape needs to be changed according to Fig. 6.10. According to the chosen PN, the unit table rows are extracted and a new table is built, setting the collected HFs as columns. Each row index is defined creating a unique ident-code from the test information related

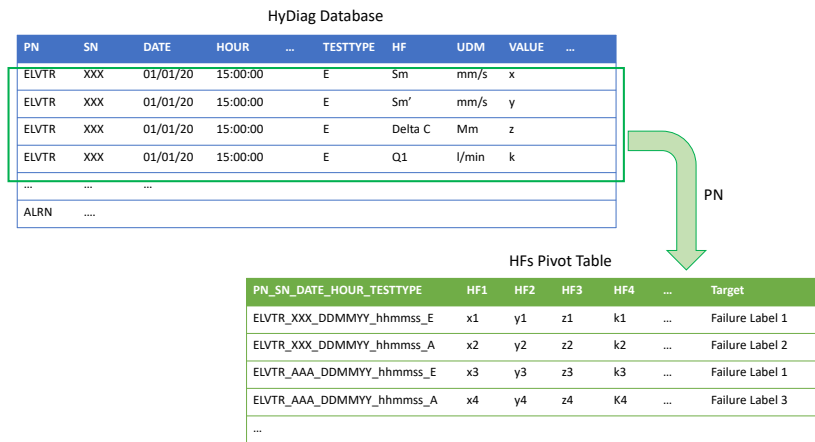


Figure 6.10: HyDiag Database reshape

columns of the HyDiag Database. In this way, the final `df_hfs` dataframe contains as much lines as the number of tests with the HyDiag procedure. It is important to notice that both Entry "E" and Exit "A" tests are taken into account, therefore it the final pivot table would include the results from the consecutive tests before and after repairing tasks in the maintenance-shops (as reported in Fig. 4.18). In this way, it is possible to collect results from both un-serviceable and recertified units that are essential to identify the nominal baseline behaviour. In fact, the re-certified units should have passed all the standard tests and therefore can be used to identify the nominal HFs patterns that facilitate any failure identification. In the `df_hfs` rows related to re-certified units are stored in a separate dataframe and they are afterwards used to identify acceptable ranges for the additional HFs extracted from the HyDiag Sequence. For instance, *Sequence* n.13 has been created ad-hoc in order to extract the linear nominal RD behaviour. All the HFs that can be extracted from this signal are not reported in any maintenance manual, but they have been properly designed in this HyDiag Sequence. Therefore there is not any indication about "acceptable" ranges, unless these limits can be extracted by the experimental results, exploring the data distribution of these HFs in the recertified-units dataframe. In the flowchart in Fig. 6.9, the red arrow shows how the fourth step output is then useful to assign the failure-labels in the last one.

Once the complete pivot table has been defined and the re-certified units identified, the last step consists in the correct failure label assignment. This specific task has been implemented in the dedicated `class_definition` function. This particular script assigns the relative failure hypothesis to the proper pivot-table row, creating an additional target column for the chosen component. For instance, the corresponding EHSV Failures in 6.2.1 are labelled according to Table 6.1:

Table 6.1: EHSV Failures Labels

	Class	Failure Label
Baseline	Serviceable EHSV	EHSV OK
Failure 1	EHSV LVDT too noisy	EHSV LVDT NOISY
Failure 2	EHSV insensitivity to supply current	EHSV GAIN TOO LOW
Failure 3	EHSV excessive step-overshoot	EHSV TOO STIFF

Beside these three observed Failures, for a correct classification, it is necessary to add specific criteria for serviceable units identification. This task needs to be executed for all the investigated components, in order to tune a Classifier which is able to distinguish failure classes from nominal serviceable behaviour.

The function `class_definition` contains all the metrics to compare the EHSV HFs with the relative baseline values and assign automatically the correct failure-label in the target column in the HFs pivot table. Compared to the traditional troubleshooting instructions, which determines whether a component needs to be replaced or not, this automatic failure-labels assignment provides the following benefits:

1. Within the `class_definition` function, both traditional and ad-hoc HFs are contemporary considered in order to assign the correct label. During the traditional test procedure, each HF is characterized by its own troubleshooting instruction without considering any correlation with other extracted results. An improvement the `class_definition` function provides is the possibility to translate with logical conditions the failure hypothesis for the relative component, considering more HFs contemporary. The extra-CMM HFs bring a substantial help in the label assignment process, since they have been designed specifically for the target component. For instance, the example code below shows how two traditional HFs (U9 and U10) and two extra-CMM ones (`snr_usek` and `f_peak_usek_null`) are contemporary taken into account to assign the proper failure label (EHSV LVDT NOISY).

Listing 6.1: EHSV LVDT NOISY label assignment

```
df.loc[((abs(df['U9']) < U_min) | (abs(df['U9']) > U_max)) &
        ((abs(df['U10']) < U_min) | (abs(df['U10']) > U_max)) &
        (abs(df['db_sv']) <= db_sv_max) &
        (
            ((df['snr_usek'] <= snr_usek_min) &
             (df['f_peak_usek_null'] > f_peak_ref)) |
            (df['snr_usek'] > snr_usek_max)
        ), comp] = 'EHSV LVDT NOISY'
```

These criteria have been defined per each failure hypothesis of each target component, and they represents the computational translation of the observed anomalous behaviour in paragraphs 6.2.1, 6.2.2 and 6.2.3, through the extracted HFs.

2. Observing the example code above, it can be noticed that both traditional and extra-CMM HFs are compared with reference values. Similarly to the standard protocol, it is necessary to define acceptable ranges that allows to discern between healthy and faulty behaviour. Considering the traditional HFs, such as *U9* and *U10* in the example (see Fig. 5.46 for more details about these HFs), these reference values might be both the traditional CMM limits but it is also possible to include the "sweet-spots" ranges identified from the Historical Database. More details about this analysis are reported in Appendix B. A different approach needs to be taken for the extra-CMM HFs. In the above example, the reference values for `snr_usek` and `f_peak_usek_null` are evaluated from the relative HFs distribution from the recertified units. Taking as reference these units, a baseline behaviour can be extracted especially from the units which are labelled as serviceable. This link is represented in the flowchart in Fig. 6.9 by the red arrow.
3. The assignment criteria may always be easily update per each target component. The numbers of failure labels that have been coded in `class_definition` strictly depends on the number of "observations" in the current set of data. Due to the relatively low number of collected measurements, it has been possible to observe the anomalous behaviours described in paragraphs 6.2.1, 6.2.2 and 6.2.3. However, increasing the number of collected data, it would be possible to define new failure-labels and code them with conditional statements as before. Furthermore, an higher number of serviceable units improves also the HFs limits definition.

Tables 6.2 and 6.3 reports the equivalent failure-labels assigned in case of RD or MSV as target component. As mentioned in 6.2.2, the RD is the only Elevator component that can be recalibrated during the test procedure. Therefore, the units with RD to be adjusted have not been considered in the following failure-labels assignment, since they cannot be considered as units with a degraded RD.

The pie plots in Fig. 6.11a, 6.11b and 6.11c show the population percentages of each class per target component. These charts highlights how actually, considering the failure labels as mutually exclusive classes, the classification algorithms would deal with an unbalanced problem. This observation needs to be taken into account especially when these algorithms needs to be tuned and compared according to their performance metrics, which are usually highly affected by the number of points per each class. More details about these analysis are reported in paragraph 6.5.2.

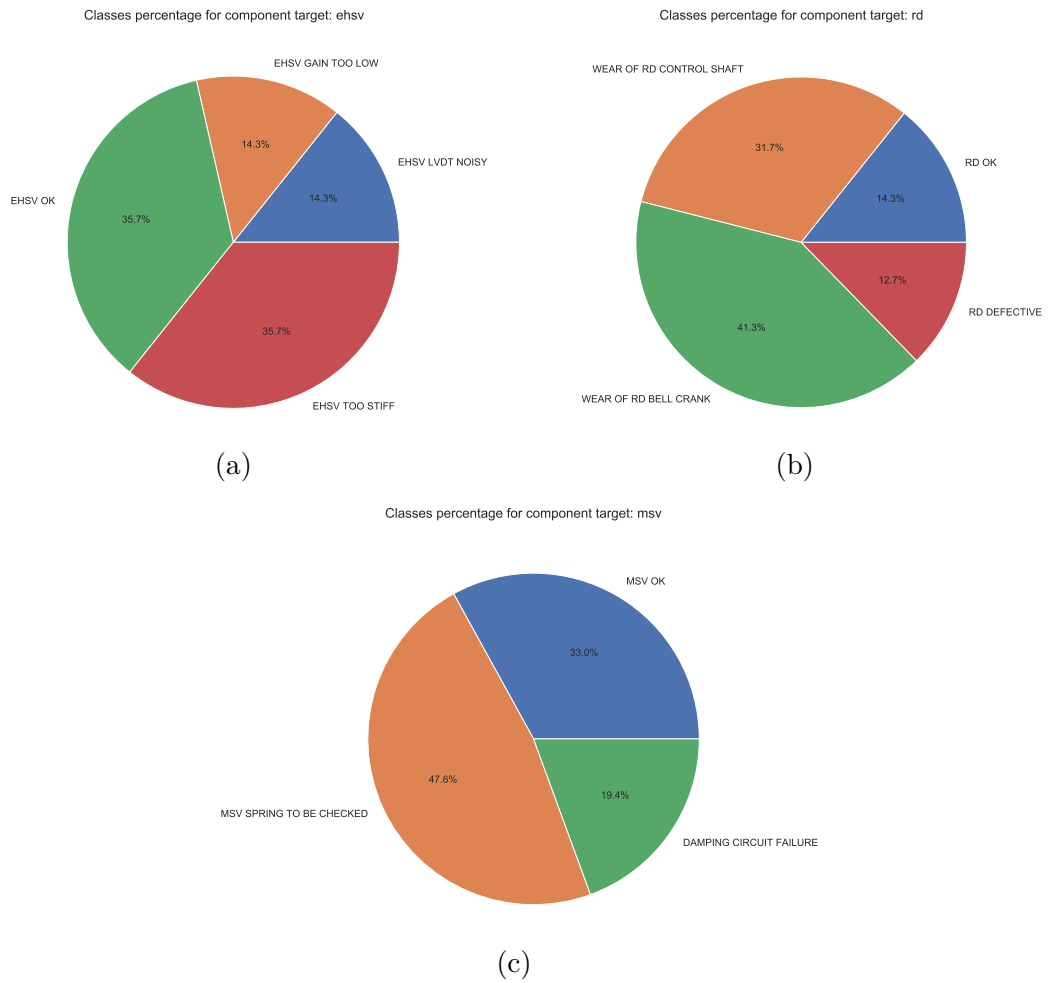


Figure 6.11: Target Component Classes populations

Table 6.2: RD Failures Labels

	Class	Failure Label
Baseline	Serviceable RD	RD OK
Failure 1	Bell Crank surface wearing	WEAR OF RD BELL CRANK
Failure 2	Control shaft surface wearing	WEAR OF RD CONTROL SHAFT
Failure 3	Disconnected RD	RD DEFECTIVE

Table 6.3: MSV Failures Labels

	Class	Failure Label
Baseline	Serviceable MSV	MSV OK
Failure 1	MSV Spring to be checked	MSV TO BE CHECKED
Failure 2	Problem in the up-stream circuit	DAMPING CIRCUIT FAILURE

6.3 Features Selection

As previously mentioned in Chapters 4 and 5, the *Sequences* have been designed in order to test all the unit components that are possible source of failures. Per each of them, both traditional and additional ad-hoc HFs have been defined and needs to be isolated from the whole HFs basket. However, some of them can be used to identify failures of more than one component and therefore need to be considered accordingly.

The pivot table columns in Fig. 6.10 includes all the defined HFs that can be used for failure identification of the target component. This dataset represents the starting point for both failure-labels assignment and evaluating the ML classifier performance for failure identification. However, it is possible that some of the considered HFs are more representative for classification tasks than others. In Data-Science application, dimensionality reduction is a key technique that can give several benefits for Classification purposes. As reported in [71], feature selection techniques facilitate data visualization and understanding, but also avoid the "curse of dimensionality" problem [72–74]. There are typically three families of feature-selection techniques [75]:

1. **Filter Methods:** they adopt variable ranking techniques for proper selection. These methods do not depend from the used Machine-Learning algorithm, but they apply statistical evaluations over the feature relevance to discern between different classes. From the computational point of view, these methods are fast and less prone to classification over-fitting.
2. **Wrapper Methods:** in these techniques the feature selection is based on a

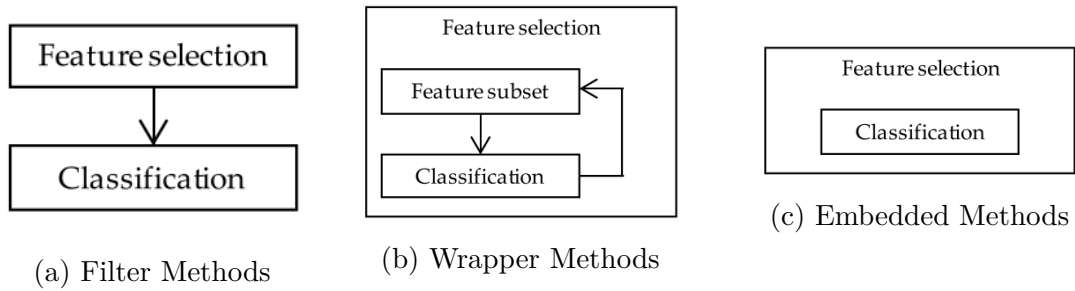


Figure 6.12: Feature Selection Methods [79]

specific classification algorithm that needs to be fit on the dataset. The strategy is to use a subset of variables that optimize the classification performance. These methods can be divided in two families:

- Sequential Selectors: these algorithms add or remove features (forward or backward selection) from the complete set of variables, until the classifier performance is maximized with the minimum subset of optimal features.
- Heuristical Selectors: techniques like Genetic Algorithms create variables subsets "by searching around in a search-space or by generating solutions to the optimization problem" [75].

3. **Embedded Methods:** in this case, the feature selection is directly implemented in the classifier training process. These techniques take into account the interaction between feature selection and classifier (like the wrapper ones) using regularization strategies, which make these algorithms less prone to overfitting. In particular, for linear classifier, the three main types of regularization are LASSO (L1 regression), RIDGE (L2 regression) and ELASTIC NETS [76–78]

Choosing the correct feature selection algorithm is an important step that needs to be taken evaluating all the pro and cons of each method-family and more specifically each algorithm in the chosen group of methods. Table 6.4 reports the strengths and weaknesses of each group, as reported in [80].

In addition to the pros and cons listed in Table 6.4, the choice of feature-selection algorithm should depend on the quality of available data and the selected classifier to be tested. Currently, the core SML algorithm for the Classifier module has not been chosen yet: in fact, in paragraph 6.4 four possible candidates are presented and compared. Therefore, a Filter Method has been preferred to both Wrapper and Embedded techniques. In this way, it is possible to have a first insight about the collected HFs and how these can statistically influence the assigned failure-label. The statistical metric to be used for filter-based selection needs to be chosen according to the kind of output the SML has to distinguish. The failure-labels

Table 6.4: Feature Selection Methods Comparison

Class	Methods	Pro	Cons
Filter Methods	F-Score, Chi square, Mutual Info	These methods are typically simple and low cost. Particularly suitable for first investigation and if the classifier has not been chosen yet.	Typically based on assumptions about the probability distribution of each feature, which are not always verified.
Wrapper Methods	Forward and Backward Selection	These methods are tailored on a specific classifier and they maximize the ML algorithm performance.	Usually they are time-consuming and not particularly suitable in case the classifier need to be modified.
Embedded Methods	L1 or L2 regularization	These techniques avoid overfitting by adding a penalty to the objective function per each additional feature.	The regulator choice depends on the kind of classifier (linear or not). Furthermore, since the penalties modify the classifier optimization function, they need to be added during the implementation.

reported in Tables 6.1, 6.2 and 6.3 are categorical variables, due to the limited and fixed number of possible values. Following the guidelines in [81], considering the numerical HFs inputs and categorical labels output, the ANOVA F-value metric can help in the selection of the best subset of optimal HFs. As reported in [82], ANOVA (Analysis of Variance) can be used to evaluate how good the features are for class separability. It is a collection of statistical models and their associated procedures used to analyse the difference among group means. The ANOVA method compares the null hypothesis H_0 of "no difference among the classes means" [83] against the opposite one H_1 where at least one class mean differs.

$$H_0 : \mu_{class1} = \mu_{class2} = \dots = \mu_{classM} \tag{6.1}$$

To verify one of these two hypothesis, it is necessary to establish a statistic metrics which allows to compare "the class-to-class variance over the within-class variance" [84]. This metric corresponds to the F-ratio. For feature selection, the F-ratio (or Fisher Discriminant Ratio) needs to determine the subset of optimal features that on the one hand maximize the distances between data points in different classes and, on the other, minimize the distance between the same class points [85].

Fig. 6.13 shows an example to define the Fisher Discriminant Ratio for a binary classification. The red and blue classes are plotted according to two reference

features on the horizontal and vertical axes. From the distribution plots, it can be asserted that feature over the horizontal dimension is a better separator than the vertical one, since the classes distributions do not overlap. Mathematically, this observation is translated considering that:

- The distance between the mean values over for the horizontal feature is bigger than the vertical one;
- The variance of each single class over the horizontal feature is lower than the vertical one.

In order to calculate the F-score per each j -th feature and evaluate the best subset of indicators, it is necessary to evaluate:

- μ_k^j and σ_k^j : respectively the mean and standard-deviation of the k -th class of the j -th feature;
- μ^j and σ^j : mean and standard-deviation of the whole j -th feature, considering all the available data points.

Considering M possible classes, each of them having n_k number of points, the j -th feature F-score x^j is calculated as:

$$F(x^j) = \frac{\sum_{k=1}^M n_k (\mu_k^j - \mu^j)^2}{(\sigma^j)^2} \quad (6.2)$$

where $(\sigma^j)^2 = \sum_{k=1}^M n_k (\sigma_k^j)^2$. Significant values of the F-ratio indicates that, for the given feature x_j , the class means are heterogeneous, therefore it is statistically relevant since it excludes the null hypothesis H_0 in Eq. 6.1 and validates the opposite H_1 one. Establishing a reference value of F-score depends on both the number of classes and samples. In particular, if the null hypothesis H_0 of the ANOVA method is fulfilled, then the F-ratio should follow an F-distribution. In this case of study, the number of samples and classes are fixed, therefore each feature can be labelled with its own F-ratio. However, increasing the number of measurements and the classes (which means the possible failure-labels), the F-ratio would necessary change: the distribution of all the possible features F-ratio depending on number of classes and points follows the so called Fisher-Snedecor distribution (or F-distribution). In particular, the Fisher-Snedecor distribution describes the how F-score depends on the degree-of-freedom of the numerator between-class variance df_b and on the same denominator within-class variance degree-of-freedom df_w , as shown in Fig. 6.14. Since it is assumed the validity of H_0 , it is possible to check the F-ratio position in the traced F-distribution to evaluate its probability p – value: in this way, it is possible to evaluate how probable the $F(x_j)$ per each feature under the assumption of null hypothesis. If this probability is low enough, according to

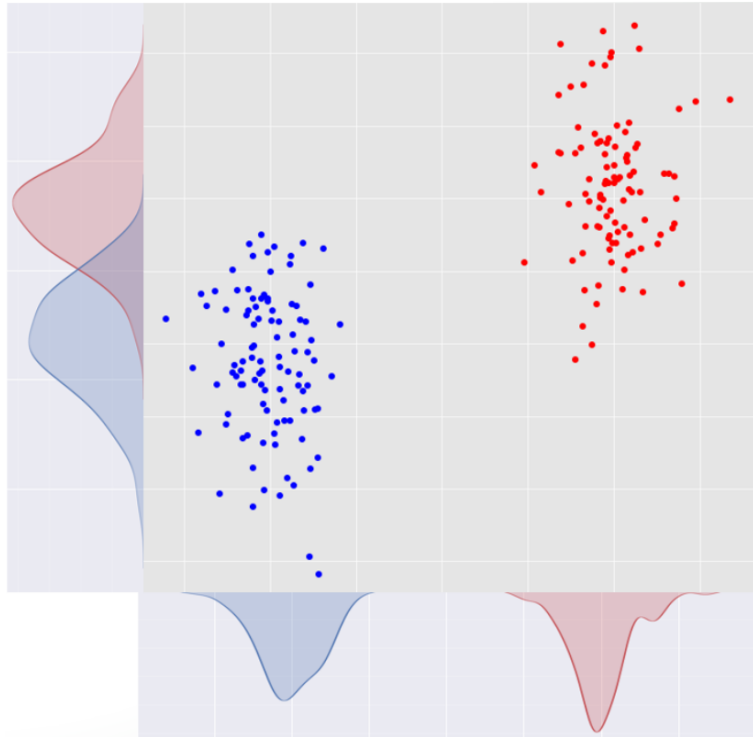


Figure 6.13: Fisher Linear Discriminant for class separation

a certain threshold, it means that the $F(x_j)$ is strong enough to discard the null hypothesis H_0 and validate the opposite one H_1 , therefore being statistically relevant. In this work, each feature ratio $F(x_j)$ would be significant if the p -value associated with it is less than 5%.

To sum up, the ANOVA F-score method is used as filter-feature selection method, ranking each collected HFs in the `df_hfs` according to associated probability per each ratio $F(x_j)$ to confute the null hypothesis H_0 of equal mean between the classes. In this work, the classes are defined through the failure-labels that can be assigned per each target component, and each HF column of the pivot table represents a peculiar feature to be tested with the ANOVA method. This algorithm results particularly robust even if the hypothesis of normally distributed data is not fulfilled [86], and this can be considered an advantage for experimental applications. However, this filter method considers each feature without taking into account any correlation between them: as univariate method, it is possible to pick a subset of optimal features with some redundant one. Therefore, the complete process for feature selection can be divided in three steps:

1. Eliminate redundant features that are linearly dependent;
2. Normalize the HFs columns in the `df_hfs` table;

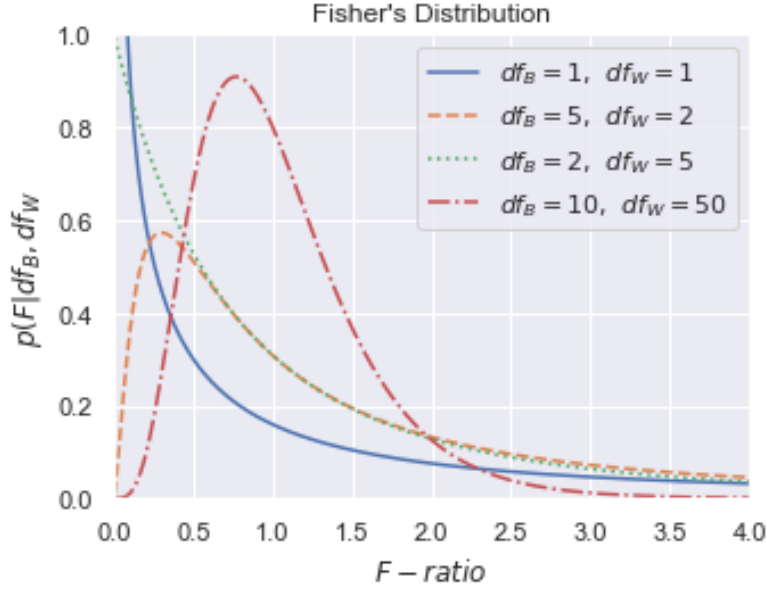


Figure 6.14: F-Distributions shapes depending on its degrees of freedom

3. Execute ANOVA F-score for HFs selection

The Multicollinearity is a common issue to be solved in case of multi-dimensional dataset, where two or more variables are highly correlated [87–89]. Having couples or groups of linear dependent features may create misleading results from the trained classifier, especially in case of linear-based algorithm. Multicollinearity exists if couples of variables are highly correlated, but even between one feature and a linear combinations of others. In general, it is possible to assert that correlation is a particular case of multicollinearity, but the vice-versa is not always true. However, for a preliminary data exploration, it has been checked the linear correlation between couples of HFs per each target component. This evaluation has been conducted calculating the Pearson ratio per each single HF against all the others:

$$\rho = \frac{\text{cov}(x_i, x_j)}{\sigma_{x_i} \sigma_{x_j}} \quad (6.3)$$

where the numerator includes the covariance $\text{cov}(x_i, x_j)$ between two generic features x_i and x_j , and the denominator is the product of their standard deviations σ_{x_i} and σ_{x_j} . This coefficient is widely used to evaluate the linear dependency between samples, normalizing the covariance value in order to have $-1 < \rho < 1$. The closest this values it to 1, the more the features have a full positive correlation and, vice-versa, the features are negatively correlated in ρ is close to -1 .

The plots in Fig. 6.15a, 6.15b and 6.15c shows the correlation Heatmap of the characterizing HFs per each target component. The HFs have been reported on

both vertical and horizontal dimensions and coloured according the colour-scale on the right side of each diagram. Each square in the resulting diagonal matrix is coloured from blue to red according to the value of ρ : the blue squared identify the negatively correlated HFs, while the red ones are the positive correlated ones. The idea is to consider all the couples of HFs that presents a Pearson score higher than the threshold $|\rho_{ref}| = 0.9$: per each of these couples, one HFs is arbitrary excluded in order to exclude the multicollinearity between dependant features. For instance, considering the RD Correlation Heatmap in Fig. 6.15b, it is possible to observe a strong correlation between the recentering positions $S2$ and $S2'$. In fact, it has been observed from the experimental data how these features are always contemporary within or outside the acceptable boundaries. Therefore, considering this HFs couple does not provide any additional information than taking into account just one of them. For this reason, one of them can be arbitrary excluded from before applying any feature-selection method for optimal HFs subset isolation.

Once the highly-correlated features have been excluded, it is necessary to scale the remaining HFs in order to make them comparable. The HFs columns in the `df_hfs` table are characterized by different unit of measure, according to the signal from which they are extracted. Therefore, feature scaling is an essential step to make the HFs readable from any feature-selection and further classification algorithm. Three main scaling methods are reported in literature [90]:

1. Min-Max Normalization: the scaling references are the minimum and maximum column values, as reported in Eq. 6.4:

$$HF_{norm} = \frac{HF - \min(HF)}{\max(HF) - \min(HF)} \quad (6.4)$$

2. Mean Normalization: compared to Eq. 6.4, the numerator reference value is the mean one:

$$HF_{norm} = \frac{HF - \overline{HF}}{\max(HF) - \min(HF)} \quad (6.5)$$

3. Z-Score Normalization: this method allows to scale all the HFs in order to have zero-mean and unit variance. In this case, the reference scaling values are the distribution mean and variance:

$$HF_{norm} = \frac{HF - \overline{HF}}{\sigma_{HF}} \quad (6.6)$$

Each column of the `df_hfs` table has been scaled using the Z-score Normalization, using the `StandardScaler` function of the Scikit-Learn python package. As

6.3 – Features Selection

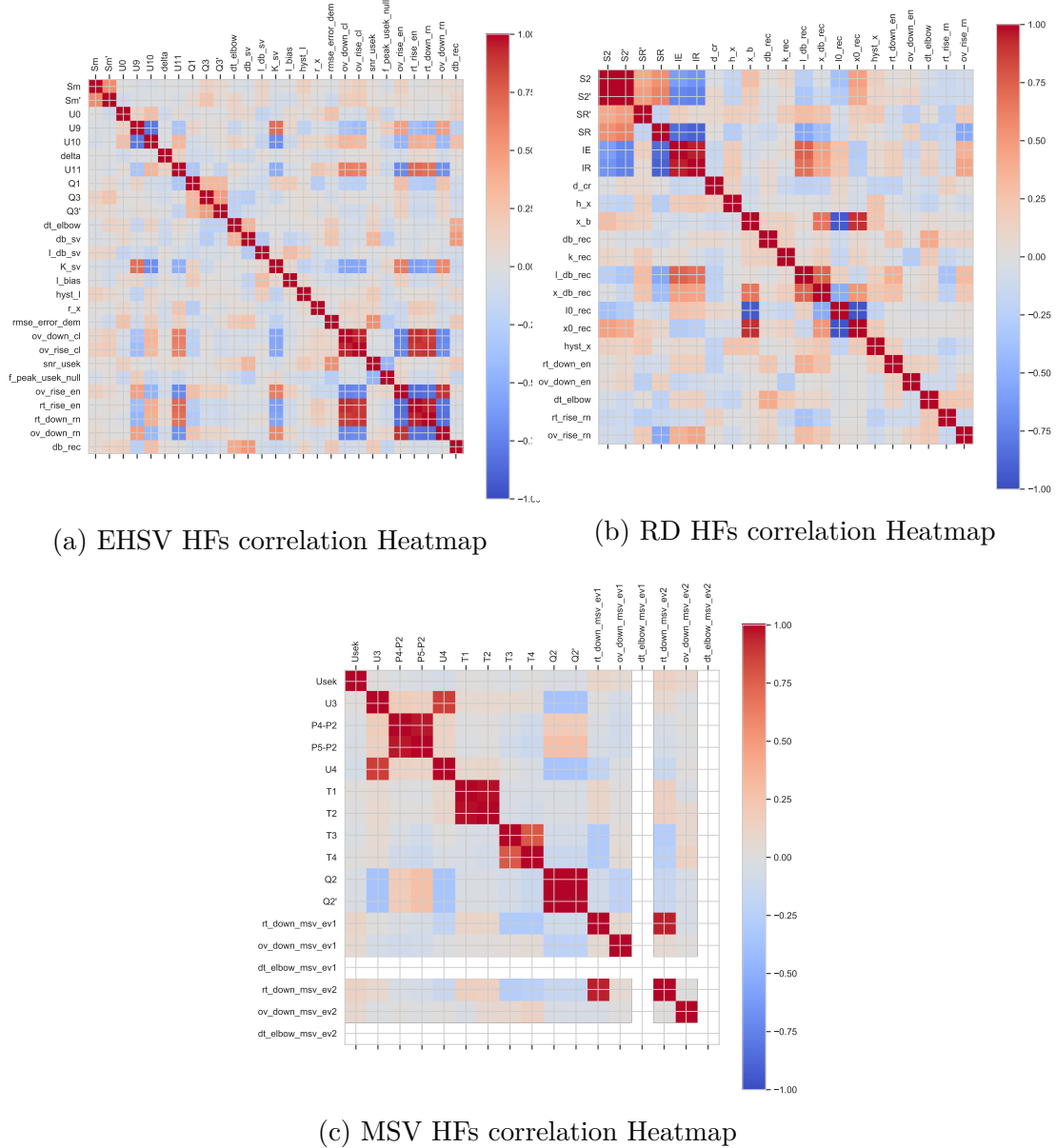


Figure 6.15: HFS Correlation Heatmap

reported in [91], this module a "wide range of state-of-art machine learning algorithms for medium-scale supervised and unsupervised problems". Due to its ease of use, large documentation and applications, it has been adopted in this work as the main source of functions and algorithms to solve data-exploration issues and compare different possible classifiers.

Once all the not-redundant remaining HF's have been normalized, the ANOVA F-score method for feature selection is launched to evaluate the most promising

HFs for the assigned classes, defined by the target component failure labels. As previously reported, the ratio in Eq. 6.2 needs to be evaluated per each HF and evaluate the corresponding p -value of the relative F-distribution, which depends on the number of classes and data-points. These tasks are implemented in the `SelectKBest(f_classif)` function of the scikit-learn python package. The code in Listing 6.2 is extracted from the main `adv_diagnose_classification`, and it shows the steps that are used to extract the `best_hfs` set of feature that present a p -value lower than 5%.

Listing 6.2: ANOVA F-score feature selection

```
# Define ANOVA F-score feature-selection method and fit it over
# complete normalized dataset and target failure-labels column
selector = SelectKBest(f_classif)
selector.fit(df_hfs_std[hfs_columns], labels_col)

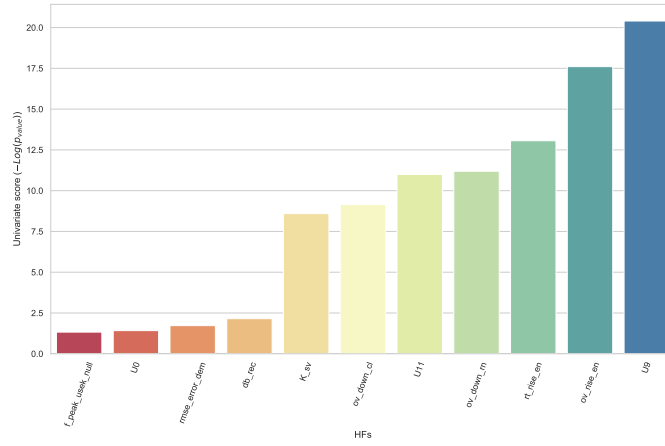
# Extract p-value associated to each F-ratio and collect them in a
# dataframe
scores = selector.pvalues_
hfs_score = pd.DataFrame(list(zip(list_hfs_uncorr, scores.tolist())),
    columns=['HFs', 'Score'])
hfs_score = hfs_score.sort_values(by=['Score'], ascending=False)

# Select features according to a set p-value treshold
THRSLD = 0.05
best_hfs = hfs_score[hfs_score['Score'] <= THRSLD]
```

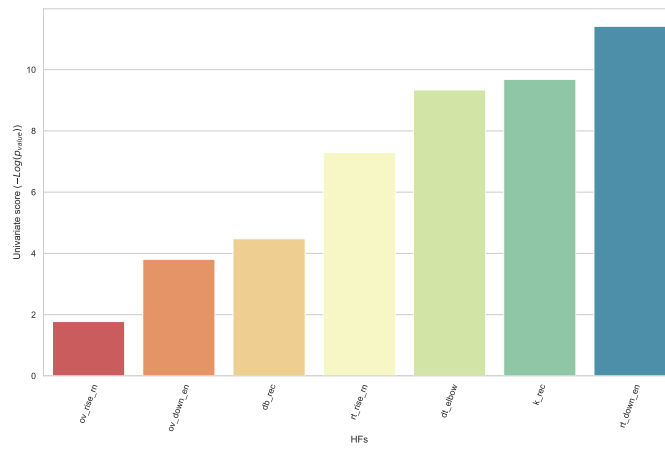
The column `labels_col` collects all the failure-labels in Tables 6.1, 6.2 and 6.3 for the selected target component. Fig. 6.16a, 6.16b and 6.16c illustrates the most promising subset of HFs per each component classification, according to the ranked logarithmic p -values extracted from the ANOVA F-score selector. These HFs have been summarized in Table 6.5. Looking at these plots and to the summary table, it is possible to notice how, per each target component, the most promising features identified by ANOVA F-score method are actually a mix between both traditional and extra-CMM HFs. This first observation suggests that, in order to have a better failure-class separation, it is important to consider more features contemporary and how the extra-CMM ones are significant. Furthermore, some of them are useful in more than one classification problem. For instance, the RD characteristic deadband amplitude `db_rec` can be exploited for both failure classification of EHSV and RD, highlighting the strict dependency between these two components.

Once the most promising HFs have been extracted, the respective columns of the normalized HFs dataframe needs to be extracted, according to the target component. At this point, this set of necessary HFs and correct failure labels represents the input required by all the SML candidate algorithms to be tested.

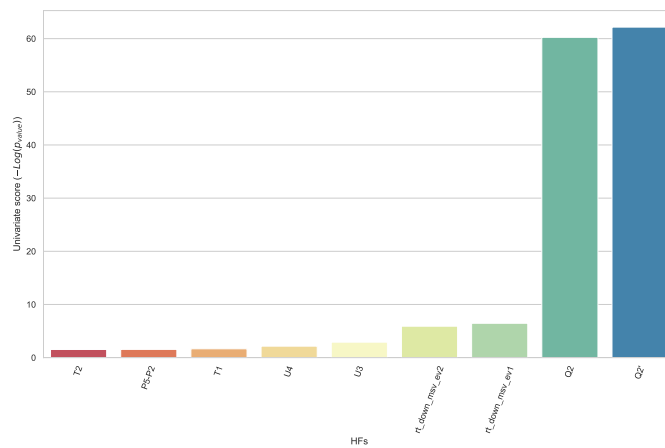
6.3 – Features Selection



(a) EHSV HF features ranked according to $-\log_{10}(p)$



(b) RD HF features ranked according to $-\log_{10}(p)$



(c) MSV HF features ranked according to $-\log_{10}(p)$

Figure 6.16: Target component HFS ranked according to $-\log_{10}(p)$

Table 6.5: Selected HFs from ANOVA F-score per each component classification

Component	Selected HFs
EHSV	$f_peak_usek_null, U0, rmse_error_dem, db_rec, K_sv, ov_down_cl, U11, ov_down_rn, rt_rise_en, ov_rise_rn, U9$
RD	$ov_rise_rn, ov_down_en, db_rec, rt_rise_rn, dt_elbow, k_rec, U11, rt_down_en$
MSV	$T2, P5 - P2, T1, U4, U3, rt_down_msv_ev2, rt_down_msv_ev1, Q2, Q2'$

6.4 Classification Algorithms

Supervised Machine Learning (SML) algorithms have been widely used in several engineering fields, with both data-driven and model-based approaches for diagnostic and prognosis purposes. For instance, in [92] four supervised learning techniques have been analysed over the same problem in order to give insights over the algorithm choice. In the mechanical world, these techniques have been widely exploited to identify failures or predict the remaining useful life of critical devices, such as bearings [93–96], electric motors [97–100] and electro-hydraulic servo-actuators. In particular, for this latter system, several studies have been conducted to implement effective failure identification algorithm, based both on model-based techniques [8, 9, 67] and data-driven approach based on experimental data [29, 101, 102]. Within the HyDiag project framework, a preliminary comparison of possible classification algorithm has been conducted in [32], based on simulated results via a High-Fidelity model. Four of the algorithms in [32] have been tested also using the experimental results in the HyDiag Database, in order to establish the accuracy of each algorithm according to the assigned failure-labels. Mathematically, the following notation can be used to define a SML algorithm for classification.

Considering a dataset with m samples $\{\mathbf{x}^{(i)}, \mathbf{t}^{(i)}\}_{i=1, \dots, m}$ where:

- $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)})$ is the vector of the corresponding n features that are collected per each $x^{(i)}$ sample;
- $\mathbf{t}^{(i)} = (t_1^{(i)}, t_2^{(i)}, \dots, t_K^{(i)})$ is the vector containing the possible K classes to define

In this work, the vector $\mathbf{x}^{(i)}$ contains the extracted HFs and $\mathbf{t}^{(i)}$ includes all the assigned failure labels per each target component. In case the $t_l^{(i)}$ classes are mutually exclusive, the classification task is a multi-class problem. A SML algorithm is built in order to output a prediction $\mathbf{y}^{(i)}$ of each sample in order to reach the closest optimal solution $\mathbf{y}^{(i)} \approx \mathbf{t}^{(i)}$, through a proper function $\mathbf{h}(\mathbf{x}^{(i)})$. Each of

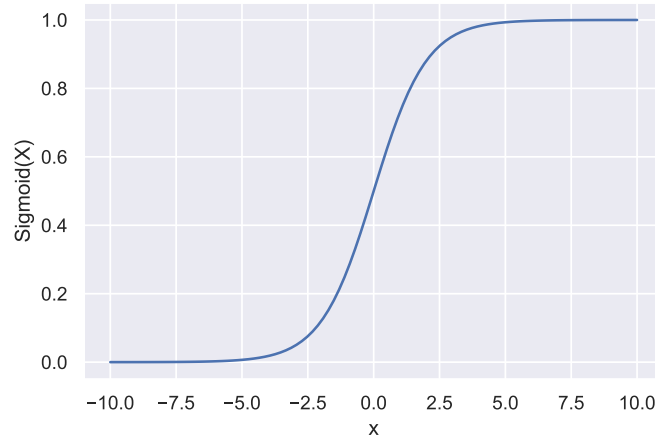


Figure 6.17: Sigmoid Function

the described algorithm in paragraphs 6.4.1, 6.4.2, 6.4.3 and 6.4.4 describes more in detail the adopted function per each candidate Classifier.

6.4.1 Logistic Regression for Classification

As reported in [91] documentation, the Logistic Regression algorithm is part of the linear model for classification, where the linear function $\mathbf{h}(\mathbf{x}^{(i)})$ can be written as in Eq. 6.7:

$$\mathbf{y}^{(i)} = \mathbf{h}(\mathbf{x}^{(i)}) = \theta_0 + \sum_{j=1}^n \theta_j x_j^{(i)} = \boldsymbol{\theta}^T \mathbf{x}^{(i)} \quad (6.7)$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1, \dots, \theta_n)$ represents the vector of linear coefficients. These parameters are obtained minimizing the cost function, defined as the sum of squared errors between the actual target $\mathbf{t}^{(i)}$ and the predicted $\mathbf{y}^{(i)}$, as reported in Eq. 6.8:

$$J(\boldsymbol{\theta}) = \sum_i^m (t^{(i)} - y^{(i)})^2 = \sum_i^m (t^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)})^2 \quad (6.8)$$

In this view, the Logistic Regression algorithm belongs to the class of Generalized Linear Model, since it calculates the probability of $\mathbf{y}^{(i)}$ to be assigned to a specific class, through an activation sigmoid function, that is by definition the logit inverse function:

$$p(y^{(i)}) = \text{logit}^{-1}(\boldsymbol{\theta}^T \mathbf{x}^{(i)}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}^{(i)}}} \quad (6.9)$$

The function in Eq. 6.9 can be written in terms of probability of correct classification p and its opposite value $q(y^{(i)}) = 1 - p(y^{(i)})$. This ratio basically explains

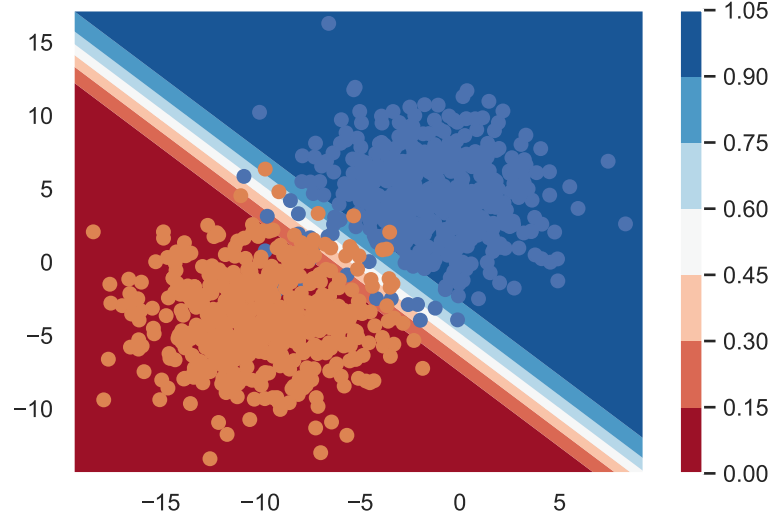


Figure 6.18: Logistic Regression probability contours [103]

the odds of a certain feature combination results to be assigned correctly or not. Mathematically:

$$\boldsymbol{\theta}^T \mathbf{x}^{(i)} = \log \left(\frac{p(y^{(i)})}{q(y^{(i)})} \right) \quad (6.10)$$

Typically, the ratio p/q is known as Odds ratio: the Eq. 6.10 shows clearly how in the Logistic regression the features combination is linear to the logarithm of the output variable odd ratio. Accordingly to the Eq. 6.10, the cost function in Eq. 6.11 can be written as:

$$J_{ce}(\boldsymbol{\theta}) = \sum_i^m \left(t^{(i)} - \log \left(\frac{p(y^{(i)})}{q(y^{(i)})} \right) \right)^2 \quad (6.11)$$

The equation 6.11 is typically referred as **cross-entropy** function, and it will be particularly important for tuning of MLP classifier in paragraph 6.4.4. Minimizing the cost function, it is possible to obtain the most likely values of parameters $\boldsymbol{\theta}$, given the assigned a-priori labels $\{t^{(i)}\}_{i=1,\dots,m}$.

The plot in Fig. 6.18 shows how the Logisti Regression plots the classification decision boundaries, with the assigned probability of assignment at two different classes. The central division white area is calculated by the $p(y^{(i)}) = 0.5$, therefore solving the equation $\boldsymbol{\theta}^T \mathbf{x}^{(i)} = 0$.

The package Scikit-Learn includes the function `LogisticRegression` that, for a multi-class problem, it used by default the One-vs-the-Rest (OvR), fitting the each class probability versus the others, with optional LASSO (11) or RIDGE (12) regularization to improve numerical stability. These methods aim to reduce the

algorithm complexity to obtain a better performance, by adding a penalization term to the cost-function for the evaluation of the $\boldsymbol{\theta}$ coefficients. The RIDGE regulator (12) in Eq. 6.12 introduces a penalty coefficient that is proportional to the sum of the square values of the evaluated θ_j coefficients, while the LASSO (11) in Eq. 6.13 evaluates the absolute ones.

$$J_{ref}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + C \sum_{j=1}^n \theta_j^2 \quad (6.12)$$

$$J_{ref}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + C \sum_{j=1}^n |\theta_j| \quad (6.13)$$

This function includes also different solvers to minimize the aforementioned defined cost functions. In this work, the two tested ones are:

- **liblinear**: it uses a coordinate descent algorithm. In this case, the optimization problem is split with the OvR approach, in order to set binary classifiers per each assigned class.
- **saga**: it uses a Stochastic Average Gradient descent algorithm [104] with additional optimization with LASSO regulator.

According to the solver, the regulators 11 or 12 can be either set in the function, by setting an appropriate penalty coefficient C to avoid overfitting. Furthermore, this function allows to chose a weighted classification, according to the number of samples per each class, in order to reduce the chances of misclassification for low populated classes. The Eq. 6.14 shows how the weight is calculated, according to the m number of samples, K classes and n_{class} datapoint per each class:

$$w_{class} = \frac{m}{K * n_{class}} \quad (6.14)$$

6.4.2 K-Nearest Neighbour for Classification

The Logistic Regressor in paragraph 6.4.1 can be defined as a parametric algorithm, since it evaluates from statistical inference the relations between features and target label. The K-Nearest Neighbors (KNN) instead does not present any parameter that needs to be statistically inferred by data distribution, but it is built by the so called Hyper-Parameters that needs to be learned from the dataset itself. Due to its simplicity, this algorithm is often considered for classification issues. For instance, in [105] it has been tested for failure recognition for aeronautical components. Considering a new data-point $\boldsymbol{x}^{(o)}$, defined by the same n features, the KNN is designed in order to assign the most probable class $\boldsymbol{y}^{(o)}$ according to the

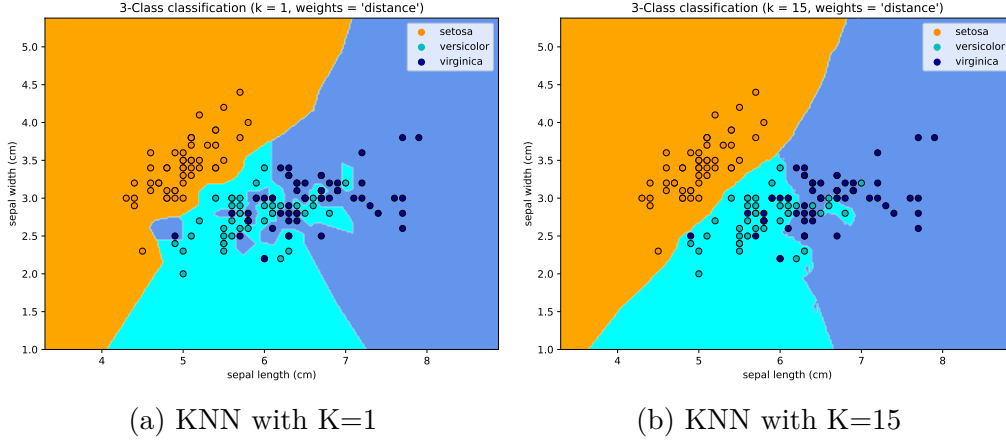


Figure 6.19: KNN Classification examples depending on K

closest k -datapoints of the complete data-set $\mathbf{x}^{(i)}$. Considering the closer points to the new one $\mathbf{x}^{(o)}$ as $N_k(x^{(o)})$, the classification problem can be written as in Eq. 6.15:

$$\mathbf{y}^{(o)} = \frac{1}{k} \sum_{x^{(i)} \in N_k(x^{(o)})}^m w(d_{oi}) \mathbf{t}^{(i)} \tag{6.15}$$

where $d_{oi} = \|\mathbf{x}^o - \mathbf{x}^i\|$ is the euclidean distance that needs to be evaluated between the new point and the others in the dataset. As reported in [91], the classification is determined from "a simple majority vote of the nearest neighbours of each point", based on the weighted distance function. The $w(d_{oi})$ represents the weight function: in this way, the nearer neighbors contribute more to determine the datapoint class. The function `KNeighborsClassifier` includes two possible options: the `uniform` one assigns a constant weight to each neighbor, while the `distance` option weights them proportionally to the inverse of their distances. In this latter case, the closer points have a greater influence than the further ones.

The example in Fig. 6.19 from the Scikit-Learn package documentation shows how this algorithm is influenced by the right choice of its hyper-parameter K . This number represents the number of K closest points to be included in $N_k(x^{(o)})$ for class evaluation. In particular, the first Fig. 6.19a shows a typical overfitting problem where each in the testing dataset has been assigned to the class of the closest one from the training dataset. In the second plot in Fig. 6.19b is obtained imposing $k = 15$, making the classifier more robust also in terms of decision boundaries delimitations. Due to its impact, this value is typically chosen for tuning the classifier itself based on the input dataset.

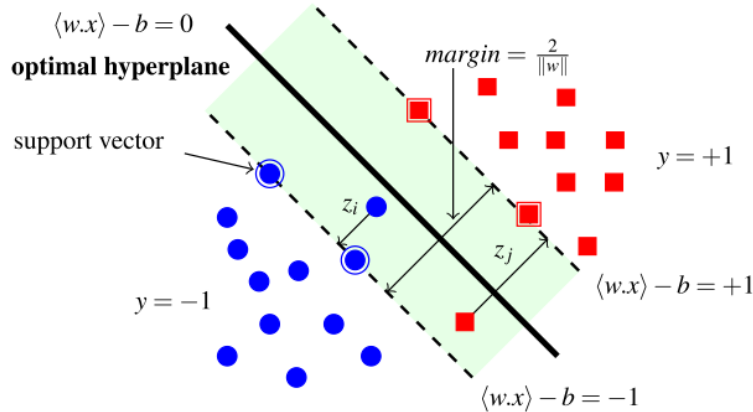


Figure 6.20: Support Vector Machine Classifier [106]

6.4.3 Support Vector Machine for Classification

Support Vector Machines (SVM) have been widely used for a large series of classification problems, such as facial recognition and text categorization [106]. As reported in [58], the distinguishing feature of such algorithms regards the hyper-parameters tuning, which can be considered as a convex optimization problem. In Fig. 6.20 the key strategy of SVM is shown for a binary classification problem: the identification of an optimal hyper-plane that separates the classes and minimizes the classification error.

In order to trace this optimal hyperplane, a SVM algorithm evaluates the margins: the smallest distance between decision boundaries and any class sample. In Fig. 6.20 they are represented with as the dashed lines that delimit the decision boundaries area. The location of these margins is determined by a subset of data points, called support vectors, that lay on these borders (the highlighted blue dots and red squares). Considering the linear function $f(\mathbf{x})$ in Eq. 6.16:

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^n \beta_j x_j \quad (6.16)$$

the hyperplane is defined by $f(\mathbf{x}) = 0$. The linear coefficients $\{\beta_j\}_{j=0,1,\dots,n}$ defines the normal vector of the hyperplane $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)^T$, that can be used to evaluate the distance between each sample point $\mathbf{x}^{(i)}$ in the training set as:

$$d_i = \frac{f(\mathbf{x}^{(i)})}{\|\boldsymbol{\beta}\|} \quad (6.17)$$

The margins definition can be completed assuming that the training data set is linearly separable in the feature space. Due to this assumption, there is at least on choice of parameters $\boldsymbol{\beta}$ that, considering the classes $y = 1$ and $y = -1$ in Fig. 6.20,

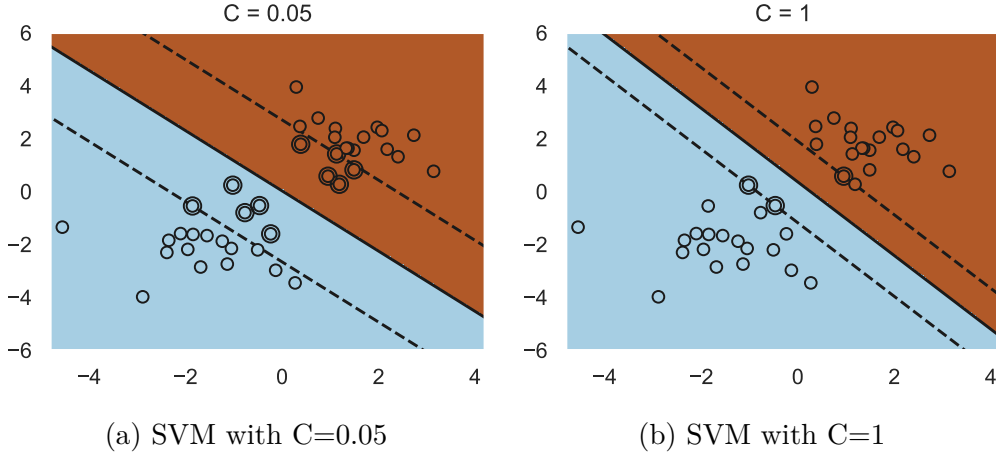


Figure 6.21: SVM Classification examples depending on C

satisfy the sign condition $f(\mathbf{x}) > 0$ for the first mentioned class, and vice versa $f(\mathbf{x}) < 0$ for the second. In general, per all the n data points, it is important to find the solutions that correctly classify them, therefore that respects the general condition: $y_i f(\mathbf{x}^{(i)}) > 0$. In [58] the SVM maximum margin solution problem is reported as:

$$\operatorname{argmax}_{\beta, \beta_0} \left\{ \frac{1}{\|\beta\|} \min_i (\beta_0 + \beta^T \mathbf{x}^{(i)}) \right\} \quad (6.18)$$

In order to simplify the solution of the problem in Eq. 6.18, it is necessary to rescale the linear coefficients β_j , according to the margin M to be maximized, in order to set the limit condition $\|\beta\| = 1/M$. The maximization M problem can be therefore reformulated as:

$$\operatorname{argmax}_{\beta, \beta_0} M \Rightarrow \operatorname{argmin}_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \quad (6.19)$$

considering the following distance constraint:

$$|d_i| = y_i \frac{\beta_0 + \beta^T \mathbf{x}^{(i)}}{\|\beta\|} \geq M \Rightarrow y_i (\beta_0 + \beta^T \mathbf{x}^{(i)}) \geq 1 \quad (6.20)$$

The expression in Eq. 6.20 is usually referred as the hyperplane canonical representation. The data points that respects the equality condition in Eq. 6.20 are the active support vectors of the hyperplane. The main typical challenge of this classifier is that typically the classes points are not perfectly separable by the hyperplane, therefore it is often necessary to temper the constraint in Eq. 6.20 by defining a degree of misclassification $\xi_i \geq 0$ so that:

$$y_i (\beta_0 + \beta^T \mathbf{x}^{(i)}) \geq 1 - \xi_i \quad (6.21)$$

If $0 < \xi_i \leq 1$, then the point $\mathbf{x}^{(i)}$ falls within the green margin areas in Fig. 6.20 while, if $\xi_i > 1$, then the point is misclassified. In order to avoid both these inconveniences, the optimization problem in Eq. 6.18 can be reformulated as:

$$\underset{\beta, \beta_0}{\operatorname{argmin}} \left(\frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^m \xi_i \right) \quad (6.22)$$

The hyper parameter C is a penalty term that acts like the inverse of regularization parameter. As reported in [58], C regulates a trade-off between minimizing the training errors and setting model complexity. As previously reported, any misclassified point is characterized by $\xi_i > 1$, therefore the sum $\sum_{i=1}^m \xi_i$ set the upper bound on the number of allowed misclassified points: increasing C , the term $\sum_{i=1}^m \xi_i$ weight in Eq. 6.22 results higher therefore the SVM would minimize the margins from the first term in Eq. 6.22. An example of the influence of C in determining the classification margins is showed in Fig. 6.21: in these plots it is possible to observe how a larger value of C makes the classifier to consider the points close to the separation line as support vectors.

The concept of hyperplane in Eq. 6.16 can be generalized in a hyper-surface equation $f(\mathbf{x}) = 0$ by the introduction of a Kernel function $K(\mathbf{x}, \mathbf{x}^{(i)})$:

$$f(\mathbf{x}) = \alpha_0 + \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}^{(i)}) \quad (6.23)$$

In case of linear SVM, the Eq. 6.23 can be traced back to Eq. 6.16 by simply considering the Kernel function as the scalar product $K(\mathbf{x}, \mathbf{x}^{(i)}) = \langle \mathbf{x}, \mathbf{x}^{(i)} \rangle$. The Scikit-learn function `sklearn.svm.SVC` allows the user to set possible four different possible Kernel functions:

1. Linear Kernel: $K(\mathbf{x}, \mathbf{x}^{(i)}) = \langle \mathbf{x}, \mathbf{x}^{(i)} \rangle$;
2. Polynomial Kernel: $K(\mathbf{x}, \mathbf{x}^{(i)}) = (\gamma \langle \mathbf{x}, \mathbf{x}^{(i)} \rangle + r)^d$
3. Radial Basis Function (RBF) Kernel: $K(\mathbf{x}, \mathbf{x}^{(i)}) = \exp(\gamma \|\mathbf{x} - \mathbf{x}^{(i)}\|^2)$
4. Sigmoid Kernel: $K(\mathbf{x}, \mathbf{x}^{(i)}) = \tanh(\gamma \langle \mathbf{x}, \mathbf{x}^{(i)} \rangle + r)$

From the Kernel functions equations, a new hyper-parameter γ is defined and needs to be tuned: it defines the influence that each training data point has in the classification. It is defined in [91] as "the inverse of the radius of influence of samples selected by the model as support vectors". A large series of examples are reported in the Scikit-Learn user guide that are useful to understand both the differences of each Kernel function, and the importance of correctly tuning the hyper-parameters C and γ .

In Fig. 6.22 three different Kernel functions are compared and it is possible to observe the extreme differences in the decision boundaries. In particular, the

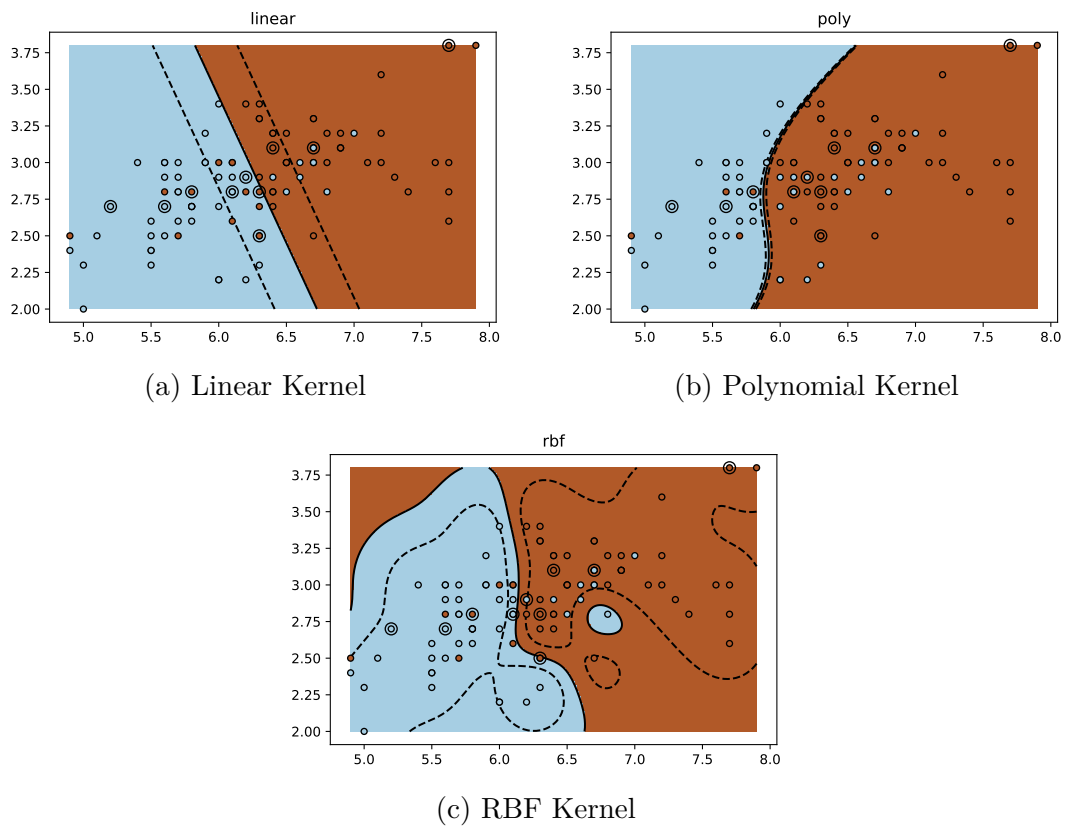


Figure 6.22: SVM Classification with different Kernels [91]

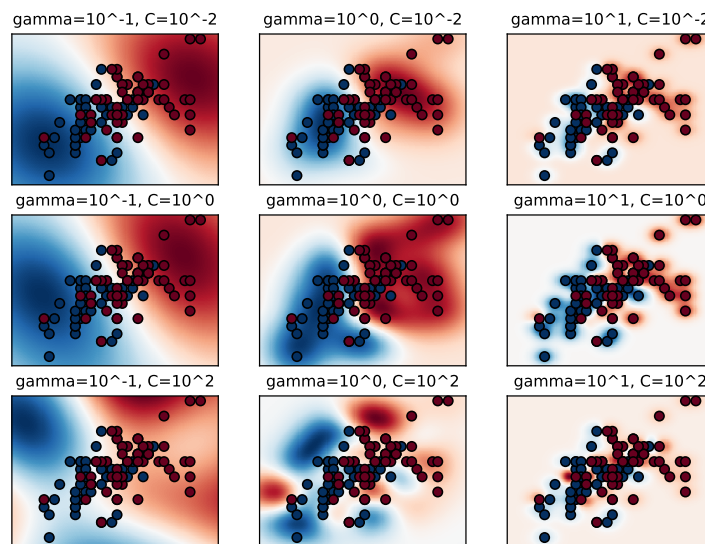


Figure 6.23: SVM with RBF Kernel: C and γ influence [91]

polynomial and RBF functions are particularly useful when the data are not linearly separable. In Fig. 6.23 the SVM algorithm has been fixed with RBF kernel function, and the influence of C and γ are analysed. If γ is too large, the tracked RBF area includes only the support vectors and the influence of C does not avoid overfitting. On the contrary, if γ is too small, the SVM does not fully extract the complexity of the data: the influence region of each support vector would include the complete training set. In this matrix of plot, the models with small γ and $\max(C) = 1$ show a better classification: the complexity of these smooth models with low values of γ can be improved by increasing the C values from $C = [0.01, 1]$

6.4.4 Multi-Layer Perceptron for Classification

The Multilayer Perceptron (MLP) is type of feedforward Artificial Neural Network (ANN). As reported in [107], this term refers to a layered structure of multiple perceptrons. The great advantage of these perceptrons is the possibility to employ arbitrary functions, while the state-of-the-art "percpetron" definition infers artificial neurons with threshold activation function. This kind of algorithm can be used to correctly perform both classification and regression tasks. For instance, these algorithms have been widely used for diagnostic purposes for engineering applications. In [101], [108] and [109] supervised MLP have been tuned to detect failures in hydraulic actuator circuit. Fig.6.24 shows the typical representation of a MLP classifier, with n inputs in the first layer and m possible outputs in the last classification layer. Between these two layers, an intermediate Hidden one is positioned and it contains specific neurons that linearly combine the previous input features $\mathbf{x} = \{x_j\}_{j=1,\dots,n}$ and are activated by specific non-linear functions. Mathematically, the input features from the initial layer are combined with a set of weights w_{kj} and bias w_{k0} parameters and then elaborated with a proper activation function $h(\mathbf{x})$ that computes a new set of features $\{H_q\}_{q=1,\dots,L}$ in the L nodes of the hidden layer:

$$H_q = h(w_{k0} + \sum_{j=1}^n w_{kj}x_j) \quad (6.24)$$

Typical activation functions for $h(\mathbf{x})$ are the logistic one in Eq. 6.18 or the hyperbolic tangent \tanh in Eq. 6.25:

$$\tanh = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (6.25)$$

The hidden layer new features H_q are then the inputs for the final output layer. Following the same approach in Eq. 6.24, the final $\mathbf{y} = \{y_l\}_{l=1,\dots,m}$ can be calculated by setting appropriate combination weights and a dedicated activation function $f(\mathbf{y})$:

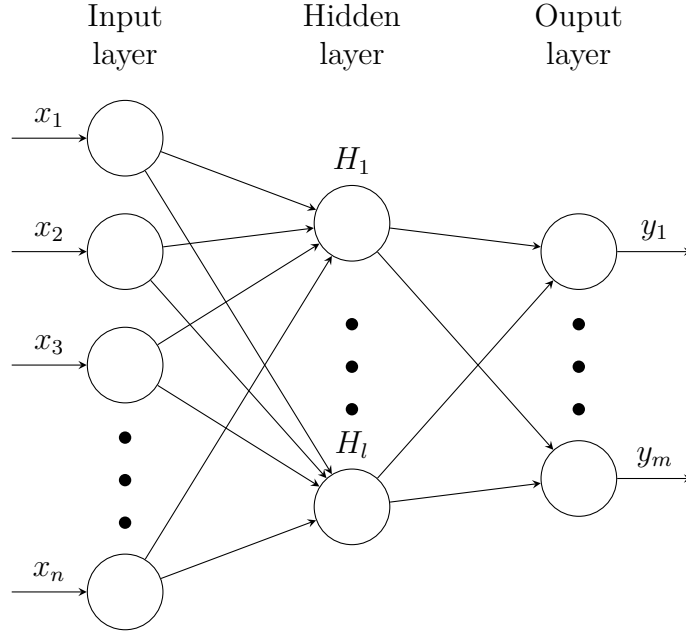


Figure 6.24: MLP structure

$$y_l = f(\beta_{l0} + \sum_{q=1}^L \beta_{lq} H_q) \quad (6.26)$$

For binary classification purposes, logistic function is typically used to activate the last output layer since a simple probability threshold of 0.5 would assign a specific data-point, described by the feature vector \boldsymbol{x} , to a class whether than the other. In case of multi-class classification problem, a recommended choice for output activation function is the *softmax* one in Eq. 6.27:

$$f(s_l) = \frac{\exp(s_l)}{\sum_{kk=1}^K \exp(s_{kk})} \quad (6.27)$$

where s_l is the l -th input to the softmax function, which corresponds to the kk -th class over a total K number of classes. The *softmax* function outputs a vector where each element corresponds to the probability of the input vector to belong at each kk class: the highest probability class becomes the assigned one.

In analogy with the Logistic Regressor in 6.4.1, the MLP algorithm can be classified as a parametric classifier, due to the statistical link that collect the input features \boldsymbol{x} with the most promising assigned class in terms of probability. In fact, it has been previously mentioned in 6.4.1 how the *cross-entropy* cost function in Eq. 6.11 would have been used to increase the MLP performance. Since the hidden-layer a typical activation $h(\boldsymbol{x})$ is the logistic function, the cost function in Eq. 6.11 can be written as follow:

$$J_{ce}(\mathbf{t}, \mathbf{y}) = - \sum_{i=1}^m \sum_{l=1}^K [t_l^{(i)} \log(y_l^{(i)}) + (1 - t_l^{(i)}) \log(1 - y_l^{(i)})] \quad (6.28)$$

In analogy with Logistic Regressor, the loss function equation 6.28 can be correct with a regularization term as follow:

$$J_{ce}(\mathbf{t}, \mathbf{y}, \mathbf{W}_2) = J_{ce}(\mathbf{t}, \mathbf{y}) + \alpha \|\mathbf{W}_2\|^2 \quad (6.29)$$

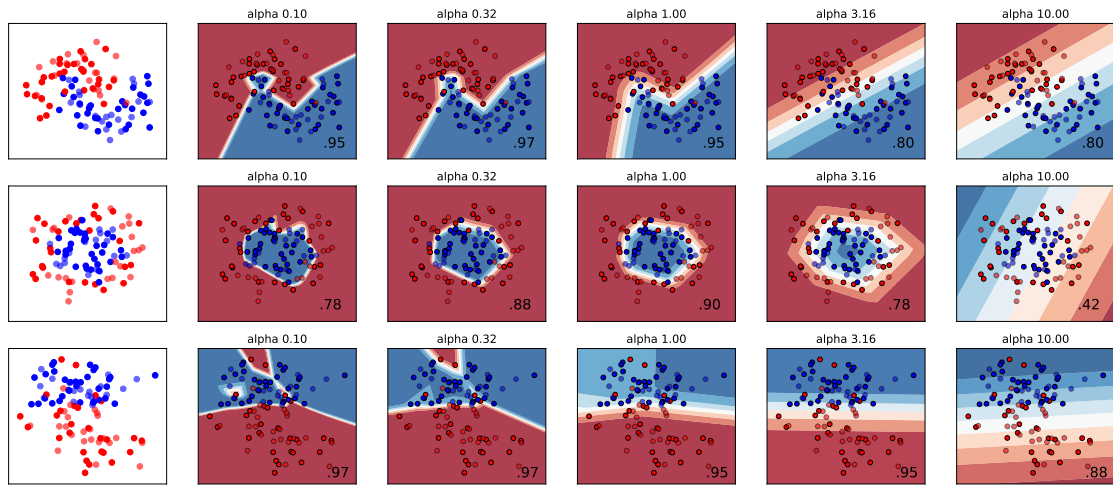
where $\|\mathbf{W}_2\|^2$ is the square module of the output weights layer \mathbf{W}_2 . The second term is equivalent to the RIDGE regularization term in Eq. 6.12 where the α parameter regulates the penalty magnitude. The Scikit-Learn function `MLPClassifier` allows the user to chose three different possible algorithms to updates the MLP parameters through the optimization of the cost function in Eq. 6.28. Starting with random values of \mathbf{W}_2 , the MLP minimizes the cost-function by updating the weights in an iterative way. After evaluating the loss, the backward step propagates the value of J_{ce} from the output layer to the previous ones: each weight is updated by progressively reducing the loss. For instance, the standard Stochastic Gradient Descent ∇J_{ce} of the loss function is evaluated at each iterative step as in Eq. 6.30:

$$\mathbf{W}^{i+1} = \mathbf{W}^i - \epsilon \nabla J_{ce}^{(i)} \quad (6.30)$$

where ϵ is an additional learning rate to set. The algorithm reaches the optimal solution when the improvement in the loss function is lower than a certain value, or after a maximum number of iterations. Other possible optimization algorithms in the `MLPClassifier` are ADAMS [110] and L-BFGS [111]. The plots in Fig. 6.25 are obtained from an example reported in the user guide of the `MLPClassifier` function. These plots are obtained with a double hidden-layers MLP and the L-BFGS solver, and they show how different values of α yield different decision boundaries. High values of the penalty term can solve problems of high variance, reducing the layers weights, creating more straight decision boundaries. On the contrary, lower values of α may fix high bias in classification, determining more complicated boundaries.

6.5 Classifiers Comparison and Results

The four presented algorithms (from 6.4.1 to 6.4.4) have been chosen as possible candidates of the core failure classification in the Advanced Diagnostic module. The potentials of these candidates have been already explored in previous related works, such as [32] and [18]. In these two latest publications, the capabilities of each algorithm have been tested over a dataset of simulated data, obtained from an extensive simulation campaign with a developed High Fidelity Model. In particular, in [32] both the possibilities of single and multiple failures classifications

Figure 6.25: MLP Classifier tuning α

have been explored, injecting in the Elevator EHSV component parametric failures in both first and second stage of this particular valve. However, as reported in paragraph 6.2, the experimental campaign in the LHT test-benches have yielded to a series of recurrent anomalous behaviours over three main unit components: the aforementioned EHSV, the mechanical RD and the MSV.

The flowchart in Fig. 6.26 includes the previously reported one in Fig. 6.9. Initially, the `adv_diagnose_classification` script gives the possibility to select the target components, extract its characterizing HFs, store them in appropriate pivot dataframes `df_hfs` and then the dedicated function `class_definition` assign the proper failure labels per each table entry. These last column in the green table in Fig. 6.10, labelled as "Target", constitutes the vector of classes \mathbf{t} the supervised classifier needs to correctly identify. Once the target column \mathbf{t} has been filled, two parallel tasks needs to be logically fulfilled. The first one, within the blue square, regards the HFs selection with the ANOVA f-score method, previously described in paragraph 6.3, with the goal to extract the list of most promising HFs for failure classification of the target component. The second task regards the `df_hfs` split into Training and Test dataframes. The necessity of splitting a dataset for evaluating a SML performance is a well established methodology to avoid the so called "overfitting" problem: avoiding misclassification on new data-point, since the algorithm has just been trained with a unique set of data. More details about this important step are reported in 6.5.1. Once these subset of points have been extracted, their columns have been normally standardized, using the same `StandardScaler` python function mentioned in paragraph 6.3, and afterwards only the output subset of HFs from the Feature Selection method are picked.

Once the complete `df_hfs` has been split, normalized and the most promising HFs extracted, then it is necessary to tune the candidate classifiers using the training

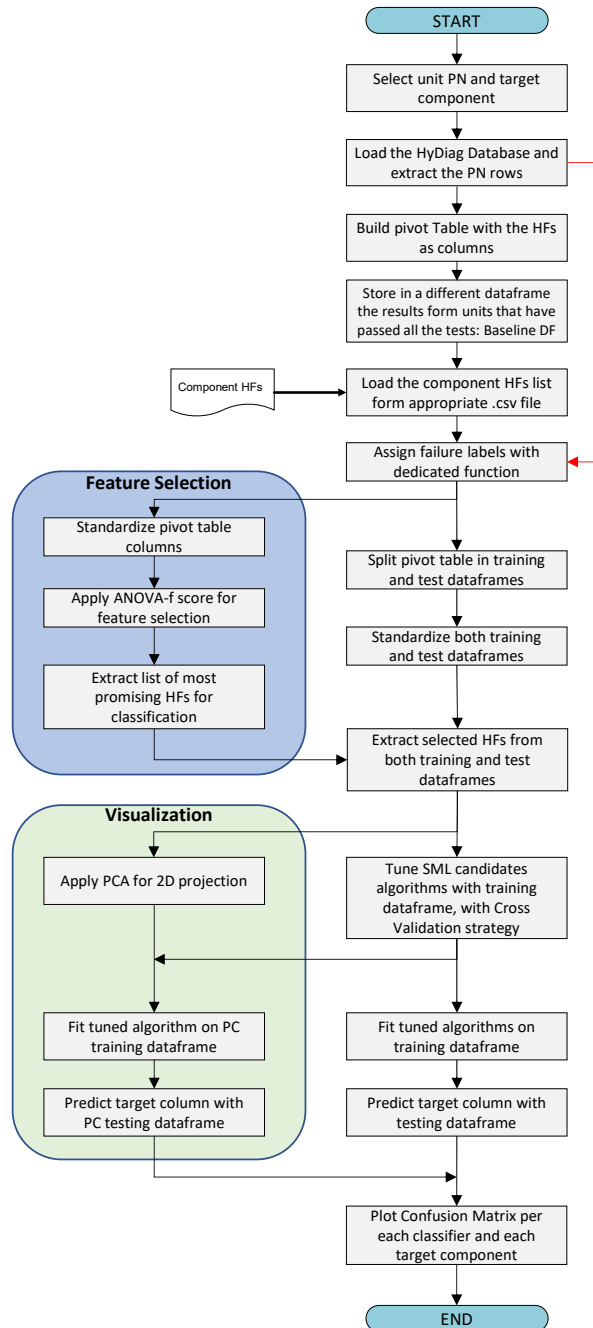


Figure 6.26: Classification Flowchart

dataset. From the descriptions in paragraphs 6.4.1 to 6.4.4, it is evident how the chosen classifier to be tested present a wide range of parameters or hyper-parameters to tune in order to perform a correct classification. Another well-established method for ML algorithm tuning is the "Cross-Validation" technique [112–115]. More details about the parameters ranges and tuning methods are described in paragraph 6.5.1. These tuned algorithms are then fit over the training dataframe with the complete set of selected HFs, and then the predicted classes \mathbf{y}_{test} are forecasted by the test set. The comparison between predicted \mathbf{y}_{test} and target \mathbf{t}_{test} failure labels are graphically reported in Confusion Matrix per each classifier. These plots summarize as table layouts visualize the performance of a SML, and they are typically used to evaluate the typical classification metrics such as the algorithm prediction Accuracy. These results and classification performance metrics are described in paragraph 6.5.2.

With the multi-dimensional datasets, as the extracted training and testing ones, it is often useful to project the data in to a lower dimensional subspace "which captures the essence of the data" [116] and for visualization purposes. The green box in Fig. 6.26 reports the steps that have been followed to project the multi-dimensional dataset into two-dimensional projection using a PCA algorithm. As reported in [117], this technique transform the set of data into a new coordinate system, by linearly combining the set of features in order to maximize the variance of data distributions along the new coordinates. Reducing the set of selected HFs with two main Principal Components coordinates, it is possible to visualize the decision boundaries each tuned classifier can identify per each failure label. This technique is in detail described in paragraph 6.5.2 and the performance of each classifier is compared before and after the PCA application.

6.5.1 Dataset split and Classifier Tuning

As reported in [91], the problem of overfitting is typically generated when the parameters (or hyper-parameters) of a SML are tuned and tested over the same dataset. Due to this methodological error, the model would not be able to correctly classify a novel set of data. The best practice to follow in order to avoid this problem are showed in Fig. 6.27. This typical cross-validation workflow indicates as first step the dataset split into training and test subset: the first one is used to fit the SML algorithm and tune its parameters, while the second one allows to evaluate the SML performance using new "un-seen" data. This procedure is easily implemented in the `adv_diagnose_classification` using the Scikit-learn function `train_test_split`. This function has been applied to the `df_hfs` dataframe, as reported in Listing 6.3:

Listing 6.3: Training and Test split

```
df_train, df_test, target_train, target_test =  
    train_test_split(df_hfs[list_selected_hfs], df_hfs[COMP],
```

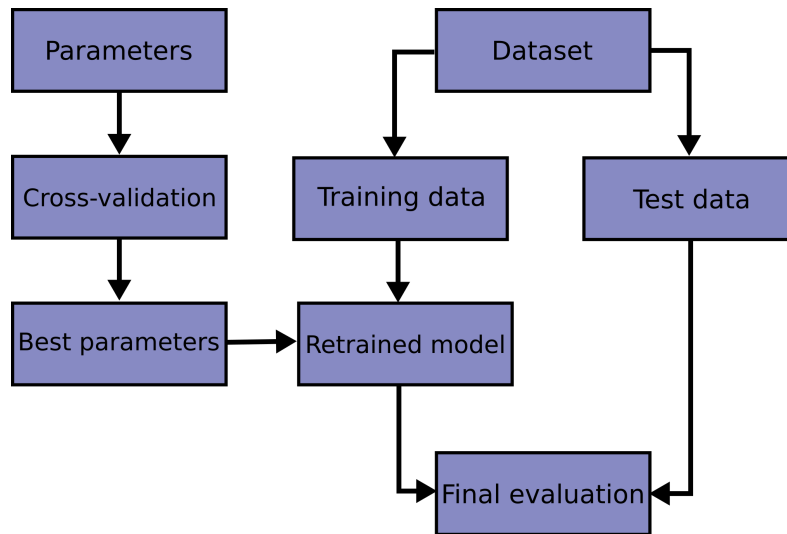



Figure 6.27: Best Practice Guideline to avoid overfitting [91]

```
stratify=df_hfs[COMP], random_state=0, test_size=0.33)
```

This function takes input the complete dataset and the relative labels-target column (`df_hfs[COMP]`), and it returns both training and test features dataset and their relative labels columns. It allows to set the split percentage, which is in this case 33% for the test set and the remaining 67% in the training one, through the setting `test_size`. Normally, the dataset rows are split randomly between the two outputs dataframes, and this procedure ensures the resulting dataset to be representative samples of the original one. However, typically when comparing different SML, it is preferred that these are trained and evaluated over the same subsets: this task is solved in Python assigning the `random_state` variable to an integer value, in order to fix the seed of pseudo-random number generator [118]. The last input `stratify` ensures that both training and test sets have a the same proportions of example per each observed class in the original dataset.

Keeping a proportional training dataset, composed by balanced samples for all the target classes in t , is fundamental for an effective tuning of the classifier. A well-established method for parameters (or hyper-parameters) evaluation is called Cross-Validation (CV) [113, 114, 119, 120]. For instance, considering the SVM hyper-parameter C introduced in Eq. 6.22, its manual set may induce a risk of overfitting on the test subset, since it still can be optimized to include new data-points as support vector of the decision hyper-planes. A possible solution for this issue can be to introduce a third "validation" set part of the initial dataset, than can be used after the training process to check its effectiveness. However, an additional split may reduce the performance training, especially if the dataset does not contain several examples per all the different classes. This issue is solved by applying a CV

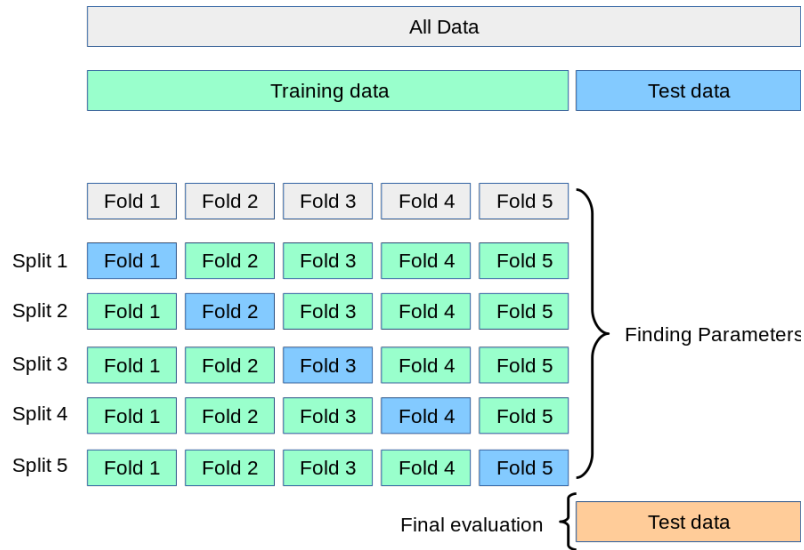


Figure 6.28: Cross-Validation Technique [91]

technique over the training set. In Fig. 6.28, the k -fold CV technique is presented: the training set is divided in k smaller sets and, per each of them:

- The classifier is trained over the $k - 1$ subsets;
- The trained classifier is then validated on the remaining one, and the performance metric is then averaged at the end of this loop.

In this process, the performance metric needs to be appropriately selected according to the user need, in terms of classification Precision and Recall. More details about the selected classifier metrics are reported in paragraph 6.5.2.

The best CV score is typically used to find the best set of classifier parameters (or hyper-parameters). This concept is graphically reported in the left branch of the workflow in Fig. 6.27: the idea is to pick the best combination of parameters, each of them defined within a range, training the classifier, according to the CV loop, in order to maximize a specific performance score. This task is fulfilled by the Scikit-Learn function `GridSearchCV`.

Fig. 6.29 shows how this function exploits Cross-Validation for building and evaluating the performance of each model, built from combination of specified input parameters. The idea is to test different models, obtained combining each parameter within each range, over k subsets of randomly sampled data-points, using CV technique for training and testing each models. The `GridSearchCV` function outputs the set of parameters that define the model with the highest model performance score, specified as input. In the `adv_diagnose_classification` script, this function has been used to tune the parameters (or hyper-parameters) of the

classifier candidates algorithms. The code in 6.4 shows both the fixed settings and the ranges of parameters that needs to be tested by `GridSearchCV`:

Listing 6.4: Classifier Parameters

```
classifiers = [
(LogisticRegression(class_weight='balanced', random_state=0), {
    'penalty': ['l1', 'l2'],
    'C': np.logspace(-4, 2, 20),
    'solver': ['liblinear', 'saga']})),
(KNeighborsClassifier(weights='distance'), {
    'n_neighbors': list(range(3, 9))}),
(SVC(class_weight='balanced'), {
    'kernel': ['rbf', 'linear', 'sigmoid'],
    'C': [1, 10, 100, 1000],
    'gamma': [1e-3, 1e-2, 1e-1, 1, 2]})),
(MLPClassifier(max_iter=1000), {
    'hidden_layer_sizes': [3, 10, 30, 50, 100],
    'activation': ['logistic', 'relu'],
    'alpha': [1e-4, 1e-3, 1e-2, 1e-1, 1]})]
```

- For the Logistic-Regression algorithm the `balanced` method has been imposed by default in order to reduce penalization for less populated classes. The other parameters to be tuned regards both the penalization strategies, if LASSO in Eq. 6.13 or RIDGE L2 in Eq. 6.12, the regularization parameter C and the optimization algorithms for the cost-function.

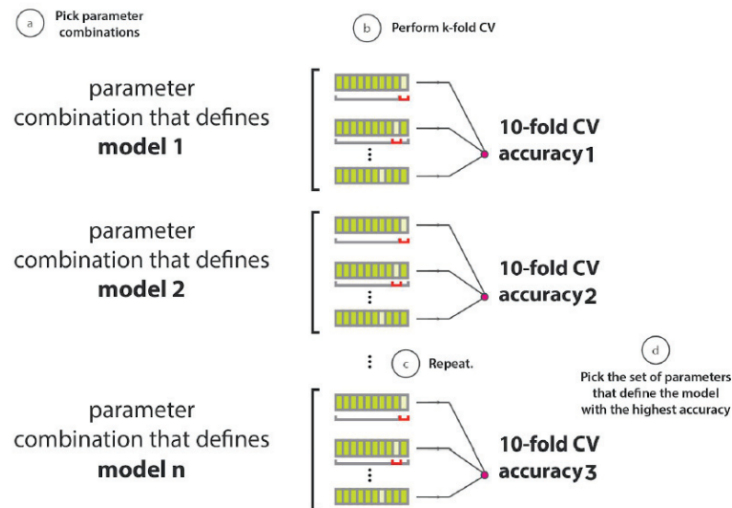


Figure 6.29: GridSearchCV strategy [121]

Table 6.6: Classifiers tuned parameters

Classifier	Parameters	EHSV	RD	MSV
LogReg	Penalty	RIDGE	LASSO	LASSO
	C	0.3	0.3	0.62
	Solver	saga	liblinear	liblinear
KNN	k	3	3	6
SVM	Kernel	linear	sigmoid	rbf
	C	1	100	10
	γ	0.001	0.001	0.01
MLP	Hidden Layer size	30	50	3
	Activation function	logistic	relu	relu
	α	0.0001	0.001	0.01

- The KNN hyper-parameter to be tuned is only the number of neighbours k to take into account for determining the class.
- The same **balanced** method has been set for the SVM, as for the Logistic-Regression one. In this case, the function **GridSearchCV** needs to compare different kernel functions, and a set of C and γ to be tested.
- The only fixed parameter of the MLP algorithm regards the allowed maximum number of iterations for back-propagation tuning of the internal neurons weights. All the other settings, such as size of hidden layer, activation function and regularization parameter α .

The classifier candidates in Listing 6.4 needs to be tuned and evaluated per all target-component datasets. In this way, it is possible to compare each classifier performance for both EHSV, RD and MSV failures classification, and decide which algorithm can be used per each target component. Since each classification problem includes peculiar characteristic and data distribution over the failure label, it would be very challenging to define a unique classifier that can be used per all of them. Therefore, with this strategy, it is possible to tailor a specific and optimized solution per each component and decide to implement all of them in the final Reasoner module in Fig. 6.1.

Table 6.6 summarizes the tuned settings of the candidates in Listing 6.4 obtained by **GridSearchCV**, per each target-component classification problem. These results have been calculated over the training dataset, considering all the selected HFs with the ANOVA-F score method per each target-component, and setting a k -fold CV process with $k = 5$.

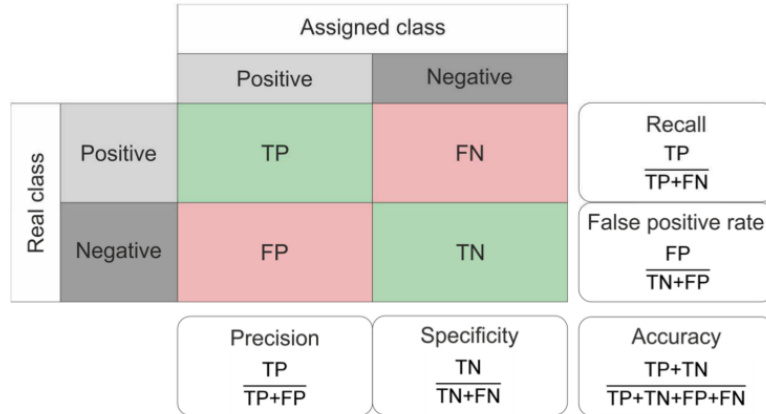


Figure 6.30: Confusion matrix and related performance measure [123]

6.5.2 Classification Results

One of the most common methods for reporting the classification results of a SML algorithm is the Confusion Matrix. In Fig. 6.30 an example for binary classification is reported. Each matrix row and column may include respectively the instances of a predicted class versus the actual classified one, or vice versa. For binary classification, each matrix cell represents a possible classification outcome [122]: the diagonal cells indicates the number of points for which the predicted and actual classes coincide, while the other off-diagonal elements are the misclassified ones. Therefore, it is easy to observe how the classifier performances are influenced by the number of points indicated in the matrix diagonal. In [91], it is reported that, by definition, any cell i, j reports the number of observation actually in the i -th group but predicted in the j -th one.

These elements are commonly referred with terms from binary classification, where typically "positive" and "negative" refer to the prediction, while "true and "false" to whether the prediction corresponds to the actual class. For this reason, the diagonal elements in Fig. 6.30 are indicated as True Positive (TP) for correct results and True Negative (TN) for correct absence of result, while the off-diagonal elements are False Positive (FP) for unexpected results and False Negative (FN) for miss-classified actual positive ones . Based on this notation, this graphical representations allows to easily calculate different performance metrics for classification [92]:

- **Accuracy:** most common used metric to evaluate classification performance, defined as ratio between the number of correct predictions and the total number of predictions (Eq.6.31):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.31)$$

This metric gives a clear indication of SML performance, however it can lead to unclear results especially for class-imbalanced data set or negative impact of misclassified results. For instance, in [124] it is reported an binary classifier for benign or malign tumor diagnose. In this case, it is highlighted how, despite the high level of accuracy of 91%, due to the imbalance of the classes, it does not provide any additional benefit in distinguishing benign from malign tumor. In addition, it has to be considered the "nature" of the obtained results since, like in this example, misclassifying a sample (benign as malign or vice versa) could have serious consequences.

- **Recall** or True Positive Rate: metric that describes the classifier capability to correctly identify the positive samples:

$$TPR = Recall = \frac{TP}{TP + FN} \quad (6.32)$$

- **Precision**: it provides a measure the portion of actually correct positive classification or, in other words, "the proportion of positive predictions that match with the true class membership" [92].

$$Precision = \frac{TP}{TP + FP} \quad (6.33)$$

- **F1 score**: a typical problem when evaluating a classifier performance regard the commonly diverging trends of both Precision and Recall. The *F1* score represents the harmonic mean of these two metrics, and it is widely used in case of imbalanced classes problem or correct classification of positive class:

$$F1 = 2 * \frac{Precision \cdot Recall}{Precision + Recall} \quad (6.34)$$

Due to the harmonic mean, the *F1* score would be always closer to the smaller value between Precision and Recall than to the larger one, penalizing any poor value of the two averaged metrics. A cons of this score in case of multi-class classification regards its sensitivity to data distribution, unless it is particularly important to correctly classify the sample of a low populated class.

The correct metric choice is fundamental not only for comparing the different candidate classifier, but also to set a reference value during the tuning process. It has been already mentioned in paragraph 6.5.1 how the CV tuning process with `GridSearchCV` requires a metric to be specified during each training dataset fold. Considering the aforementioned metrics, the choice is based on two possible metric candidates: accuracy and *F1* score. For multi-class problem, the Scikit-Learn package allows the user to set three significant possible ways to average the *F1*

score in Eq. 6.34 per each class, in order to obtain a unique harmonic mean result from Precision and Recall of each class:

- **micro**: the Precision and Recall in Eq. 6.34 are calculated evaluating the individual TP, FP and FN per each i -th class:

$$Precision_{micro} = \frac{\sum_i^K TP_i}{\sum_i^K TP_i + \sum_i^K FP_i} \quad (6.35)$$

$$Recall_{micro} = \frac{\sum_i^K TP_i}{\sum_i^K TP_i + \sum_i^K FN_i} \quad (6.36)$$

This measure gives a significance to each class population.

- **macro**: in this case, the Precision and Recall of each class are averaged to evaluate a global macro values:

$$Precision_{macro} = \frac{1}{K} \sum_i^K Precision_i \quad (6.37)$$

$$Recall_{macro} = \frac{1}{K} \sum_i^K Recall_i \quad (6.38)$$

This metric can be used to know how the model performs overall across all the data points, without considering the classes population.

- **weighted**: this method differs from the previous since it consider a weighted average by the number of true instances per each label, making the macro-average more imbalance sensitive.

Considering the failure classification problem for the described target component, the **micro** $F1$ score can be considered as a good candidate for metric score, since the failure classes are imbalanced due to the few number of collected measurements but also to the large possible observable failures. In fact, even increasing the number of measurements and therefore samples in the dataset, it needs to be considered also the possibility to observe new failure that needs to be labelled and assigned at each measurement. This phenomena would therefore increase the level of granularity of observed failures but also may lead to less populated groups of samples with the same label that need to be compared to a possible larger set of no-faulty units. However, considering the Eq. 6.35 and 6.36 for N multi-class problem, if any instance needs to be classified in one unique class, therefore any FN per one class can be considered as a FP for an "adjacent" one in a $N \times N$ confusion matrix. Therefore, with this hypothesis, the micro-average $F1$ score coincides with the global Accuracy. For these reasons, the overall Accuracy score has been taken into account as reference metrics for classifier tuning and comparison.

The confusion matrices from Fig. 6.31a to 6.33d report the multi-class classification results that have been obtained from the tuned classifiers with `GridSearchCV`, considering the complete set of HFs selected with the ANOVA F-Score method. These number of each square represent the percentage of classified TP, TN, FP or FN per each class, considering the test subset. In addition, both the Accuracy and F1 macro-averaged score are reported per each classifier. These scores have been summarized in Table 6.7. The classification confusion matrices reports the classifier results in terms of overall accuracy, and they gives an effective insight of the algorithms potentials in terms of discerning between different failure labels. In particular, the percentage of correctly or wrongly classified failure labels highlight how the each classifier can find some difficulties when distinguish between contingent classes. For instance, in the EHSV classification, it can be noticed that all the tuned classifier are not able to distinguish between the "EHSV LVDT NOISY" and "EHSV GAIN TOO LOW" class. This can be physically explained by the great influence the noise level of U_{sek} in the EHSV LVDT has over the EHSV hysteresis curve in *Sequence* n.26. Therefore, this observation gives an insight for further investigation about the noise level of the EHSV LVDT secondary coil, and whether it is determined by external disturbances or by internal sources such as the first stage Torque Motor. In Appendix A a detailed investigation about the noise effect over LVDT demodulation has been conducted, however this classification issue suggests to perform some closer study via an experimental campaign. On the other hand, the candidates classifiers perform better for both RD and MSV failure classification problems, such as the SVM for both the targets components. It is important to remember however how for both these latter components a more accurate inspections of the possible faults or degradations have been conducted from samples of defective components, while it has not been possible to perform the same inspection on the EHSV.

These kind of observation can be deduced by the reported numbers in the classification matrices, but it becomes really difficult to visualize these misclassification problem in a multidimensional space, such as the one defined by the selected HFs with ANOVA F-score in paragraph 6.3. A possible solution would be to apply a further dimensionality reduction method to better visualize the potential of each

Table 6.7: Classifier Performance Metrics

Classifier	EHSV failure classif.		RD failure classif.		MSV failure classif.	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
LogReg	0.77	0.71	0.79	0.83	0.75	0.74
KNN	0.77	0.63	0.79	0.83	0.90	0.92
SVM	0.85	0.67	0.84	0.87	0.87	0.89
MLP	0.69	0.67	0.74	0.79	0.90	0.92

tuned classifier. In the light of this, the Principal Component Analysis (PCA) has been adopted in order to project the set of data per each target component into a new two-dimensional coordinate system. Before showing how this algorithm works, it is important to assert that the PCA is actually considered as a dimensionality-reduction technique, but from a different perspective of the aforementioned feature-selection methods. These latter methods in 6.3 are actually developed to select an optimal subset from the available features that are statistically relevant for classification purposes. The PCA creates a new set of M principal components by linearly combining the uncorrelated available HFs and therefore it define M new features from these combinations that, in this particular classification case, do not represent any physical variable. The main purpose of the PCA algorithm is to minimize the averaged distance between each datapoint and the new defined coordinates. In this case, the PCA has been used for visualization purposes: the first two components ($K = 2$) are therefore used to project the M_{HF_s} dimensional space over a bidirectional space, and therefore visualize the classifiers decision boundaries.

There are two equivalent definition of Principal Component Analysis (PCA):

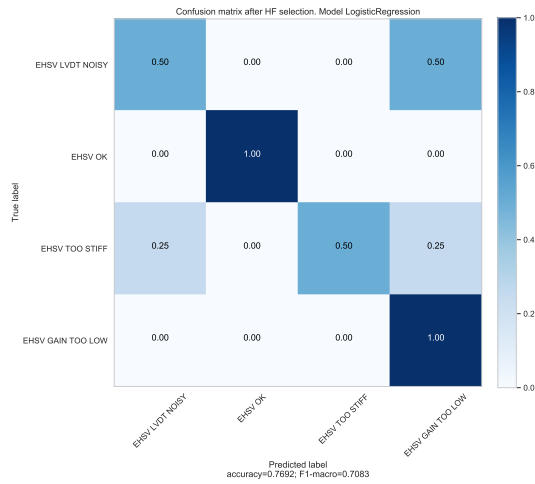
1. This algorithm aims to find the orthogonal projection of the data into a lower dimensional space, defined by D orthogonal *principal components*, in order to maximize the projected datapoints variance over the *principal components* directions.
2. Alternatively, the PCA is defined as the linear projection that minimize the average squared distance between the data and their projections.

Both the definitions are correct, and in [58] both of them are mathematically explained. The explaining plot in Fig. 6.34 shows how the data points would be better described if projected into the bidimensional space defined by the green and red *principal components*, maximizing the variance along both the new dimensions according to the first definition. Mathematically, considering the dataset $\{\mathbf{x}_n\}_{n=1,\dots,N}$ with N points and M defining features, the algorithm goal is to project each point into a lower dimensionality space $K < M$ while maximizing the variance. Considering the one-dimensional projection $K = 1$, each point can be projected taking as reference the unit vector of the \mathbf{u}_1 with M dimension, so that each point is projected onto a scalar value $\mathbf{u}_1^T \mathbf{x}_n$. Along this dimension, it is possible to determine the mean of the projected data $\mathbf{u}_1^T \bar{\mathbf{x}}$ and variance as:

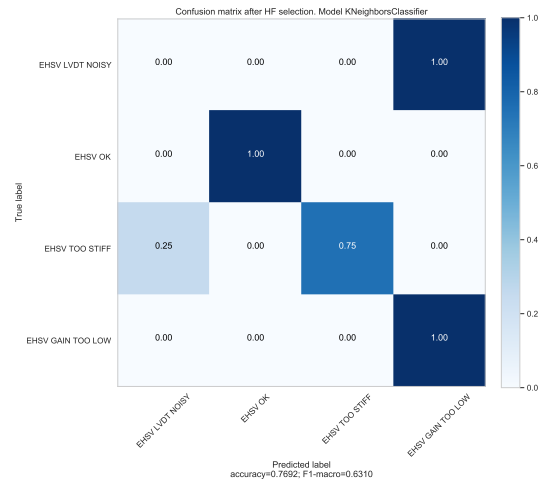
$$\frac{1}{N} \sum_{n=1}^N \{\mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \bar{\mathbf{x}}\}^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \quad (6.39)$$

where \mathbf{S} represents the covariance matrix:

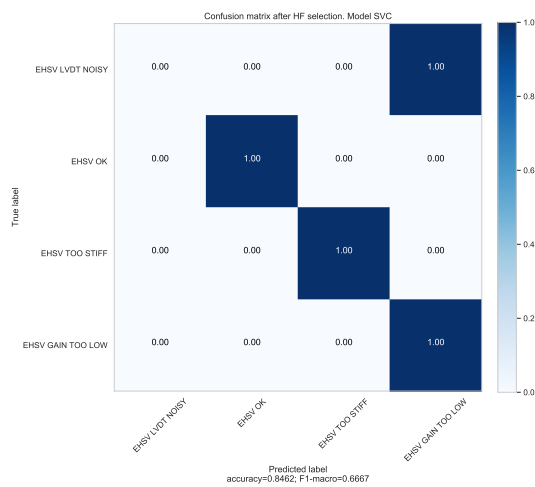
$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T \quad (6.40)$$



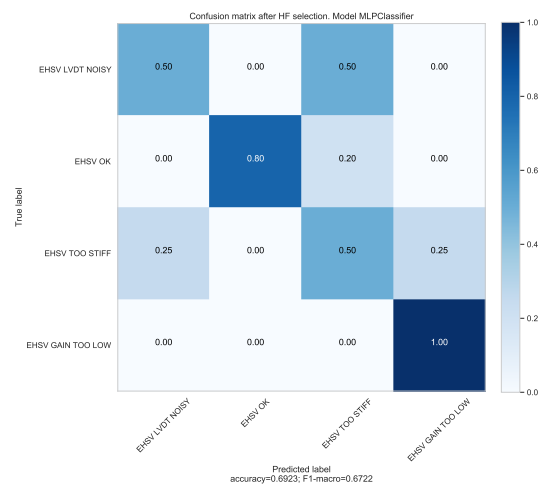
(a) EHSV LogReg Confusion Matrix



(b) EHSV KNN Confusion Matrix

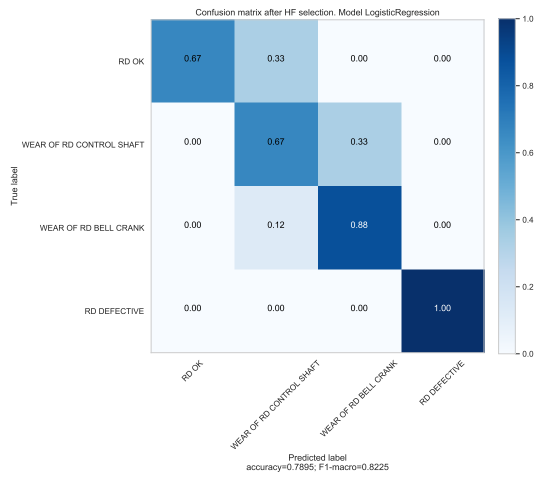


(c) EHSV SVC Confusion Matrix

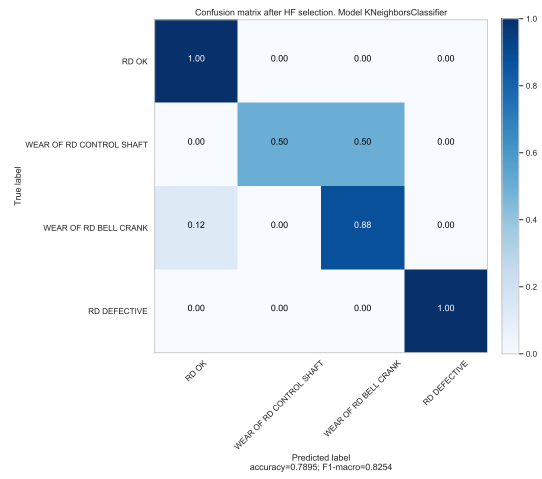


(d) EHSV MLP Confusion Matrix

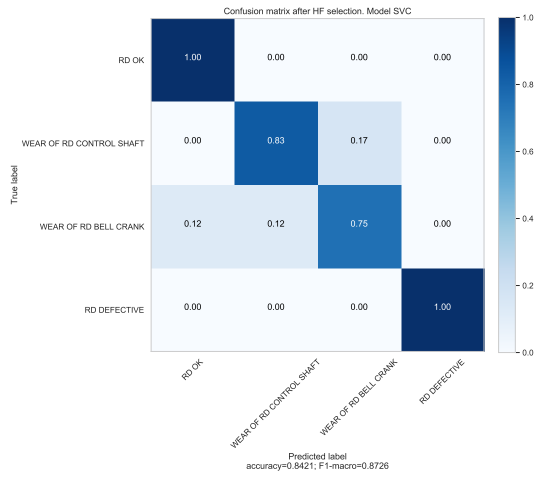
Figure 6.31: EHSV target component: Classifiers Confusion Matrices



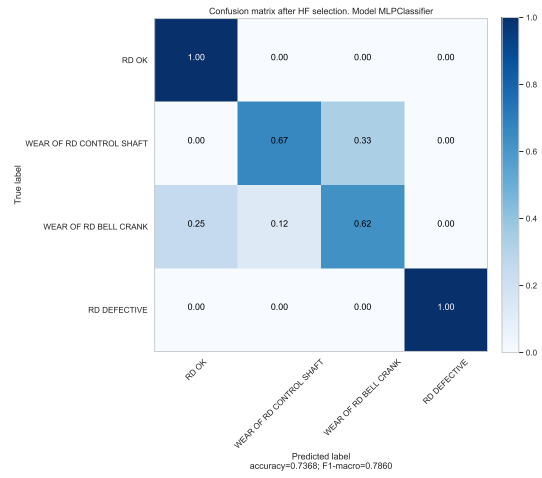
(a) RD LogReg Confusion Matrix



(b) RD KNN Confusion Matrix

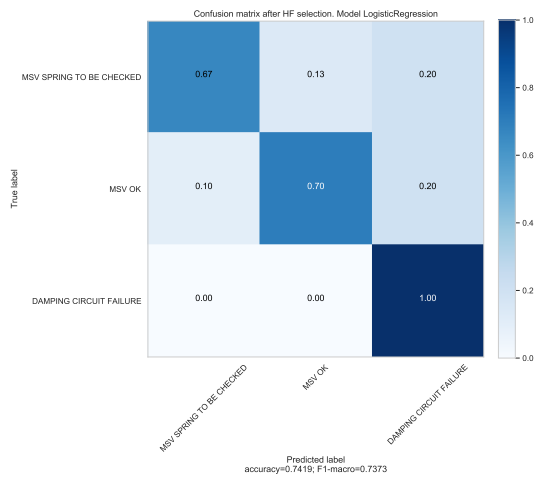


(c) RD SVC Confusion Matrix

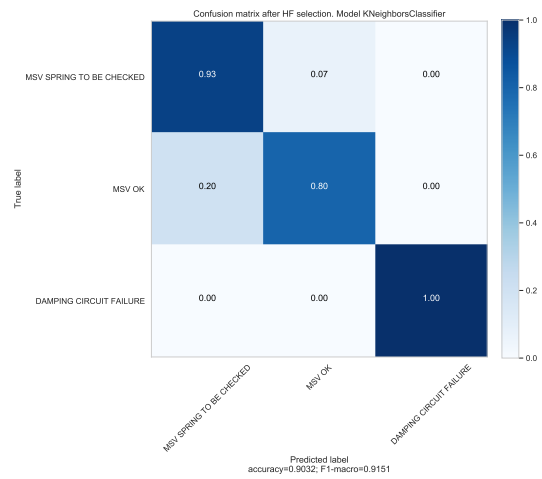


(d) RD MLP Confusion Matrix

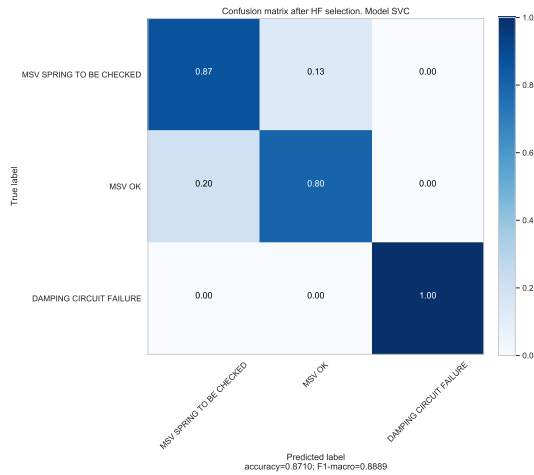
Figure 6.32: RD target component: Classifiers Confusion Matrices



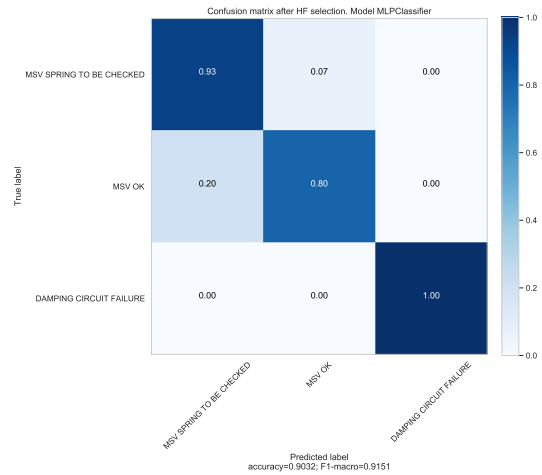
(a) MSV LogReg Confusion Matrix



(b) MSV KNN Confusion Matrix



(c) MSV SVC Confusion Matrix



(d) MSV MLP Confusion Matrix

Figure 6.33: MSV target component: Classifiers Confusion Matrices

The goal is to obtain the correct direction of \mathbf{u}_1 that maximize the projected variance $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$. The Lagrange multiplier λ_1 is introduced to correct the second term in Eq. 6.39 as follow:

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1) \quad (6.41)$$

By deriving the Eq. 6.41 with respect to \mathbf{u}_1 in order to find the maximum, it is possible to assert that the unit vector \mathbf{u}_1 needs to be an eigenvector of the covariance matrix \mathbf{S} :

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \rightarrow \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \quad (6.42)$$

From Eq. 6.42, it is possible to assert that the variance is maximized when the \mathbf{u}_1 is equal to the eigenvector with the largest eigenvalue λ_1 . This eigenvector is therefore the first principal component and these considerations can be extended to all the possible $K - 1$ orthogonal directions defined by ascending sorting the \mathbf{S} matrix eigenvalues.

As already mentioned, this technique has been adopted in this work to reduce project each component dataset into a two-dimensional equivalent space, in order to allow a clearer visualization of the decision boundaries each tuned classifier define. Each component dataframe, defined by each set of selected HFs from the ANOVA

2-dimensional dataset with principal components

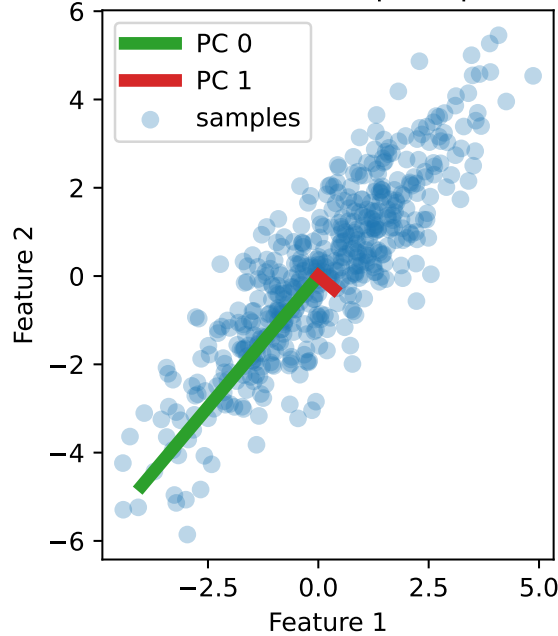


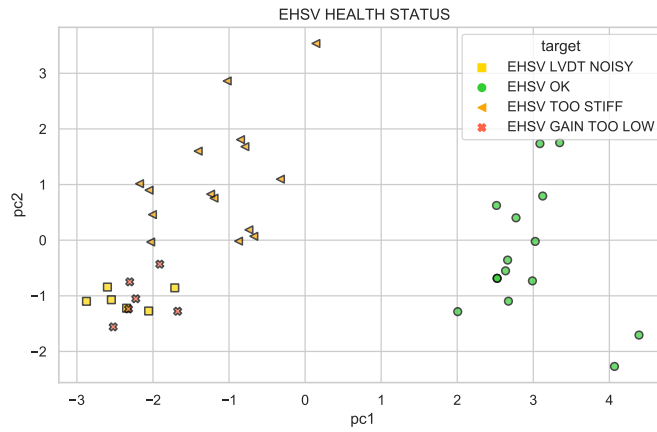
Figure 6.34: Two Principal component over example dataframe [91]

F-score method, has been processed using the `sklearn.decomposition.PCA` function, in order to extract the first two principal components. The scatter plots from Fig. 6.35a to 6.35c illustrates how the failure classes points are distributed in the principal components *pc1* and *pc2* space. Looking at these plots, it is already possible to intuitively trace decision boundaries that clearly separate the serviceable components (labelled as "OK" in green) with the faulty ones. Within the wider group of degraded components, typically coloured from yellow to red, it is in some cases possible to distinguish separable groups, like for the RD component in Fig. 6.35b, while in other cases the different groups are too overlapped, like for the EHSV case in Fig. 6.35a. However, these results are actually coherent with the classifier performances in the previous confusion matrices. For instance, the recurrent misclassification between the "EHSV Lvdt Noisy" and "EHSV Gain too low" is also visible in this two-dimension projection where yellow and red data points are overlapping in the same region. In case of the RD, the separation between classes is clearer than the previous case: the "RD Defective" class, which gather the cases of units that are not recentering at all, is way further than the other classes; while all the units with wearing phenomena in the control shaft or bell crank are gathered together in separable but still adjacent groups.

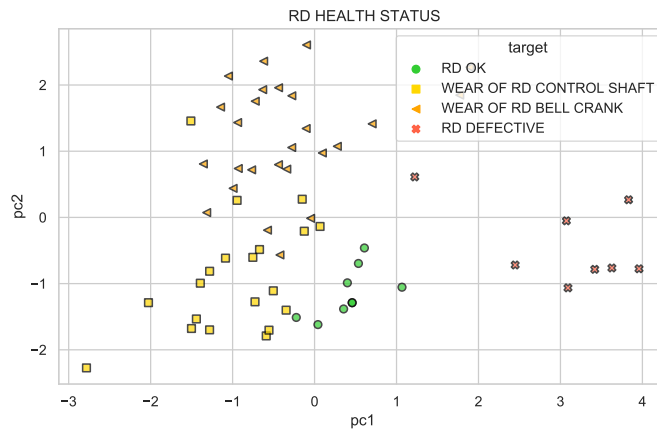
Looking at the points distributions of Fig. 6.35a, 6.35b and 6.35c, it is simple to distinguish separable areas by human eye and therefore it is important to find within the candidates classifiers the best solution that can replicate such class separation in the best way possible. For this reason, each tuned candidate classifier has been fit over a training set that has been built by picking the same datapoints of the previous classifier performance analysis. Since the goal is to visualize the decision boundaries between the different failure label classes, the classifiers have not been tuned again over the new reduced principal components dataframe.

The plots in Fig. 6.36a, 6.36b and 6.36c illustrates the decision boundaries the candidates tuned classifier are able to trace over the projected data points in the principal components space. The different coloured areas denotes the areas in the projected space where the classifier is able to discern a specific class. The borders between these areas are the identified decision boundaries and circled points represents the testing dataframe elements.

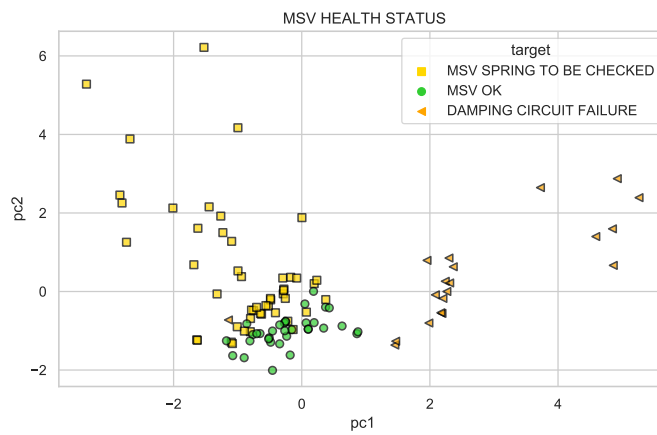
Looking at these plots, it is possible to observe how the projection in the two-dimensional principal components space can give significant insights about the data distribution, however the tuned classifier do not always perform better in case of low-dimensional space. For instance, in the new principal component space, the Logistic Regressor is not able to distinguish between the aforementioned overlapping classes in the EHSV classification problem in Fig. 6.36a, while the KNN is affected by overfitting when discerning between red and yellow points. Analogous consideration can be made when observing the Logistic Regressor decision boundaries in the second subplot in Fig 6.36b, which is not able to define a larger decision area for the "RD Ok" class. In this case, any misclassification of serviceable components,



(a) EHSV data points distribution after PCA

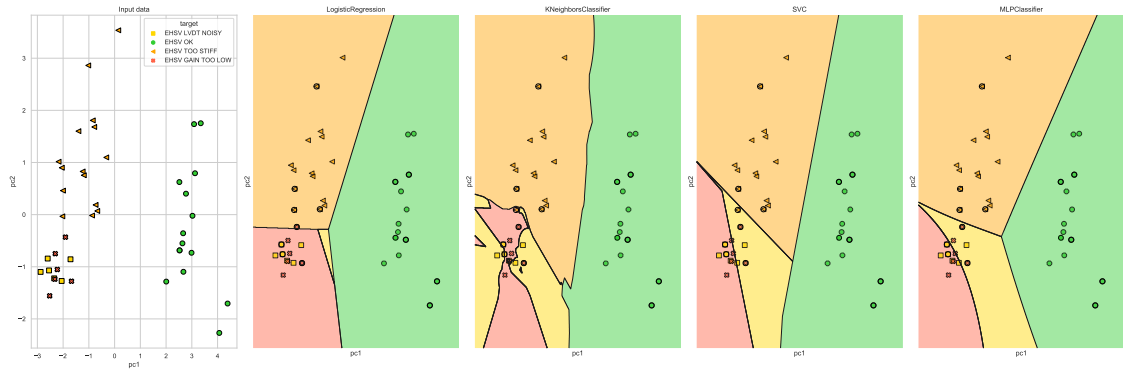


(b) RD data points distribution after PCA

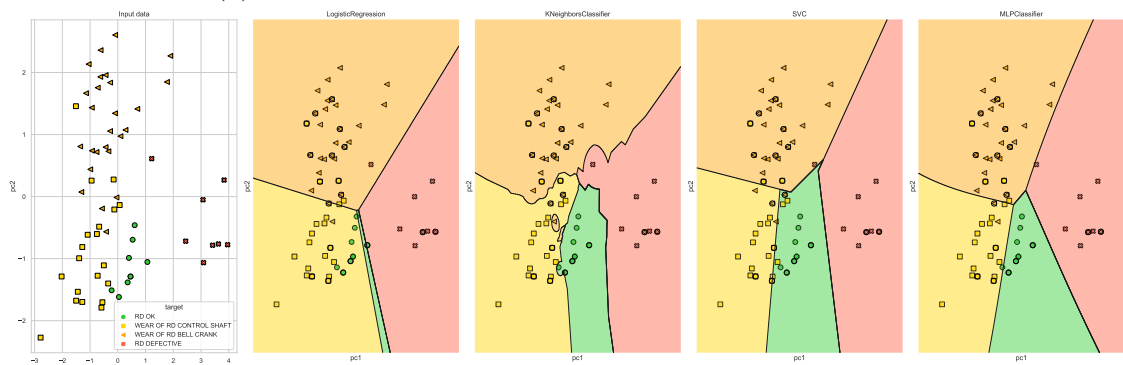


(c) MSV data points distribution after PCA

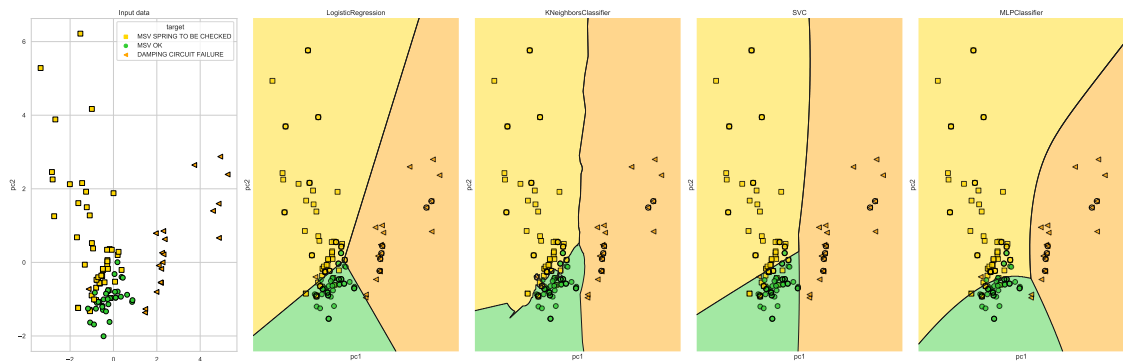
Figure 6.35: Data points distribution per each target component



(a) Classifiers decision boundaries for EHSV dataframe



(b) Classifiers decision boundaries for RD dataframe



(c) Classifiers decision boundaries for MSV dataframe

Figure 6.36: Classifier decision boundaries after PCA

generally labelled as "Ok" with green dots, has to be penalized more than any misleading classification in the other failure classes. A well-traced decision boundaries are instead traced by the SVM with linear kernel and the MLP classifier for all the target components problems.

Chapter 7

Conclusion

The presented Ph.D. research project aims to demonstrate the feasibility and explain all the required necessary steps to develop an advanced diagnostic system for FBW flight control systems. The described approach has been developed and tested in several aspects on the A320 elevator, aileron and spoiler servoactuators. However, this methodology has been designed in order to be scaled and adaptable to other FBW units of other aircraft. This aspect is also strategical from a market perspective, since the hydraulic technology for primary FBW systems will be still extensively used in the next decades fleet. Due to the crucial safety role of these components and the high volume of repairing events, the standard maintenance procedures need to be improved in terms of testing automation, data collection and effective automatic failure identification. These three key enablers have been in detail described in the previous chapters while, in this last one, a comprehensive summary is reported of both strengths and flaws of each of them. Furthermore, the guidelines for future project development are suggested in the last paragraph.

7.1 Summary

The main goal of the presented research project, in collaboration with Lufthansa Technik, was to design and implement novel testing methodologies that aim to improve and automate diagnostic tasks for electro-hydraulic FBW systems. In this work, the A320 primary flight control systems have been considered as project use-cases, however the explained methods and techniques can be adapted per other FBW systems of different aircraft. During the three-years HyDiag Project, four main milestones have been reached and they are hereby summarized in paragraphs 7.1.1, 7.1.2, 7.1.3 and 7.1.4

7.1.1 Automatic and Scalable testing procedure

The first significant achievement is represented by the described testing procedure reported in Chapter 4. After an extensive study of the physical characteristics of the target FBW systems (described in Chapter 2) and their testing procedures, it has been noticed how the project-use cases units share several structural components and therefore also testing procedures for maintenance purposes. In this framework, an automatic test has been designed in order to obtain a modular testing protocol, built upon parametric command sequences that are concatenated in order to create a unique and tailored excitation signal. These signals have actually been designed with the multiple goals to reduce the testing time but at the same time to cover most the prescribed tests in the standard procedure and add new possible signals which target specific unit components. These multiple goals have been all successfully achieved thanks to the constant improvements of the signal *Sequences* from the preliminary design phase, to the middle prototyping test in the H-Lab research lab and the final implementation in the LHT repairing shops (both described in Chapter 3). More specifically, the following achievements in this field are worthy to be mentioned:

- The designed signals covers more than 75% of the standard testing procedures, and additionally they includes some specific "extra-CMM" testing signals which are helpful to investigate the health status of specific target components.
- The HyDiag Signals have been fully implemented in the LHT shops test-rig and their effectiveness have been proved over more than 100 FBW units tested with this method. The testing time in this final form have been drastically reduced from 90' of the standard manual procedure up to maximum 30' for the longest Elevator signal. It is important also to mention that these signals have been tailored according to the feedbacks received directly from expert LHT operators, which have given a valid contribution to adapt the HyDiag Signals to the LHT test-bench. In case of testing facilities upgrade, the HyDiag test can be shorten even to 20': several tests in this perspective have been conducted in the research H-Lab.
- The HyDiag Sequences, the building blocks of the testing signals, are automatically developed by a unique MatlabTM generation code and therefore they are easy to be adapted and parametrized. Each Sequence shape is uniquely defined in terms of type of command to move the main ram (closed or open loop), supply pressure signal and the eventual signals to the Electrovalves for switching the unit working modes.

7.1.2 Automatic Health-Features extraction from experimental data

After completing the automatic test, it is fundamental to analyse the collected results in order to detect particular signal features that are characterizing for any anomalous behaviour of the unit component. This task involves a preliminary phase of definition of these so called Health Features, starting from the study of each unit CMM and the physical interaction between components, and the implementation of an automatic way to extract them and properly store. This phase has been in detail described in Chapter 5, where the HyDiag software structure and fundamentals algorithms have been described. This particular tool has been fundamental for the complete development of the HyDiag protocol itself, since it is considered a central node for both the test phase and the following data collection and analysis phase.

The signal post-processing software, developed in MatlabTM, has been implemented in two versions: the first one with a GUI interface for off-line data processing, and a second one that has been tested and debugged for an on-line signal processing in the LHT shops test-rig. For this latter case, the setting phase of the testing computer and measurement channels has required significant effort and collaboration with other colleagues from the partner company. In both the versions, the software includes specific algorithms of signal processing that include automatic recognition of signal change-points, peaks finding and frequency or spectral analysis. These techniques allow to automatically extract the HFs from the collected signals: these features includes both the traditional ones from the standard procedure, but more importantly some additional ones that target specific critical components. For instance, the *Sequences* n.13 and n.26 of the Elevator signal have been designed specifically to extract the health conditions of both RD and EHSV. At the end of the signal processing phase, these set of HFs are then collected and reported in a unique HyDiag Report. This document is also printed in two different versions: a ready-to-read version directly delivered to the operator, with a complete set of troubleshooting instructions, and a tabular version for data-analytics purposes. The whole collected reports in this latter format lay the foundation for the further data-collection framework and classifier implementation with ML techniques.

7.1.3 Data collection of experimental results and test reports

Another important achievement of this research project regards the design of a robust data-collection protocol for both test measurements and extracted results from the HyDiag software. This whole infrastructure has been described in paragraphs 5.6 and 6.1. In particular, in the first one the whole setting of data and

reports folders have been described. These mentioned "HyDiag Data" and "HyDiag reports" folders are automatically populated by the aforementioned HyDiag software at the end of each test. Their coherent division in subfolders, identified with uniquely defined labels, gives a great advantage in terms of traceability of a particular performed test and allows the user to easily retrieve data about a certain unit during a consecutive test. The "HyDiag Data" folder contains all the extracted measurement channels from each signal Sequence, and it represents a valuable source of information that have been largely used during the phases of development and debug of the HyDiag software. The "HyDiag reports" folder instead represents the core source of information at the base of the so-called HyDiag database, described in 6.1. As shown in Fig. 6.2 this database contains both information of the performed tests and each extracted HFs, but also important information coming from the repairing shops. In this way, it is possible to have a complete overview about the tested component in a unique environment.

7.1.4 Failure recognition through ML algorithms

The HyDiag database represents the principal source of information for training and testing the core algorithm of the Classifier in the Diagnostic module in Fig. 6.1. The concept is to implement an intelligent algorithm, based on Machine Learning technique, that is able to identify any failure within the tested unit and assert the health-status of each part. In this work, the main steps for training, testing and evaluate the performance of four different Supervised ML algorithms candidates have been presented. Firstly, the supervised strategy has been preferred over the un-supervised technique since it allows the chosen algorithm to "replicate" the knowledge of the user, which has to assign specific labels based on a-priori knowledge or experimental observations. For instance, in paragraph 6.2 it has been described the procedure to assign specific failure labels, based on observed degradations over dismantled components (f.e. RD and MSV) and physical hypothesis (f.e. EHSV). Through the correct assignment of these labels, based on the characterizing HFs of each target component, it is possible to train any SML to replicate these results over a large set of data. In this framework, four candidates algorithms have been considered as core function of the Failure classifier module. In order to perform an effective failure recognition through ML techniques, the following steps have been defined and developed:

- Data preprocessing and failure labels assignment through dedicated Python functions
- Features selection with ANOVA F-score method for dimensionality reduction. This step results particularly important firstly to avoid the so-called "curse of dimensionality" problem and extract the most statistically significant HFs that ensures a good separability of the defined classes, through the assigned

failure labels. This method has been preferred over any other technique (f.e wrapper methods) since it is a valid technique that allows to detect significant features without any preliminary information of the SML to be adopted. Therefore, as first approach to the problem, it has been considered a suitable solution.

- Classifiers parameters and hyper-parameters tuning through cross-validation technique. Each of the candidates algorithms is characterized by a set of parameters and solver settings that need to be correctly tuned in order to maximize the classifier performances. This tuning process has been implemented through Cross-Validation techniques over the training portion of the whole dataset. This method is broadly recognized as a suitable technique to avoid overfitting problems when testing the chosen classifier. In this work, per each candidate, ranges of parameters and performance metrics to optimize have been defined and the Cross-Validation tuning process developed through dedicated Python function.
- The tuned classifier performance have been tested considering the whole set of selected HFs. In order to evaluate the performance, per each target component, the multi-class confusion matrices have been reported that show the correctly labelled and misclassified results per each defined class.
- In order to visualize the decision boundaries that each classifier is able to define, the whole dataset has been projected in a two dimensional space using the Principal Components method. Even if this technique is widely used also for dimensionality reduction purposes, in this case it has been initially considered as valid tool to visualize the class separability and how each classifier works when defining the proper decision boundaries for classification. The last paragraph in Chapter 6 reports both the results for multi-dimensional problem and the ones obtained after the PCA.

7.2 Future Development

The presented work presents several margins of improvement in different fields, which are actually dependent from each other. In order to have a deeper understanding of the possible improvements, it is better to evaluate how to obtain a better failure recognition through ML algorithms. The key-point of the future developments is basically centred over the amount of experimental data which are collected through the HyDiag Sequence. The more data are collected, the better the failure labels assignment can be. As previously mentioned in Chapter 6.3, the defined Failure labels, assigned at each tested unit, are strictly dependent on the number of measurements: due to the structural complexity of each unit

and the huge amount of possible failures, it cannot be excluded to encounter new possible anomalies as long as the experimental campaign is extended. Therefore, even the assignment criteria can be updated and improved by looking directly at the time-responses from each collected measurement channels, and very likely add other target-component that have not been considered (such as the main ram, the electrovalves etc.). This latter point brings the improvement analysis to another observation over the extracted HFs: in the future, there could be the necessity to extract new unseen HFs and discard some defined ones. This consideration influence both the structure of the HyDiag Software and also the shape of the signal, that could need new defined Sequences or different parametrization of the existing ones. Any change or improvement in these two cardinal stages of the test procedure are all related to the number of collected data. Other possible improvements regard surely the tuning and choose of the correct SML to use as Classifier and relative Feature selection technique. In this work, as first approach to the problem, four promising SML algorithms have been compared and a Filtered method for feature selection has been tested. However, increasing the number of datapoints and consequently the complexity of the classification problem, it could be possible that other SML techniques (such as Decision Trees or Random Forest Classifier) may need to be evaluated with other feature selection techniques (such as the wrapper-ones). About the classification problem, in Chapter 6 it has been presented as a simple multi-class one where all the classes are mutually exclusive. Due to the physical complexity of the problem, it is difficult to think about isolated failures that do not influence each others. Therefore, a possible improvement of the classification problem may regard the class-influence definition or a multi-label approach.

Once these improvement have been tested and implemented, and a final testing-pipeline defined, with the correct testing signal, extracted features and fine-tuned classifier, the last improvement regards the collection of all these information in a new version of the presented HyDiag report. This report could then include an health-status map of each unit component, the experimental results, and smart troubleshooting instructions that includes both mandatory indications about broken components to replace and recommendation about failures in early stages. This latter form would be the final recap of all the informations coming both from the classifier module, and from the previous "history" of the tested components. These historical information have been considered in this work for different analysis (as reported in Appendix B), however they need to be considered properly to enrich the set of information per each tested unit.

Appendix A

Synchronous Demodulation

A.1 Synchronous Demodulation - Integral Formulation

As previously described in 2.6, the AC output of a 4-wires LVDT U_{ab} needs to be demodulated in order to extract both magnitude and direction of the linked component movement. The synchronous demodulator returns "...the magnitude of the DC component proportional to the amplitude of the fundamental frequency component of the periodic signal to be detected" [125]. This DC component is evaluated at each instant of time, and its sign indicates the direction of the LVDT magnetic core movement. In this section, an integral formulation of the expected output from the demodulator is presented. Due to the periodicity of both primary and secondary AC voltages, the results are always presented during the first period T of the sinus waves. Through an integral formulation, it is possible to compare how close the time-series outputs are with their reference scalar values. The presented results include both the effects of a phase-shift ϕ between primary and secondary and noisy-harmonics with their own random amplitudes and phases.

A.1.1 Nominal conditions: no phase-shift between signals

Let consider the following expressions for the secondary U_{ab} and primary U_p voltages:

$$U_{ab} = A_s \sin(2\pi ft) \quad (\text{A.1})$$

$$U_p = A_p \sin(2\pi ft) \quad (\text{A.2})$$

where A_s and A_p are their relative amplitudes, f is the supply frequency of the primary voltage. Due to the mutual inductance, this frequency characterizes also the secondary voltage shape. Since their periodicity, let consider the first period from 0 to T in order to calculate the average value of the rectified signal.

$$U_{rect} = \frac{1}{T} \int_0^T (\text{sign}(U_p)) U_{ab} dt = \frac{1}{T} \left(\int_0^{T/2} U_{ab} dt - \int_{T/2}^T U_{ab} dt \right) \quad (\text{A.3})$$

Substituting the equation A.1 in A.3:

$$U_{avg} = \frac{2}{T} \left(\int_0^{T/2} U_{ab} dt \right) = \frac{2}{\pi} A_s \quad (\text{A.4})$$

Eq. A.4 represents the expected amplitude of the filtered rectified signal. This scalar value is proportional to the secondary amplitude with a factor of $\frac{2}{\pi}$. From this result A.4, the rms value of the secondary coil can be easily evaluated:

$$U_{demod} = U_{rms} = \frac{A_s}{\sqrt{2}} = \frac{\pi}{2\sqrt{2}} U_{avg} \quad (\text{A.5})$$

Figure A.1 shows the comparison between the output from a digital synchronous demodulator with the results of equations A.5:

- Red line: primary AC voltage of equation A.2 with $A_p = 10V$ and $f = 1976Hz$;
- Green-dashed line: raw secondary AC voltage of equation A.1 with $A_s = 7V$ and same frequency of the primary one;
- Green line: rectified secondary voltage according to the sign of the primary voltage;
- Black-dashed line: nominal mean rectified value evaluated according to equation A.4;
- Blue line: nominal rms value calculated from equation A.5. Both this scalar value and the previous one have been reported per each instant of time;
- Violet line: output of the synchronous demodulator.

The adopted filter in the digital demodulator is a second order zero-phase lag Butterworth filter, with a cut off frequency $f_{cut} = 200Hz$, which is almost 1/10 of the carrier one. This filter has ensured high accuracy in the final estimation of the demodulated voltage.

In particular, Fig. A.2 reports how the filter order influence the percentage error between nominal integral value and filtered one. In particular, the curves refers to the mean voltage obtained as output from the filter. The second column values in Fig A.2 are calculated as reported in eq. A.6:

$$err = 100 * \frac{U_{avgintegral} - U_{avgfiltered}}{U_{avgintegral}} \quad (\text{A.6})$$

As it is possible to observe from both time-series and table in Fig. A.2, the second-order filter ensures the smallest percentage error.

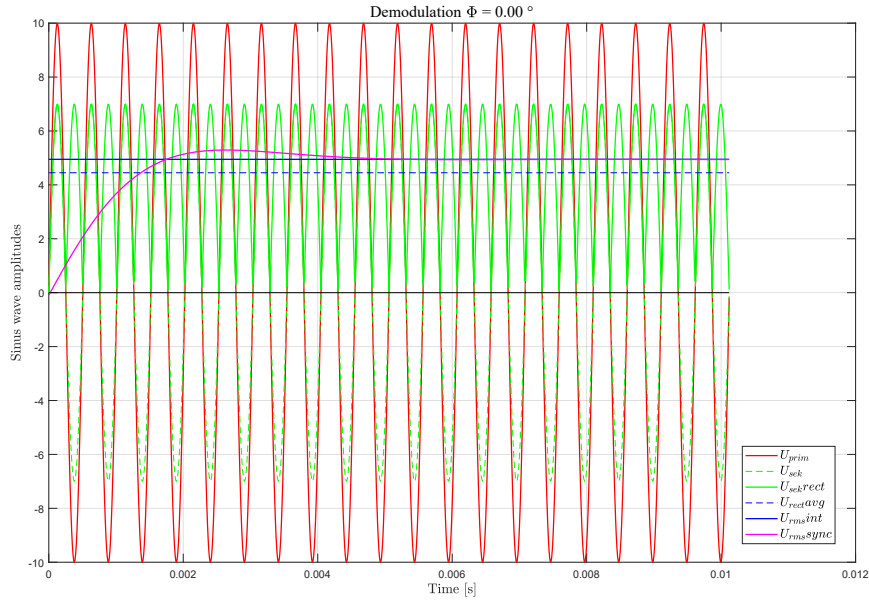
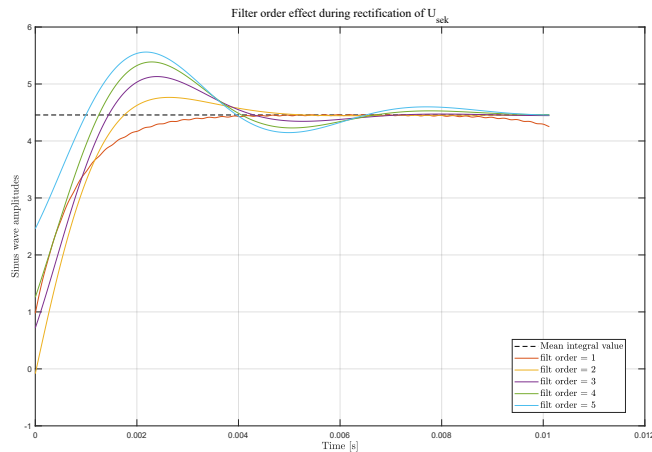


Figure A.1: Synchronous Demodulation ($\phi = 0$)



Filter order	% error
1	0.65
2	0.15
3	0.67
4	2.15
5	2.43

Figure A.2: Effect of filter order on Synchronous Demodulator

A.1.2 Phase-shift between signals

The results in paragraph A.1.1 are evaluated in ideal nominal condition with no phase-shift between primary and secondary voltage. In this case, it has been proved that the synchronous demodulator returns almost the exact RMS value from the integral expression, with an error of 0.15%. However, as already reported in 2.6, this technique is particularly affected by a phase-shift between primary and secondary voltages, in particular during the rectification step. It is possible to

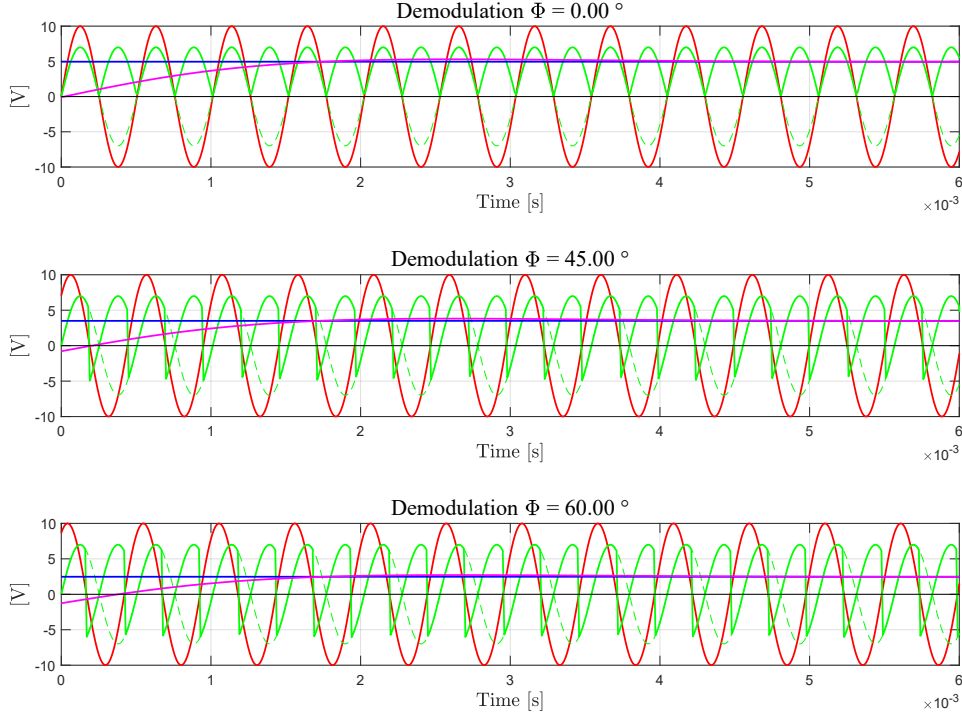


Figure A.3: Synchronous Demodulation with variable phase-shift

quantify this effect by solving the equation A.4 considering a certain angle $\phi > 0$:

$$U_{rect|\phi} = \frac{2}{T} \left(\int_0^{T/2} A_s \sin(2\pi ft + \phi) dt \right) = \frac{2}{\pi} A_s \cos(\phi) \quad (\text{A.7})$$

From this equation A.7, the rms value extracted by the demodulator is lower than the nominal by a factor of $\cos(\phi)$:

$$U_{rms|\phi} = U_{rms|\phi=0} \cos(\phi) \quad (\text{A.8})$$

Figure A.3 shows how the rms value decreases due to the phase-shift ϕ . The second and third subplots illustrates how the rectified signal (continuous green line) presents periodic counter-sign peaks which contribute to underestimate the final rms value.

A.1.3 Noise effect

All the previous considerations in paragraphs A.1.1 and A.1.2 are based considering a perfect sinus-wave as secondary LVDT coil input. In reality, due to the kind of 4-wires connection, the shape of this input may be affected by external noise. In this case, it can be demonstrated that the synchronous demodulator response is

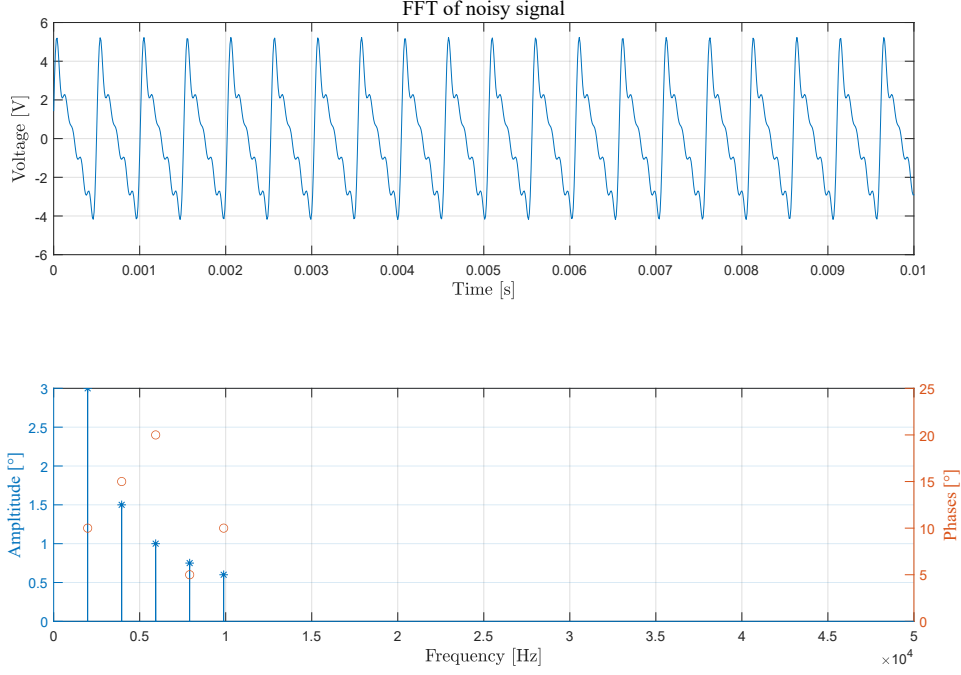


Figure A.4: Noisy signal and its frequency spectrum

influenced by both amplitude and phase of each odd-order harmonics. Let consider a noisy periodic signal which can be written, using Fourier series decomposition as:

$$U_{ab} = A_{s0} + \sum_{n=1}^N A_{sn} \sin(2\pi nft + \phi_n) \quad (\text{A.9})$$

Let exclude any steady part of the signal ($A_{s0} = 0$) and let truncate the equation A.9 at the third harmonic $n = 1, 2, 3$:

$$U_{ab} = A_{s1} \sin(2\pi ft + \phi_1) + A_{s2} \sin(4\pi ft + \phi_2) + A_{s3} \sin(6\pi ft + \phi_3) \quad (\text{A.10})$$

The rectification of the signal in eq. A.10 includes both the effects of the first and third harmonic and their relative phase. These contributes can be calculated by solving the integral in eq. A.11:

$$U_{rect} = \frac{2}{T} \left(\int_0^{T/2} U_{ab}(t) dt \right) = \frac{2}{\pi} \left(A_{s1} \cos(\phi_1) + \frac{A_{s3}}{3} \cos(\phi_3) \right) \quad (\text{A.11})$$

The result in eq. A.11 shows that the even-order harmonic effect has been cancelled during the rectification, while all the other harmonics influence the rectified

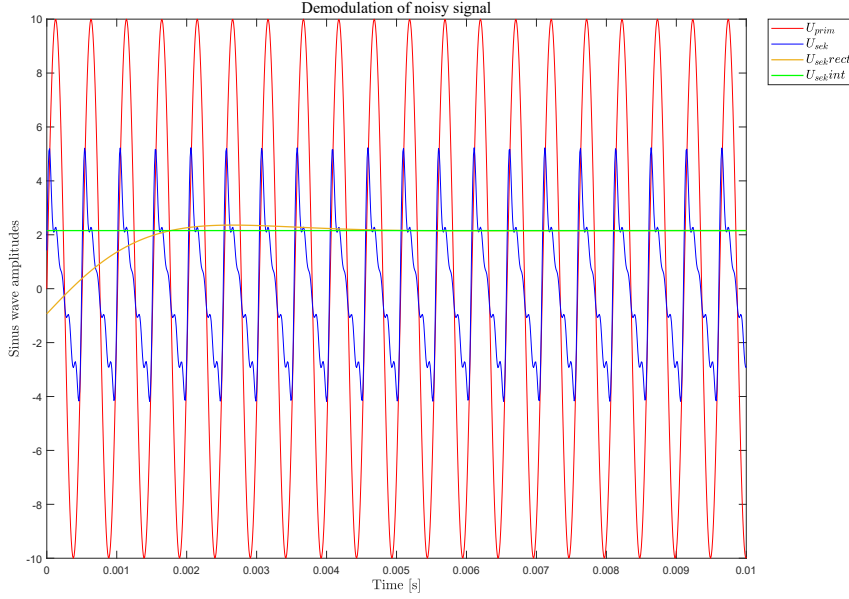


Figure A.5: Rectification of noisy signal

amplitude with both their phases (as in eq. A.3) and amplitudes, which are scaled with a factor of $1/n$. This result can be generalized as follow:

$$U_{rect} = \frac{2}{\pi} \left(\sum_{n=1}^{N_{odd}} \frac{A_{sn}}{n} \cos(\phi_n) \right) \quad (\text{A.12})$$

with n is the odd-harmonic order index. As the previous analysis in eq. A.5, the output rms value from the synchronous demodulator is obtained multiplying the result A.12 with the gain $\frac{\pi}{2\sqrt{2}}$. In order to validate the eq. A.12, the results from the integral formulation and from a digital synchronous demodulator have been compared. The input signal has been built considering five different harmonics with their own amplitude and phase-shift. In Fig. A.4 both the raw signal and its relative frequency spectrum have been reported. In particular, the second subplot shows both amplitudes and phases of each harmonic. Fig. A.5 reports the result of both rectification. The red and the blue lines represents the raw periodic primary U_{prim} and secondary U_{sek} signals. The green and yellow lines instead shows the integral result of eq. A.12 and the output of the synchronous demodulator. The overlapping of these two curves demonstrates the accuracy of expected integral formulation A.12 with the demodulator output.

The results from eq. A.12 can be generalized for all the harmonics in the frequency spectrum which are characterized by their proper amplitude, frequency and phase shift. Beside the contribute of the n odd-order, the other k harmonics with a non-integer order in the Fourier series A.9 influence the demodulation. In

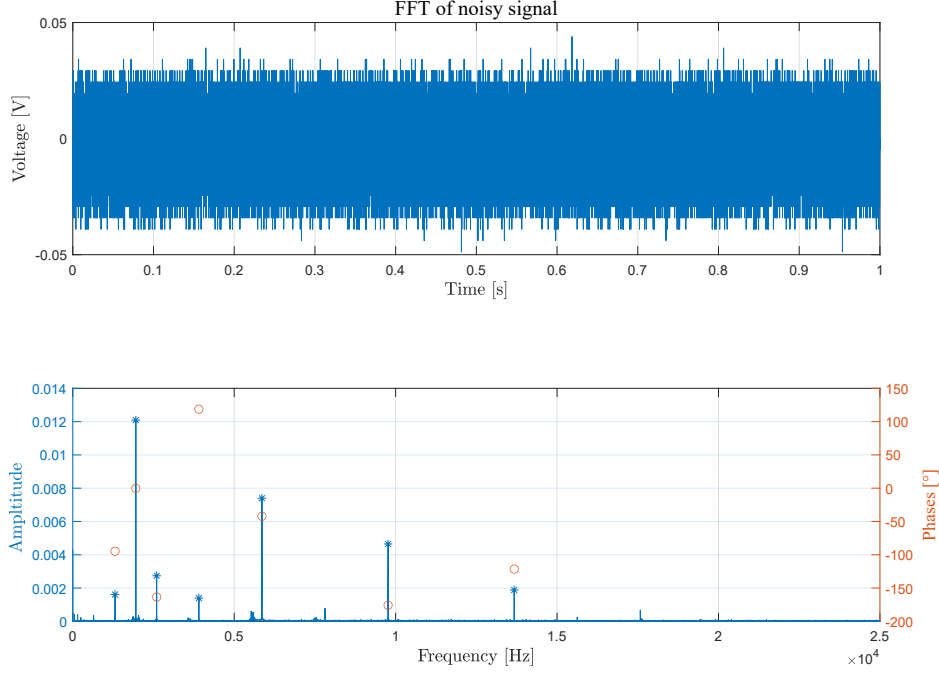


Figure A.6: 4-wires secondary LVDT coil signal and its frequency spectrum

fact, referring to each of these harmonics with its frequency f_k , Amplitude A_{sk} and phase ϕ_k , the noise-effect is expressed expanding the eq. A.12 as follow:

$$\begin{aligned}
 U_{rect} = & \frac{2}{\pi} \left(\sum_{n=1}^{N_{odd}} \frac{A_{sn}}{n} \cos(\phi_n) \right) + & (A.13) \\
 & + f_1 \sum_{k \in R-n}^{\infty} \frac{A_{sk}}{\pi f_k} \left(1 - \cos \left(\frac{\pi f_k}{f_1} \right) \right) \left[\sin \phi_k \cdot \sin \left(\frac{\pi f_k}{f_1} \right) - \cos \phi_k \cdot \sin \left(\frac{\pi f_k}{f_1} \right) \right]
 \end{aligned}$$

where f_1 is the frequency of the first principal harmonic.

In order to validate the results in eq. A.13, a real noisy signal from a 4-wires LVDT has been considered, whose time-series and frequency spectrum are reported in Fig.A.6. In particular, this signal is a small portion of the EHSV Elevator LVDT in its null position. The signal has been sampled with a $f = 50kHz$, and the digital synchronous demodulator parameters are still the one reported in the previous paragraph A.1.1.

Comparing the frequency spectrum in Fig. A.4 with the one in Fig. A.6, it is evident that the latter presents n-integer odd harmonics at multiple frequency of f_1 , but also it is filled with other k harmonics, especially around the area of the first one. All these harmonics are not discarded during both rectification and

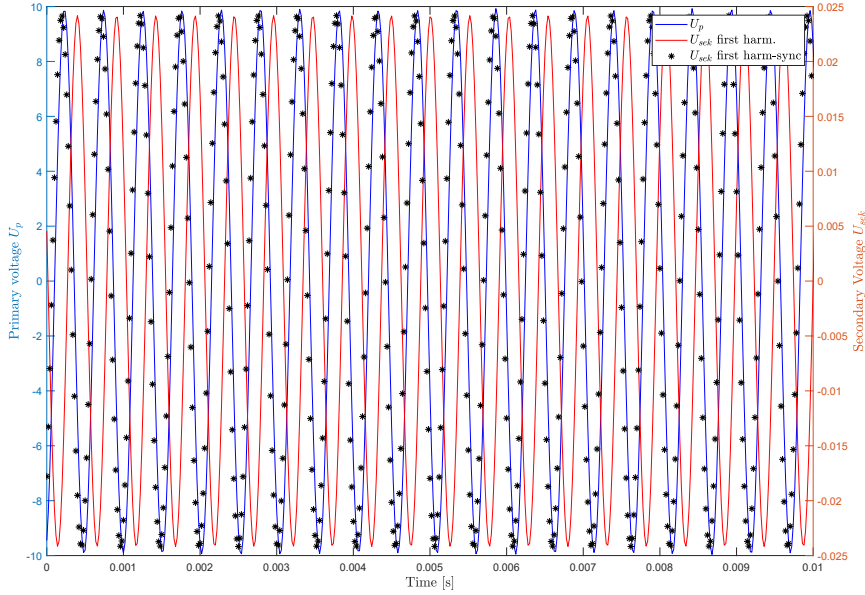


Figure A.7: Alignment of first harmonic of U_{sek} signal with U_p

filtering steps, but they need to be included in the second term of the sum in eq. A.13. The integral formulation need to be compared with the output from the developed digital synchronous demodulator, which need to be slightly modified in order to eliminate the phase shift between primary coil voltage U_p and the first carrier frequency harmonic of the secondary one. This preprocessing step results fundamental in order to have a correct evaluation of the other harmonics phases. In this way, re-aligning the secondary signal with the primary one, the first harmonic contribute in eq. A.13 takes into account a correct phase ϕ_1 value, without any slicing error of the recorded signals. The plot in Fig. A.7 shows the results of this alignment process between U_p and the first harmonic, at the same carrier frequency of the primary voltage, of the noisy signal U_{sek} . Using cross-correlation as in [126], it has been possible to eliminate the phase-shift between the reference blue line and red one, which represent the first harmonic of the frequency spectrum in Fig. A.6. The black-dashed line represents the final result of this preprocessing step. Once evaluated, the equivalent time-lag between signals, the raw vector has been aligned accordingly.

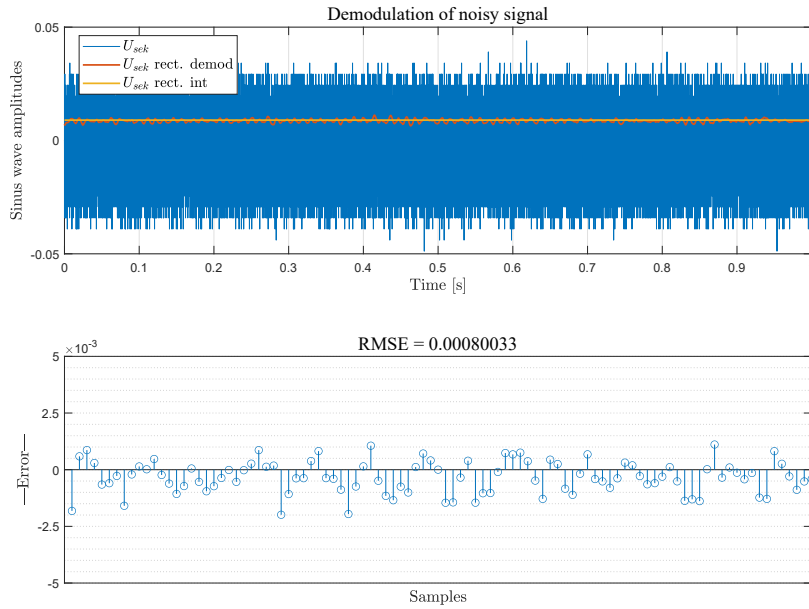
The first subplot of Fig. A.8a shows three main lines:

- Blue one: preprocessed noisy raw signal from the secondary coil. This time-series has been preprocessed and aligned to the primary voltage;
- Red line: output from the digital synchronous demodulator;
- Yellow line: result of the integral formula A.13.

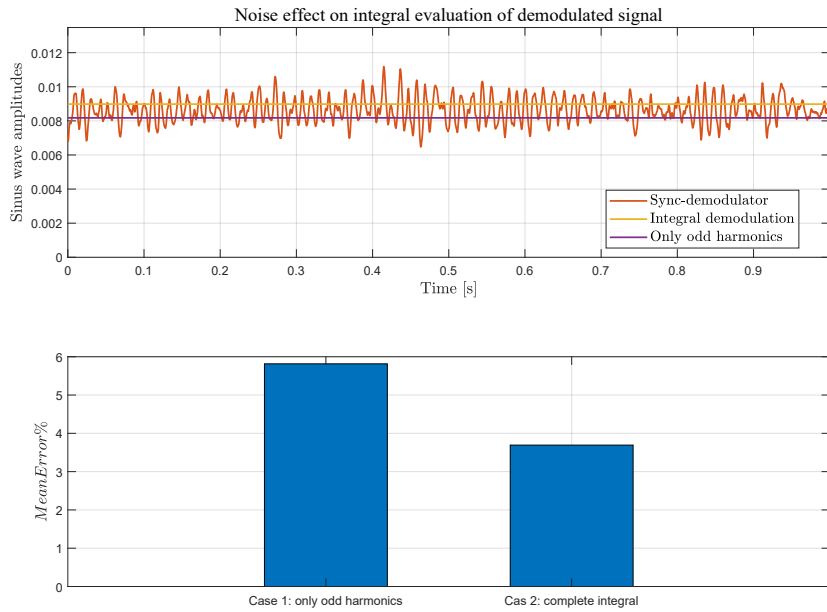
The comparison between these two last lines show how robust the integral formulation is compared to the real time-series. The second subplot reports the absolute value at each sample between red and yellow line, while in its title the Root Mean Squared Error is reported:

$$RMSE = \sqrt{\sum_{i=1}^N \frac{(\hat{x}_i - x_i)^2}{N}} \quad (\text{A.14})$$

This metric is typically used to measure the error between a predecided value and a quantitative measured one. In this case, it quantifies how spread the observed demodulator output is from the expected integral one: the smaller it is, the more the variables are correlated. The second subplot in A.8a reports the absolute error every n sampled points. Imposing a threshold to the average absolute or rms error, can be considered an health indicator which helps to identify a possible fault in the LVDT itself or in the connected spool. The last plot in Fig. A.8b shows the difference between the results of equations A.12 and A.13 in case of noisy signal. In case of high level of noise, using just the contribute of all the odd-order harmonics can estimation error of about 6%, which decreases around 3.5% when considering also the noisy harmonics effect.



(a) 4-wires secondary LVDT coil demodulation



(b) Integral demodulation comparison: with or without the noise effect

Appendix B

Health Features limits reduction

In paragraph 5.6 it has been mentioned how the HyDiag reports has been designed in order to include information from both the Diagnostic Module and results from the analysis of previously collected data. This latter aspect implies the reports results from the standard CMM testing procedure. This Appendix reports the results from the identification process of possible fixing sweet-spots in the CMM limits. The idea behind this analysis was born from a recurrent observation of the standard test reports. In fact, it has been noticed how some units which are initially labeled as No Fault Found (NFF) , since they have passed all the CMM tests, may return in the LHT shop after few Flight Hours (FH) for further testing and fixing. This inconvenience can be avoided defining acceptability sub-ranges within the suggested CMM limits ranges. These latter intervals are essential to re-certify the units, however they lead to binary labelling of each test (PASSED or FAILED) which seems to be not enough. For this reason, the identification of possible sweet-spot within the suggested CMM test ranges can be useful to identify how close each HF is to the limits, and if it falls within a "risk-area" where the interested sub-component may need a further investigation. This kind of analysis has been conducted comparing the NFF results with the recertified units ones, labelled as "Ausgang" test. Paragraph B.1 explains the Historical database structure, the next one B.2 illustrates the methods used for preprocessing and clean the collected data, while the last B.3 shows the results and sweet-spot identification.

B.1 Historical Database

The Historical Database represents a useful source of information since it collects all the reports information from 1996 of all the tested units, including the use-cases PFCS. As reported in [31], it has been built mining different set of datas from different databases, as shown in Fig. B.2. Six different LHT databases are queried to extract not only the CMM HFs reports, but other important set of data such as

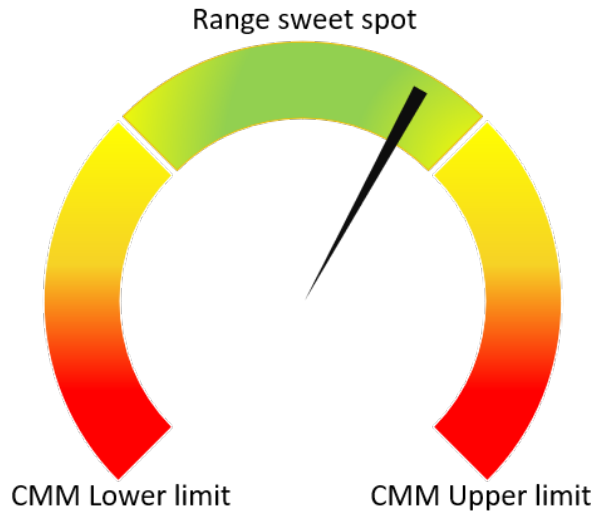


Figure B.1: Sweet spot identification

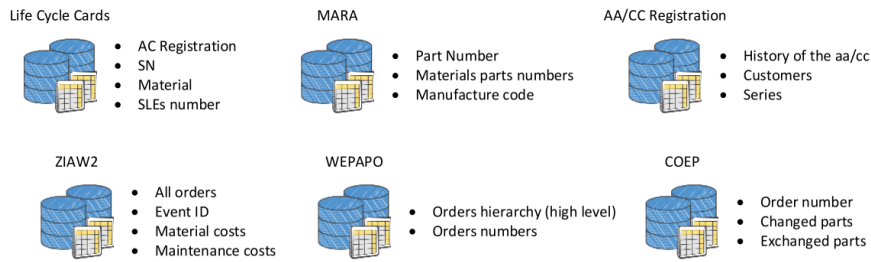


Figure B.2: Source databases for the Historical one [31]

the material quantity, Time Since Installation (TSI) , cost and exchanged parts.

Joining all these information, the final Historical Database is exported in *.xls* format for an easier processing. The latest version used for this analysis includes all the recorded test from September 1995 to January 2020. This collection of data has been already analyzed in previous works. In [31], the author has demonstrated how the average TSI, which is used to estimate the time range between two consecutive SLE, drastically decreases after the first repairing event. Furthermore, the author has given an important contribution during the identification of the most critical components, evaluating per each PN which were the most exchanged parts in the past. This analysis has supported the design process of the HyDiag testing procedure, designed in order to evaluate in particular the health status of these parts. Another useful study has been conducted in [33], where the HyDiag reports have been compared with the traditional one, highlighting both the coherence of the extraction methods and the impact the automatic HF's extraction makes in terms

of avoiding any reporting mistake from the operator.

In this Appendix, the focus is related to the HF's distribution between NFF units and recertified ones, in order to identify any optimal adjustment range within the suggested CMM one.

B.2 Data cleansing and pre-processing

Before starting, a preliminary cleansing and preprocessing of the Historical Database data are necessary to extract the useful results for the sweet-spot identification. First of all, the database file in its *.xls* format is converted in *.csv*, then it is processed by two *Python* scripts which need to be executed consecutively to obtain the final result:

1. `db_build_histodb_fix.py`: it performs all the necessary fixing over the table columns.
2. `hist_db_cmm_shrink.py`: it takes as input the fixed output of the previous script and executes the sweet-spot range identification just for NFF and recertified units.

B.2.1 Data cleansing with `db_build_histodb_fix.py`

It may happen the information stored in the Historical Database might be incomplete, due to the mining and joining process from different data sources or to human error during reports compiling. The goal of `db_build_histodb_fix.py` is to improve the quality of this database and preprocess the data for further CMM limits analysis. The high-level flowchart in Fig. B.3 illustrates the strategy used in `db_build_histodb_fix.py`. Firstly, the code loads and imports the latest database version in *.csv* format, stored in the proper folder, as Pandas dataframe [`reback2020pandas`]. This Python library allows the user to query the dataframe, process the data row and column-wise and merge the information between two or more dataframes. All these features are used by the `db_build_histodb_fix.py` to improve the quality of the collected data.

The dataframe contains 39 columns with several information about the tested unit, such as the HF's test results, removal and testing dates, exchanged parts, SLE index and TSI in hours and days. To improve the processing process, it is necessary to assign to each the proper data type. For example, the flowchart shows the subprocess of converting the PN and SN columns into strings, the dates related columns into date-type, SLE index in integer number and the `numbervalue` column into float. This latter one is particularly important since it includes the CMM HF's numeric value extracted during the standard tests. Another important column which needs to be fixed if data are missing is the TSI one. This parameter is a good estimator

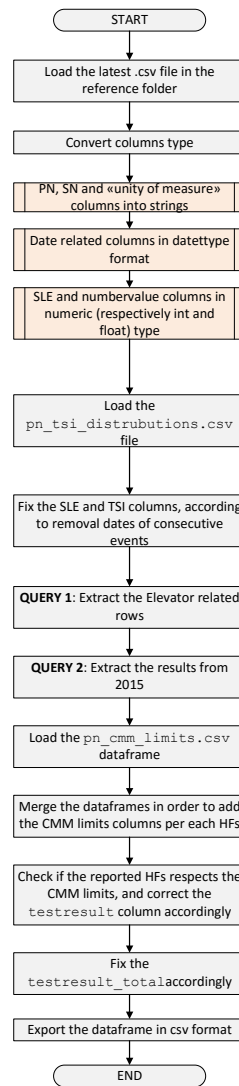


Figure B.3: Simplified flowchart for data cleaning

of the tested unit life, since it is calculated between two consecutive SLEs. Any missing data in this column is a lack of knowledge of the quality of the previous SLE.

In order to fix this eventual missing information, the script checks per each tested unit (identified with a unique label `pn_sn`) how many SLE the unit has experienced and if there is any missing value in the TSI column. It may happen to have units which presents just the first SLE, because they are still installed

CMM Test N.	HyDiag test name	HyDiag sequence	CMM test name	HDb Test N.	HF's name	U.o.m	Lower limit	Upper limit	Flag HyDiag
-------------	------------------	-----------------	---------------	-------------	-----------	-------	-------------	-------------	-------------

Table B.1: `pn_cmm_limits.csv` structure

on board and did not receive any fixing in the LHT shops yet. In this case, if the TSI information is missing, the code replaces the relative empty TSI cell with a conventional value stored in the previously loaded `pn_tsi_distribution.csv` dataframe. This short table collects, per each PN, the descriptive statistic features of the TSI column per each SLE, excluding NaN values. Using the Pandas function `describe`, it is possible to extract per each SLE the mean and maximum values and the first, second and third quartiles. In particular, the second one is used as reference value to replace in case the TSI cell of the first SLE is empty. If the missing TSI value is referred to a following SLE (second, third etc.), its value is calculated using as reference the removal date of the previous one. Actually this method is also used if the TSI value is anomalously low, for example less than 38 FH. This reference value has been estimated taking as reference an average 2000 FH per year. After fixing the TSI column values, the dataframe can be queried in order to extract the necessary results for the CMM limits evaluation. For this analysis, only the Elevator reports after a reference date (e.g. 01/01/2015) have been extracted from the whole dataframe. Once this sub-dataframe has been extracted, in correspondence to each CMM HF's, the suggested limits values are inserted merging the obtained dataframe with the reference `pn_cmm_limits.csv` one. This file has been compiled manually and it is loaded also by the HyDiag software for report compiling.

This dataframe in Table B.1 contains the following columns:

1. CMM Test N.: with this field, three columns have been grouped. Together they form the correct CMM test enumeration composed by Test (N), Sub-test (M) and Measurements numbers (K).
2. HyDiag test name: this field contains the CMM test name which is then reported in the csv version of the HyDiag report. The name structure is build from the CMM Test Numbers columns as `cmm_N_M_K`.
3. HyDiag Sequence: this column contains the HyDiag Sequence number where the relative CMM test is included.
4. CMM Test name: it contains the original CMM test name.
5. HDb Test N.: together with CMM Test N., it allows to link the correct CMM enumeration with the one reported in the Historical Database. It has been indeed observed that the aforementioned CMM test numbers N, M and K often differ from the one contained in the Historical Database.

6. HFs name: HFs name (e.g: "Sm" is the total speed in extended stroke).
7. U.o.m: HFs unity of measure.
8. Lower limit: CMM HFs lower limit. If the CMM does not report any lower limit, a default value of 0 is imposed by default
9. Upper limit: CMM HFs upper limit.
10. Flag HyDiag: binary column value: 1 if the CMM test is included in the HyDiag procedure, and 0 otherwise.

The merge between this dataframe and the Historical data one is made on the CMM Test N. and HDb Test N. columns. In this way, the bigger Historical Database is enriched with information regarding the HyDiag procedure but more importantly it includes also the Lower and Upper limits columns per each HFs. These two columns are used in the script to perform an additional check over the reported HFs. In fact, the last two process in Fig. B.3 refer to both fixing the `testresult` and `testresult_total` columns. The first one contains binary values: 1 if the relative CMM test has passed and 0 otherwise. The second column instead reports the same binary value considering the whole HFs per that unit: the whole measurement is labelled as Failed (therefore 0) if at least one of the collected test is failed as well. This additional test over the reported results ensure to correctly identify both the NFF units, which have been tested just once and they have success in all the CMM tests, and the re-certified units, which needed a previous fixing or replacement of faulty components. These kind of units are the main actors of the further analysis of sweet-spot identification within the CMM ranges. The last task performed by the script is the export of the new fixed dataframe in a new `.csv` file.

B.3 Sweet-spots identification in limits range

Once the `db_buld_histodb_fix.py` has exported the final fixed dataframe in `.csv` format, it can be used to identify the HFs sweet-spots within the suggested CMM ranges. This task is performed by the Python script `histo_db_cmm_shrink.py`. As previously anticipated, the main actors of these study are both the NFF units, labelled as "EA" (Eingang-Ausgang) units, and the certified ones labeled as "A" (Asugang). All the units which belong to these categories have one common characteristic: they have passed all the tests. The EA units represents the units which have been shipped in the LHT shop and they have passed all the CMM tests during the first trial. The A units are instead the ones which have been certified after being tested and repaired. However, it is possible that this test-label (EA or A) is wrongly reported in the Historical Database, due to one or more misreported failed test. Vice versa, it is possible that some test procedures have been labelled as Entry

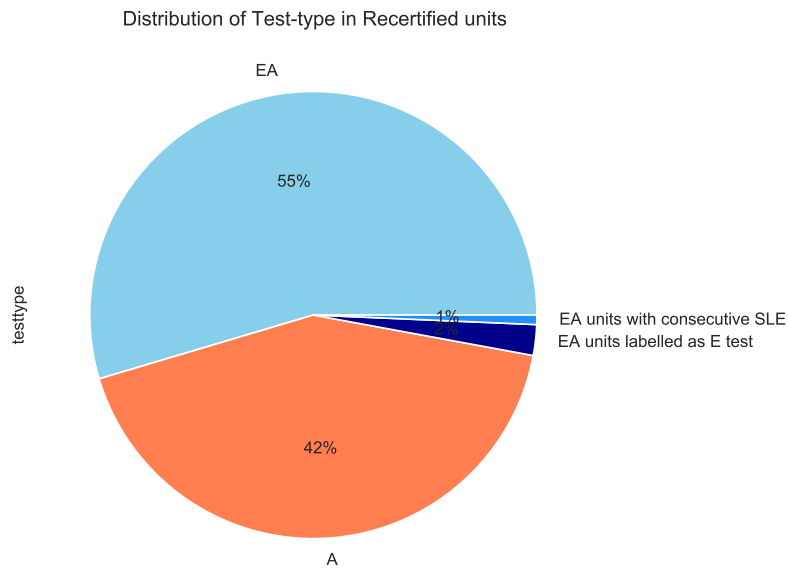


Figure B.4: NFF and Recertified units

Test (E) but they actually should have been marked as EA because they do not present any failed units. In order to avoid these possibilities, the first part of the `histo_db_cmm_shrink.py` present these pre-processing phases. From the updated `testresult_total` column, the code checks if the units labelled as passed are also marked as EA or A. If a NFF unit has been wrongly marked as E, then the code includes it in the EA ones. With the same approach, all the units with at least one failed test and wrongly labelled as NFF or certified are discarded by this analysis. Therefore, only rows of the corrected marked EA and A units are selected.

Another interesting observation regards all the NFF units that actually presents a consecutive SLE event after few FHs. These particular units need to be marked and in detail analyzed, since they actually represent a missed chance of repairing. This particular group of units could have presented an incipient fault in one or more sub-components which have not been correctly identified by the CMM tests. This inconvenience can be avoided by the identification of sweet-spots and risky area within the CMM ranges, discouraging a binary classification of each test.

The pie-plot in Fig. B.4 shows the percentages of units which have passed all the CMM tests, therefore labelled as NFF (EA) tests or Recertified (A) test after repairing. It is possible to notice how actually the NFF units are 55% of this group. This considerable percentage can be explained considering the standard protocol followed when the units are disembarked from the aircraft. As reported

in [17], the NFF rate can be particularly high especially in case of redundant on-board architecture, such as for the Elevator. If the FCC detects a failure in one of the two Elevator on one side of the horizontal stabilizer, the protocol recommend to disembark both the units for an additional check. This procedure ensures high safety level, however it drastically increases the NFF ratio, since it is very likely that at least one of the disembarked unit does not present any failure. The percentage of the wrongly labelled NFF units which actually should have been labelled as Entry tested (E) is about 2% of the overall population. Almost 1% of NFF units present a consecutive SLE event after few HFs. Even if this class percentage content is the smallest, it has driven the authors to focus on the recertified and repaired units (A tests) for the identification of possible sweet-spots in the CMM ranges. This conservative approach is also driven by the fact that actually the background of the repaired units can be easily tracked and the exchanged or repaired components information are collected within the Historical Database. Taking as reference the Ausgang tests results, the distributions of the CMM HFs included in HyDiag have been analyzed. In particular, the sweet-spots identification in the CMM ranges has been conducted comparing the IQR scores of the HFs distributions for both EA and A tests. The IQR Score represents the interquartile range and it measures the statistical dispersion of the data:

$$IQR = Q3 - Q1 \tag{B.1}$$

where Q3 and Q1 are the upper and lower quartiles. The HFs with $IQR_A < IQR_{EA}$ has been chosen for the sweet-spot identification, penalizing the NFF units with wider dispersion around the median value. According to this criteria, it has been possible to identify the sweet-spot within the CMM ranges for 15 CMM HFs which are also included in the HyDiag procedure. Fig. B.5 and B.6 illustrate the results of this analysis over two representative HFs: IE is the nominal bias current value in extended position, and SR' represents the recentering speed from totally extended stroke. As already mentioned in Chapters 5 and 6, these HFs have been used for the RD health status detection, so it becomes particularly useful to evaluate the best possible range of adjustment for these two. Therefore, they have been selected and presented in this work to demonstrate the results of this analysis.

The first subplot represents the swarm-plot and boxplot graphs for both HFs distributions for EA and A tests. This kind of graph is widely used in statistics to visually represent the data distributions and each quartile [127]. The amplitude of the central box represents the IQR score. The values have been normalized taking as reference the maximum value of the range, therefore the right-limit of the CMM range. Both the nominal limits are signed with continuous horizontal red lines in the first subplot and accordingly with vertical red lines in the second subplot. The green dashed lines represents the IQR score limits of the recertified units, and they mark the beginning and the end of the sweet-spot range. In the lower histogram, all the bars within this range are coloured in green. The upper and lower whiskers in

the box-plot are signed with vertical black dashed lines in the lower histogram. The upper whisker is evaluated as the greatest value in the distribution which is smaller than $Q3 + 1.5IQR$ while, in the same way, the lower whisker is estimated as the smallest distribution value which is greater than $Q1 - 1.5IQR$. These values delimit the yellow areas in the lower histogram which can be considered as acceptable ranges but with higher risk than the best central spot. The values between the whiskers values and the CMM limits have to be considered with an higher level of risk or may even represent an incipient fault in the tested component.

The validation of this CMM range partition method is showed directly by the two possible class of units identified in the first subplot swarmplot. Each point represent a NFF or recertified unit, and the red dots in the EA distribution represents the NFF units which have been lately tested for further repair (the 1% of unit in the pieplot B.4). It is possible to observe how actually most of these dots are outside the optimal range identified by the green-dashed line, therefore they represent a missed opportunity of further investigation since a possible incipient fault has not been correctly detected.

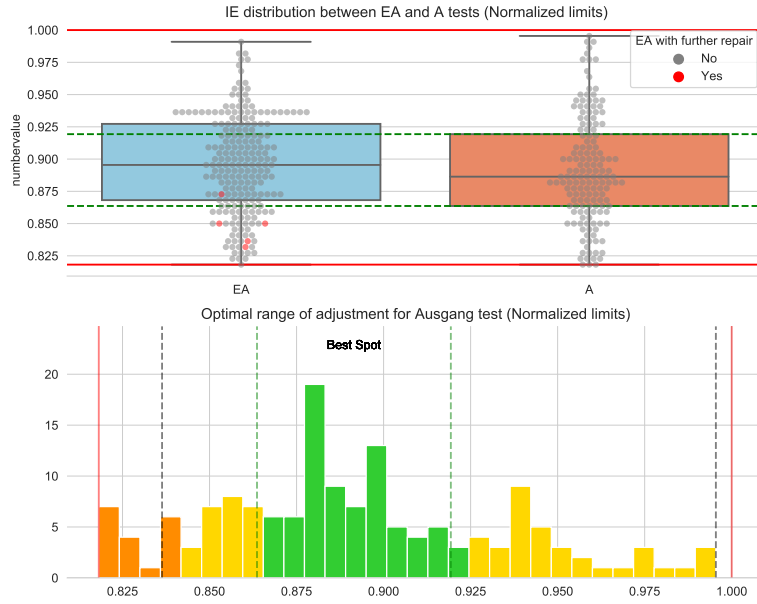


Figure B.5: IE sweet-spot range identification

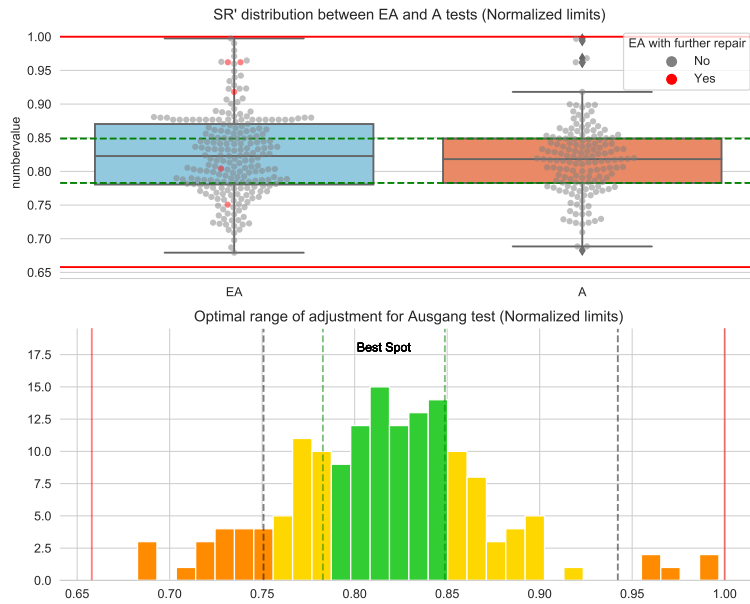


Figure B.6: IE sweet-spot range identification

Nomenclature

Acronyms / Abbreviations

ANN Artificial Neural Network

APU Auxiliary Power Unit

ATA Air Transport Association

CAGR Compound Annual Growth Rate

CBM Condition-Based Maintenance

CFDS Centralized Fault Display System

CM Condition Monitored Maintenance

CMM Component Maintenance Manual

CV Cross-Validation

DV Display Variable

EHSV Electro-Hydraulic Servovalve

EIS Electronic Instrument System

ELAC Elevator and Aileron Computers

EV Electro-Valve

FAA U.S. Federal Aviation Administration

FAC Flight Augmentation Computer

FCC Flight Control Computer

FCDC Flight Control Data Concentrators

FCS Flight Control Systems

<i>FH</i>	Flight Hours
<i>FMEA</i>	Failure Mode & Effects Analysis
<i>FN</i>	False Negative
<i>FP</i>	False Positive
<i>HF_s</i>	Health Features
<i>HP</i>	High Pressure
<i>HT</i>	Hard Time action
<i>IBIT</i>	Initiated Built-In Test
<i>ISO</i>	International Standard Organization
<i>KNN</i>	K-Nearest Neighbors
<i>LHT</i>	Lufthansa Technik
<i>LHT</i>	Lufthansa Technik
<i>LP</i>	Low Pressure
<i>LPF</i>	Low Pass Filter
<i>LVDT</i>	Linear Variable Differential Transformer
<i>MLP</i>	Multilayer Perceptron
<i>MRO</i>	Maintenance Repair and Overhaul
<i>MRO</i>	Maintenance Repairing and Overhaul
<i>MSG</i>	Maintenance Steering Group
<i>MSV</i>	Mode Switching Valve
<i>NFF</i>	No Fault Found
<i>NFF</i>	No Fault Found
<i>OC</i>	On Condition action
<i>OEM</i>	Original Equipment Manufacturer
<i>OEM</i>	Original Equipment Manufacturer

<i>PAC</i>	Pressurization Air Conditioning
<i>PCA</i>	Principal Component Analysis
<i>PFCS</i>	Primary Flight Control Systems
<i>PHM</i>	Prognostic and Health Management
<i>PSD</i>	Power Spectral Density
<i>PTU</i>	Power Transfer Unit
<i>PV</i>	Process Variable
<i>RAT</i>	Ram Air Turbine
<i>RCM</i>	Reliability-Centered Maintenance
<i>RD</i>	Recentering Device
<i>RMSE</i>	Root Mean Squared Error
<i>RUL</i>	Remaining Useful Life
<i>RVDT</i>	Rotary Variable Differential Transformer
<i>SEC</i>	Spoiler and Elevator Computer
<i>SFCC</i>	Slats Flaps Control Computer
<i>SFCS</i>	Secondary Flight Control Systems
<i>SML</i>	Supervised Machine Learning
<i>SVM</i>	Support Vector Machines
<i>THS</i>	Trimmer Horizontal Stabilizer
<i>TM</i>	Torque Motor
<i>TN</i>	True Negative
<i>TP</i>	True Positive
<i>TPR</i>	True Positive Rate
<i>TSI</i>	Time Since Installation
<i>TST</i>	Time Since Installation

Nomenclature

UUT Unit Under Test

UUT Unit Under Test

VC Virtual Channels

Bibliography

- [1] F. C. C. Goncalves and L. G. Trabasso. “Aircraft preventive maintenance data evaluation applied in integrated product development process”. In: *Journal of Aerospace Technology and Management* 10 (2018), pp. 1–15. ISSN: 21759146. DOI: 10.5028/jatm.v10.706.
- [2] A. Mornacchi. “Design and Development of Prognostic and Health Management System for Fly-by-Wire Primary Flight Control Electrohydraulic Servoactuators”. Ph.D. degree in Mechanical Engineering. Politecnico di Torino, 2016.
- [3] C. Spreafico, D. Russo, and C. Rizzi. “A state-of-the-art review of FMECA including patents”. In: *Computer Science Review* 25 (2017), pp. 19–28.
- [4] J.F. Van Leeuwen et al. “Risk analysis by FMEA as an element of analytical validation”. In: *Journal of pharmaceutical and biomedical analysis* 50.5 (2009), pp. 1085–1087.
- [5] M. L. Chiozza and C. Ponzetti. “FMEA: a model for reducing medical errors”. In: *Clinica chimica acta* 404.1 (2009), pp. 75–78.
- [6] A. Mornacchi, G. Vachtsevanos, and G. Jacazio. “Prognostics and Health Management of an Electro-Hydraulic Servo Actuator”. In: *Annual Conference of the Prognostic and Health Management Society*. 2015, pp. 1–12.
- [7] A. De Martin, G. Jacazio, and M. Sorli. “Enhanced Particle Filter framework for improved prognosis of Electro-Mechanical flight controls Actuators”. In: *European Conference of the Prognostic and Health Management Society 2018*. Utrecht, The Netherlands, 2018, pp. 1–10.
- [8] S. Autin et al. “Feasibility Study of a PHM System for Electro-hydraulic Servo- actuators for Primary Flight Controls”. In: *Annual Conference of the Prognostic and Health Management Society*. 2018, pp. 1–19.
- [9] A. Nesci et al. “Detection and Prognosis of Propagating Faults in Flight Control Actuators for Helicopters”. In: *Aerospace* 7.3 (2020), p. 20. DOI: 10.3390/aerospace7030020.

- [10] J. Banks et al. “How engineers can conduct cost-benefit analysis for PHM systems”. In: *IEEE Aerospace and Electronic Systems Magazine* 24.3 (2009), pp. 22–30. ISSN: 08858985. DOI: 10.1109/MAES.2009.4811085.
- [11] J. P. Sprong, X. Jiang, and H. Polinder. “Deployment of Prognostics to Optimize Aircraft Maintenance-A Literature Review”. In: *Annual Conference of the Prognostic and Health Management Society*. Scottsdale, Arizona, USA, 2019, pp. 1–12.
- [12] G. Jacazio. *Prognostics and Health Management - Framework and Applications*. Turin, 2018.
- [13] A. Singh, N. Thakur, and A. Sharma. “A review of supervised machine learning algorithms”. In: *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*. Bharati Vidyapeeth, New Delhi as the Organizer of INDIACom - 2016, 2016, pp. 1310–1315. ISBN: 9789380544199.
- [14] F. Bateman, H. Noura, and M. Ouladsine. “Active fault detection and isolation strategy for an unmanned aerial vehicle with redundant flight control surfaces”. In: *Mediterranean Conference on Control and Automation - Conference Proceedings, MED’08*. July. Ajaccio, France, 2008, pp. 1246–1251. ISBN: 9781424425051. DOI: 10.1109/MED.2008.4602147.
- [15] T. Brotherton et al. “Generic integrated PHM / controller system”. In: *IEEE Aerospace Conference Proceedings*. Vol. 2005. 858. Big Sky, Montana, USA: IEEE, 2005, pp. 3427–3437. ISBN: 0780388704. DOI: 10.1109/AERO.2005.1559645.
- [16] W. F. Bonnice. “Intelligent Fault Diagnosis and Failure Management of Flight Control Actuation Systems.” 2019.
- [17] O. Ritter et al. “Intelligent diagnostics for aircraft hydraulic equipment”. In: *European Conference of the Prognostics and Health Management Society*. Utrecht, The Netherlands, 2018, pp. 1–13.
- [18] R. Gentile et al. “Electro-hydraulic servoactuators failure identification: Health features extraction through a high-fidelity physical model”. In: *International Journal of Mechanics and Control* 21.1 (2020), pp. 91–100. ISSN: 15908844.
- [19] R. Isermann and P. Ballè. “Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes”. In: *Control Engineering Practice* 5 (1997), pp. 709–719. ISSN: 0967-0661. DOI: 10.1016/S0967-0661(97)00053-1.
- [20] M. Azam et al. “In-flight fault detection and isolation in aircraft flight control systems”. In: *IEEE Aerospace Conference Proceedings*. Vol. 2005. Big Sky, Montana, USA: IEEE, 2005, pp. 3555–3565. ISBN: 0780388704. DOI: 10.1109/AERO.2005.1559659.

- [21] V. Igenewari, Z. Skaf, and I. K. Jennions. “A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities”. In: *Annual Conference of the Prognostic and Health Management Society*. Scottsdale, Arizona, USA, 2019, p. 10.
- [22] O. J. Mengshoel et al. “Diagnosing Faults in Electrical Power Systems of Spacecraft and Aircraft”. In: *Twentieth Innovative Applications of Artificial Intelligence Conference*. 2008, pp. 1699–1705.
- [23] Dr Bryon Wicks and Mr Richard Eustace. “The Insertion of Advanced Diagnostic Technologies in an Ageing Fleet of Engines , to Improve Engine / Aircraft Availability and Mission Reliability”. In: 2011.
- [24] T. Cooper et al. *Global Fleet & MRO Market Forecast Summary*. Tech. rep. 2017.
- [25] *Free Radical: Innovative configurations and propulsive concepts for the 2030s*. URL: www.cleansky.eu/free-radical-innovative-configurations-and-propulsive-concepts-for-the-2030s (visited on 05/06/2021).
- [26] *Data revolution in aviation - Innovation*. URL: www.airbus.com/public-affairs/brussels/our-topics/innovation/data-revolution-in-aviation.html (visited on 05/06/2021).
- [27] M. Harrison. *Big Data: the race is on! But what is the end goal?* Tech. rep. November. 2018.
- [28] Airbus. *Skywise brochure*. Tech. rep. 2019.
- [29] C. S. Byington, M. Watson, and D. Edwards. “Data-driven neural network methodology to remaining life predictions for aircraft actuator components”. In: *IEEE Aerospace Conference Proceedings*. Vol. 6. Big Sky, Montana, USA, 2004, pp. 3581–3589. ISBN: 0780381556. DOI: 10.1109/AERO.2004.1368175.
- [30] J. Speri. *Base Maintenance - Aircraft Maintenance*. May 2021. URL: www.lufthansa-technik.com/base-maintenance.
- [31] F. Marino. “Advanced Fault Detection and Classification on Primary Flight Control Actuators”. M.Sc. Degree. Politecnico di Torino, 2017, pp. 1–144.
- [32] D. Bruno. *Development of health features and automatic fault diagnosis methods for primary flight control actuators*. 2018.
- [33] G. Carezza. *Automatic failure diagnostics techniques and health features investigation for electro-hydraulic actuators*. 2020.
- [34] Airbus. “Flightdeck and systems briefing for pilots”. 1998. URL: http://www.smartcockpit.com/docs/A320%7B%5C_%7DFlight%7B%5C_%7DDeck%7B%5C_%7Dand%7B%5C_%7DSystems%7B%5C_%7DBriefing%7B%5C_%7DFor%7B%5C_%7DPilots.pdf.

- [35] F. Rombolà. *Development of new health features and automation of tests for primary flight control actuators using a collaborative robot*. 2019.
- [36] S. Antika et al. *Method for Lubricating High Pressure Hydraulic System Using Phosphate Ester Hydraulic Fluid*. 2004.
- [37] Solutia Co. *Fire-resistant hydraulic fluids*. Tech. rep. 10 , Nov. 1981. 1981, p. 68.
- [38] E. Urata. “Influence of unequal air-gap thickness”. In: *Journal of Mechanical Engineering Science*. Vol. 221. 2007, pp. 1287–1297.
- [39] S. Boissenin and E. Salavy. “A320 Family/Evolution of ground spoiler logic”. In: *Safety First - The Airbus Safety Magazine* 9 (2010), pp. 3–6.
- [40] H. Zumbahlen. *Linear Variable Differential Transformers*. 2017.
- [41] W. Chu. “Infinite Series Identities on Harmonic Numbers”. In: *Results in Mathematics* 61.3-4 (2012), pp. 209–221. ISSN: 14226383. DOI: 10.1007/s00025-010-0089-2.
- [42] National Instruments. *Measuring Position and Displacement with LVDTs*. 2016. URL: <http://www.ni.com/white-paper/3638/en/>.
- [43] M. G. Ruppert et al. “A review of demodulation techniques for amplitude-modulation atomic force microscopy”. In: *Beilstein Journal of Nanotechnology* 8.1 (2017), pp. 1407–1426. ISSN: 21904286. DOI: 10.3762/bjnano.8.142.
- [44] J. Szczyrbak and S. Products. *LVDT Signal Conditioning Techniques*. Tech. rep. 1997, pp. 1–18.
- [45] R. M. Ford, R. S. Weissbach, and D. R. Loker. “A novel DSP-based LVDT signal conditioner”. In: *IEEE Transactions on Instrumentation and Measurement* 50.3 (2001), pp. 768–773. ISSN: 00189456. DOI: 10.1109/19.930452.
- [46] W. Petchmaneelumka, P. Mano, and V. Riewruja. “LVDT demodulator based on amplitude detector”. In: *3rd International Conference on Control and Robotics Engineering, ICCRE 2018*. 2018, pp. 218–221. DOI: 10.1109/ICCRE.2018.8376468.
- [47] F. F. Yassa and S. Garverick. “A Multichannel Digital Demodulator for LVDT/RVDT Position Sensors”. In: *IEEE Journal of Solid-State Circuit* 25.2 (1990), pp. 441–450.
- [48] M.G. Ruppert, D.M. Harcombe, and S.O.R. Moheimani. “High-Bandwidth Demodulation in MF-AFM : A Kalman Filtering Approach”. In: *IEEE/ASME Transactions on Mechatronics* 21.6 (2016), pp. 2705–2715. DOI: 10.1109/TMECH.2016.2574640.
- [49] D. De Stefanis. *Advance failure detecion and diagnostics on primary flight control servo actuators augmented by collaborative robotics*. 2020.

- [50] A. Raviola. *Automatization of primary flight control actuators maintenance procedures using collaborative robotics*. 2018.
- [51] G. Vachtsevanos, F. L. Lewis, and M. Roemer. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems -*. John Wiley & Sons, Inc., 2006, p. 456. ISBN: 978-0-471-72999-0.
- [52] M. Liu. *CDF Home Page*. URL: cdf.gsfc.nasa.gov/ (visited on 10/29/2020).
- [53] P. D. Welch. “The Use of Fast Fourier Transform for the Estimation of Power Spectra”. In: *Digital Signal Processing 2* (1975), pp. 532–574.
- [54] A. Schuster. “On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena”. In: *Journal of Geophysical Research* 3.1 (1898), p. 13. ISSN: 0148-0227. DOI: 10.1029/tm003i001p00013.
- [55] F. J. Harris. “On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform”. In: *Proceedings of the IEEE* 66.1 (1978), pp. 51–83. ISSN: 15582256. DOI: 10.1109/PROC.1978.10837.
- [56] P. Podder et al. “Comparative Performance Analysis of Hamming, Hanning and Blackman Window”. In: *International Journal of Computer Applications* 96.18 (2014), pp. 1–7. DOI: 10.5120/16891-6927.
- [57] R. Killick, P. Fearnhead, and I. A. Eckley. “Optimal detection of change-points with a linear computational cost”. In: *Journal of the American Statistical Association* 107.500 (2012), pp. 1590–1598. ISSN: 01621459. DOI: 10.1080/01621459.2012.737745. arXiv: 1101.1438.
- [58] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006. ISBN: 978-0-387-31073-2.
- [59] M. V. Patil and A.M. Nageswara Yogi. “Importance of Data Collection and Validation for Systematic Software Development Process”. In: *International Journal of Computer Science & Technology* 3.2 (2011), pp. 260–278.
- [60] Y. Roh, G. Heo, and S.E. Whang. “A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective”. In: *IEEE Transactions on Knowledge and Data Engineering* (2019), p. 1. ISSN: 1558-2191. DOI: 10.1109/TKDE.2019.2946162.
- [61] T. Wei et al. “Research on Data Acquisition of Communication Equipment Based On Machine Learning”. In: *2019 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. 2019, pp. 106–112. DOI: 10.1109/ICCNEA.2019.00030.

- [62] A. Vogelsang and M. Borg. “Requirements Engineering for Machine Learning: Perspectives from Data Scientists”. In: *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. 2019, pp. 245–251. DOI: 10.1109/REW.2019.00050.
- [63] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Ed. by Elsevier. Second Edi. Vol. 2. 2003. DOI: 10.1016/0370-2693(94)90548-7.
- [64] R. Saravanan and P. Sujatha. “Algorithms: A Perspective of Supervised Learning Approaches in Data Classification”. In: *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. Iicccs. IEEE, 2018, pp. 945–949. ISBN: 9781538628423.
- [65] B. Eryilmaz and B. H. Wilson. “Modeling the Internal Leakage of Hydraulic Servovalves”. In: (2000).
- [66] A. Mornacchi and M. Vignolo. “Health Management System for the Hydraulic Servoactuators of Fly- by-wire Primary Flight Control Systems”. In: *European Conference of the Prognostics and Health Management Society*. 2014.
- [67] A. C. Bertolino et al. “EHSA Primary Flight Controls Seals Wear Degradation Model”. In: *IMECE*. Ed. by ASME. Pittsburgh, PA, USA: ASME, 2018, pp. 1–12. DOI: 10.1115/IMECE2018-87080.
- [68] M. Dalla Vedova, P. Maggiore, and M. Sorli. “Identification of Precursors of Servovalves Failures for Implementation of an Effective Prognostics”. In: *4th International Conference on Recent Advances in Aerospace Actuation Systems and Components*. Toulouse, France, 2010.
- [69] L. Borello et al. “A Prognostic Model for Electrohydraulic Servovalves”. In: *Annual Conference of the Prognostics and Health Management Society*. 2009.
- [70] L. Borello and G. Jacazio. “Mathematical Models of Electrohydraulic Servovalves for Fly-by-wire Flight Control Systems”. In: *Mathl Computer Modelling II* (1988), pp. 563–569.
- [71] I. Guyon and A. Elisseeff. “An Introduction to Variable and Feature Selection”. In: *Journal of Machine Learning Research* 3 (2005), pp. 1157–1182. ISSN: 00032670. DOI: 10.1016/j.aca.2011.07.027.
- [72] T. Yiu. *The Curse of Dimensionality. Why High Dimensional Data Can Be So Troublesome*. 2019. URL: towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e (visited on 03/18/2021).
- [73] Y. Frances and H. Sloan. “Lifting the curse of Deimensionality”. In: *Noitce of the AMS* 52.5 (2005), pp. 50–57. ISSN: 00368733.
- [74] M. Koepen. *The Curse of Dimensionality*. Tech. rep. 3. 2018, pp. 614–615. DOI: 10.1097/ALN.0000000000002350.

- [75] G. Chandrashekar and F. Sahin. “A survey on feature selection methods”. In: *Computers and Electrical Engineering* 40.1 (2014), pp. 16–28. ISSN: 00457906. DOI: 10.1016/j.compeleceng.2013.11.024. URL: <http://dx.doi.org/10.1016/j.compeleceng.2013.11.024>.
- [76] S. Alelyani, T. Jiliang, and L. Huan. “Feature Selection for Clustering: A Review.” In: *Data Clustering: Algorithms and Applications* 29 (2013), pp. 110–121.
- [77] S. J. Kim et al. “An interior-point method for large-scale L1-regularized least squares”. In: *IEEE Journal on Selected Topics in Signal Processing* 1.4 (2007), pp. 606–617. ISSN: 19324553. DOI: 10.1109/JSTSP.2007.910971.
- [78] J. Friedman, T. Hastie, and R. Tibshirani. “Regularization Paths for Generalized Linear Models via Coordinate Descent”. In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. ISSN: 0014-4886. arXiv: 0908.3817.
- [79] A. Suppers, A. J. van Gool, and H. Wessels. “Integrated chemometrics and statistics to drive successful proteomics biomarker discovery”. In: *Proteomes* 6.2 (2018). ISSN: 22277382. DOI: 10.3390/PROTEOMES6020020.
- [80] G. Azevedo. *Feature selection techniques for classification and Python tips for their application*. 2019. URL: towardsdatascience.com/feature-selection-techniques-for-classification-and-python-tips-for-their-application-10c0ddd7918b (visited on 03/21/2021).
- [81] J. Browniee. *How to Choose a Feature Selection Method For Machine Learning*. 2020. URL: machinelearningmastery.com/feature-selection-with-real-and-categorical-data/ (visited on 03/21/2021).
- [82] M. Desai, S.V. Patel, and B. Prajapati. “ANOVA and Fisher Criterion based Feature Selection for Lower Dimensional Universal Image Steganalysis”. In: *International Journal of Image Processing* 10.3 (2016), pp. 145–160.
- [83] J. P. Verma. “Sports Research with Analytical Solution Using SPSS®”. In: *Sports Research with Analytical Solution Using SPSS*. Ed. by John Wiley and Sons. First Edit. 2016. Chap. Independen, pp. 100–153. DOI: 10.1002/9781119206767.
- [84] K. J. Johnson and R. E. Synovec. “Pattern recognition of jet fuels: Comprehensive GC \times GC with ANOVA-based feature selection and principal component analysis”. In: *Chemometrics and Intelligent Laboratory Systems* 60.1-2 (2002), pp. 225–237. ISSN: 01697439. DOI: 10.1016/S0169-7439(01)00198-8.
- [85] Q. Gu, Z. Li, and J. Han. “Generalized fisher score for feature selection”. In: *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011* (2011), pp. 266–273.

- [86] M. J. Blanca et al. “Non-normal data: is ANOVA still a valid option”. In: *Psicothema* 29.4 (2017), pp. 552–557. ISSN: 1886144X. DOI: 10.7334/psicothema2016.
- [87] A. Alin. “Multicollinearity”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.3 (2010), pp. 370–374. ISSN: 19395108. DOI: 10.1002/wics.84.
- [88] N. Zhao et al. “High-dimensional variable screening under multicollinearity”. In: *Stat* 9.1 (2020), pp. 1–11. ISSN: 20491573. DOI: 10.1002/sta4.272.
- [89] A. F. Lukman et al. “Modified ridge-type estimator to combat multicollinearity: Application to chemical data”. In: *Journal of Chemometrics* 33.5 (2019), pp. 1–12. ISSN: 1099128X. DOI: 10.1002/cem.3125.
- [90] J Han, J Pei, and M Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011. ISBN: 9780123814807. URL: books.google.it/books?id=pQws07tdpjoC.
- [91] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [92] C. Sharpe et al. “A Comparative Evaluation of Supervised Machine Learning Classification Techniques for Engineering Design Applications”. In: *Journal of Mechanical Design* 141.12 (2019). ISSN: 1050-0472. DOI: 10.1115/1.4044524.
- [93] A. Giantomassi et al. “Hidden Markov model for health estimation and prognosis of turbofan engines”. In: *Proceedings of the ASME Design Engineering Technical Conference*. Vol. 3. PARTS A AND B. 2011, pp. 681–689. ISBN: 9780791854808. DOI: 10.1115/DETC2011-48174.
- [94] V. Vakharia, V. K. Gupta, and P. K. Kankar. “Ball Bearing Fault Diagnosis using Supervised and Unsupervised Machine Learning Methods”. In: *The International Journal of Acoustics and Vibration* 20.4 (2015). DOI: 10.20855/ijav.2015.20.4387.
- [95] M. Elforjani and S. Shanbr. “Prognosis of Bearing Acoustic Emission Signals Using Supervised Machine Learning”. In: *IEEE Transactions on Industrial Electronics* 65.7 (2018), pp. 5864–5871. ISSN: 02780046. DOI: 10.1109/TIE.2017.2767551.
- [96] F. Shen et al. “Bearing fault diagnosis based on SVD feature extraction and transfer learning classification”. In: *Proceedings of 2015 Prognostics and System Health Management Conference, PHM 2015* 1 (2016). DOI: 10.1109/PHM.2015.7380088.

- [97] M. Z. Ali et al. “Machine learning-based fault diagnosis for single and multi-faults in induction motors using measured stator currents and vibration signals”. In: *IEEE Transactions on Industry Applications* 55.3 (2019), pp. 2378–2391. ISSN: 19399367. DOI: 10.1109/TIA.2019.2895797.
- [98] K. Li et al. “A novel fault diagnosis algorithm for rotating machinery based on a sparsity and neighborhood preserving deep extreme learning machine”. In: *Neurocomputing* 350 (2019), pp. 261–270. ISSN: 18728286. DOI: 10.1016/j.neucom.2019.03.084.
- [99] S. M. Tayyab et al. “Intelligent fault diagnosis of rotating machine elements using machine learning through optimal features extraction and selection”. In: *Procedia Manufacturing* 51 (2020), pp. 266–273. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.10.038. URL: doi.org/10.1016/j.promfg.2020.10.038.
- [100] X. Wu et al. “A hybrid classification autoencoder for semi-supervised fault diagnosis in rotating machinery”. In: *Mechanical Systems and Signal Processing* 149 (2021), p. 107327. ISSN: 10961216. DOI: 10.1016/j.ymsp.2020.107327. URL: doi.org/10.1016/j.ymsp.2020.107327.
- [101] W. J. Crowther et al. “Fault diagnosis of a hydraulic actuator circuit using neural networks—an output vector space classification approach”. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 212.1 (1998), pp. 57–68. ISSN: 0959-6518. DOI: 10.1243/0959651981539299.
- [102] M. Mazzoleni, Y. Maccarana, and F. Previdi. “A comparison of data-driven fault detection methods with application to aerospace electro-mechanical actuators”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 12797–12802. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.1837. URL: doi.org/10.1016/j.ifacol.2017.08.1837.
- [103] J. Brownie. *Plot a Decision Surface for Machine Learning Algorithms in Python*. URL: machinelearningmastery.com/plot-a-decision-surface-for-machine-learning/ (visited on 04/02/2021).
- [104] M. Schmidt, N. L. Roux, and F. Bach. “Minimizing Finite Sums with the Stochastic Average Gradient”. In: (2013), pp. 1–45.
- [105] Zhang Y. Xu Y. M. and Chen L. N. “Research of genetic training algorithm for identifying mechanical failure modes within the framework of case-based reasoning”. In: *Chinese Journal of Aeronautics* 18.2 (2005), pp. 122–129. ISSN: 10009361. DOI: 10.1016/S1000-9361(11)60316-6.

- [106] T. N. Do. “Automatic Learning Algorithms for Local Support Vector Machines”. In: *SN Computer Science* 1.1 (2020), pp. 1–11. ISSN: 2662-995X. DOI: 10.1007/s42979-019-0006-z. URL: doi.org/10.1007/s42979-019-0006-z.
- [107] S. Lek and Y. S. Park. “Multilayer Perceptron”. In: *Encyclopedia of Ecology, Five-Volume Set*. 2008, pp. 2455–2462. ISBN: 9780080914565. DOI: 10.1016/B978-008045405-4.00162-2. URL: en.wikipedia.org/wiki/Multilayer_perceptron.
- [108] T. T. Le, J. Watton, and D. T. Pham. “An artificial neural network based approach to fault diagnosis and classification of fluid power systems”. In: *Journal of Systems and Control Engineering* October 1996 (1997), pp. 307–317.
- [109] J. McGhee, I. A. Henderson, and A. Baird. “Neural networks applied for the identification and fault diagnosis of process valves and actuators”. In: *Measurement* 20.4 (1997), pp. 267–275. ISSN: 02632241. DOI: 10.1016/S0263-2241(97)00043-2.
- [110] D. P. Kingma and J. Lei Ba. “Adam: a Method for Stochastic Optimization”. In: *Proceedings of the ICLR 2015*. San Diego, 2015. ISBN: 9781450300728. DOI: 10.1145/1830483.1830503. arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.
- [111] A. Mokhtari and A. Ribeiro. “Global convergence of online limited memory BFGS”. In: *Journal of Machine Learning Research* 16 (2015), pp. 3151–3181. ISSN: 15337928. arXiv: 1409.2045.
- [112] Y. Zhang and Y. Yang. “Cross-validation for selecting a model selection procedure”. In: *Journal of Econometrics* 187.1 (2015), pp. 95–112.
- [113] C. Bergmeir and J. M. Benitez. “On the use of cross-validation for time series predictor evaluation”. In: *Information Sciences* 191 (2012), pp. 192–213.
- [114] A. Arlot S. Celisse et al. “A survey of cross-validation procedures for model selection”. In: *Statistics surveys* 4 (2010), pp. 40–79.
- [115] C. Schaffer. “Technical Note: Selecting a Classification Method by Cross-Validation”. In: *Machine Learning* 13.1 (1993), pp. 135–143. ISSN: 15730565. DOI: 10.1023/A:1022639714137.
- [116] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [117] A. P. Daga et al. “On the use of PCA for Diagnostics via Novelty Detection : interpretation , practical application notes and recommendation for use”. In: *European Conference of the Prognostics and Health Management Society*. Turin, Italy, 2020, pp. 1–13.

- [118] J. Brownlee. *Introduction to Random Number Generators for Machine Learning in Python*. 2018. URL: machinelearningmastery.com/introduction-to-random-number-generators-for-machine-learning/ (visited on 04/15/2021).
- [119] C. Schaffer. “Selecting a classification method by cross-validation”. In: *Machine Learning* 13.1 (1993), pp. 135–143.
- [120] A. W Moore and M. S. Lee. “Efficient algorithms for minimizing cross validation error”. In: *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 190–198.
- [121] G.S.K Ranjan, A.K. Verma, and S. Radhika. “K-nearest neighbors and grid search cv based real time fault monitoring system for industries”. In: *2019 IEEE 5th international conference for convergence in technology (I2CT)*. IEEE. 2019, pp. 1–5.
- [122] T. Fawcett. “An introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010.
- [123] S. Rothe, S. F. Wirtz, and D. Soeffker. “About the reliability of diagnostic statements : fundamentals about detection rates , false alarms , and technical requirements”. In: *AKIDA*. November 2018. 2016, pp. 159–174.
- [124] Google. *Classification: Accuracy*. 2020. URL: developers.google.com/machine-learning/crash-course/classification/accuracy.
- [125] J. F. Sutton. *Synchronous Demodulator*. 1991.
- [126] Mathworks. *Align Signals Using Cross-Correlation - MATLAB & Simulink - MathWorks Italia*. URL: it.mathworks.com/help/signal/ug/align-signals-using-cross-correlation.html (visited on 09/15/2020).
- [127] Upton G. J G and Cook I. *Understanding statistics*. English. Oxford: Oxford Univ. Press, 1996. ISBN: 0199143919 9780199143917 019914351X 9780199143511.

This Ph.D. thesis has been typeset by means of the T_EX-system facilities. The typesetting engine was pdfL^AT_EX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete T_EX-system installation.