Politecnico di Torino

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

Doctoral Dissertation
Doctoral Program in Aerospace Engineering (37^{th}cycle)

Definition, Design and Optimization of Aircraft On-Board System Architectures Considering Multiple Life Cycle Stages

By

Carlos Cabaleiro de la Hoz

Supervisor(s):
Prof. Marco Fioriti, Supervisor
Prof. Sabrina Corpino, Co-Supervisor
Dr.-Ing. Björn Nagel (DLR), Co-Supervisor

Doctoral Examination Committee:
Prof. Danilo Ciliberti, Referee, University of Naples Federico II
Prof. Gianpietro di Rito, Referee, University of Pisa
Prof. Pierluigi della Vecchia, Board Member, University of Naples Federico II
Prof. Paolo Maggiore, Board Member, Politecnico di Torino
Prof. Davide Ferretto, Board Member, Politecnico di Torino

Politecnico di Torino
2025

[This page has been intentionally left blank]

Declaration

I hereby declare that, the contents and organization 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.

Carlos Cabaleiro de la Hoz
2025

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

This Ph.D. thesis would not have been possible without the help, assistance and guidance of many people. First of all I would like to thank my university, Politecnico di Torino, and the German Aerospace Center (DLR) for giving me the chance to perform this research.

I would like to thank my supervisor, Prof. Marco Fioriti, for all the guidance, support, availability and assistance during all these years of work. Also, to my mentor at DLR, Luca Boggero, for all the support and encouragement to become a better researcher. To my colleagues at DLR-SL, with special mention to Jasper and Pina, for all the help and support. And to my colleagues at Politecnico.

Many thanks to Jennifer Ramm, from DLR-MO, for the amazing collaboration that we carried out. And special thanks to the three students that we supervised in common: Simone Dell'Anna, Feliciano Scarcelli and Niccolò Ninotta. This Ph.D. thesis would not have been possible without their extraordinary and successful work.

Thanks to all my friends and colleagues who were also doing a Ph.D. at the same time, for sharing their tips and experiences and for making me have a better perspective on how to approach this experience. Special mention to Kilian Swannet, David Planas and Alberto Racionero.

Last but not least, I would like to thank my family for all the unconditional support during this stressful period in which I moved from one country to another several times. And to all my friends, old and new, for all the fun that we had during this time.

Abstract

On-board systems play a critical role in ensuring the safe and efficient operation of an aircraft. They include all the systems that perform a needed function for the aircraft, such as control or cabin air conditioning, as well as the systems that are needed by other systems, such as power generators and distributors. They represent a significant part of the overall aircraft design, directly influencing performance factors such as mass and fuel consumption. Their impact extends beyond the operational phase, affecting various stages throughout the aircraft life cycle (such as manufacturing, maintenance or disposal). The architecture of these systems is determined early in the design process, and it has a profound effect on the final product. Such architecture consists of various subsystems, components, and the connections among them. Given the huge number of possible feasible solutions, the architectural design space is often large and complex, and automation is essential to effectively explore it.

This Ph.D. research focuses on developing a methodology to evaluate the performance, maintenance, and certification aspects of on-board system architectures during early design phases. The proposed methodology enables the automated assessment of innovative system architectures by linking their design space model to a multi-objective and multi-disciplinary evaluation framework. This framework filters out non-certifiable architectures at an early stage, reducing the design space and saving computational time by preventing unnecessary evaluations.

Two application cases are shown. One assesses the three disciplines of interest for an innovative architecture concept of high-lift devices, comparing it to the conventional case and showing strong potential. The other example illustrates the roll control function of a flight control system. It demonstrates that the effective filtering of the design space successfully identifies the most promising architectures and uncovers valuable trade-offs. This process helps engineers in the decision making process during the conceptual design of on-board system architectures.

[This page has been intentionally left blank]

Contents

List of Figures **ix**

List of Tables **xiii**

Nomenclature **xv**

1 Introduction **1**

2 Literature Review **5**

2.1 On-Board Systems Definition 5

2.1.1 Power Generation Systems 8

2.1.2 Power Transformation and Distribution Systems 8

2.1.3 Power Consuming Systems 10

2.2 OBS Architectures Examples: Conventional, MEA & AEA 12

2.3 On-Board Systems Evaluation 17

2.3.1 On-Board Systems Evaluation: Performance 18

2.3.2 On-Board Systems Evaluation: Certification 21

2.3.3 On-Board Systems Evaluation: Maintenance 28

2.4 Science Gaps & Research Questions 33

3 Methodology & Implementation **36**

3.1 General Overview 36

3.2 System Architecting: Architectures Generation 40
3.2.1 Implementation 42
3.3 Certification: Architectures Filtering & Evaluation 44
3.3.1 Reliability Block Diagram 49
3.3.2 Implementation 74
3.4 Performance: Architectures Evaluation 79
3.4.1 Implementation 84
3.5 Maintenance: Architectures Evaluation 88
3.5.1 Implementation 90
3.6 Optimization Process: Metrics Evaluation 91
3.6.1 Implementation 92
3.7 Trade-Off Analysis: Metrics Synthesis 93
3.7.1 Implementation 95

4 Application Case 1: High-Lift Devices 96
4.1 Motivation & Context 96
4.2 Methodology Adjustments 103
4.3 Results 110

5 Application Case 2: Flight Control System, Roll Control 118
5.1 Motivation & Context 119
5.2 Design Space Modelling & Methodology Adjustments 124
5.3 Results 135
5.4 Trade-off Analysis 148

6 Conclusions 155

References 160

List of Figures

1.1 Mitigating the knowledge paradox 2

1.2 Scheme of the main objective of this Ph.D research, showing the three disciplines of interest from the overall aircraft life cycle 4

2.1 Conventional and more electric OBS architectures 14

2.2 More-electric and all-electric OBS architectures 14

2.3 Examples of different age-reliability patterns for aircraft components represented by Weibull distributions 26

2.4 Breakdown structure of an aircraft total operating cost 31

3.1 Simplified methodology schema. The proposed filter covers two of the research questions, while the trade-off block tackles the remaining one 37

3.2 Extended design structure matrix of the methodology 39

3.3 Architectures example: gas pipeline 42

3.4 Design space of the gas pipeline closing system. 44

3.5 Examples of different RBDs configurations 51

3.6 RBD; H-configuration schema 52

3.7 H-configuration schema partitions 53

3.8 Example of an architecture with an H-configuration RBD. The figure shows the hydraulic system of the Airbus A320 and its flight spoilers, it also contains the resulting RBD 54

3.9 Example of an architecture with an H-configuration RBD. The figure shows the fuel system of the Airbus A330 and its simplified RBD . 55
3.10 RBD; S-configuration schema 56
3.11 S-configuration schema partitions 56
3.12 S-configuration schema, partitions of partition 1S 57
3.13 Example of an architecture with an S-configuration RBD 57
3.14 Example of an architecture with an S-configuration RBD. The figure shows the electrical system of the Airbus A320 and its simplified RBD for the DC Essential function 58
3.15 RBD; V-configuration schema 58
3.16 V-configuration schema partitions 59
3.17 Example of an architecture with an V-configuration RBD. The figure shows the electrical system of the Airbus A320 and its simplified RBD for the DC Bat function 60
3.18 RBD; T-configuration schema 60
3.19 T-configuration schema partitions 61
3.20 T-configuration schema, partitions of partition 2T 61
3.21 Example of an architecture with an T-configuration RBD 62
3.22 RBD; R-configuration schema 62
3.23 Component attributes card 65
3.24 Example on nodes numbering for a generic RBD 66
3.25 Logic of the RBD automation algorithms 67
3.26 RBD reduction example: iteration 1 69
3.27 RBD reduction example: iterations 2 and 3 69
3.28 RBD reduction example: iterations 4, 5, 6 and 7 70
3.29 RBD reduction, second example: first iteration 71
3.30 RBD reduction, second example: rest of iterations 71
3.31 Example of a system with various functioning modes 73

3.32 Example of automated translation from physical architecture to RBD 73

3.33 ACOBS - Automated preliminary Certification of aircraft On-Board Systems 74

3.34 Example of translation of an architecture from ADORE into a RBD, H-configuration 77

3.35 Example overview of the process from architecture definition to RBD solving 78

3.36 Systems with more than one top function 79

3.37 XDSM of the evaluation framework, simplified version without convergence loop 83

4.1 Scheme of a conventional flap architecture for a commercial aircraft 98

4.2 Scheme of the proposed innovative flap architecture. 100

4.3 RBD scheme for the conventional flap architecture 104

4.4 RBD scheme for the innovative flap architecture 105

4.5 RBD scheme for the case considering jamming-tolerant strategies. This case is added for discussion but has not been assessed in the analysis. 105

4.6 RBD schema for the innovative slat architecture 106

4.7 MPD tasks associated to the flaps for the conventional architecture; tasks that disappear for the innovative architecture are marked with a red dot, the ones that change are marked with an orange triangle . 108

4.8 MPD tasks associated to the slats for the conventional architecture; tasks that disappear for the innovative architecture are marked with a red dot 108

4.9 MPD tasks associated to the EWIS for high-lift devices for the conventional architecture; tasks that change for the innovative architecture are marked with an orange triangle 109

4.10 OBS architecture of the two aircraft with conventional OBS 111

4.11 OBS architecture of the two aircraft with all-electric OBS 112

5.1 Reference architecture for flight control system roll control; Airbus A320 ailerons and spoilers 120

5.2 Reference architecture for the distribution systems; Airbus A320 . . 122

5.3 Design space model; zoom in the top function and quantities of interest, screenshot from ADORE 124

5.4 Design space model; zoom in the FCS, screenshot from ADORE . . 125

5.5 Design space model; zoom in the hydraulic system, screenshot from ADORE 126

5.6 Design space model; zoom in the electrical system, screenshot from ADORE 127

5.7 Design space model; zoom inside the spoiler hydraulic actuator component, screenshot from ADORE 128

5.8 Design space model; zoom inside unconventional components, screenshots from ADORE 129

5.9 Design space model; summary of all the architectural choices, screenshot from ADORE 130

5.10 System mass breakdown by architecture; with MTOM 137

5.11 Linear dependency between MTOM and fuel burn for the specific application case 138

5.12 Visualization of the results, screenshot from ADORE 140

5.13 Visualization of results for the optimization run with NSGA-II genetic algorithm 142

5.14 Sensitivity analysis of the NSGA-II genetic algorithm parameters for the application case 147

5.15 Utility functions for the MTOM attribute 151

5.16 Utility functions for the MMH attribute 151

5.17 Utility functions for the probability of failure attribute 151

5.18 Value result for each trade-off point for each of the proposed cases . 153

List of Tables

2.1 ATA 100 chapters definition for aircraft systems 7

3.1 Dedicated FCS tool: input and mass results for three different test cases 88

4.1 Assumed input parameters for the reference aircraft; Airbus A320neo 103

4.2 Failure rates values for the components of the high-lift device system, taken from the Quanterion Automated Databook (NPRD-2016) . . 103

4.3 Results for both OBS baselines calculated with ASTRID 107

4.4 Mass results of the flight control system and on-board systems of the four studied aircraft 113

4.5 Engine off-takes and bleeding results of the flight control system and on-board systems of the four studied aircraft 114

4.6 MTOM and fuel burn results of the four studied aircraft 115

4.7 Probability of failure results of the different high-lift devices concepts 115

4.8 MPD-MMH results for the high-lift devices concepts 116

4.9 MPD-MMH results of the four studied aircraft 116

5.1 Assumed input parameters for the reference aircraft; Airbus A320neo 123

5.2 Failure rates values for the components of the flight spoiler system, taken from the Quanterion Automated Databook (NPRD-2016) . . . 128

5.3 Design space statistics 129

5.4 ASTRID results for the three OBS baseline architectures 132

5.5 Summary of the architectures analyzed in the initial design of experiments 136

5.6 Results for the 11 architectures of the design of experiments 137

5.7 Optimization results analysis 141

5.8 Pareto Front points from the optimization run with NSGA-II genetic algorithm 145

5.9 Trade-off points 149

5.10 Trade-off results for the four different cases 152

Nomenclature

λ	Failure Rates
AC	Alternating Current
ACE	Actuator Control Electronics
AEA	All-Electric Aircraft
AMC	Acceptable Means of Compliance
APU	Auxiliary Power Unit
ATA	Air Transport Association
CER	Cost Estimation Relationship
CS	Certification Specifications
DC	Direct Current
DMC	Direct Maintenance Cost
DOC	Direct Operating Cost
EASA	European Aviation Safety Agency
EBHA	Electric Backup Hydraulic Actuator
ECS	Environmental Control System
EHA	Electro-Hydrostatic Actuator
EMA	Electro-Mechanical Actuator
EWIS	Electrical Wiring Interconnection System
FAA	Federal Aviation Administration
FCS	Flight Control System
FHA	Functional Hazard Assessment
FMECA	Failure Modes, Effects and Criticality Analysis
FTA	Fault Tree Analysis
GA	Genetic Algorithms
GB	Gearbox
GCU	Generator Control Unit

HSA	Conventional Hydraulic Servo Actuator
INCOSE	International Council on Systems Engineering
IOC	Indirect Operating Cost
IPS	Ice Protection System
MAU	Multi Attribute Utility
MDO	Multi-Disciplinary Design Optimization
MEA	More-Electric Aircraft
MMH	Maintenance Man Hours
MPD	Maintenance Planning Document
MTOM	Maximum Take-Off Mass
MTOW	Maximum Take-Off Weight
MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
NPRD	Non-electronic Parts Reliability Data
OAD	Overall Aircraft Design
OBS	On-Board Systems
OEM	Aerospace Manufacturing Company
OEW	Operational Empty Weight
PCE	Power Control Electronics
PDU	Power Drive Unit
PTU	Power Transfer Unit
R	Reliability
RAMS	Reliability, Availability, Maintainability and Safety
RAT	Ram Air Turbine
RBD	Reliability Block Diagram
SFC	Specific Fuel Consumption
TL	Torque Limiter
TLARs	Top Level Aircraft Requirements
TOC	Total Operating Cost
TR	Transformer Rectifier
XDSM	Extended Design Structure Matrix
ZSA	Zonal Safety Analysis

Chapter 1

Introduction

Air passenger traffic has grown worldwide around a 9% every year since 1960 and is expected to keep growing in the following years [1]. Also, air transportation represents a 2% of human CO2 emissions, and it is expected to grow to a 3% by 2050. Non-CO2 emissions, such as contrails, are expected to increase as well [2]. Owing to this, there is pressure on aircraft manufacturers to improve the performance of future aircraft in terms of air pollution and noise to achieve the worldwide objectives regarding climate change [1]. As a result, the aviation industry faces the challenge of reducing fuel consumption in the near future. New aircraft concepts are being explored to address this goal. Innovations regarding on-board systems (OBS) technologies are also under investigation to help to achieve such fuel reduction. For example, more-electric and all-electric aircraft are adopting new electrified systems that could potentially lower such emissions, and even operating costs [3]. Hydrogen-powered aircraft require new fuel system designs to store and deliver this alternative fuel. Hypersonic vehicles, on the other hand, introduce the need for new systems, such as thermal controls, which were not required in previous aircraft designs. Many of these new aircraft concepts could shape the next generation of aviation, all of which will depend on innovative systems.

On-board systems play a significant role in an aircraft's performance, impacting aspects such as mass, power off-takes, and fuel consumption, all of which affect direct operating costs. New systems could offer benefits like enhanced performance, increased reliability, and reduced costs. OBS are critical to an aircraft's mission, supporting functions like communication, navigation, and flight control. These

systems must be certified and maintained to ensure their reliability. OBS include the power generation, distribution, and consumption systems. However, on-board systems are getting more complex and integrated, specially in new aircraft concepts where electrification is a key of the development [4].

An architecture is a model that defines the structure and behavior of a system [5], it is usually represented by its components and connections among them. Properly exploring different architectures of a system during conceptual design allows to increase the design freedom and add more knowledge. This brings strong benefits to the design process [6]. One challenge when designing on-board systems is the vast number of potential solutions, resulting in a huge architectural design space with millions, or even billions, of possible solutions for one design. To effectively explore this design space, some level of automation and filtering is necessary. A system architecting approach is recommended, as it allows for the structured creation and exploration of various architectures without bias [7, 8]. This concept is shown in figure 1.1. It shows the improvements in knowledge that come from increasing the design freedom in early stages of the design process (from continuous line to dashed line).

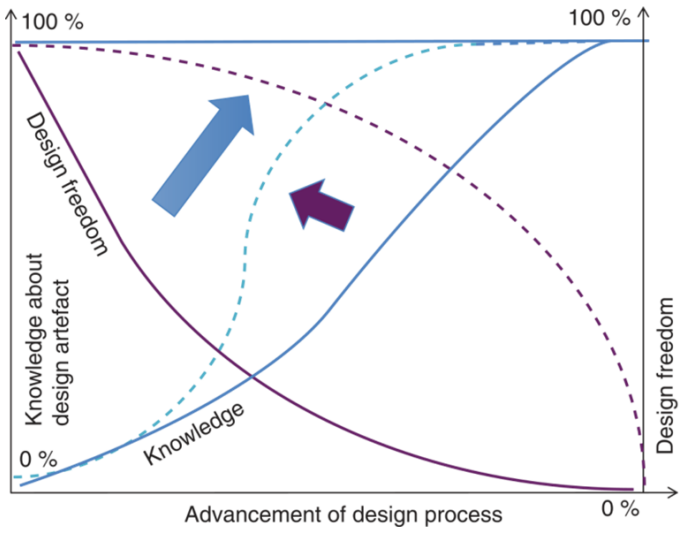


Fig. 1.1 Mitigating the knowledge paradox, from [9].

Another thing that can add knowledge during early design stages is considering the whole product life cycle, as opposed to considering only the performance of a system as it was typically done in the past. This means that all the different

phases from the conceptual design of a product until its retirement are considered. A different breakdown of the life cycle stages can be had depending on the product. However the one proposed by the International Council on Systems Engineering (INCOSE) [10] is used for this analysis. The different stages are proposed as it follows: concept, development, production, utilization, support and retirement [11]. The ISO 14040 for life cycle assessment [12] also emphasizes the need to not consider only a product utilization during design, but to include more external aspects such as recycling and reuse, waste treatment, or energy supply. Other new trends such as the digital twin aim at achieving a digital duplicate of a product throughout its whole life cycle. The implementation of this technology could accelerate the development of new products, reduce the time for their testing and certification, and accelerate the start of production [13]. Digital twin applications could potentially be used during design phases to support later stages such as manufacturing, operations and disposal [14, 15]. Another important concept is concurrent engineering. It is a methodology regarding the designing and development of products that aims to shorten product development time, which is crucial in the aerospace industry. Reduction of times and costs along the aircraft life cycle is a permanent priority. It achieves it by dividing the development process into specific domains or disciplines and creating a multidisciplinary framework where all those domains from the different stages are run simultaneously in parallel instead of consecutively. This decreases the product development time [16–18], also helping to mitigate the knowledge paradox.

The overall objective of this Ph.D research is to improve the knowledge on aircraft on-board systems during conceptual design by simultaneously assessing various stages of the life cycle. During this conceptual phase the architecture of the system is defined and the goal is to assess the impact that this decision has in three different stages: certification, operations and maintenance. The rest of stages are left for further research. Figure 1.2 shows in an schematic way the approach of this study.

The names of the life cycle stages have been slightly changed and particularized for an aircraft in order to be more specific. The concept and development stages are here divided into design and certification. The certification stage is typically included inside development but has been extracted now in order to highlight it. The production phase follows and it usually contains phases such as manufacturing and assembly. The utilization stage is represented by the aircraft operations, while the support is represented by the maintenance, repair and overhaul block. Lastly the

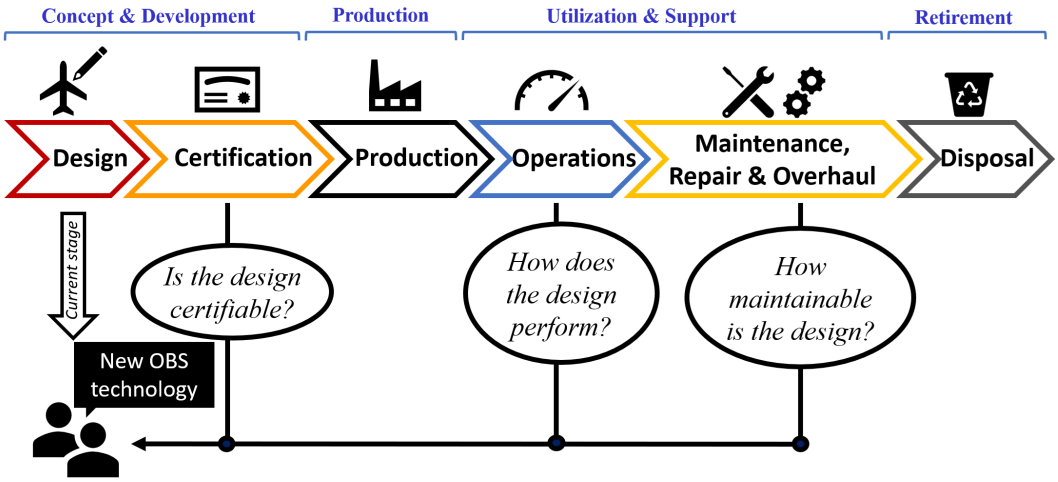


Fig. 1.2 Scheme of the main objective of this Ph.D research, showing the three disciplines of interest from the overall aircraft life cycle. The stages represented on top in blue are as specified by INCOSE.

retirement stage for an aircraft is the disposal. As seen in the figure, this research focuses only on three stages (i.e., certification, operations and maintenance). This three stages are also called disciplines in this manuscript. As a summary, this Ph.D. research focuses on developing a methodology to evaluate the performance, maintenance, and certification aspects of on-board system architectures during the early design stages. This multi-disciplinary analysis provides a clearer understanding of the impact of each architecture by assessing multiple stages of the product life cycle simultaneously. It enables the automated evaluation of innovative architectures by linking their huge design space model to a multi-objective evaluation framework which filters out non-certifiable architectures at an early stage, reducing the design space and saving computational time by preventing unnecessary evaluations.

This thesis is organized in six chapters, and each chapter is further divided into sections. Chapter 1 introduces the research, while chapter 2 shows the state of the art on the main topics, showing the gaps in literature and stating the research questions. Chapter 3 explains the developed methodology with the selected implementation. The two application cases are shown in chapters 4 and 5. Lastly, chapter 6 contains the conclusions of the whole study.

Chapter 2

Literature Review

This chapter contains the literature review. Section 2.1 is focused on the on-board systems definition. Here it is shown how to properly group the different systems into more generic categories, so that these systems can be organized by functionality. A detailed explanation about each of the most common aircraft on-board systems is also presented. Section 2.2 explains the concept of on-board system architecture, and shows some examples. These include architectures from conventional to more-innovative ones, explaining the current tendencies and innovations in the field. Section 2.3 reviews the methods and models used for on-board systems evaluation, regarding performance, certification and reliability analysis, and maintenance aspects. Lastly, section 2.4 summarizes everything and provides, in a structured way, the main science gaps regarding the topic and formalizes the research questions.

2.1 On-Board Systems Definition

An aircraft can be typically divided into three parts: structure (including aerodynamic devices), power plant (engines, propulsion) and systems. For the scope of this thesis, aircraft on-board systems (OBS) are defined as those systems required to operate an aircraft efficiently and safely. This includes the systems that perform a direct aircraft-needed function (such as control the aircraft) or the systems that are needed by other systems (such as power distributors). All these systems are commonly called "subsystems" as well [19, 20]. This notation is used to avoid confusion with the word systems. A wing or horizontal tailplane can also be called system, or even

subsystem. To avoid confusion and narrow things down the term "on-board systems" is chosen and used. This term has also been used in literature [21–25].

Different definitions can be used in order to split the different OBS into categories. The Air Transport Association (ATA) proposed a common and standardized numbering system in 1956 for commercial aircraft documentation. Different parts and systems of the aircraft are here defined and specified in different chapters from 0 to 100. This representation is generally known as "ATA 100". This common reference allows easy and fast understanding and communication about specific commercial aircraft parts, providing benefits for everybody involved (i.e., maintenance personnel, technicians, engineers, suppliers...). For these reasons the ATA 100 numbering system is frequently used in the aircraft-related industry. Regarding OBS specifically, all of them are defined in ATA chapters from 20 to 50. The whole ATA definition for these systems can be found in table 2.1.

The ATA numbering system is well-defined and detailed, however it refers to more systems other than just the OBS and it lacks hierarchical structure. A more generic OBS definition is needed in order to have a more general view and to easily include new systems that might appear as a result of new technologies or aircraft concepts. The proposed definition in this thesis is the one suggested by [26, 20]. Here all the different subsystems (or OBS) are included into three main categories depending on their functionality. This allows a more function-oriented definition that also helps to better understand the on-board systems structure. The definition is represented as follows:

- **Power Generation Systems:** this category includes all the subsystems that generate power. Examples of these are the engines (plus generators), auxiliary power unit (APU) and ram air turbine (RAT).
- **Power Distribution Systems:** these systems are in charge of power distribution and transformation. Their main goal is to distribute power from the power generation systems to the power consuming systems. The most relevant subsystems under this category are the hydraulic, electrical and pneumatic systems [19].
- **Power Consuming Systems (Users)**: this category includes the systems that perform a specific function of the aircraft (e.g., allow communications with ground). They require power provided by the power distribution systems in

order to function properly. These subsystems vary from one aircraft to another and might change widely with new aircraft concepts. Common examples are the flight control system, environmental control system or avionics. As an example, a new system that might be needed in future aircraft is the thermal and energy management system. This system is not needed in commercial aircraft right now but would be needed for example in hypersonic aircraft [27].

Table 2.1 ATA 100 chapters definition for aircraft systems

ATA Number	**ATA Chapter Name**
ATA 21	AIR CONDITIONING And PRESSURIZATION
ATA 22	AUTO FLIGHT
ATA 23	COMMUNICATIONS
ATA 24	ELECTRICAL POWER
ATA 25	EQUIPMENT /FURNISHINGS
ATA 26	FIRE PROTECTION
ATA 27	FLIGHT CONTROLS
ATA 28	FUEL
ATA 29	HYDRAULIC POWER
ATA 30	ICE AND RAIN PROTECTION
ATA 31	INDICATING / RECORDING SYSTEM
ATA 32	LANDING GEAR
ATA 33	LIGHTS
ATA 34	NAVIGATION
ATA 35	OXYGEN
ATA 36	PNEUMATIC
ATA 37	VACUUM
ATA 38	WATER / WASTE
ATA 39	ELECTRICAL - ELECTRONIC PANELS AND MULTIPURPOSE COMPONENTS
ATA 40	MULTISYSTEM
ATA 41	WATER BALLAST
ATA 42	INTEGRATED MODULAR AVIONICS
ATA 43	EMERGENCY SOLAR PANEL SYSTEM (ESPS)
ATA 44	CABIN SYSTEMS
ATA 45	ONBOARD MAINTENANCE SYSTEMS (OMS)
ATA 46	INFORMATION SYSTEMS
ATA 47	INERT GAS SYSTEM
ATA 48	IN FLIGHT FUEL DISPENSING
ATA 49	(AIRBORNE) AUXILIARY POWER UNIT
ATA 50	CARGO AND ACCESSORY COMPARTMENTS

The concept of on-board systems has been introduced. The different systems have been structured and grouped into more generic categories. On-board systems directly affect the aircraft operational empty weight (OEW) and power budget, as stated in [26]. This has an impact on the total fuel consumption, aircraft development and maintenance cost, all of which affect the direct operation cost (DOC). The

contribution is significant and can reach around a 30% of the total DOC value [28]. This explains the relevance of properly studying and evaluating the OBS since their impact on the aircraft design is not negligible. In the following subsections the OBS structure is further investigated going more into detail for the most common aircraft on-board systems, providing examples for each. After this, some concepts of on-board system architectures are provided to the reader for a better understanding of the connections and links among them.

2.1.1 Power Generation Systems

These systems are in charge of the power generation for the rest of the OBS, they are also commonly called prime movers. The engines (together with their corresponding generator and pumps) are mainly in charge of this function since they provide most of the power utilized by the aircraft. Some systems serve as back-up power sources when the engines fail or cannot be used [26]. One example is the auxiliary power unit (APU), a fuel-based power source used to provide energy for non-propulsive related functions. Another example is the ram air turbine (RAT), a small wind turbine used in emergencies. This device is deployed and generates power from the airstream caused by the speed of the aircraft. It is usually connected to a hydraulic pump or to an electrical generator. Batteries can be also fit under this category but are generally considered as part of the electrical system since they provide secondary power from a consumable source [29] and their low utilization time does not allow them to power the aircraft systems for long periods. Lastly, an example of a new technology that could become a main power source for future aircraft are the fuel cells [30].

2.1.2 Power Transformation and Distribution Systems

These systems are in charge of taking the energy provided by the power generation systems, transform it and distribute it to the users. For most aircraft these systems are the hydraulic system, electrical and pneumatic. Some authors also consider a mechanical system [20], but this can also be considered as part of the corresponding users. For example, a flight control system with mechanical connections (pulleys, cables, levers, and rods) can be considered to have a mechanical distribution system, or these devices can be considered as part of the flight controls themselves. The same can be applied to the high-lift devices that have mechanical shafts and gearboxes

connecting the different actuators. A common and central mechanical distribution system is not present in any case, so it is not considered as a distribution system itself. The three distribution systems are now explained.

Electrical System

This system transforms the mechanical torque from the engine into electrical power thanks to the engine-driven generators, which are commanded by generator control units. It can also get energy from the APU generator, RAT generator or the batteries. The different users receive the corresponding power through different electrical lines. Another function of this system is to adapt properly the current from the power source to the user. The users can require power in alternating current (AC) or in direct current (DC), and at different voltage levels. Components such as the transformer rectifiers or static inverters are needed to change and adapt the current correspondingly for each function [31]. As an example to the reader, a conventional aircraft generator (with an integrated drive generator) can generate power at 115V AC [32], or it can also generate at 28V DC. New engine designs based on permanent magnets can generate at 230V AC [32], or even at 270V DC. The current needs to be later adapted to be supplied to the users. Some examples of users can be the avionics, which typically work with 28V DC or 115V AC. Other examples of users are the cabin lights, fuel system pumps or certain components of the ice protection. Depending on the users and the generation, different current conversion and transformation strategies must be implemented. Other components that can be found are the primary and secondary power distribution units, and the cables.

Hydraulic System

The hydraulic system transfers hydraulic power around the aircraft. Some main tanks store the hydraulic fluid, which is moved by hydraulic pumps though a system of pipes, filters and valves. Accumulators are used in order to provide extra power in certain situations, such as emergencies. Some aircraft use power transfer units to transfer the fluid from one line to another and increase the level of redundancy. The pumps can be engine-driven pumps or pumps connected to other prime movers such as electric motors or the RAT [33]. Some users of this system can be the actuators

such as the ones for the flight controls, braking, landing gear retraction or cargo doors.

Pneumatic System

The pneumatic system extracts hot and pressurized air from the engine through a bleeding, and then supplies it to the pneumatic users. This air can be provided to the environmental control system, to be used to condition the cabin, or to the ice protection system to avoid ice formation in certain areas. This system is mainly composed of pipes and valves that carry the airflow from one point to another. The engine bleed penalizes the specific fuel consumption, raising slightly the fuel burn. For this reason other concepts take the air from air inlets instead of from the engine, extra compressors are needed in this case and the aircraft drag is slightly penalized as a result.

2.1.3 Power Consuming Systems

The main power consuming systems of interest for commercial aircraft are here briefly introduced and explained to the reader.

Flight Control System (FCS)

The flight control system includes all the aircraft control surfaces such as the ailerons, elevator, rudder and spoilers. These surfaces were typically moved by mechanical pulleys in the past but have evolved and are mostly powered by actuators in general aviation nowadays. The actuators can be from different types but these are commented later together with the architectures examples. This system also includes the high-lift devices (i.e., flaps and slats), with all the components involved in the movement of such surfaces. This OBS is used in the application cases in chapters 4 and 5 and is explained into more detail there.

Landing Gear System

This system includes all the landing gear functions, which are extension, retraction, wheel steering and wheel braking. New concepts such as the electric taxi system can

also by included here. Regarding its main components, these include the different actuators, valves and friction disks, among others.

Environmental Control System (ECS)

This system takes air from some source and prepares it to meet the necessary conditions inside the cabin (i.e., correct temperature, humidity and pressure). It is composed of a series of pipes, air packs, mixers, valves and manifolds. Different strategies can be used in order to control the air parameters (e.g., simple, three wheel or four wheel bootstrap cycle [34, 6]). Other concepts such as the vapor cycle can be used to control the air conditions in the avionic bay [35]. New studies tend towards concepts in which the air is not taken from the engine, hence not penalizing fuel burn [36].

Ice Protection System (IPS)

The ice protection system prevents ice formation in certain areas of the aircraft. These areas are mainly the wing, engine nacelles and tail, but also include small zones such as probes. Conventional aircraft use the air from the pneumatic system in order to warm up those regions and melt the potential ice formation. Other conventional solution consist of inflating certain areas with cold gas to directly break the ice formation [37]. New concepts use electrothermal (i.e., heating the surface with electric resistances) or electromechanical (i.e., vibrating the surface with small actuators) solutions to avoid icing [38], and these do not need a pneumatic system in order to do it. Other concepts are trying passive ways of achieving this like for example covering the surfaces with anti ice coatings or sprays [38], this could potentially completely remove the IPS.

Fuel System

This system is in charge of providing fuel to the engines during the whole mission. It achieves this by storing the fuel in different tanks and transferring it with valves and pumps. If an APU is present this system must also provide the necessary fuel to it. Fuel pumps are usually electrical and redundancies are achieved with cross-feed valves among tanks [39]. This system is also in charge of controlling the center

of gravity with the fuel in certain aircraft concepts. This system might completely change in future aircraft that might not use fuel (e.g., electric aircraft).

Avionics

The avionics are the electronic devices that allow the aircraft to perform key functions such as communications, navigation, displays or management of systems. They vary from one aircraft to another depending on the mission and are electrically powered. They can include devices as for example the GPS, radio, autopilot, cockpit displays, traffic alert and collision avoidance system, flight recorders, health and usage monitoring system, among others.

Other Systems

Other OBS can include for instance the fire protection, cabin lights, oxygen, water waste, in-flight entertainment, galleys, or furniture. Other OBS is for example the thrust reverser, which can be considered as part of the engine system, or as part as the flight control system, or the braking system or as a separated system itself. New systems that might appear in new aircraft concepts such as the thermal management system could also be added in the future.

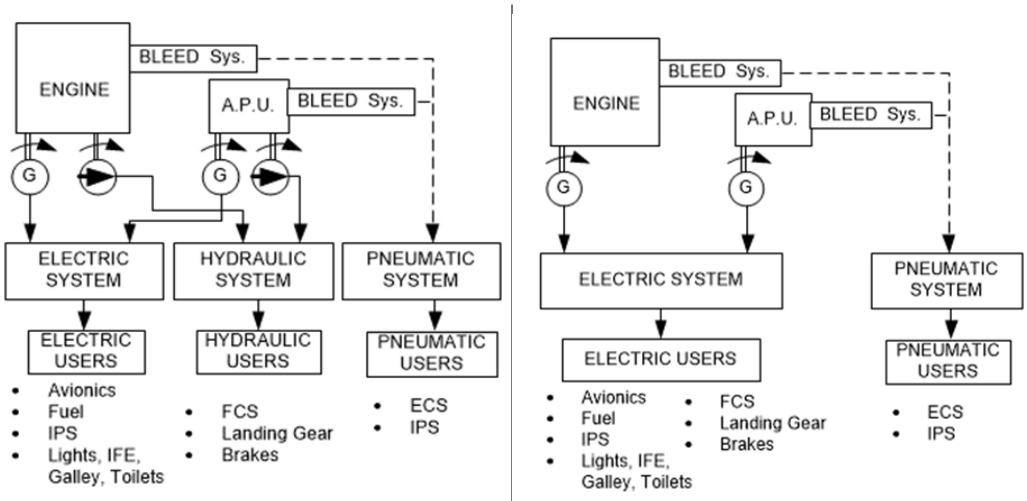
2.2 On-Board System Architectures Examples: Conventional, More-Electric & All-Electric Aircraft

The importance of on-board systems as part of the aircraft has been highlighted in the previous section. One initial and key part of OBS design is the architecture definition of the system of interest (e.g., FCS or ECS). In the early stages of the design of any complex system, the architecture of that system is defined [40]. The decisions taken in this stage highly impact the final performance of the product [40]. As defined by [5]: "A system architecture is the conceptual model that defines the structure, behavior, and more views of a system". It is basically a formal description of the system that allows to better model and structure the behavior of it. A system architecture consist of all the system components and subsystems that function in order to characterize the overall system [41], this includes the connections among

them and the redundancies. Architecting is defined as "the process by which a system is created, designed, built" [42]. System architecting can be seen as one of the phases in the whole systems engineering process [7] and its relevance cannot be neglected [8]. As stated by [42]: "The success or failure of many civil and defense systems depends mainly on their architecture". Regarding on-board systems, the architecture definition is as a result a key part in the feasibility and success of the system.

Some examples of on-board system architectures are now shown. These examples are not too detailed and represent just a high-level concept of the whole aircraft OBS architecture. Figure 2.1a shows a conventional OBS architecture. This kind of architecture can be found in aircraft such as the Airbus A320. The reader can see how the different power consuming systems are powered by the typical three power distribution systems. The engine and APU work as power generation systems. Moving to figure 2.1b the reader can notice how the hydraulic system disappears in this more-electric architecture concept. This comes as a result of fully electrifying the flight control system, landing gear and brakes. The advantages of this architecture concept rely on the removal of the heavy hydraulic components, which causes a benefit to the overall mass, even if the electrical system has now a higher power demand [24, 43]. This can also lead to advantages in maintenance. Figure 2.2b shows the concept of the all-electric OBS architecture. Here not only the hydraulic system disappears, but also the pneumatic system. Now there is no direct bleeding to the engine and the pneumatic users are powered by external compressors. This benefits the fuel consumption, since no bleeding is now penalizing the specific fuel consumption. However the external compressors need new inlets in the aircraft structure that potentially penalize the aerodynamics, increasing the drag [20, 26]. This trade-off is key in order to assess the benefits of this new concept. Finally, another more-electric architecture is shown in 2.2a. In this case the hydraulic system exists but is now powered by the electrical system [24, 43]. Several studies have been conducted in order to further investigate these four OBS architecture concepts in terms of performance [23]. Other studies have started assessing other domains as well, such as certification or noise [3].

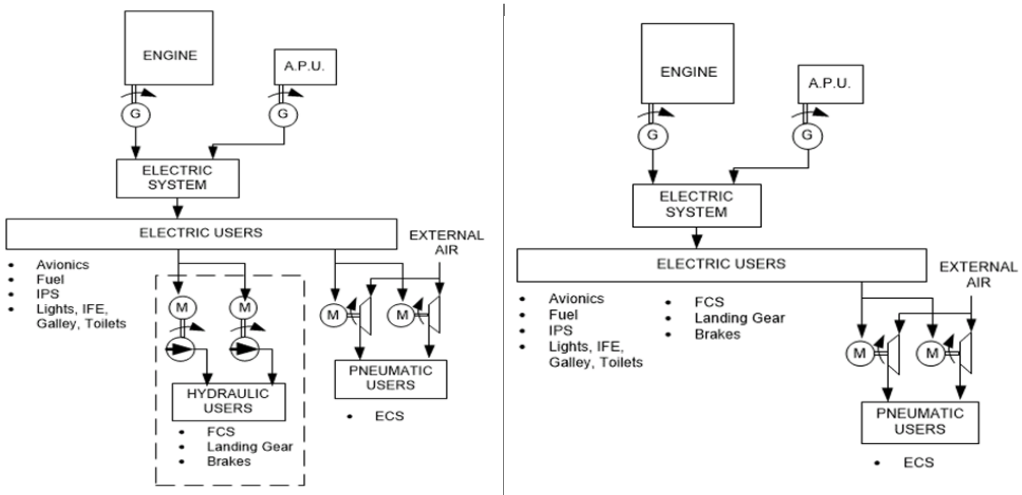
The concepts of more-electric aircraft (MEA) and all-electric aircraft (AEA) have been directly introduced though the previous example. These concept architectures represent the current innovation tendency for aircraft OBS. MEA architectures are found in some commercial aircraft (e.g., A350, B787, A380). However a full removal of the hydraulic system is yet to be achieved, the latest aircraft have achieved only



(a) Conventional architecture

(b) More-electric architecture, example 1

Fig. 2.1 Conventional and more electric OBS architectures, from [43]



(a) More-electric architecture, example 2

(b) All-electric architecture

Fig. 2.2 More-electric and all-electric OBS architectures, from [43]

a reduction of the hydraulic system (e.g., the Airbus A350 has two hydraulic lines instead of three like the A320 thanks to the electrification of some part of the FCS). Before explaining the advantages of these concepts it is important to briefly comment some of the most important enabling technologies for such concepts.

One of the main enablers are the actuators. Conventional actuators for flight controls or landing gear consist of a hydraulic actuator that receives hydraulic power

from a central hydraulic line. These are typically called hydraulic servo actuators (HSA) [44] and they require the presence of a hydraulic system in order to function. One type of more-electric version of an actuator is the so called electro-hydrostatic actuator (EHA). This actuator is also based on hydraulic power but each actuator has its own local hydraulic deposit that is powered by a local electrically driven motor [19, 44]. This allows the actuator to function without a central hydraulic system, it only needs a source of electric power to activate the electrical motor, move the local hydraulic fluid and, as a result, move the actuator. This technology is already being used in commercial aircraft for primary surfaces of the flight control system, and it is claimed to be easier to maintain than conventional ones [45]. Another similar actuator is the electric backup hydraulic actuator (EBHA) [19, 44]. This concept has two functioning modes. In normal operation it functions as a HSA, receiving power from a central hydraulic line. In case of failure of the line it enters backup mode and a local unit provides power as for the EHA. This concept is used to increase the reliability of the actuator through redundancies, however it does not manage to remove the need of a central hydraulic system. This type of actuator is used in current aircraft such as the Airbus A380. Lastly, electro-mechanical actuators (EMA) [46] are fully electric versions of actuators. They consist of a ball-screw assembly connected to a local gearbox assembly powered by an electric motor. Hydraulic components are completely removed. These actuators are currently being used in commercial aircraft for spoiler actuation and other secondary functions such as cargo doors or thrust reversals.

Some commercial aircraft have been mentioned before. The Airbus A320 uses HSAs for all the flight control system actuators. The Airbus A350 moves towards a more-electric concept in which some actuators from the ailerons, rudder and elevator are EHAs, and two spoilers are actuated by EBHA. A similar trend is seen in the Airbus A380, with EHAs and EBHAs being also used. EMAs are also currently being utilized, both A350 and A380 use them in the thrust reversal [44]. Another example of more-electric vehicle is the Boeing 787 which utilizes a bleed-less ECS concept and electro-thermal IPS [38]. It also utilizes the EMAs for the actuation of some spoilers [47]. EMAs are also expected to be used for commercial aviation in landing gear functions such as steering, but the maturity level is still not high enough [48].

Other enabling technologies of the MEA and AEA concepts are those related to the electrical system. Old studies [49] indicated the following key OBS concepts

to enable a MEA fighter: internal engine starter/generator, electric primary flight control actuation, integrated emergency power unit and a fault tolerant electric power management and distribution motor drive system. These technologies are currently being implemented and the most important challenge at the moment is improving the power electronics [50]. One of the main improvements is being able to generate power at 270V DC [51], this would translate in a potentially lighter electrical system since AC systems require larger cables than DC ones. Furthermore, DC systems are found to be robust enough for modern and future aircraft systems and are considered high performance power electronic systems. The needed conversion for the users from a generation at 270V DC to 28V DC, 230V AC and 115V AC is possible and guaranteed with current devices [52].

The advantages of the more and all-electric concepts reside mainly on the weight reduction that they can achieve. Removing the need of a hydraulic and/or pneumatic system can translate in noticeable savings for MTOM. Also the more efficient use of electrical energy means that these new systems can enable future aircraft to be quieter and more fuel efficient [50]. Also, the need for a bleed air system disappears without a pneumatic system, this leads to a significant improvement in the efficiency of the turbine [50, 53] and in the fuel consumption as a result. Some authors have estimated a potential benefit of a 6% weight reduction for an optimized MEA architecture [54], although other studies have more mitigated expectations [24]. In any case a reduction in weight is always achieved [55]. Hence, hydraulic systems are heavy owing to the needed infrastructure (e.g., piping), also they have the disadvantage of the potential leakage of dangerous and corrosive fluids [56]. Electrical power does not require such a heavy infrastructure. However, a big disadvantage is that it provides a lower power density and manifests a higher risk of fire (in the case of a short circuit) [56]. As a result the objective for future aircraft is to replace most of the main OBS that are currently non-electrically powered (e.g. environmental controls and engine start), with new electrical ones. This could improve a variety of aircraft specifications apart from mass, such as emissions, efficiency and reliability [57]. Maintenance cost is also expected to be reduced [58, 31], as well as some other stages of the life cycle [59]. As a final remark, it is generally acknowledged that the big benefits of these new technologies will be reached with the full electrification of the whole systems (i.e., AEA) [20]. However, a gradual and progressive electrification is preferred by industry for a better and smoother transition and risk [20].

2.3 On-Board Systems Evaluation

Regarding aircraft OBS most analyses focus only on performance aspects (e.g., mass, power, fuel needed). However this analysis is not complete since a system can have excellent performance and at the same time a high development and/or manufacturing cost. Maintenance cost also depends on the components (e.g., actuators). All these factors affect the direct operating cost. Accounting for the impact of these domains during early stages of design of aircraft development could give a better insight of how feasible a system is. Several studies have emphasized the importance of assessing more life cycle stages rather than just performance. Some specific examples are product cost management [21], safety [20], or RAMS (Reliability, Availability, Maintainability and Safety) [60]. As said by [60], expanding the knowledge on RAMS analysis during conceptual design "helps the designer in the trade-off of the system architecture and technologies, reduces the cost of product development and the time to market". It is then generally known that adding more knowledge about the different life cycle phases is a must in future analyses. In terms of maintenance some studies focus on making comparison among conventional and innovative components, but most rely on qualitative results and almost none assesses the impact at aircraft level. For certification, barely any information is available and most studies just refer to standards without providing specific examples. Each of the three disciplines are explained with more details in the following sections, adding more knowledge to them.

Focusing now on preliminary design of OBS, the first step is to define the possible architecture. Typically, only a limited set of pre-defined architectures is evaluated when analyzing on-board systems [20, 26, 61]. Sometimes the architecture generation is even based on boolean decisions, not reaching component information but staying on a very high-level definition [62]. This leads generally to results based on weighted values based on such decisions, not on results from simulation. Other methods generate architectures with more details but always based on some parameter selection, and not on the system architecting principles [63]. From these small number of options, one architecture is selected based on which best meets the design objectives and provides optimum results. In these cases, each architecture can be evaluated individually, ensuring that all devices satisfy the redundancy requirements specified by the designers. However, future analyses should consider a broader exploration of architectures. This expanded approach would reach a better and

broader exploration of the solutions, potentially leading to novel, more optimal designs for on-board systems architectures that go beyond the limitations of prior studies.

Other studies focus on a very detailed description of the architecture, reaching component level. But no design space is provided [64]. The vast number of possible configurations while defining on-board systems connections, and thus their architectures, must be accounted for. However, evaluating all potential solutions is not viable. Optimization algorithms are commonly applied to help identify promising solutions without the need to examine every possible solution. To reduce computational times and ensure that only viable and interesting architectures are assessed, effective pre-filtering is essential, this allows the exclusion of architectures that do not meet connectivity or redundancy requirements. As a result, assessing numerous architectures requires model automation, meaning that both the tools and the links between them must be automated to facilitate the evaluation of many configurations. Although semi-automated methods for generating on-board system architectures have been developed in the past [65], they have since evolved to try to achieve fully automated models [66]. However, this is still a gap in literature, as the automated evaluation of multiple automatically-generated architectures has not yet been fully realized, creating a disconnection between architecture generation and evaluation [55, 67].

The motivation to increase the level of detail of the OBS architectures during preliminary design was enhanced after the realization of the AGILE 4.0 project [68]. In this project only a small number of interesting architectures were evaluated in some application cases (one conventional, two MEA, and one AEA). The concept of the four of them came from experience and previous knowledge. Increasing the design space and expanding the knowledge by adding more life cycle stages was highlighted as a further need. Each of the three disciplines of interest are further developed in the following sections.

2.3.1 On-Board Systems Evaluation: Performance

The first discipline that is analyzed is performance (as representative of the operations life cycle stage). Regarding OBS this discipline can be summarized as and estimation of the mass and fuel penalties that the specific system has. Mass directly affect the

aircraft maximum take-off mass (MTOM) [19], fuel penalty is directly affected by the power off-takes and engine bleeding that the OBS need [43]. Some systems can also have an impact in the aircraft drag [26, 63]. Several methods and tools exist in literature to estimate these parameters and can be grouped in different levels depending on their level of detail and scope.

These levels are commonly called in literature as Level-0, Level-1 and Level-2 [69, 70]. Level-0 methods consist of relatively simple empirical estimations for the OBS mass and power consumption. In general they provide linear equations for each system, such as the ones proposed in [71]. These equations depend on different aircraft parameters. For example the landing gear mass depends on the aircraft type, landing weight and lading gear length. The fuel system mass depends on the fuel weight, maximum fuel capacity and the total number of engines. All the systems mass consist of a linear equation that provides results based on statistical analysis of different aircraft. These type of methods are able to provide fast results, however they cannot be used to assess novel technologies, and they do not reach component level. Other most common and well-known examples of Level-0 methods are for example the Torenbeek [72], Roskam [73] or the most modern Raymer [74].

Level-1 methods follow a more detailed approach usually based on more advanced physics [69] including component scaling regression equations (e.g., power-to-mass ratios [75]) or linear simulations (e.g., linear time invariant simulations). The design method consist of modeling the power consuming systems first from some top level aircraft requirements that come from the aircraft definition and mission analysis (e.g., take-off, climb, cruise, descent, and landing). Then the distribution systems are estimated based on the power requirement needed by the power consuming systems (i.e., hydraulic, pneumatic and electrical). The power required by the distribution systems has an impact on the power generation systems (i.e., bleeding and off-takes). Different sizing cases must be assessed such as one engine inoperative or one generator inoperative [26]. Depending on the OBS architecture, the mass and power requirements change and this has an impact on the aircraft. This method successfully evaluates innovative architectures and can reach component level, even accounting for individual component efficiencies. This methodology has already been used in literature [26, 70, 63], even for the evaluation of MEA and AEA concepts [76].

Level-2 methods achieve a more detailed design, even at conceptual stages. They are based on complex geometry based approaches that can even estimate the

dimensions of the components. Parameters such as material properties and densities are hence needed. Complex non-linear equations (e.g., non-linear time invariant simulations with Simulink or Modelica), computational fluid dynamics calculations or finite element models are sometimes needed and used. As a result extensive system knowledge is required. Some Level-2 methods use open libraries for components such as the ones in the software Modelica. Sizing methods do not stick to just one level and can mix different levels depending on the application case. As an example, ASTRID [70] is a tool for OBS that allows to select Level-0 or Level-1 per each on-board system. Some OBS can even reach Level-2, such as the environmental control system [77]. For instance, the environmental controls can be sized with a lot of detail following a Level-2 approach, then the rest of the user can be following lower levels. Systems without big architecture impacts are typically sized by level-0 methods (e.g., avionics), while other more impactful systems might need higher levels.

One important factor that needs to be addressed during design is that the implementation of a new technology at a OBS level has an impact at an aircraft level, and this might be difficult to assess [78]. This is generally known as the the snowball effect. This was already emphasized in previous studies [24] and it is explained in this paragraph. The estimation of the OBS mass usually depends on different TLARs, an initial estimation of the MTOM is needed in order to evaluate systems such as the landing gear (the wheel and strut sizing depends directly on the aircraft landing weight). After the OBS are sized the aircraft can be sized. A better estimation of the MTOM is now available and can be used to re-size again the OBS. This initiates an iterative loop that can be stopped once convergence is found. This iterative design process between OBS design and aircraft design has been addressed in other studies [6, 68]. The theoretical explanation of this effect is that when a component (or system, or part) increases its weight, the overall impact on the global mass is higher than just that increase. This comes from the fact that when component is installed, the structure that holds that component needs to be reinforced accordingly, increasing the mass of the structure itself. For this reason an improvement in a subsystem mass can potentially have a higher impact on the total mass reduction. Some studies have analyzed this effect [3].

Fuel burn is also affected in two ways. From one side the change in aircraft weight has a clear impact on the fuel needed to perform the mission. From another side the fuel consumption is affected directly when bleeding and off-takes are taken

from the engine. On-board systems are typically powered by engine off-takes and/or engine bleeding [23]. The higher they are, the higher the penalty on fuel consumption. If a system requires a lot of power off-take from the engine, it seems reasonable to think that the amount of fuel consumed from it would be high. Some new OBS architectures try to minimize these penalties [23, 43].

Lastly, it is important to assess the change in the OBS as a whole, even when just one is changing. For example, just changing the FCS has an impact in the distribution systems. But it is not until the landing gear is also electrified that the hydraulic system can be removed. This effect must be accounted for and was already emphasized in [79]. The benefits of electrifying the FCS and the landing gear together are bigger than the sum of electrifying both separately. To conclude, the performance discipline has already been studied quite extensively in literature and several tools and methods are available.

2.3.2 On-Board Systems Evaluation: Certification

Certification is an important step of a product life cycle. A technology with the best performance cannot be used if it does not pass the certification requirements. Adding certification aspects in early stages of OBS design can provide interesting insight and remove from the analysis some architectures that do not meet the minimum requirements to be certifiable. The main step of the certification process is to meet the requirements given by the certification specifications. For commercial aircraft these are specified by EASA (European Union Aviation Safety Agency) in the CS-25 [80], or in the CS-23 [81] (for aircraft lighter than 5700 kg or 8600 kg depending on the category). These specifications are complemented by the acceptable means of compliance (AMC). Other regulatory entities such as the FAA (Federal Aviation Administration) in the USA propose similar standards.

The main issue with the certification specifications regarding OBS architectures is that they do not provide specific guidelines per each system, making it difficult to understand which requirements to apply. For OBS most of the requirements are too detailed (e.g., maximum temperatures or loads in certain components), or too vague (e.g., saying that redundancies are needed but not providing the minimum number of redundancies needed for a component without needing to perform a RAMS analysis). This is generally not usable at early stages of design, however the objective of this

analysis is to find the requirements that have an impact at an architecture level. For the OBS the most important one is the minimum required reliability. Certification authorities establish a minimum reliability threshold for a system that has to be guaranteed through the use of redundancies. These redundancies and connections significantly impact the resulting architectures of the OBS. Certification authorities provide rules, guidelines and requisites for the safe and correct functioning of such systems, however requirements about the connections themselves are not explicitly specified. Some guidelines regarding connectivity rules can be inferred from other requirements or found in appendices or AMCs. A detailed analysis is key to be sure that all the necessary conditions are achieved.

As said, proving compliance with the certification specifications is a must but can be difficult owing to the vague statements and non-categoric rules thereby specified. For this reason SAE International developed some standards that help and aid on this task. These technical standards are the ARP-4754 [82] (Guidelines for Development of Civil Aircraft and Systems) and ARP-4761 [83] (Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems and Equipment). Both are aerospace recommended practices that, used in conjunction, demonstrate compliance with the FAA airworthiness regulations for transport aircraft, and they are also harmonized with the ones from EASA. Hence the advantages are clear. Certifying a system following the ARP documents serves as means to prove the airworthiness for both entities and provides more specific guidelines than the certification specifications. Even aircraft software is designed following these standards [25]. ARP-4754, deals with the development processes that can be used to support certification of aircraft systems, they are intended to be used for the whole development cycle from system requirements through systems verification. ARP-4761 defines a process to assess the safety of a system with common modeling techniques.

These techniques are RAMS (Reliability, Availability, Maintenance and Safety) methods [84], mostly focused on safety analysis. They are parts of a systematic approach that can be executed for a specific architecture of a system. Some parts of this process are now presented as an example:

- FHA (Functional Hazard Assessment): this method identifies the main failure modes of a system and assigns some level of severity and risk to them. It can be considered as a qualitative method and can be difficult to automate.

- FTA (Fault Tree Analysis): this method aids to define the architecture of a system and its mechanisms of failure. It can be used to allocate safety budgets via top-down approach [85] or to define the probability of a failure event, via bottom-up approach. An example of this method applied to an electrical system can be found in [86]. The main disadvantage resides in its difficulty to be automated and executed for a generic architecture.

- FMECA (Failure Modes, Effects and Criticality Analysis): serves to identify possible occurring failures for each component. This method requires a lot of detailed information that is not yet available at this stages of design [87].

- ZSA (Zonal Safety Analysis): used to ensure that the equipment installed in certain zones of the aircraft meet adequate safety standards and does not interfere with the safe functioning of another system (e.g., check that if several tubes pass through a common area a leakage of one might influence the others). This kind of analysis is performed at later stages, once the dimensions and positioning of components are known [88].

The results from all the different analysis converge into a final safety assessment, applied to the whole system [60]. All these methods are used to check that the design is robust enough, meaning that there are enough redundancies to ensure the safe operation of the aircraft under critical failure scenarios [89]. The main issue relies in that many parts of the process provide qualitative results, cannot be easily executed for a generic architecture definition [90], or need information that is not available during preliminary design. This creates difficulties to automate the process and limits its applicability in early design phases.

Other current lines of research are those that try to link safety assessment with other analysis, such as model based safety assessment [60, 91, 92], through SysML modelling [93], or establishing a link between RAMS and prognostics and health management [94]. This relies on the monitoring of system parameters to be able to identify the causes of failures [94]. Another important aspect for future safety analysis is the capability to reach component level. An architecture definition must describe the system, subsystems and components and reach each of them individually. This allows to identify better the critical components with higher failure rating and take action (e.g., increase redundancies around them). Old RAMS methods were

based on top-down approaches based on allocation [95]. Component level was not reached, they consider the subsystem as a conjunct and not as a group of individuals.

The literature [20] and the technical standards [82] emphasize the need of evaluating systems performance and RAMS analysis since the early stages of design. However, as said before, these methods do not provide simple quantitative results, they need a lot of detailed information and cannot be automated, needing human supervision in a lot of steps. This makes it difficult to integrate them in conceptual design phases. For this reasons they are not considered for this study. However, without a certain level of detailed information it is not possible to perform any kind of certification or safety analysis. A method that achieves a middle point between too detailed and simple enough is the reliability block diagram (RBD). The RBD technique works perfectly under this context since it manages to represent components and connections among them, and only requires failure rates as an input, which is quite reduced. It also allows automation, hence it was chosen for this study for all these reasons.

A reliability block diagram is a graphical representation that shows the logical connections among the components of a system [96]. RBS are standardized and can be solved only by knowing the component connections and their failure rates. Several books and studies explain how to build and solve them [97, 98]. There are even formal technical standards for RBD [99]. The main issue when solving them is that there are configurations apart from series and parallel, such as the r-out-of-n configuration [100], which makes it difficult also to build them. Non-conventional RBDs configurations are very common on aircraft systems and hence must be accounted for in this study. There are several ways and methods to solve RBDs [98] such as the minimal cuts [101, 102], minimal paths [103], or Markov [104]. However these methods do not allow to perform automation in an easy way when the non-conventional configurations are involved. The complexity of their definition needs human supervision in some steps and do not manage to automatically read, reduce and solve generic RBDs, which is a must of this study. Also the RBDs become quite difficult to understand. Other studies have also tried to solve this problem. A method was proposed in [105], which allows to solve in an automated way one non-conventional RBD configuration. However, this method is based in the making of a matrix which complexity increases exponentially with the number of components. This makes it unfeasible for huge design spaces. Other studies tried to also automate the RBD solving [106], but only accounting for the series and

parallel configurations. Some examples of RBDs for aircraft systems are present in literature such as the one in [107], but there is no link to the distribution systems, which makes it incomplete for this analysis. Nevertheless, a new issue appears when the distribution systems are included, which is repeating components in the same RBD. This provides false results as stated by the standards [99]. As mentioned, such mistake is quite common when a system is represented together with the corresponding distribution system (e.g., a FCS connected to the hydraulic system) [84, 54] since the number of components grows and connections start to be difficult to manage. A new method is proposed in the methodology which solves the issues here presented without increasing the required input and complexity.

Therefore, there are two main inputs that must be provided to solve a RBD. One is the connection among components, the other are the components failure rates. The estimation of the failure rates is key for the correct modeling of the system. However, these values are extremely difficult to estimate since they generally require experimental data and testing. Furthermore, these are generally sensitive data that companies do not share with the public. A common way to model failure rates is the Weibull distribution [100]. This distribution allows to model component failure following different curves that can be modified by just adjusting three parameters. The equation that represents the model is shown in equation 2.1:

$$F(t) = 1 - e^{-(\frac{t-x_0}{\alpha})^{\beta}} \tag{2.1}$$

Equation 2.1 shows the three-parameter Weibull cumulative distribution function, where:

- $F(t)$ is the probability that a random unit taken from the population of a certain component fails at a certain time t
- β and α are correspondingly the shape parameter and the characteristic life, both parameters that adjust the curve shape
- x_0 is the location parameter [108], or offset [109]. It defines the location of the origin of the distribution in time and for this analysis it can be assumed as zero

Modifying these parameters allows to create different models. Typically, these curves are modeled following bathtub curves [109], which represent three separated

regions. Initially there is an infant mortality region, here the component fails more than usual owing to undetected manufacturing defects. Then the curve follows a constant value that corresponds to random failures. Lastly, the probability of failure increases after a certain time due to reaching a wear-out region. Figure 2.3 shows different distributions that can be modelled. Curve A shows the three regions previously commented. Curves B and F are representations or just two of those regions. Curve E shows the case in which the curve becomes an exponential, meaning that only random failures are represented.

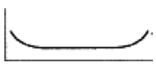
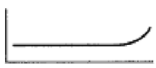
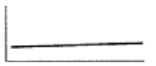
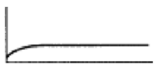
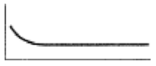
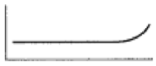
Wear-out characteristic			Prevalence among aircraft components	Best mathematical model
A		Bathtub curve	4%	3 Weibull distributions
B		"Pronounced wear-out region"	2%	Weibull
C		"Gradually increasing"	5%	Weibull with extended tail
D		"Low... followed by a quick increase"	7%	Weibull
E		"Constant probability of failure"	14%	Exponential
F		"Infant mortality"	68%	Weibull
G		Pronounced wear-out region characterized by fatigue life	Most structural components	Weibull

Fig. 2.3 Example of different age-reliability patterns for aircraft components represented by Weibull distributions, from [109]

Weibull distributions are quite flexible and can be tailored by adjusting the correspondent parameters. The main problem is that adjusting β and α can be difficult [100]. For aircraft OBS, it is reasonable to assume that manufacturing defects are detected prior to the utilization of the system, and that the system is replaced before reaching its wear-out region. This leads to a constant region characterized by random failures, which corresponds to a β parameter equal to 1 [110] and an α parameter equal to $\frac{1}{\lambda}$. As a result the the Weibull function becomes an exponential with constant failure rates (i.e., λ) [110], as shown in curve E of figure 2.3. These assumptions mean in practice that the components are located in the second part of the bathtub curve. Here production defects are not present

anymore, the component is substituted before entering the wear-out region and the failures are mainly random [96].

Other more sophisticated methods consider that there might be components in redundancy that do not decay (i.e., the time does not pass on them until the component is needed and starts to be used). These methods allow to stablish warm and cold redundancies depending on the strategy used for that component [111], and they are specially useful when analyzing the systems fault-tolerance (i.e., capability to function even when a fault is present [112]). This is also useful to analyze the operational reliability of a system, but for conceptual and preliminary design assuming constant failure rates does not have a huge variation of the results [110]. Concluding, all the previous complex methods for failure rate modeling are useful to evaluate transients states of the component, for conceptual design assuming constant failure rates provides valid enough results [110].

Most studies that provide failure rates for aircraft systems give constant values for each component. Some examples for FCS can be found in [54, 113, 114], for electrical system [115, 116], other common components such as pumps and motors [117], or even at subsystem level [118]. These values differ from each other sometimes and are extracted from other studies or analyses. In order to maintain a common source for all the components that are considered in this analysis, the "Quanterion Automated Databook: Non-electronic Parts Reliability Data (NPRD)" from 2016 is used. It provides failure rate data for a variety of components including mechanical and electromechanical assemblies. It gives detailed data sorted by part type, quality level, environment and data source. The data here contained represents a compilation of military, commercial and industrial applications. It includes part descriptions, quality level, application environments, point estimates of failure rate, data sources, number of failures, total operating hours, miles, or cycles, and detailed part characteristics. The exact values that are used are reported in each application case chapter.

Summarizing, the certification discipline is preliminary assessed by checking the compliance with the certification specifications. The most important and difficult check for conceptual design is the safety of the system (i.e., checking that it meets the minimum reliability required). This can be done by using the reliability block diagram technique, which only requires the connection among components and their failure rates to be done. This approach yields quantitative results, potentiating

its suitability for integration into an optimization framework and perform trade-off analysis with other disciplines. Automation is also achieved thanks to the methodology proposed in the following chapter. As a conclusion, the exploration and utilization of the certification discipline during early design stages is a current gap in literature, and one focus of this study. Other recent studies are investigating this line of research as well [119].

2.3.3 On-Board Systems Evaluation: Maintenance

Maintainability is an important aspect of aircraft design. Maintenance can represent typically between a 10% and a 25% of the direct operating cost (DOC) of an aircraft [120–122]. Design for maintainability shall consider aspects including assembly, visibility and accessibility, among others, needing a high level of detail in order to be fully addressed. As a result, it can be a challenging task to be carried out at early design stages [123]. Maintenance aspects are linked to RAMS analysis, as explained before, where concepts such as operational reliability can be used to to support maintenance planning [124]. Some maintenance standards exists such as the MIL-HDBK-472: maintainability prediction [125], which aims at providing a list of current maintainability prediction procedures. Or the MIL-DBK-470A: designing and developing maintainable products and systems [126], which gives information to help the reader understand maintainability in the context of an overall systems engineering effort. Different maintenance strategies can be applied for different components, such as preventive or predictive maintenance [127]. Several policies also apply, such as failure-based, time-based, inspection-based policies, etc. [128]. Furthermore, maintenance component indexes such as the mean time to repair (MTTR) [129] can vary even depending on the Mach number of the aircraft [130]. All this makes it difficult to create a model of maintenance cost for conceptual stages of design. This sections aims at focusing at only the needed and relevant parts and providing a method that can provide fast results while reaching a proper level of detail.

Starting from the definition, maintenance is defined as the combination of all actions taken during the life cycle of a product, that have the objective to retain, or restore it, to a state in which it can perform the required function [131]. Airlines have the obligation to created a maintenance program that satisfies the minimum requirements imposed by the law. The first step to define such program is the

Maintenance Review Board Report. This document, supplied by the Type Certificate holder, provides all the fundamental scheduled maintenance tasks for a specific aircraft type. Complementing this, regulatory authorities issue the Certification Maintenance Requirements and Airworthiness Limitations to mitigate excessively hazardous failure conditions and ensure the integrity of safety critical components. Together, these three essential documents are merged into the Maintenance Planning Document (MPD). This document represents the scheduled maintenance as planned by the airline, and it contains the maintenance tasks that are needed in order to fulfill it. These maintenance activities ensure that the corresponding components are repaired or replaced before failing. However, this document does not represent the full maintenance plan. It mainly serves as a basis or reference of the minimum scheduled maintenance required [33]. Nevertheless, unexpected events that compromise the airworthiness can still happen, such as bird strikes or unpredictable system failures. These events require operators to perform unscheduled maintenance, which is not specified in the MPD and serve to restore the original condition of the aircraft. As a summary, maintenance tasks can be grouped into scheduled and unscheduled maintenance.

Regarding scheduled maintenance, tasks can be grouped into different check intervals, depending on the occurrence of each. In general, airlines partition such intervals into four different check types: A-, B-, C- and D-Checks [132]. Each check has maintenance tasks that are more invasive than the previous. They are now explained with more detail:

- **A-Checks:** these checks generally include fast and easy tasks that are performed often, such as visual inspections or filter replacements. Typically done every 750 flight hours (FH) or 750 flight cycles (FC).
- **B-Checks:** they are no longer commonly used in modern aviation, their tasks have been redistributed between A-Checks and C-Checks [131].
- **C-Checks:** these checks include accurate testing of some systems, lubrication of components, or other tasks that require the dismounting of some parts of the aircraft. They are commonly done every 15 and 24 months, but depend entirely on the airline planning.

- **D-Checks:** during these checks the aircraft structure is evaluated in depth for testing, having as a result long task times. These are usually carried out every 6 to 12 years.

Completing the maintenance task categorization, another naming can also be used, which includes line, base and shop maintenance. Line maintenance includes pre-flight, transit, daily and weekly checks, together with the A-checks. It can be done generally in some spaces close to the parking area, or in such area itself. Base maintenance comprehends the C and D-checks that require qualified personnel and specific facilities and tools. Shop maintenance gathers all the tasks, also from C and D-checks, that require in-depth knowledge of components, that can not be repaired locally and need to be sent to the component manufacturer [131].

As mentioned before, maintenance cost represents a noticeable part of the DOC. This paragraph aims at locating maintenance cost inside the Total Operating Costs (TOC) of an aircraft. A general division of the TOC considers the Direct Operating Cost (DOC) and Indirect Operating Costs (IOC). DOC depends on the utilization and includes things such as crew salaries, direct maintenance and fuel. IOC considers those costs that are not directly related to the flight schedule, such as depreciation, rent, traffic services, administrative expenses, etc... Figure 2.4 shows this breakdown with a diagram. It is noticeable how maintenance does not fit entirely into direct or indirect, having a contribution on each. Direct Maintenance Cost (DMC) derives from the maintenance actions, usually divided into Labor Cost (LC) and Material Cost (MC) [33]. Indirect Maintenance Cost (IMC) includes the costs that are not related to a specific maintenance task, for example the purchase of tools or facilities that are needed to perform the maintenance.

Conceptual design stages lack enough information to properly assess IMC, most methods focus on the estimation of the DMC and assume the IMC as a proportion, or directly related, to the DMC, and not as an independent result. However, DMC estimation is also linked to severe uncertainties because it is highly dependent on the aircraft utilization, which depends on the airline. The flight hours (FH) have been identified as one of the most impactful and important parameters for this cost [133]. It is important for manufacturers to be able to estimate the DMC during preliminary design stages to properly know how promising a technology might be. Some cost estimation techniques are available in literature to evaluate the DMC. A proper

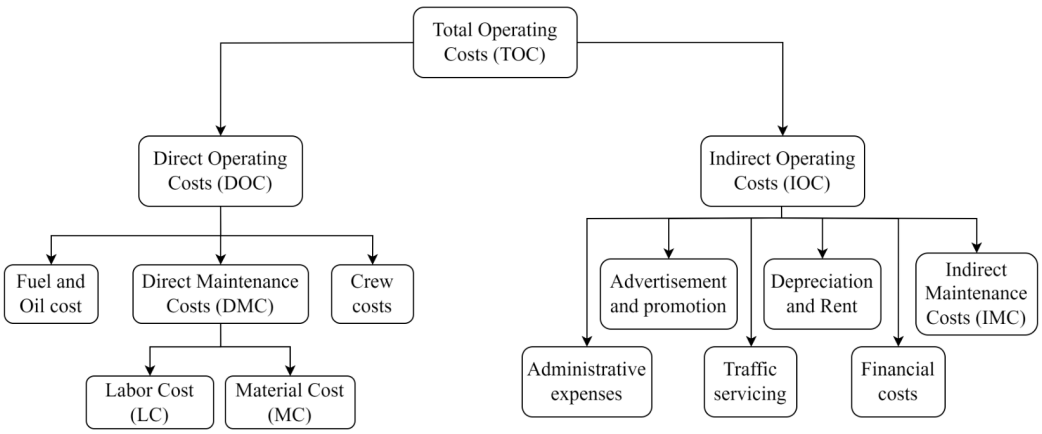


Fig. 2.4 Breakdown structure of an aircraft total operating cost, from [131]

review of these methods was already initiated by Dell'Anna [33] and completed by Ninotta [131], a summary is now provided.

DMC can be estimated following a parametric cost estimation technique [134]. This is based on a statistical database that, once analyzed, defines a series of estimators known as Cost Estimation Relationships (CERs) or cost drivers. These have and represent the highest correlation to the cost. As a result, the DMC can be estimated directly from these cost drivers though a series of equations or CERs. This method simplifies and reduces the whole method to two steps. The first step consists of creating such CERs though a detailed analysis and calculations. Second step consist of just using such CERs previously defined. The difficulty of the first step (creating the CERs) relies on actually finding the proper cost drivers that are representative and sensitive to the DMC. The advantage is that this method provides realistic results since these are based on a real database. Some of these methods are for example those of the NASA-95 [135] or others [135], these are however quite old and outdated. These cost drivers represent old aircraft but get outdated once new technologies are added. This represents the main disadvantage of the method since it needs constant updates. Some authors have tried to update already existent methods [136] or to come up with new innovative ones [137], in order to be able to assess new technologies for future aircraft. Other more recent studies started to develop CERs to make them sensible to the new OBS architectures [138]. However, the biggest gap at the moment when applying these methods to OBS is that they are all based on conventional architectures. MEA and AEA are yet not fully assessed by any existing method. Some studies try to cover this gap and assess the maintenance

cost of such new technologies [139]. However, the main issue is that these methods stay at a concept level, not really reaching component level. This does not manage to properly reach an architecture level for the OBS. For example, a reduction in maintenance cost of around a 1% was calculated in [139] if the actuators of an Airbus A320 are changed from HSA to EHA. But this mainly comes from the difference in MTOM and fuel, not from the component change. Covering this gap motivated the work performed in [33] and [131], two master thesis developed in the context of the analysis performed in this PhD research. These studies contributed to cover this gap in literature and created a CERs method to assess the impact of MEA and AEA architectures of OBS. One important issue that was also taken care of is making a method that can assess the impact of each OBS separately, being able to evaluate the individual impact of electrifying each system alone, as well as all the systems as a whole.

More specifically, Dell'Anna [33] managed to estimate the difference in DMC between two A320 aircraft, one conventional and other one with electrically actuated FCS. The proposed methodology managed to reach component level and was based on the MPD analysis. The MPD analysis consist of comparing both aircraft and removing the tasks related to components that are not needed anymore when moving from the conventional concept (e.g., hydraulic components), and later adding and adjusting new tasks for the new components involved (e.g., actuators and generators). When a task is removed no further information is needed. When a task is modified, the change in maintenance time is modified proportionally following the change in failure rates (i.e., using an analogy based in the failure rate change between the conventional and the more-electric components). Ninotta [131] continued this work expanding the methodology to all the OBS and creating the appropriate CERs for the DMC. The results from the MPD analysis were mitigated by using insight from expert interviews. As a result a surrogate model is provided, which can assess the maintenance cost change of an aircraft when the OBS are electrified (individually and as a whole). This method fully covers the gap in literature and is used for this PhD study. Some small adjustments are needed and explained in the corresponding chapters.

2.4 Science Gaps & Research Questions

This section summarizes the previous sections of this chapter and highlights the main science gaps. The research questions and objectives of this Ph.D. thesis are also enunciated.

The first science gap is related to the system architecting part. Typically, a small number of OBS architectures are defined during conceptual design stages. These architectures are later evaluated and rated based on the results. Some studies are moving towards automated ways to generate the architectures, being able to generate a larger number of solutions. One objective of this Ph.D. is to be able to automatically generate OBS architectures following the system architecting principles. Hence creating an architectural design space from which different solutions can be generated. This huge design space allows to better explore the solutions, generating non-biased architectures. Another gap that is covered by this is automatically connecting the evaluation framework to the architectures generation. This allows to generate OBS architectures from the design space and assess them, all in an automated way. This also allows to introduce such framework into an optimization process. The main advantage of this approach is being able to explore more the design space and different solutions for a certain system, which might show interesting innovative solutions. Another important gap to cover is that the OBS architectures are usually defined just on a subsystem level, not reaching a component level. Achieving a level of detail that reaches components and connections among components is key to assess new technologies. This gap can be covered thanks to the system architecting approach, which allows to reach the required level of detail.

The next science gap is related to the evaluation of OBS architectures. Most studies assess only performance aspects during conceptual design, leaving the rest of the disciplines for later stages. This study aims at enhancing the current state of the art by providing a methodology to also evaluate certification and maintenance aspects at the early stages of the design. A new issue appears which is on how to perform optimization and decision making when moving from a single-discipline problem to a multi-disciplinary one. The objective is to find out how to trade different objectives that come from different disciplines and stages of the life cycle. Which objectives or results are more important, and if there are results that are more important, is an issue that needs to be addressed.

Regarding maintenance, the main science gap is that there are no detailed and quantitative methods to assess OBS architectures (specially more- and all-electric ones). Most studies provide qualitative expectations about new technologies, but no method to back up such results. This affects the capability to reach component level with the analysis, not being able to assess the impact of changes in components. Other issue is that maintenance cost is divided into several contributions (e.g., scheduled, unscheduled, direct, indirect, etc...) which makes it difficult to estimate in early stages of design. The main focus of this analysis is on the direct maintenance cost, which provides a good initial representation of the whole maintenance cost.

Concerning certification, the requirements that apply to the on-board systems architectures are generally quite vague and non-specific. Most of them are qualitative checks, such as having enough back-up systems. However the most relevant quantitative certification requirement is the one that specifies the minimum reliability that a system must have. The reliability block diagram technique is chosen among all the available methods since it can provide the needed result without an enormous number of input, making it feasible for conceptual design. One of the main gaps covered in this Ph.D. thesis is making this method automated so that it can be connected to the architectures generation framework. Another important characteristic of the certification discipline is that it can be used as a filter that removes those architectures that do not meet the certification specifications, reducing the valid design space and allowing the assessment of only those architectures with real potential and feasibility.

The main research question is now formulated:

How can new on-board system architectures be identified by considering simultaneously performance, certification and maintenance aspects during early stages of design?

The main research question can be quite generic and vague, so it is subdivided into three more specific questions:

• How to determine whether innovative and automatically-generated on-board system architectures are not certifiable?

• How to reduce and filter the huge design spaces that characterize the on-board system architectures design?

• How to select the best architectures and perform trade-off analysis with results that come from different life cycle stages (i.e., performance, maintenance and certification)?

The following chapters of this thesis tackle the answer to these questions.

Chapter 3

Methodology & Implementation

The methodology proposed for this Ph.D. is explained in this chapter. A methodology is defined by [140, 141], as a "collection of related processes, methods, and tools". This section explains the different processes and methods used for the analysis here proposed, and it also shows the tools that are used for this specific implementation. Another implementation of the same methodology would be possible with different tools, however this chapter focuses on showing only the implementation done by the author and suggesting alternative tools found in literature when possible.

Section 3.1 shows the general overview of the methodology with different diagrams to better understand it. A breakdown of the methodology into several parts is done and each of the parts is explained in a corresponding section. Section 3.2 shows the starting point where the generation of architectures is performed. Sections 3.3, 3.4 and 3.5 show how to evaluate each of the three disciplines of interest. Section 3.6 explains how to close the optimization framework and section 3.7 focuses on the trade-off analysis that can be done once the optimum architectures have been found.

3.1 General Overview

The objective of the methodology is to evaluate metrics of certification, performance and maintenance of on-board system architectures during early stages of design. This methodology needs to reach component level and to be able to automatically generate and evaluate huge number of architectures. The methodology consist of three main parts and it is represented in a very schematic way in figure 3.1.

The first part of the methodology schema is the architecting and optimization framework, this consists of a design space connected to optimization algorithms. This part is represented by the block in the figure called "OBS architectures generation" and the arrows below the image. This part consists of an iterative process in which several architectures are generated, then evaluated by the evaluation framework, and lastly assessed by the optimization algorithms to select the most promising or optimum ones. The input of this part is a design space of the system of interest, from which the different architectures are generated. The output is a series of optimum architectures based on the metrics obtained in the evaluation framework. This part is further explained in sections 3.2 and 3.6. The second part is the already mentioned evaluation framework, this consists of a series of assessments, one per discipline, making a total of three blocks in the image. These three blocks represent each one of the disciplines, having the corresponding blocks for certification (i.e., certification filter and evaluation), performance (i.e., performance evaluation), and maintenance (i.e., maintenance evaluation). This part of the methodology can be summarized as getting an architecture as input, and providing some metrics for the three disciplines as output. This second part is further developed in sections 3.3, 3.4 and 3.5. Lastly, the third and last part of the methodology is the trade-off analysis. Here the input is the list of optimum architectures found by the optimization algorithms, and the output is a trade-off analysis that supports the decision making process. This means, it helps to identify the best architectures from the optimum ones based on different scenarios created by the designer. This part of the methodology is developed more into details in section 3.7. The three parts summarized result in an "architecting and optimization framework", an "evaluation framework" and a "trade-off analysis".

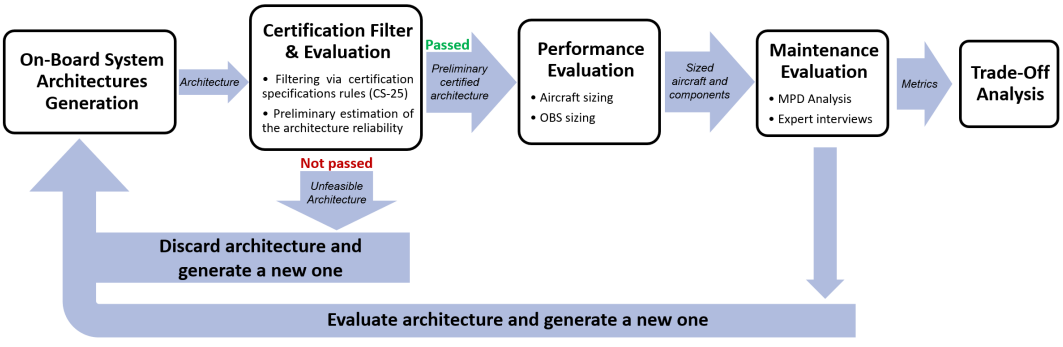


Fig. 3.1 Simplified methodology schema. The proposed filter covers two of the research questions, while the trade-off block tackles the remaining one

A simple example of the whole process is provided in this paragraph to explain better the methodology diagram from figure 3.1. The first step is to define the inputs and outputs for the "architecting and optimization framework". As input a design space must be given, this can be a design space of any kind (e.g., diagram, list of coded rules). This represents a series of possible architectures that could be used to build the system of interest. One example can be a decision diagram that contains a hundred possible solutions for a steering system. This decision diagram can contain choices such as the number of actuators per landing gear strut, the typology of such actuators and/or the distribution systems to which they are connected. As output for the architecting and optimization framework, the different optimization objectives and constraints must be provided. For now it can be simplified with the example before saying that the steering system has three optimization objectives, one for performance, one for certification and one for maintenance. The "evaluation framework" obtains one architecture as input per iteration, and provides the required metrics. Following the steering system example such metrics could be the system mass (for performance), the system reliability (for certification) and the system maintenance cost (for maintenance). The process is iterative and generates one random architecture from the design space per iteration. This architecture is evaluated by the evaluation framework, and the required metrics (i.e., optimization objectives) are provided as a result. One interesting aspect of the certification evaluation is that it can already determine if an architecture is feasible or not, for instance, if one architecture has a reliability lower than a certain margin. This allows to skip the evaluation framework already since this architecture does not need to be further evaluated. This concept is explained with more details in the corresponding section. The optimization algorithms receive the optimization objectives and generate a new architecture based on them. The process continues until the algorithms determine that enough architectures have been evaluated, and they provide the optimum ones. These optimum points are lastly analyzed in the trade-off analysis in order to determine which ones are the best given certain scenarios.

A more technical way to represent the methodology is the eXtended Design Structure Matrix (XDSM), an standardized method used to visualize multi-disciplinary design optimization (MDO) processes. In these diagrams each tool is represented by a block on the diagonal. The input for each tool is contained in the vertical lines, the outputs in the horizontal ones. This method allows to quickly identify the interactions and information exchange among tools, the running order of them

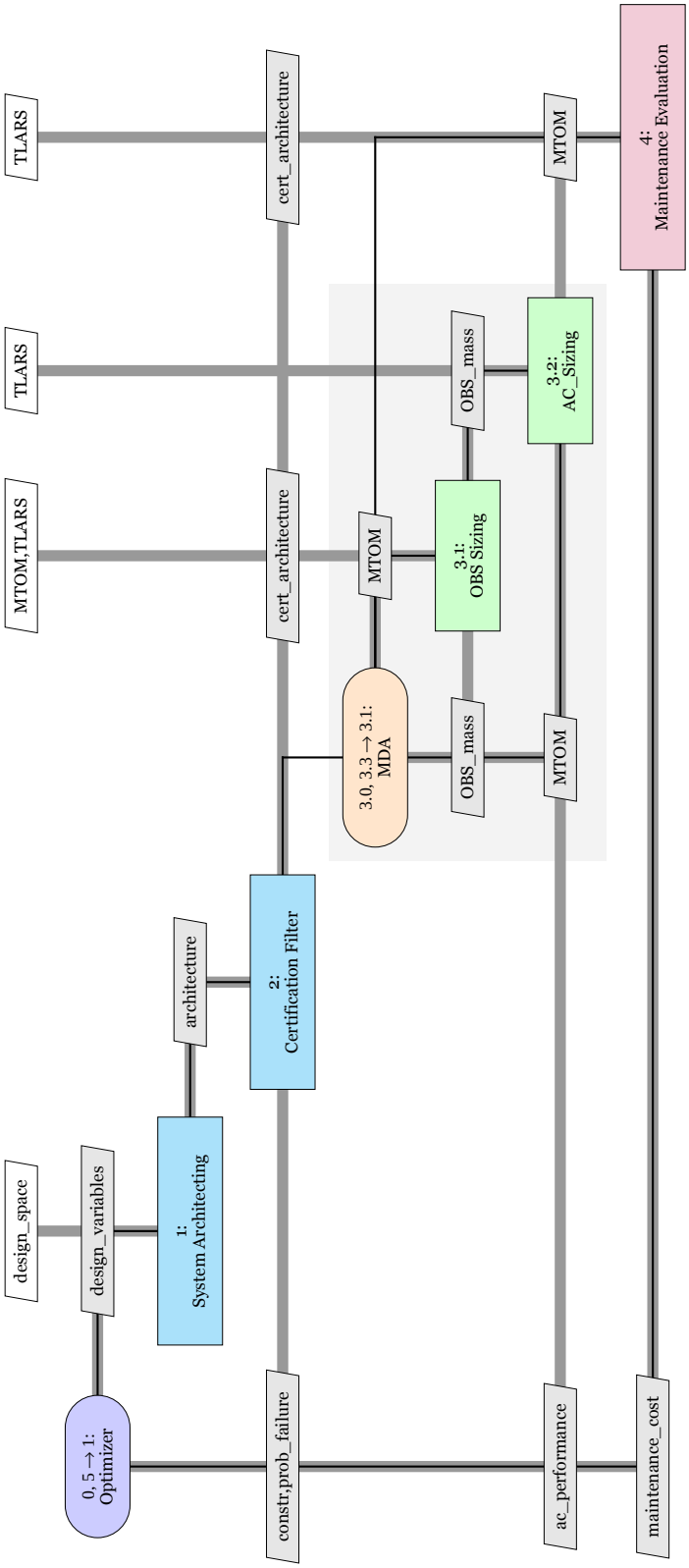


Fig. 3.2 Extended design structure matrix of the methodology

and the overall picture of the optimization framework. Figure 3.2 shows the XDSM of the methodology, which was created with a DLR in-house tool called MDAx [142, 143]. The first block (Block 0,5) represents the optimizer. On each iteration the optimization algorithms propose new design variables based on the results of the optimization objectives obtained in previous iterations. Such design variables are interpreted and generate one specific and unique architecture from the design space. This task is done in the system architecting block (Block 1). Once the architecture for that iteration is generated, it is sent to the first discipline of the evaluation framework, which is certification. The architecture is hence evaluated in the certification filter (Block 2) which provides two results. One is the probability of failure of the architecture, and the other one are the constraints. The constraints are linked to the maximum probability of failure specified in the certification specifications but also contain other certification checks. If the architecture is detected as non-certifiable, the constraints communicate this to the optimization algorithms. This allows to skip the rest of the evaluation framework and discard the architecture as non-feasible. If all constraints are fulfilled, then the architecture is considered as preliminarily certified and it is given to the other disciplines for evaluation. The next block is represented as a conjunct of three blocks which summarize the performance evaluation (Blocks 3.0,3.3: MDA/Converger; 3.1: OBS Sizing and 3.2: Aircraft Sizing). These blocks represent another iterative process that needs to be solved in order to properly assess performance. This includes the snowball effect previously explained in chapter 2 section 2.3. The aircraft and the on-board systems are sized with such iterative process once convergence is achieved. Some top level aircraft requirements (TLARs) are needed as input, such as some geometric characteristics or mission requirements. Lastly, maintenance can be assessed in the last block (Block 4) and the results can be forwarded to the optimizer to continue with further architectures. The trade-off analysis is not represented here but it is carried out after the optimization is finished. Each of the steps (and blocks) are explained into detail in the following sections.

3.2 System Architecting: Architectures Generation

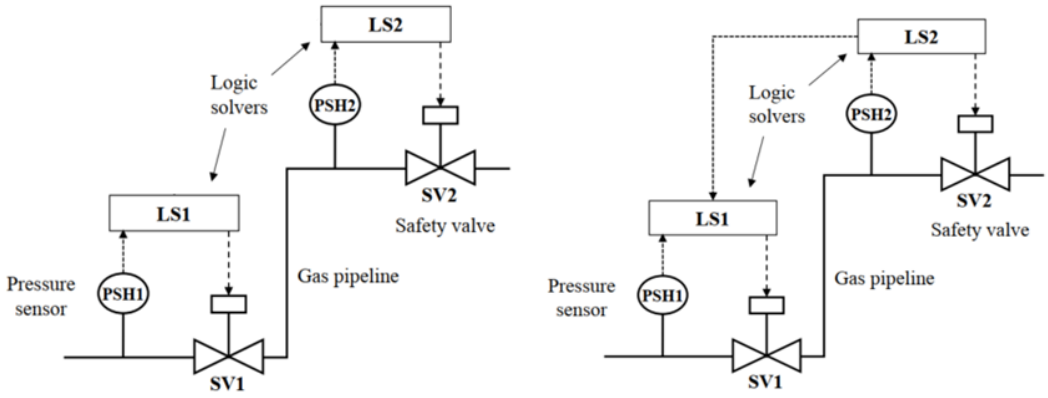
This section explains with more details the part of the methodology regarding the system architecting part, specifically the on-board system architectures generation.

It corresponds to the first block of the schema in figure 3.1 and to Block 1 in the XDSM diagram of figure 3.2.

The main objective of this section is to successfully build the design space of the system of interest. This design space represents all the possible architectures that can build the system. There are multiple ways to create a design space but the one proposed by Crawley [7] is used for this analysis. This method is based on the well-established principles of system architecting. The process starts by identifying the main function (i.e., top function) that the system of interest must fulfill (e.g., a retraction system must deploy the landing gear successfully, an IPS must remove the ice from a certain region of the aircraft, etc...). A series of alternative components are assigned to fulfill this function. However, these components generate subsequent derived functions that need to be fulfilled by other components. An example is provided with the function to deploy the landing gear, such function can be fulfilled by for example a hydraulic actuator (HSA) or an electric one (EHA or EMA), depending on the decision chosen each of the actuators need different functions. A HSA needs to be supplied with hydraulic power, an EMA needs to be supplied with electrical power. This way the design space is built, based on the decisions regarding the component selection. New components create new induced functions, and these functions are derived and connected to further components. As a result, the design space is created from a functional perspective, this removes biased solutions that could be implemented by the designer unconsciously [7]. For more insight on how to built design spaces the reader is referred to [7, 8, 144].

Hence, the design space is modeled following the principles of system architecting and it represents the architecture decisions that are involved in the later optimization problem. An example of architecture is given in figure 3.3, which is presented in [98]. The figure shows two architectures of a gas pipeline closing system. The main idea is that this pipeline needs to be closed if a certain pressure is reached inside. There are two pressure sensors (i.e., PSH in the figure) which work separately and independently. Such sensors are connected to a logic solver correspondingly (i.e., LS in the figure), which is in substance a small computer that interprets the registration of the pressure and determines whether or not to actuate the next components. These are the servo-valves (i.e., SV in the figure), which can close the pipeline. Figure 3.3a represent the simple case in which two assemblies of PSH-LS-SV work together in parallel, this means that there is a redundancy in case the assembly 1 or 2 fail, however if one of the three components of the assembly

malfunctions all the assembly is lost. Figure 3.3b shows another architecture with the same components but an extra redundancy. In this case the logic solver from the second assembly can provide the closing signal to the other logic solver. As result if the second servo-valve and the first pressure sensor fail the system can still function, increasing the reliability. This system, even if simple, explains quite well the concept of architecture and redundancy. Two architectures can differ by having different components, but also by having the same components but different connections. This effect makes the design space grow exponentially when the number of components of a system increases.



(a) Architecture 1 (b) Architecture 2

Fig. 3.3 Architectures example: gas pipeline closing system, from [98]

These two architectures can be modeled in a common design space, and it is shown in the implementation subsection of this section. Typically the first step is to create the design space and then generate the architectures. However, in order to better explain the concept to the user these two architectures are presented first, and the design space is given later.

3.2.1 Implementation

The implementation of this step of the methodology is done with ADORE [8, 144], a DLR in-house tool that allows to graphically model, create and modify architectural design spaces supporting the design process. It also links the design space to the optimization problem and to automatically generate architectures. ADORE even includes some optimization algorithms that can be automatically connected to the

architectural design space. Functions are modeled, and different components can be inserted as potential fulfillers of such functions. Furthermore, additional information about components can be added, such as the number of instances and attributes. When there is a function that can be fulfilled by two or more components, a decision node is inserted to represent the choice of architectures. The architectural design space is then created with these architectural decisions, which are automatically identified and mapped to the optimization problem. This approach allows the modeling of complex concepts. From the design space, all the different possible physical architectures of the system can be generated, and such generation of architectures becomes an automated process.

An example on how to create an architectural design space is shown in figure 3.4, which represents both architectures from figure 3.3. The first step is to define the top function. The system of interest is a gas pipeline and the top function is to close the gas pipeline (if a certain pressure is registered inside). Function names are reduced for simplicity and marked with a number next to them for a better tracking. The top function is fulfilled by two components in parallel, for this a component called multi-fulfillment is used ("multi" in the figure). This component does not represent a decision, it means that both components are present in the architecture, not one or the other. Such components are "servo valve 1" and "servo valve 2", which are both present in all the architectures. Each servo valve needs to be activated by something, for that each generates its own induced function "activate servo valve x" (functions 2 and 3). As seen in the schema, the component that fulfills this function is a logic solver. Each servo valve has its own solver (i.e., servo valve 1 -> logic solver 1; and servo valve 2 -> logic solver 2). Correspondingly, each logic solver needs the function "give pressure measurement to logic solver x", since the pressure value is their needed input. The component that fulfills function 4 "give pressure measurement to logic solver 2" is pressure sensor 2, as seen in figure 3.3. Lastly, there are two options to fulfill function 5 "give pressure measurement to logic solver 1". One is directly with pressure sensor 1, the other one is with a redundancy between pressure sensor 1 and logic solver 2. As a result, two possible architectures can be generated from this design space depending on the decision of how to fulfill function 5. Expanding this design space would be quite easy to do just by adding more decisions or components. For instance function 4 could be solved the same way as function 5, with a redundancy to logic solver 1. This would increase the number of architectures to four. Adding a third conjunct of valve-solver-sensor would also

increase the number or architectures. This shows that once a base design space is created, modifying and enhancing it is quite straightforward. More examples of a design space are shown in chapter 5 once the application case is presented, this case shows a more realistic design space with a a huge number of decisions and architectures.

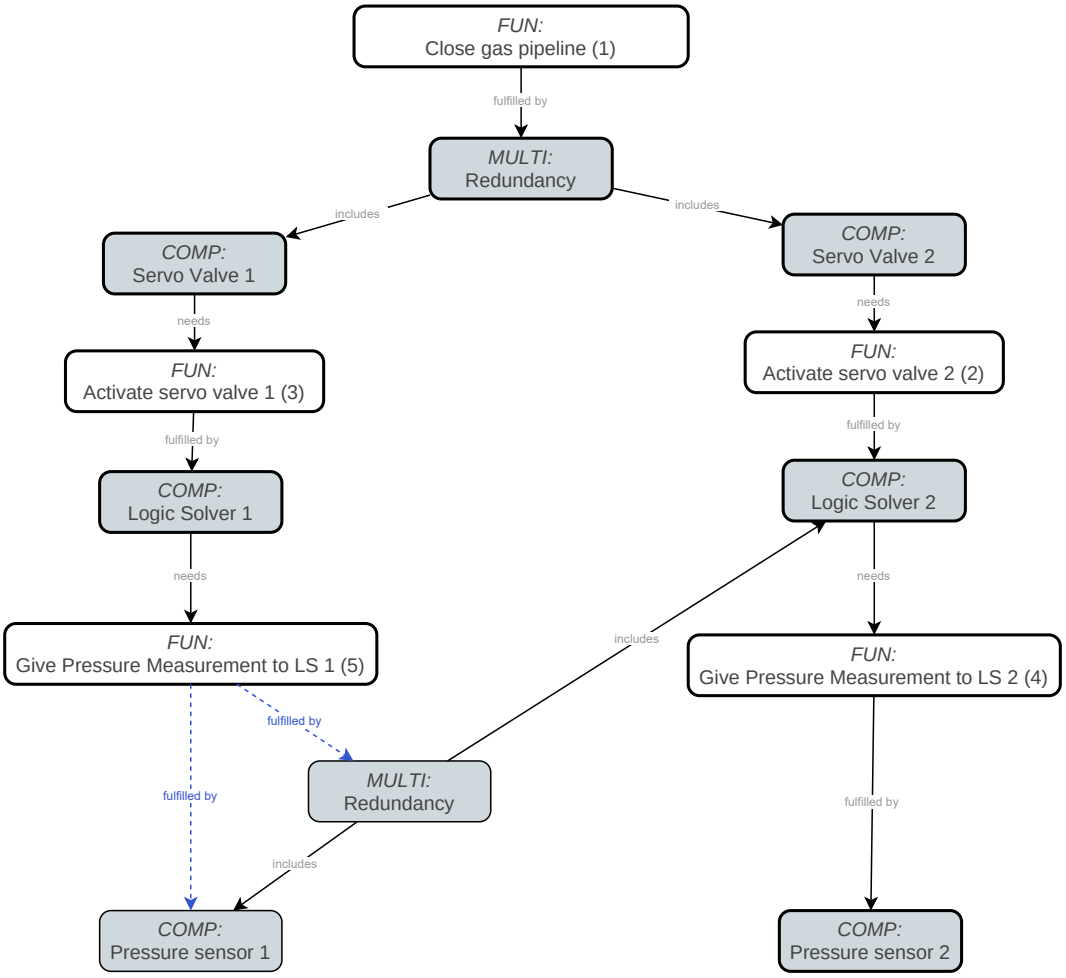


Fig. 3.4 Design space of the gas pipeline closing system from figure 3.3, modeled in ADORE

3.3 Certification: Architectures Filtering & Evaluation

This section explains with more details the part of the methodology regarding the architectures evaluation part, specifically the certification evaluation. It corresponds

to the second block of the schema in figure 3.1 and to Block 2 in the XDSM diagram of figure 3.2. It can be seen how this part of the methodology can be used as a filter, the rest of the evaluation can be skipped if certain conditions are not met.

The main objective of this section is to establish the metrics related to the certification discipline, explain how to connect this part with the architectures generation and how to properly filter the architectures according to certain constraints. Some parts of this section were already disseminated in [145] and [146], a detailed and merged overview is now provided. Some key aspects that this part of the methodology must achieve are to allow multiple architectures evaluation, be easily automated, reach component level and to be usable for innovative technologies.

The first thing to do in order to check certification aspects is to gather the adequate requirements that the OBS architectures need to fulfill to be certified. This task is quite challenging and can be divided into two steps, first selecting the proper certification specifications and then choosing the proper requirements. As explained in section 2.3.2, two main entities establish the certification specifications for commercial aircraft depending on the region (i.e., EASA and FAA). Both have the same problem for the OBS architectures, the requirements are too generic or too detailed. Most requirements express conditions that cannot be yet assessed during conceptual design. Some examples can be the maximum temperatures that a component can bear, electromagnetic compatibility issues that some avionics need to surpass, zonal analysis that constrains how close some pipes, how to avoid temperature interference, etc. These requirements cannot be assessed during conceptual design stages since only the OBS architecture has been defined, needing to be checked at later stages. The relevant requirements at this level are those that affect the connectivity among systems and components, and the redundancies. This task represents a big gap in literature. Typically these architectural requirements are created from experience, replicating architectures that were already certified in current aircraft. An example, extracted from [20], is now shown to the reader for a better understanding of the issue. Here some rules and guidelines are provided for the design of the ailerons and elevator of commercial aircraft (taken from [20]):

> "Control surfaces such as ailerons and elevators, which are flight-critical, are provided with two actuators per panel... Each actuator is supplied by a single power system. Then:"

a) “If the aileron group and/or the elevator group is powered by the same type of power (i.e., either hydraulic or electric but not both), then three such power systems are required"

b) “If the aileron group and/or the elevator group is powered by both types of power (i.e., both hydraulic and electric), then two power systems of each type are required”

These rules are extrapolated from good practices seen in literature and provide exactly the information that is needed. They constrain the connections and establish redundancy rules. However, the purpose of this analysis is to extract the rules and requirements that come directly from the certification specifications, not from experience. Using existing architectures as reference can have an influence and establish bias in the results, blocking some potential innovative solutions. The objective is to only follow what is written in the certification specifications, not what comes from old practices. The main focus of this study is on commercial civil aircraft, for this reason the "CS-25: Large Aeroplanes" [80] is chosen. The objective is to find and extract the requirements that directly affect the OBS architectures, which can be qualitative or quantitative. The list is quite short since most requirements cannot be assessed on conceptual design stages, they are listed now.

CS-25 Requirements

The most important requirement for OBS architectures which is related to reliability is the one that provides guidelines on catastrophic failure conditions. There is a statement repeated several times through all the CS-25 document that says as follows: "catastrophic failure conditions must be extremely improbable". More precisely for on-board systems this can be found under the paragraph CS-25.1309: Equipment, systems and installations. The most important parts are here shown for the reader:

• CS 25.1309(b): "The aeroplane systems and associated components, considered separately and in relation to other systems, must be designed so that:"

- (1)"Any catastrophic failure condition

  • (i) is extremely improbable; and

- (ii) does not result from a single failure"

The first requirement (CS 25.1309(b)(1)(i)) states that "any catastrophic failure condition is extremely improbable", such condition shall be quantified. The instructions on how to do this are found in the "CS-25 AMC-Subpart F, Chapter 7: failure condition classification and probability terms". Precisely under the AMC 25.1309 System Design and Analysis:

- AMC 25.1309(7.c): Quantitative Probability Terms:
  - (1) Probability Ranges
    - (iv) "Extremely improbable failure conditions are those having an average probability per flight hour of the order of $1x10^{-9}$ or less."

Another important and relevant sizing condition is found in CS 25.671(d). This requirement provides further safety requirements for the landing gear and flight control system. It says as it follows:

- "The aeroplane must be designed so that, if all engines fail at any time of the flight:
  - (1) it is controllable in flight;
  - (2) an approach can be made;
  - (3) a flare to a landing, and a flare to a ditching can be achieved; and
  - (4) during the ground phase, the aeroplane can be stopped."

Lastly, CS 25.729(c) gives more restrictions to the landing gear design with a requirement that states:

- Emergency operation. "There must be an emergency means for extending the landing gear in the event of –
  - (1) any reasonably probable failure in the normal extension and retraction systems; or

– (2) the failure of any single source of hydraulic, electric, or equivalent energy supply."

Summarizing now, three main requirements are extracted and formulated from the previous information.

➤ The first requirement is named "No single point-of-failure" and comes from CS-25.1309(b)(1)(ii). It says that the single failure of a single component must never lead to the failure of the whole subsystem. This implies that some redundancies are needed for safety, such as components in parallel or redundant power supplies. Even if the architecture meets the minimum reliability requirements if one single failure makes the system fail it cannot be certified. Other paragraphs emphasize more this condition. For example CS-25.671(c)(1) applies to the flight control system and states that a single failure cannot risk the continuous safe flight. Or CS-25.745(c) which provides the same statement for the aircraft steering function. As a general outcome, it can be stated that the certification specifications limit the single failure condition, not allowing it to compromise the functionality of any subsystem. This requirement is considered qualitative since the result is a yes-or-no output.

➤ The second requirement is named "Maximum probability of catastrophic event" and comes from the combination of CS-25.1309(b)(1)(i) and AMC-25.1309 (7.c)(1)(iv). It says that the loss of one subsystem functionality leads to a catastrophic failure and must occur with an average probability lower than $1x10^{-9}$ per flight hour. This requirement is considered quantitative since the result of the architecture is a probability of failure that is compared to the minimum established, having a positive or negative margin. This one is quite difficult to assess and there is no simple way to do it. SAE-ARP 4754 and SAE-ARP 4761 guidelines provide a standard process to calculate this value, but as explained before this is not possible in conceptual stages of design due to a lack of detailed information. A simplified but standardized method is needed to be able to preliminary estimate the probability of failure of a given system. The reliability block diagram is suggested for this purpose, as expressed before in section 2.3.2. The steps and guide on how to assess this condition are explained later in the following sections.

➤ The third requirement is named "Need for back-up system" and comes from CS-25.671(d). It says that there must be at least one back-up system that provides power if all the engines are lost, more specifically for the flight control system and landing gear. This ensures the correct functioning of such systems if all the engines

fail. This leads to the necessity to have an APU or RAT, or another redundant power source in the aircraft that is non dependent on the engines. This condition is also a yes-or-no output and is generally quite easy and straightforward to assess.

There could be a fourth requirement that applies to the landing gear. No application cases for the landing gear are shown in this analysis but it is still written for future analysis. This requirement comes from from CS-25.729(c)(2) and says that the landing gear extension cannot be connected to a single line. No matter if such line is electrical or hydraulic. This implies that at least two lines are needed for the landing gear extension, and if the extension is fulfilled by only one actuator there must be another means of extending the landing gear in the event of failure (e.g., gravity of hand pump).

All these requirements are extracted from the CS-25 but can also be re-used for other certification specifications since they are quite generic. They represent the most important requirements to be considered during the design of OBS architectures. As said before, the requirements that express more specific conditions are left out of the scope of this analysis since they do not affect the OBS connections or redundancies. An example is CS-25.831(a), this requirement provides a minimum airflow value that needs to be provided by the ECS. This type of requirements can only be assessed at later stages of design, once the OBS are fully sized.

A summary of this part of the methodology is provided in this paragraph. The input is a certain architecture of a system, the outputs are the probability of failure of the architecture and a set of constraints. Such constraints are the three previously mentioned requirements: single failure condition, catastrophic failure condition (or minimum reliability) and back-up systems. If any of the requirements (or constraints) are not passed the architecture is discarded. This saves computational time and allows to filter the architectures, evaluating only the ones with potential for certification. The RBD is used for the calculation of the probability of failure and minimum reliability condition, it is further explained in the following section.

3.3.1 Reliability Block Diagram

The reliability block diagram (RBD) is a technique supported by standards [99] that allows to estimate the reliability (and probability of failure) of a certain architecture. This can be used to calculate the probability of failure of a subsystem and compare

it to the minimum established in the certification specifications by the certification authorities. This section focuses on explaining how to solve a RBD, which common problems need to be addressed (e.g., non-conventional configurations), how to automate the process and how to translate an architecture into a RBD configuration.

It is important to understand that one system can fulfill different functions (e.g., the FCS provides roll, pitch and yaw control). One RBD represents only one function per diagram, having one RBD for each of the different functions that the system fulfills. As a result, it is incorrect to think that one RBD represents one system. It only represents one of the functions of the system. This idea is repeated through this manuscript but must be clarified now before explaining further the concept of RBD. One system can have then one or more RBDs that represent its different functionalities, this fact is further explained later in the RBD translation subsection. For now, it is important to highlight that even if one RBD is shown, this does not represent the entirety of the system of interest.

In order to solve a RBD, the first step is to know the components involved and the connections among them. Each component can be synthesized into a reliability value (R) that can be estimated from its failure rates [97] following equation 3.1:

$$R(t)_{component} = 1 - F(t) = e^{-\lambda t} \tag{3.1}$$

Where as explained in section 2.3.2, the time dependency can be simplified by using the values per hour. Which corresponds to λ = failure rates = $1/MTTF$; expressing the Mean Time To Failure (MTTF) in hours. This results in R, which is the reliability of the component, needed to solve the RBD.

A group of components form a conjunct. Depending on how they are connected among them they can be structured in a certain configuration or another one. The most typical, common and well-known RBD configurations are the series and parallel. These configurations are quite simple and easy to evaluate. However it is not uncommon to find other non-conventional configurations in aerospace systems, some examples are shown in figure 3.5.

Series and parallel configurations are the most simple ones, and can be solved following equations 3.2 and 3.3, which can be found extensively through literature [97–100]. These equations can be used for conjuncts of two or more components in both cases.

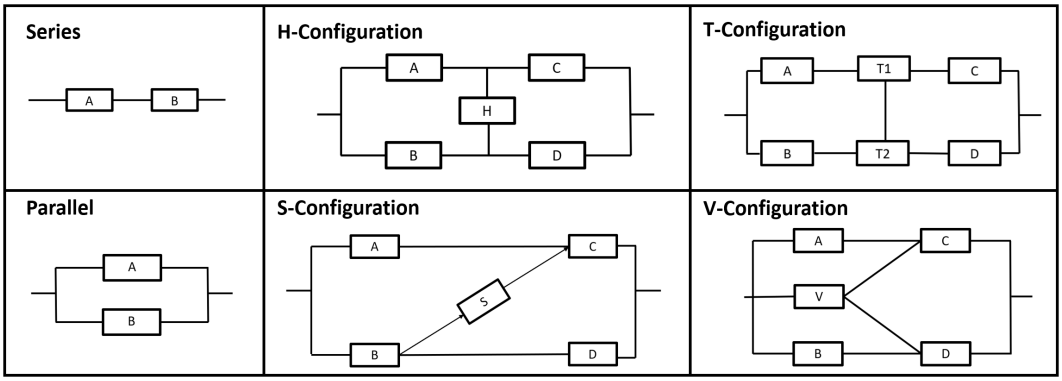


Fig. 3.5 Examples of different RBDs configurations

$$R_{series} = \prod_{i=1}^{n} R_i \tag{3.2}$$

$$R_{parallel} = 1 - \prod_{i=1}^{n} (1 - R_i) \tag{3.3}$$

The reliability of two components in series (component A and component B) is the product of their reliabilities ($R_A \cdot R_B$). As an interesting note, in the hypothetical case that one component never fails ($R = 1$) the reliability of the conjunct is the reliability of the other component. Furthermore, if one component fails the whole conjunct fails. This configuration is the most simple one but at the same time the least reliable. An example of two component in series can be an engine and an engine-driven generator, or an electric motor and an electric-driven pump. On the other hand, the reliability of two components in parallel (that are statistically independent) is $1 - (1 - R_A) \cdot (1 - R_B)$. If for example component A never fails, then the whole conjunct never fails. If component A fails, the reliability of the conjunct is the reliability of component B. This configuration establishes the concept of redundancy in its simplest ways. An example of components in parallel can be one control surface that can be moved by two actuators in parallel, or an electric actuator that is connected to three different electrical lines.

The following subsections show some of the most common non-conventional configurations focusing on the ones typically found in aerospace systems. Some of them come from literature while others are an original contribution of this Ph.D. A unified and merged section is provided, also showing examples and how to solve

them. The names of each of the configurations are not standardized and can be found differently through literature. The names are given following the authors preferences.

Non-conventional RBD configurations: H-configuration

The H-configuration, also called bridge configuration [97], has its name due to its characteristic shape. It is shown in figure 3.6 in its most simple form. Five components are connected, having component H in a non-conventional position. This configuration can function even if components A-D fail, or if A-C fail, since H is a bi-directional component that serves as a double redundancy. This configuration cannot be solved with direct equations and a statistical approach is needed [97], deriving partitions based on the probability of failure of certain components and then merging the results of such partitions with their corresponding probabilities. This concept now explained step by step.

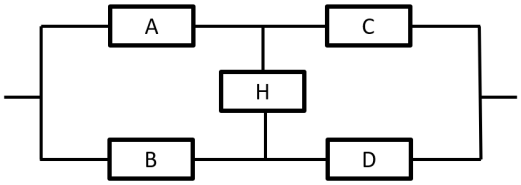


Fig. 3.6 RBD; H-configuration schema

The key to solve such non-conventional configurations is to partition them into solvable pieces and later sum the probability of each of them happening. For this specific configuration, it can be partitioned into two more simple configurations, as seen in figure 3.7. One partition represents the case in which component H is always functioning, the other one the case in which this component failed. The Bayes theorem of conditional probability is now used, as supported by the IEC-61078 reliability block diagram standards [99]. This theorem applied to this case says that the reliability of the whole configuration is the reliability of partition 1 multiplied by the probability that component H functions (reliability of H) plus the reliability of partition 2 multiplied by the probability that component H does not function (probability of failure of H). This leads to equation 3.4.

$$R_{H-configuration} = R_{partition_{1H}} R_{component_H} + R_{partition_{2H}} (1 - R_{component_H}) \qquad (3.4)$$

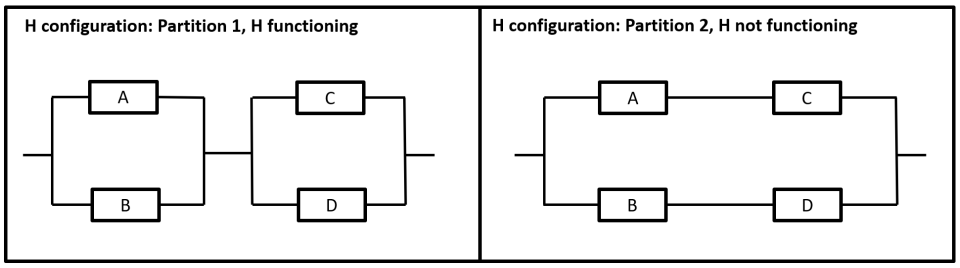


Fig. 3.7 H-configuration schema partitions

The reliability of partitions 1 and 2 can be derived from the series and parallel equations, resulting in equations 3.5 and 3.6.

$$R_{partition_{1H}} = [1-(1-R_A)\cdot(1-R_B)]\cdot[1-(1-R_C)\cdot(1-R_D)] \tag{3.5}$$

$$R_{partition_{2H}} = [1-(1-R_A\cdot R_C)\cdot(1-R_B\cdot R_D)] \tag{3.6}$$

Some examples of components that work bi-directionally as an H-component and that can lead to H-configurations are cross-feed valves and power transfer units (PTU). Cross-feed valves are used as a redundancy in fuel systems and allow to transfer fluid from point A to B and the other way around. Same effect happens for the PTU in the hydraulic system, they allow to transfer fluid in two directions if a failure is detected in one of the lines. Some examples are now provided, showing the architecture of the system and the resulting RBD.

The first example is about the flight spoilers of the Airbus A320, and it is shown in figure 3.8. The image is divided into three parts. One shows the control surfaces and the hydraulic lines that they are connected to, the other the detailed view of the hydraulic system and lastly the resulting RBD. The ailerons are removed from the example at this point for simplicity, leaving only the flight spoilers (i.e., spoilers 2, 3, 4 and 5). Colors are used in the images to help the reader understand better the connections among components. The RBD is assessing the reliability of the function "control roll with spoilers", it can be seen how the four spoilers pairs (i.e., left and right) are set in parallel. Each of them is connected to their corresponding hydraulic line (i.e., blue, yellow or green) and the PTU is located between two of them (yellow and green). This component creates a non-conventional RBD configuration that is

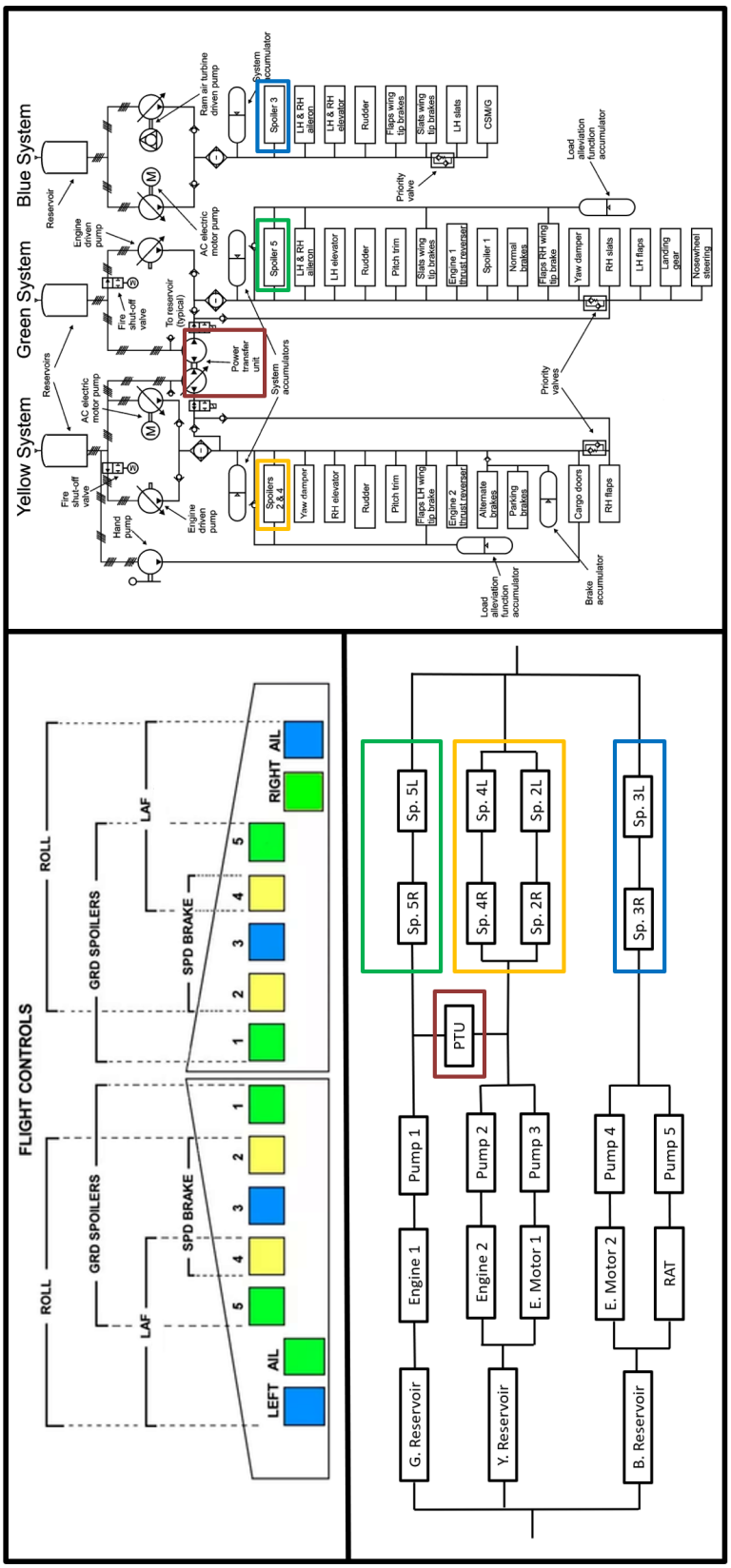


Fig. 3.8 Example of an architecture with an H-configuration RBD. The figure shows the hydraulic system of the Airbus A320 and its flight spoilers, it also contains the resulting RBD

common in conventional aircraft. This enhances the importance of considering such non-conventional configurations in the analysis.

Another example is provided for the fuel system of the Airbus A330, shown in figure 3.9. This example is quite simplified and the number of components has been reduced in order to just show the root of non-conventional configuration. The RBD is assessing the reliability of the function "to provide power to the engines in normal mode", represented in the architecture by a thick black line that goes from engine 1 to engine 2. This means that components such as the trim tank or the APU are now excluded for simplification. It can be seen how the cross-freed valve (X-feed) creates another H-configuration in the RBD.

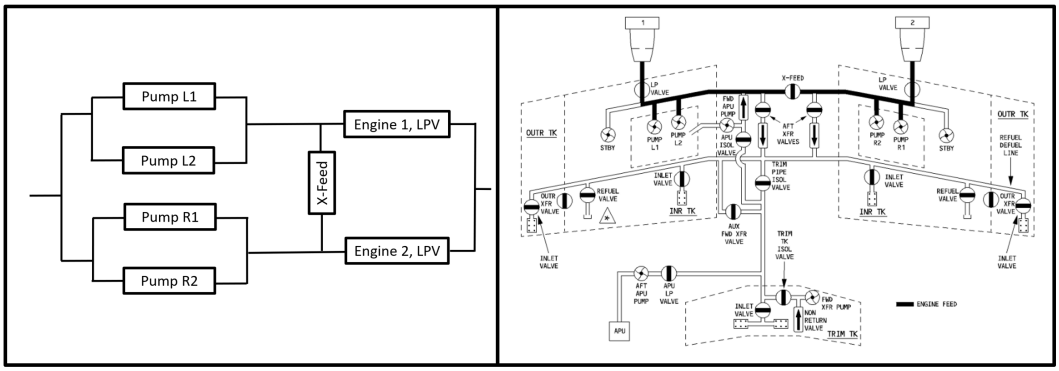


Fig. 3.9 Example of an architecture with an H-configuration RBD. The figure shows the fuel system of the Airbus A330 and its simplified RBD

Non-conventional RBD configurations: S-configuration

The S-configuration is similar to the H-configuration but without the bi-directionality characteristic. Figure 3.10 shows its schema. Also five components are connected, having component S in a non-conventional position. The redundancy for this configuration comes from the possibility to still function if components A and/or D fail. A statistical approach is also needed in order to solve this case.

As for the previous configuration, the key is to divide the configuration into partitions. This is done in an analogous ways as before, considering the reliability when component S fails plus the one when component S never fails. This leads to equation 3.7 and to the partitions represented in figure 3.11.

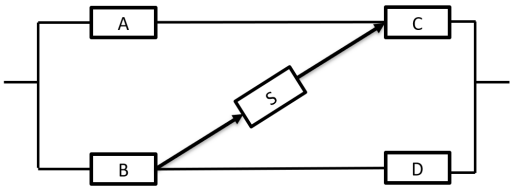


Fig. 3.10 RBD; S-configuration schema

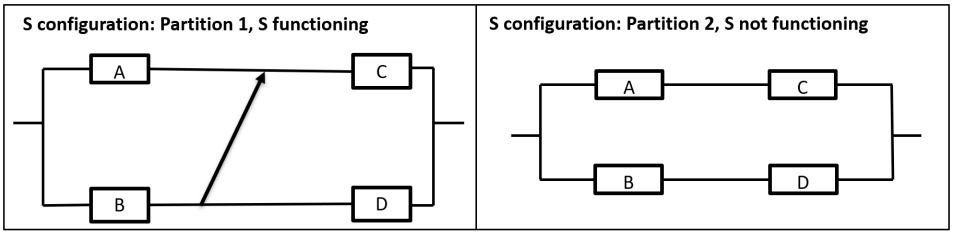


Fig. 3.11 S-configuration schema partitions

$$R_{S-configuration} = R_{partition_{1S}} R_{components_S} + R_{partition_{2S}} (1 - R_{components_S}) \tag{3.7}$$

The reliability of partition 2S can be derived from the series and parallel equations, resulting in equation 3.8.

$$R_{partition_{2S}} = [1 - (1 - R_A \cdot R_C) \cdot (1 - R_B \cdot R_D)] \tag{3.8}$$

However, partition 1S leads again to a non-conventional configuration that is an S-configuration in which the S-component always works. This partition can be further decompound into two other sub-partitions, where component B is the root of the non-conventionality. Equation 3.9 shows the resulting expression and figure 3.12 the corresponding sub-partitions.

$$R_{partition_{1S}} = R_{partition_{1.2S}} R_{component_B} + R_{partition_{1.1S}} (1 - R_{component_B}) \tag{3.9}$$

Both sub-partitions of partition 1S are easy to evaluate since they are just two component in series and in parallel. The final result for the S-configuration is

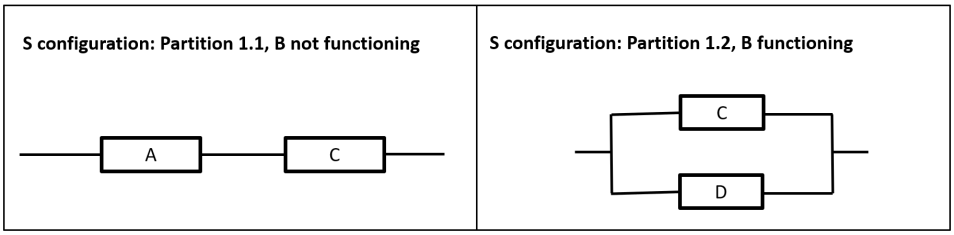


Fig. 3.12 S-configuration schema, partitions of partition 1S

shown in equation 3.10. This statistical approach provides the same result as other RBD-reduction methods, as suggested in [99], validating the methodology.

$$\begin{aligned} R_{S-configuration} &= R_A \cdot R_C + R_B \cdot R_D - R_A \cdot R_B \cdot R_C \cdot R_D + \\ &+ R_S \cdot (R_B \cdot R_C + R_A \cdot R_B \cdot R_C \cdot R_D - R_A \cdot R_B \cdot R_C - R_B \cdot R_C \cdot R_D) \end{aligned} \tag{3.10}$$

An example of a S-configuration can be found in figure 3.13, with the example that was already explained in figure 3.3b. The function that the RBD is assessing is "to close the gas pipeline if the pressure raises from a certain margin". This configuration shows that very simple systems can still have non-conventional RBD configurations.

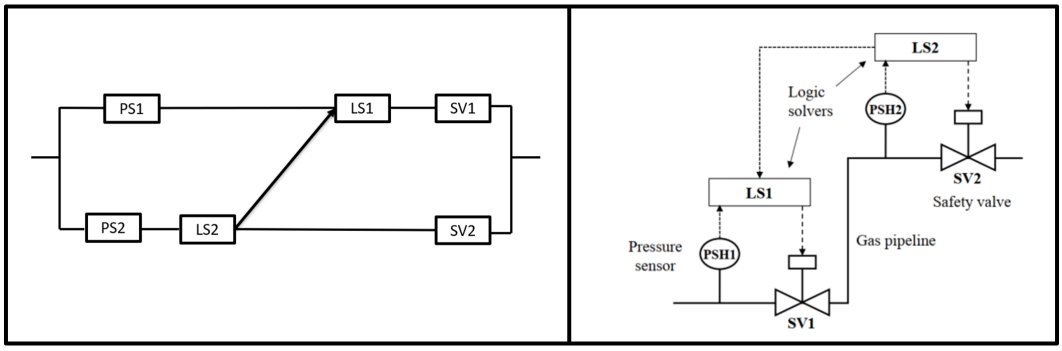


Fig. 3.13 Example of an architecture with an S-configuration RBD

An aeronautical example of the S-configuration is shown in figure 3.14. This figure shows the electrical system of the Airbus A320, and the RBD assesses the function "to provide power to the DC essential line". This line works in a particular way since it can receive power from AC1, AC essential or from the batteries directly. The connection among AC1 and AC essential creates the S-configuration (without a component in between). This non-symmetry comes from the functioning mode of

the system, in normal mode AC1 is used for DC1 and DC essential, while only AC2 is used for the DC2.

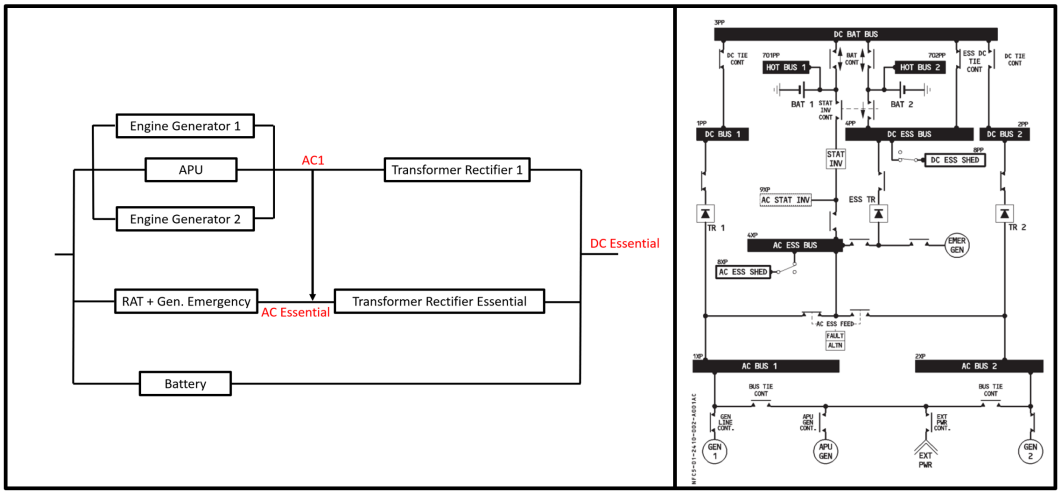


Fig. 3.14 Example of an architecture with an S-configuration RBD. The figure shows the electrical system of the Airbus A320 and its simplified RBD for the DC Essential function

Non-conventional RBD configurations: V-configuration

The V-configuration has its name owing to its shape. It is represented in figure 3.15. Also five components are interlinked in this case, having component V in the non-conventional position. This component works as a double redundancy that can substitute either component A o B if they fail. It can also be solved statistically as the previous configurations.

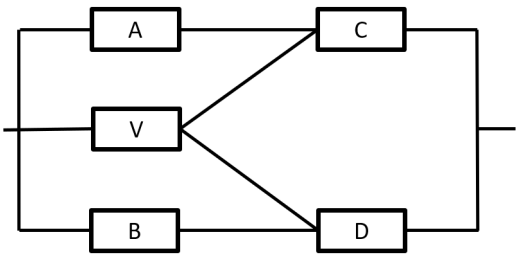


Fig. 3.15 RBD; V-configuration schema

The partitions are done as before. One results when component V fails and the other when it is always working, as seen in figure 3.16. Equation 3.11 shows the mathematical expression.

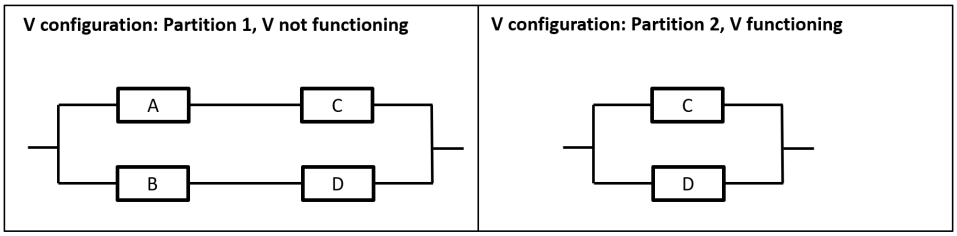


Fig. 3.16 V-configuration schema partitions

$$R_{V-configuration} = R_{partition_{2V}} R_{component_V} + R_{partition_{1V}} (1 - R_{component_V}) \quad (3.11)$$

The reliability of partitions 1 and 2 can be derived from the series and parallel equations, resulting in equations 3.12 and 3.13.

$$R_{partition_{1V}} = [1 - (1 - R_A \cdot R_C) \cdot (1 - R_B \cdot R_D)] \quad (3.12)$$

$$R_{partition_{2V}} = [1 - (1 - R_C) \cdot (1 - R_D)] \quad (3.13)$$

Some examples of components that work as an V-component and that can lead to V-configurations are auxiliary power units (APU), as shown in figure 3.17. This figure shows again the electrical system of the Airbus A320, now assessing the "DC batt function". In this case the APU works differently as for the case previously presented. Now AC1 and AC2 work in parallel, and either of them can provide power to the objective line. This comes from the functioning mode of the system, which allows both lines to provide power to the DC batt. Now the APU works in redundancy with both in a particular architecture, resulting in a V-configuration.

Non-conventional RBD configurations: T-configuration

The T-configuration is similar to the H-configuration but with two components acting as a bridge. Figure 3.18 shows the corresponding schema. Six components are connected in this case, since two are needed for the redundancy, which are T1 and T2. The redundancy for this configuration comes from the possibility to still function if components A-and-D or B-and-C fail, as long as both T1 and T2 are functioning.

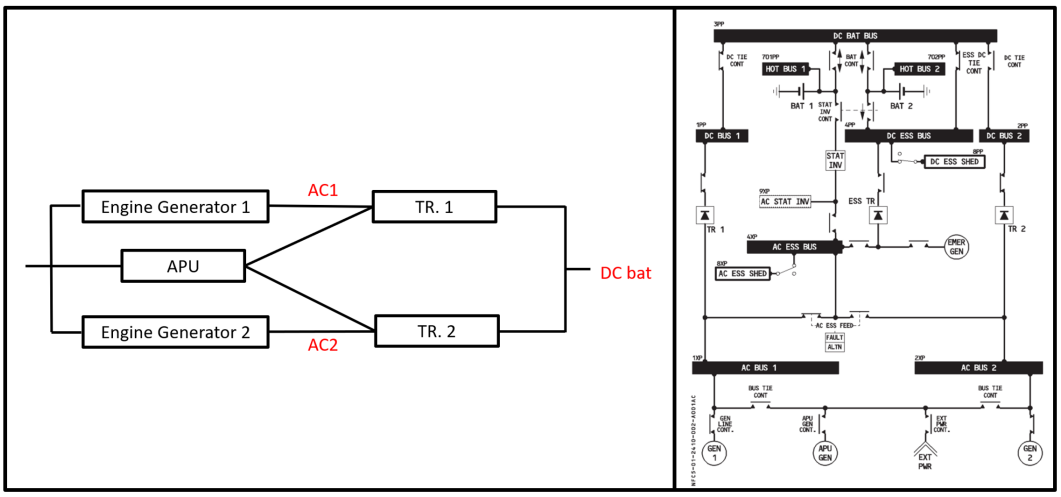


Fig. 3.17 Example of an architecture with an V-configuration RBD. The figure shows the electrical system of the Airbus A320 and its simplified RBD for the DC Bat function

This architecture is not seen in literature but can be obtained by modifying an S-configuration as shown later. A statistical approach is also needed in order to solve this case.

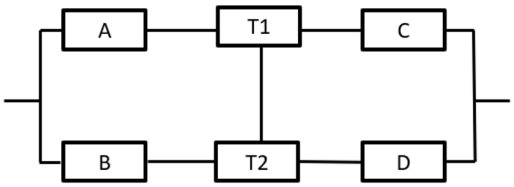


Fig. 3.18 RBD; T-configuration schema

As for the previous configurations, the objective is to divide the configuration into partitions. This case leads to a similar one such as the S-configuration. One initial partition is performed, around component T1, and a posterior partition around T2 is needed. This results in equation 3.14 and figure 3.19.

$$R_{T-configuration} = R_{partition_{2T}} R_{component_{T1}} + R_{partition_{1T}} (1 - R_{component_{T1}}) \quad (3.14)$$

The reliability of partition 1T is just three components in series, hence multiplying the reliabilities of components B, T2 and D. For partition 2T a new sub-partition must be done in order to reduce the complexity of the new RBD that is again another

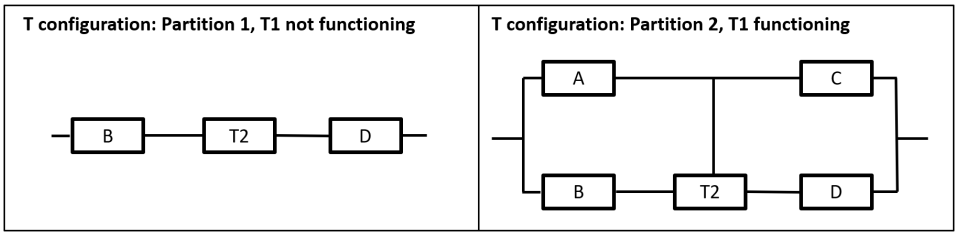


Fig. 3.19 T-configuration schema partitions

non-conventional configuration. This sub-partition is shown in figure 3.20 and represented by equation 3.15.

$$R_{partition_{2T}} = R_{partition_{2.2T}} R_{component_{T2}} + R_{partition_{2.1T}} (1 - R_{component_{T2}}) \tag{3.15}$$

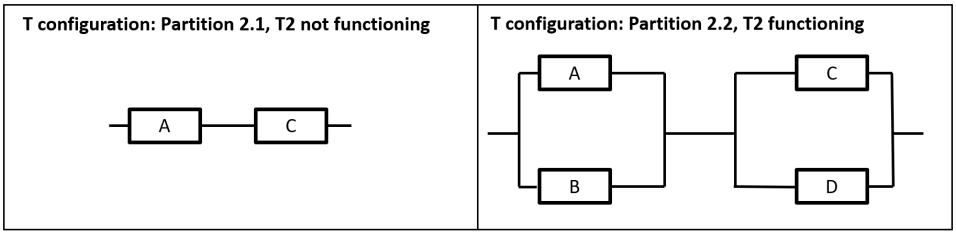


Fig. 3.20 T-configuration schema, partitions of partition 2T

Both sub-partitions of partition 2T are easy to evaluate since are just component in series and in parallel. Partition 2.1T is just the multiplication of $R_A \cdot R_C$, while partition 2.2T is shown in equation 3.16.

$$R_{partition_{2.2T}} = [1 - (1 - R_A) \cdot (1 - R_B)] \cdot [1 - (1 - R_C) \cdot (1 - R_D)] \tag{3.16}$$

An example of the T-configuration is shown in figure 3.21. This example is obtained by modifying the previously seen gas pipe architecture. Now a redundancy is added since both logic solvers can communicate, creating a bi-directionality. This leads to the T-configuration, in which both components must work so that the redundancy is active (LS1=T1, LG2=T2).

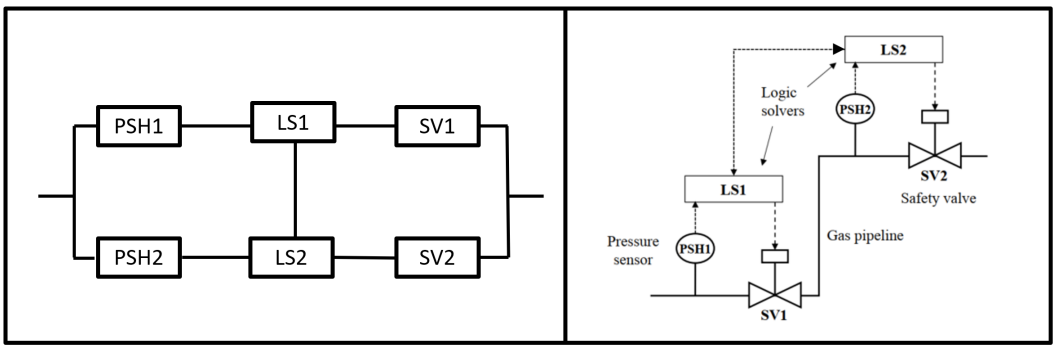


Fig. 3.21 Example of an architecture with an T-configuration RBD

Non-conventional RBD configurations: R-configuration (r-out-of-n)

The R-configuration is also commonly known as r-out-of-n, or m-out-of-n [99, 98], or even k-out-of-n [97, 100]. The name R-configuration is used here. This configuration is like a parallel configuration in which not only one of the components must function. The number of total components is specified by the "n", while the number of components that need to function in order to make the system function is specified by the "r". Figure 3.22 shows three examples. One generic with an undetermined number of components, another one in which 2-out-of-5 must function, a a last one in which 2-out-of-4 must function. The solving of some of these examples can be found in literature through the use of statistical analysis. However, no generic and uniform method or equation is found for a generic R-configuration in which both r and n can be any numbers. A numerical analysis is developed and shown in this section.

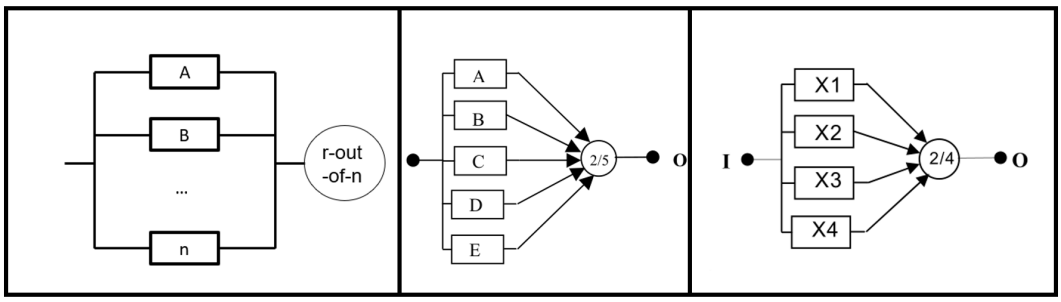


Fig. 3.22 RBD; R-configuration schema

When a R-configuration has identical elements it can be solved by an analytical expression, shown in equation 3.17.

$$R_{r-out-of-n} = \sum_{i=0}^{n-r} \frac{n!}{i!(n-i)!} R^{n-i}(1-R)^i \tag{3.17}$$

Here the R value represents the reliability of the components, which is the same for all since they are identical. This equation is taken from literature [99], however no equation can be derived for the case in which the components are different. This creates a gap in literature. These cases are solved statistically, as for the other non-conventional configurations. However, in this case an analytical equation is not existent. The solution for the automation of this configuration was the development of specific algorithms that solve it numerically with the use of Boolean truth tables, as suggested in the standards [99]. These algorithms are included in the software that was developed for this analysis. Some examples of this non-conventional architecture are cases in which 2 engines out of 4 must function, or cases in which 2 flight computers out of 3 are needed for a specific function.

RBD automation

The importance of automation in all steps of the methodology has already been highlighted. The automated solving of RBDs is not found in literature, and no standard or proposed method is available. A new method is developed for this analysis that allows to automatically read and solve RBDs from an architecture. This method is based on the RBD configurations that were commented previously, however it allows fast and easy additions of new configurations if needed. This section focuses on the component definition that allows the automated solving of RBDs, while the next section explains how to perform that automated translation from a physical architecture to a RBD-configuration. As a result, in this section the main input is a RBD configuration, and the focus is on how to properly define the components so that all the connections can be defined and the RBD can be solved automatically.

The method proposed is based on the iterative reduction of the initial RBD into consequent smaller configurations. For instance merging two components that are in parallel into one single conjunct, and so on. Some examples are shown later. The key to achieve this is on the correct and proper definition of the components following certain rules. Each component has seven attributes that are:

- Name: this attribute is not completely needed but supports the designer during the implementation phase, hence it is recommended. It is juts the component name such as "actuator" or "generator".

- ID: this is a unique number assigned to each component. It is needed by the algorithms to understand the total number of components that are involved in a configuration and to properly understand which components or conjuncts are reduced on each iteration. Each component has one ID that cannot be repeated. It is assigned with natural numbers starting from 1. The order in which they are assigned does not matter.

- Type: this attribute marks non-conventional components to assist the algorithms. For example an H-configuration is marked with an "H", a V-configuration with a "V", and so on. Some specific examples are shown later. The important thing is that the specific tag that is given here to these components is understood later by the algorithms during the implementation.

- Failure Rates: the failure rates of the component are given under this attribute. This is needed to estimate the final reliability of the configuration. Depending on the configuration and failure rates, the iterative process reduces the block into smaller conjuncts according to the equations presented in the previous section.

- Node to Fulfill: nodes are the enabling attribute of the automation process. They define the configuration (connections among components), and are just natural numbers. Each component is located between two nodes. The node to fulfill refers to the node after the component (this can be also be seen as the function that the component fulfills). The rules on how to properly define the nodes are further explained later.

- Node Needed: this attribute provides the missing node for the component. This is the node before the component (or function needed by the component).

- Auxiliary: the auxiliary attribute is needed by some configurations (e.g., V-configuration or R-configuration). They provide the extra information needed for those cases.

The attributes have been introduced, and a summary is displayed in figure 3.23. If all the attributes are properly defined for each of the components, the RBD can be solved automatically following the algorithms that are explained later.

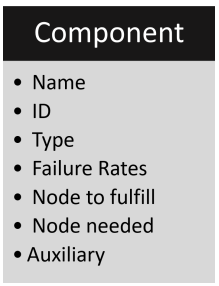


Fig. 3.23 Component attributes card

Before explaining the algorithms, it is important to further explain the concept of the nodes. This is explained with the example from figure 3.24. This example does not represent any architecture in particular and serves only as a reference for the nodes numbering and definition. Nodes represent the connections among components. They cannot be repeated. There has to be always at least one component between two nodes and there has to be always only one node between two components. The initial node, which is typically represented on the far left, must be assigned with the number zero. The last node, which is usually on the far right, must have the node 1. The rest of the nodes are consequently numbered with natural numbers from 2, until finishing. The number of each other node is not important as long as it is unique and not repeated. For instance, in figure 3.24 nodes 2 and 5 could be exchanged. However it is important to notice that no natural number can be skipped, this is important for the algorithms later. Component L for instance has as "node to fulfill" node 4, and as "node needed" node 7. Components A, B and C have the same node to fulfill (node 1), and the same node needed (node 2). Nodes are needed for the proper reduction fo the RBD into smaller configurations, properly merging components into conjuncts and then conjuncts into other conjuncts. For more examples on the node numbering the reader is referred to the last part of this section where some examples of RBDs are shown with components in black and nodes in red. For more information on the attributes for the different non-conventional configurations the reader is referred to the implementation section.

The algorithms logic for RBD automation is now explained. First, the number of components and nodes is counted. This can easily be done thanks to the information

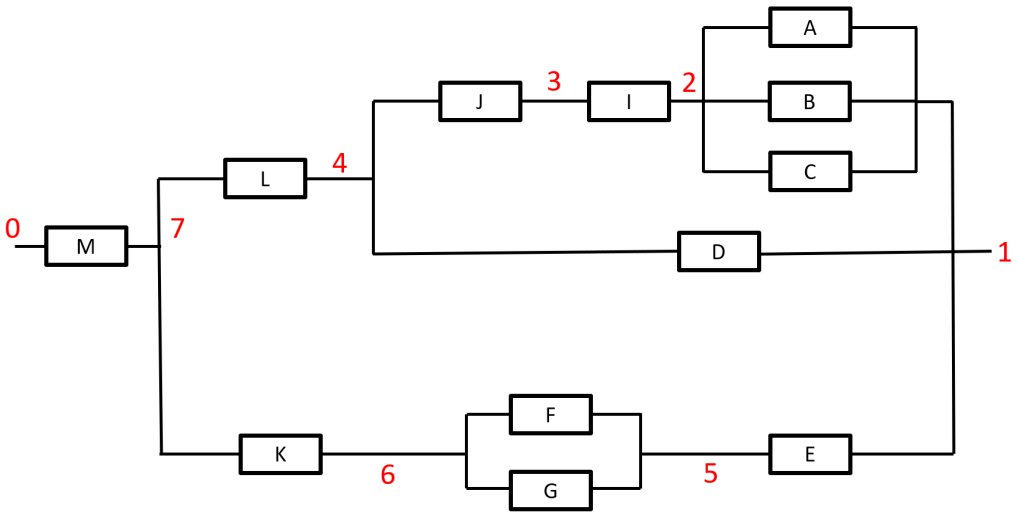


Fig. 3.24 Example on nodes numbering for a generic RBD

stored in the components attributes. After this, all components are transformed into conjuncts. Conjunct are just components or grouping of components with the same attributes as the components. For instance, a component "B" can only represent one component, but a conjunct can represent component "B" or components "B+C" merged together. This allows to separate the initial components from the posterior grouping of them, it also helps at tracking which components are merged with which and in which order. The main difference is that a component has failure rates, while a conjunct transforms this value into a reliability value directly. This makes it easier to use the equations and merge components among them. As a result, all components have been transformed into conjuncts (one conjunct per component at the moment) and they have a reliability value. The algorithms can now be applied by looking at some predefined structures (i.e., RBD configurations). Conjuncts can be merged with others if such structures are found. This follows an iterative process until only one conjunct is achieved. For instance, if two components (A and B) are in parallel, the algorithms first transform them into corresponding conjuncts (conjunct A and conjunct B). They now have a reliability associated, and are merged following the equations for parallel configurations. The result is a single conjunct (conjunct A-B), which reliability is equivalent as the parallel among A and B. If there is another component (C) in series with A and B, then the algorithms first do the parallel conjunct (A-B) and then merge it in series with conjunct C, resulting in conjunct C-A-B. The RBD gets progressively reduced by merging conjuncts among them through a series of iterations. The process is explained in figure 3.25. The iterative

process continues until all the RBD has been reduced into one conjunct. This final conjunct represents the reliability value of the whole system.

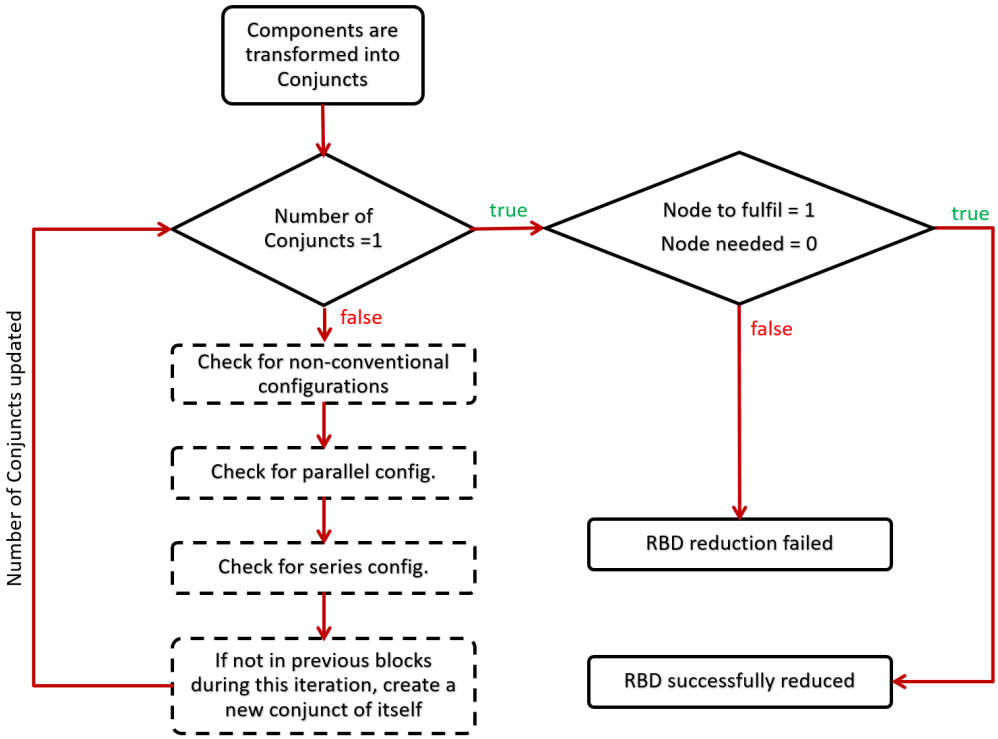


Fig. 3.25 Logic of the RBD automation algorithms

The iterative process shown in figure 3.25 is now explained with more details. The first step, as explained before, is to transform all the components into conjuncts. Then the iterative reduction process begins, looking for specific configurations and merging the conjuncts until only one remains. This conjunct represents the reliability of the whole system. There is one check after the process is finished in order be sure that the algorithms correctly reduced the diagram. It consist of checking that the nodes are correct and that initial function corresponds to node 0, and the final function to node 1. If the condition is true the reduction was successfully achieved. This is just used as a safety double check. The iterative loop is now further developed into steps:

- Check for non-conventional configurations: here the algorithm looks for non-conventional configurations. Precisely in a specific order which is H, S, V, T and R. By checking the nodes and types of the conjuncts it is possible to understand the different configurations in the RBD. The case for the H-configuration

is now explained but the others are done in an analogous way. After finding an H-type conjunct the algorithm reads the nodes and looks for components attached to those nodes until finding the correct other 4 components. Another check on the further nodes of those components is performed, this ensures that the configuration is a pure H-configuration that can be reduced. If for instance one of the nodes of the H-components is linked to a parallel configuration, the algorithms do not reduce it yet and skip to the following step. The configuration is reduced at a later iteration. Once a conjunct is reduced it receives the new attribute ID, starting from 1. If there are two H-configurations then both conjuncts have IDs 1 and 2, and so on. If no H-configuration is found this step is omitted.

- Check for parallel configurations: this step looks for parallel configurations, which means two conjuncts in parallel. One internal loop checks for a certain condition which is that two conjuncts share the same node needed and node to fulfill. If a third conjunct also shares the nodes it is left on hold for the next iteration (i.e., iteration 1 merges A-B; iteration 2 merges C-A-B, all in parallel). The conjuncts receive a new ID, the ones left on hold also receive a new ID. These IDs start from a count started in the previous step, if there was an H-configuration reduced the counter starts now at 2. Else it starts giving IDs from 1.

- Check for series configurations: this step looks for conjuncts in series. The logic is similar to the one used in the parallel configurations. Conjuncts are reduced 2-by-2, leaving on hold the case with three conjuncts in series. The logic is a bit different in this case, instead of checking the conjuncts it checks the nodes. If a conjunct has the current node as node to fulfill it increases a counter. If a conjunct has the current node as node needed another counter increases. A condition is evaluated after looping through all the conjuncts for each node. If the total amount of conjuncts with that node as node needed is one, and the total amount of conjuncts with that node as node to fulfill is one, that means that those two conjuncts are in series and they get reduced and assigned with a new ID.

- If not in previous blocks, create a new conjunct: this step makes sure that the re-numbering of the conjuncts is consistent. It sees which conjuncts have

not been reduced on this iteration and provides new IDs for them. At each iteration the IDs gets smaller since more conjuncts get merged.

This fully explains the algorithms logic but an example is now given to the reader since it helps to understand better the logic. Figures 3.26, 3.27 and 3.28 show and example of the iterative process for RBD reduction. On each iteration the RBD gets smaller and some nodes disappear. It can be seen how the triple conjuncts in parallel are solved by performing a double parallel between two of them and them doing another double parallel between the third component and the conjunct with the two previous ones. Also the H-configuration stays on hold until only five conjuncts are involved. This example takes 7 iterations to be fully reduced.

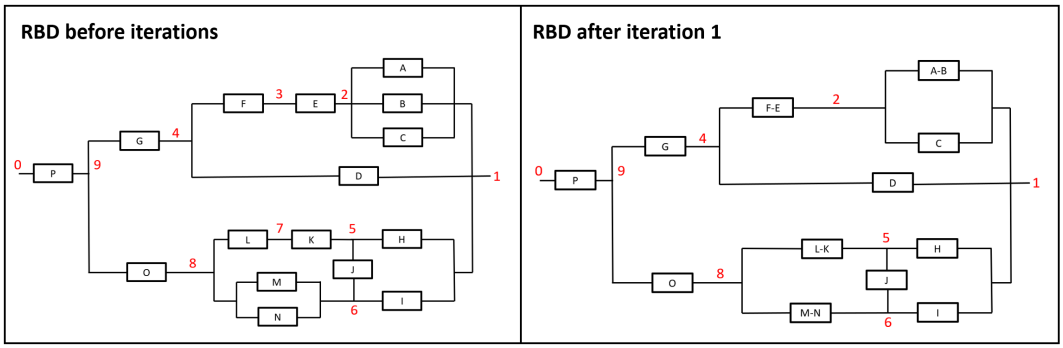


Fig. 3.26 RBD reduction example: iteration 1

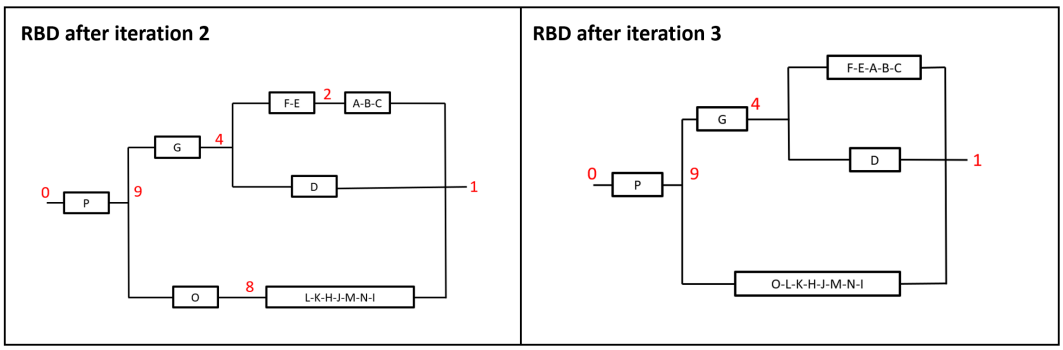


Fig. 3.27 RBD reduction example: iterations 2 and 3

The first and second iterations are now explained with more details. There are initially 16 components, which are transformed into conjuncts (i.e., from failure rates to reliability). The algorithms look for non-conventional configurations and they find an H-conjunct. However there are more than five conjuncts around this one, so

they stay on hold for now, not merging this conjunct. Then they check for parallel configurations and they reduce conjuncts A and B and M and N into two conjuncts (A-B and M-N respectively), with the IDs 1 and 2. Series configurations are now searched and the algorithms reduce two more conjuncts (F-E and L-K) with IDs 3 and 4. The rest of conjuncts are unchanged during this iteration, their IDs are now changed from 5 to 12. Leaving a total of 12 conjuncts for the next iteration. During the second iteration the algorithms immediately understand that the H-configuration is now around only 5 components and can be reduced. It performs such reduction creating one conjunct (LK-J-MN-H-I) with the ID number 1. Then conjunct AB is merged with conjunct C in parallel (C-A-B), and gives it ID number 2. There are no series conjuncts found during this iteration. The rest of conjuncts are numbered again with new IDs from 3 to 7. There are only 7 conjuncts left now that are merged in the following 4 iterations.

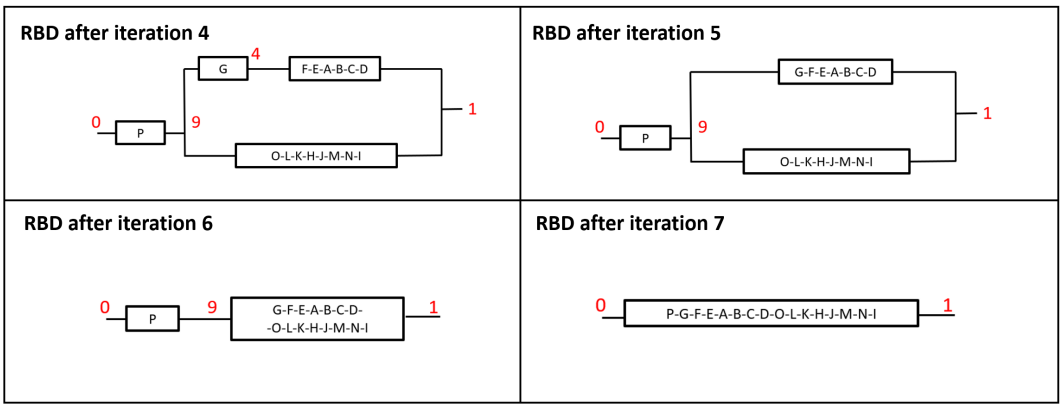


Fig. 3.28 RBD reduction example: iterations 4, 5, 6 and 7

Another example of the process is shown in figures 3.29 and 3.30. Here the RBD components are named with numbers instead of letters, it can be seen that there are initially 23 components and 13 nodes. This case can be reduced in just five iterations and contains one H-configuration, one S, and one V.

The first iteration only reduces some parallel and series conjuncts. While the second one manages to merge most of them already. This makes this example converge faster than the previous one.

This concludes the algorithms for RBD automation. The next section covers the topic on how to translate from a physical architecture into a RBD configuration that can be solved automatically.

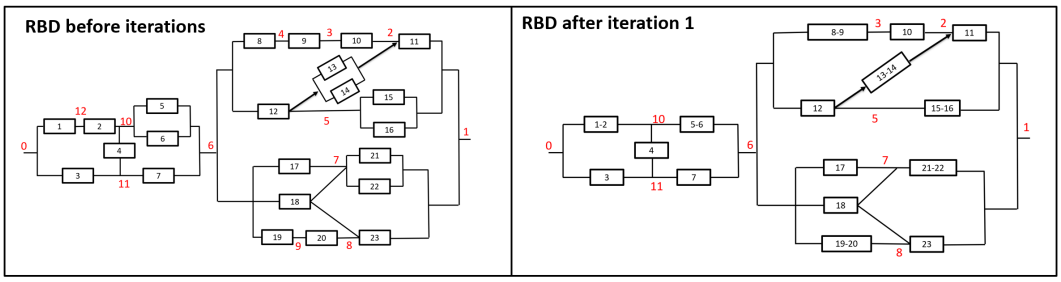


Fig. 3.29 RBD reduction, second example: first iteration

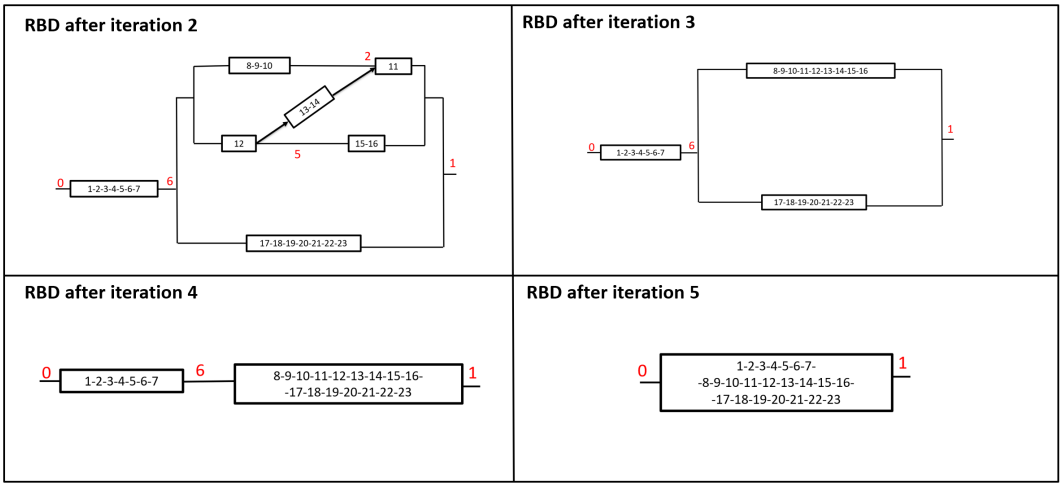


Fig. 3.30 RBD reduction, second example: rest of iterations

RBD translation

As mentioned before, and as already published in [146], a key task of the methodology is to be able to translate a physical architecture into a RBD configurations. The RBD configuration of a system is not the same as its physical/logical architecture. Furthermore, a system can have several functioning modes that translate into several RBDs. A consistent way to generate RBD configurations from an architecture of a system is needed, this represents a strong gap in literature. Some examples of this translation can be seen in [147], however the automation of this process is difficult as mentioned by the author: "Unfortunately, rules for the generation of a RBD (i.e., from a physical/logical architecture) do not exist because it is difficult to guarantee that they would always deliver the right model". Assigning logical components to components in the RBD diagram is easy. However, knowing the exact positions of these components in the RBD is a process that is tough to automate, since it is difficult to check the correctness of the result. One architecture can be derived into different RBDs, depending on the functionality of the system and the functioning modes.

A simple example of this is now provided to the reader. Figure 3.31 shows a system proposed in [147]. It consist of a tank connected to two lines that transfer the fluid to another tank. These lines have one pump each, and a flow meter. Both flow meters are connected to a flow control system. From this simple system, two functioning modes can be extracted. One line can be the main line of transport, having the other as a redundancy. This leads to functioning mode 1, as represented in the figure. However, it could happen that both lines need to work at the same time for time constraints. This means that the second line is not added as a redundancy, but as a requirement to pump the fluid faster from one tank to the other. This leads to functioning mode 2. Both functioning modes have different logical representations that lead to different RBDs. This issue creates problems to automate the process.

Despite this, for aircraft OBS an approach is developed in this analysis. These systems have very specific and well-known functionalities. This allows to recognize some patterns and to automate the RBD creation. One main issue arises however. The solution is completely implementation-oriented, which means that it is fully particularized for the specific software utilized in the analysis. The patterns are identified in the physical architectures generated by ADORE and in the input generation required by the RBD tool. The nature of this problem makes it tough to

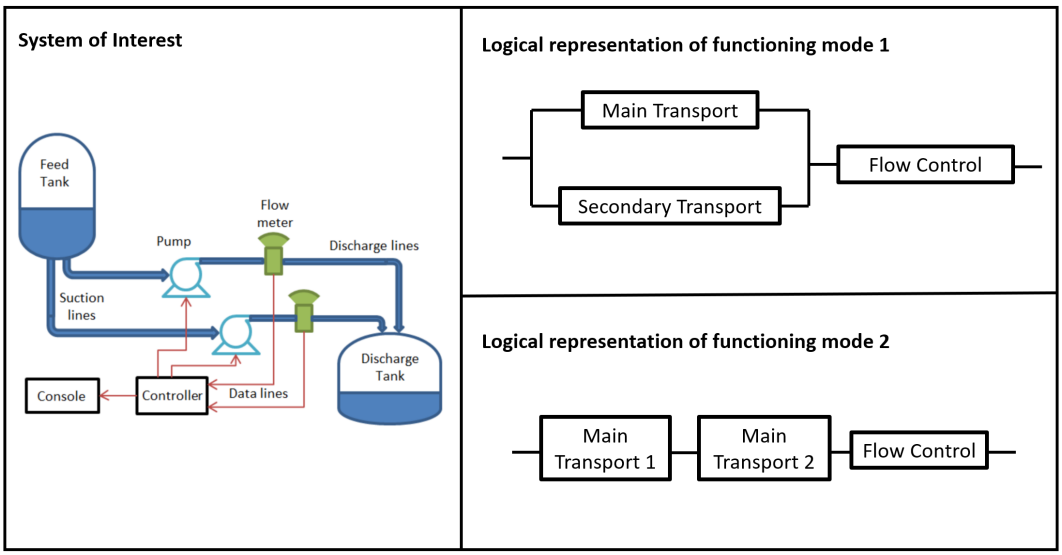


Fig. 3.31 Example of a system with various functioning modes; system of interest from [147]

create the required automation for a generic case, but a solution is found for the proposed implementation. For this reason, the rest of examples are commented and presented directly in the implementation section. One example of the process is shown here in figure 3.32, with the gas pipeline example introduced before in section 3.2. This system has only one function, to close the gas pipeline if a certain pressure is detected. This leads to one physical architecture (represented with ADORE) with one functioning mode. The translation to RBD is automated.

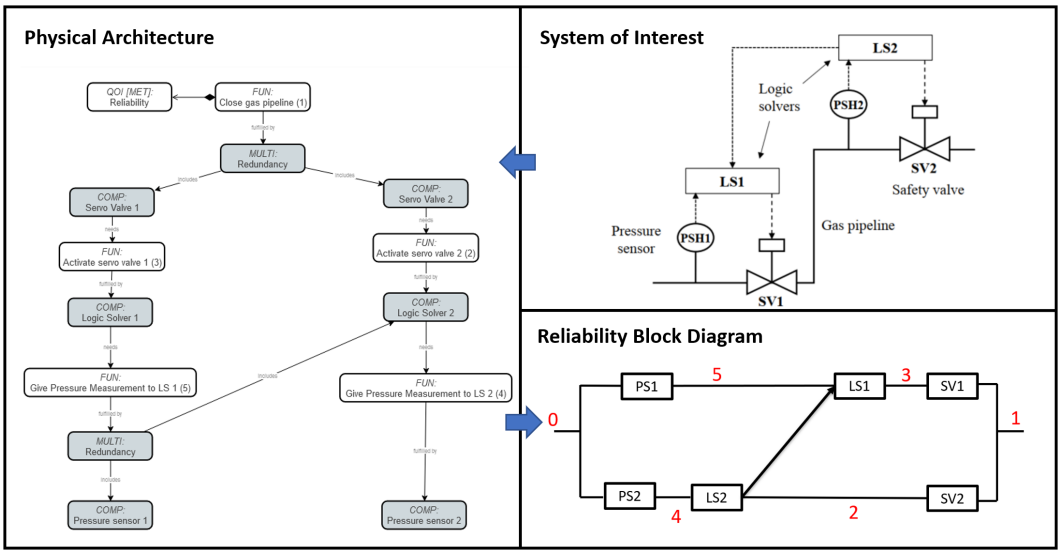


Fig. 3.32 Example of automated translation from physical architecture to RBD; from [146]

3.3.2 Implementation

The implementation of this section consist of several parts. First, the CS-25 requirements are used as constraints, having a 1 if they are passed and a 0 if not. This allows to automatically connect the results with the optimizer. For the RBS solving, a tool is developed, which is called ACOBS (Automated preliminary Certification of aircraft On-Board Systems, figure 3.33). This tool was developed specifically for the analysis of this PhD, and it is written in python. This tool takes care of two main things: the automated evaluation of RBDs and the automated translation of architectures from ADORE into RBDs. Both aspects are explained now in the following subsections. After addressing them separately, a common overview and summary of the tool steps and processes is given. Lastly, an important aspect is raised about how to address systems with more than 1 top functions.

Fig. 3.33 ACOBS - Automated preliminary Certification of aircraft On-Board Systems

ACOBS, automated evaluation of RBDs

This tool just replicates the equations, methods and algorithms already explained in section 3.3.1. Some further information about the tool is here provided on how to model the non-conventional configurations.

H-configurations are easy to model. The attribute type "H" is given to the bi-directional components so that the algorithms understand how to handle them. If there are two components in series acting as an "H-component" then both must have the type "H". An example of this is a power transfer unit that is modeled as two pumps in series. In this case both pumps must have the "H" type and the algorithms understand that they must be reduced into one conjunct of type "H". For the nodes it is not relevant for this component which one is the "node needed" and which one the "node to fulfill", because of the bi-directionality of the component. So if a cross-feed valve is located between two nodes (e.g., nodes 2 and 3) it does not matter

which node is needed and which one is to fulfill. Both cases work the same for this configuration.

S-configurations are similar to the H case. The component must be marked with an "S". In this case it is important to distinguish between the node to fulfill and node needed since they identify the direction of the S-configuration. Some of these configurations are created without a component linking them. An example is represented in partition 1 of figure 3.11. In this case a fake component must be added with the type "S" and failure rates equal to zero. This will allow the algorithms to understand the configuration without affecting the results.

V-configurations are more difficult to model and they need an auxiliary attribute. The type of the key component must be "V", both nodes must now be the two nodes to fulfill, and the node needed is now specified with the auxiliary. As an example, a component in V that is located among three nodes (node 4 on the left, and nodes 6 and 7 on the right) will have the following attributes: node to fulfill 6, node needed 7 and auxiliary 4. Nodes 6 and 7 can be exchanged, however it is important to leave node 4 as auxiliary to avoid confusion with other non-conventional configurations. This fully represents this non-conventional configuration.

T-configurations also need an auxiliary attribute. For this case two components are involved, and connected in the T-shape. Both components are configured as usual, having the regular nodes. However the node among both of them is not represented with a natural number as before. This is done for the algorithms to not thing that an H configurations is found. This "node" is marked with the auxiliary attribute with a letter. For instance, a T-configuration with two components T1 and T2 will have normal nodes for each of them (e.g., T1 has node needed 4 and node to fulfill 5; T2 has node needed 8 and node to fulfill 7) but both components have the same auxiliary (e.g., auxiliary of T1 and T2 is "t12"). If there is another T-configuration inside the same RBD a different tag for the auxiliary must be used (e.g., T3 and T4 have both auxiliary "t34"). This fully covers this configuration.

R-configurations also need an auxiliary attribute. This time it only provides the number "r" of the "r-out-of-n". For instance, a configuration with a 4-out-of-6 components needs all those six components to have the attribute "R" and the auxiliary "4". The rest is taken care by the algorithms since "n" is retrieved from the component count (i.e., how many components are among those two nodes).

ACOBS, automated translation of architectures from ADORE into RBDs

As mentioned before, the translation between the architecting part and the RBD needs to be tailored specifically for a given implementation. Here, different rules are created to map functions to nodes and components to RBD components. These rules are created so that all the non-conventional configurations are understood directly from their architecture shape, without the need to specify the type in the component. This means that the user must just create the design space in ADORE and on every architecture generation the algorithms check the type of connections among the components and understand which type of non-conventional RBD configurations are present. Then they automatically assign the corresponding type to the key components so that the RBD reduction algorithms can automatically solve the configuration.

This set of rules for translation is not defined here since it is only usable for this specific implementation (i.e., ADORE with ACOBS). However some examples are provided. One representative example is the one resulting from having two components in parallel. For instance, components A and B in parallel, each with a corresponding induced function that needs to be fulfilled by component C. For a RBD perspective, this architecture is the exact same as one in which components A and B generate the same induced function that is fulfilled by component C. From an architecting point of view these architectures have different functions but the same components. From the RBD point of view it is important to stablish the rules so that the algorithms realize that both architectures have the same RBD configuration in which the two induced functions merge into one single node. Rules such as this one are stored inside ACOBS and some examples of architecture translation for non-conventional configurations are now presented.

Figure 3.34 shows an example with a H-configuration. This is a simplified example of a system of flight spoilers. Spoilers 2 and 4 are powered by the yellow line or PTU. Spoiler 5 is powered by the green line or PTU. These redundancies are specified through the connections among components and functions in the architectural design space. The algorithms automatically assign the PTU component with an H-type and the RBD is solved. An example for the S-configuration was already presented in figure 3.32. For a T-configuration the difference with 3.32 is just an extra redundancy from logic solver 2 to logic solver 1. The nodes and auxiliary attributes are automatically filled. The only configuration that needs proper definition

in the architecting phase is the R-configuration. For this one a quantity of interest must be given in ADORE as an input. This quantity says how many components of the R-configuration must function (i.e., the r quantity of the r-out-of-n). The rest is fully automated.

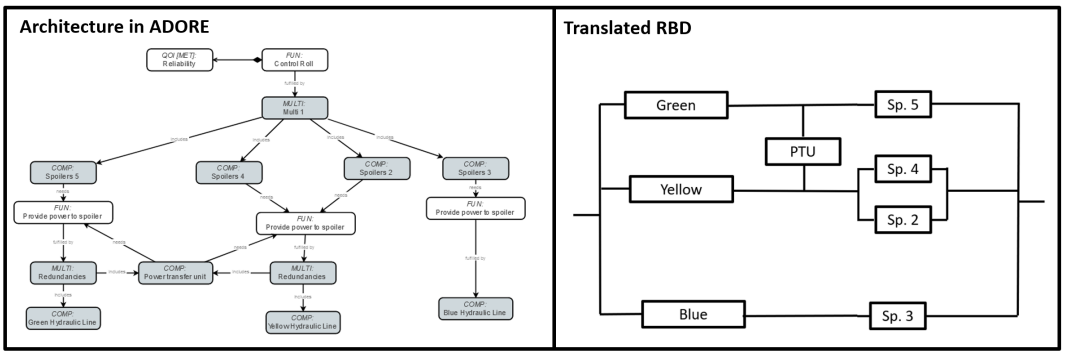


Fig. 3.34 Example of translation of an architecture from ADORE into a RBD, H-configuration

ADORE-ACOBS, overall view

This subsection provides a common overall overview of the process from the point of view of the implementation. Figure 3.35 shows such process in four steps:

1. Design Space Model: This model is created with ADORE. The user creates it manually through the user interface. Each component must have a quantity of interest assigned with the corresponding failure rates.

2. Architecture generation: ADORE automatically generates one architecture from the design space per iteration.

3. Translation to RBD: ACOBS reads the architecture. It translates functions to nodes according to a set of rules and it transforms the components with failure rates into conjuncts with reliabilities. It also assigns non-conventional types to some components if needed. The RBD configuration is built.

4. RBD reduction and solving: ACOBS reads the RBD and automatically solves it by using the reduction algorithms. A final value of probability of failure is provided to the architecture.

After this process, ACOBS evaluates the three constraints specified before: minimum reliability, single failure condition and back-up systems. For the minimum

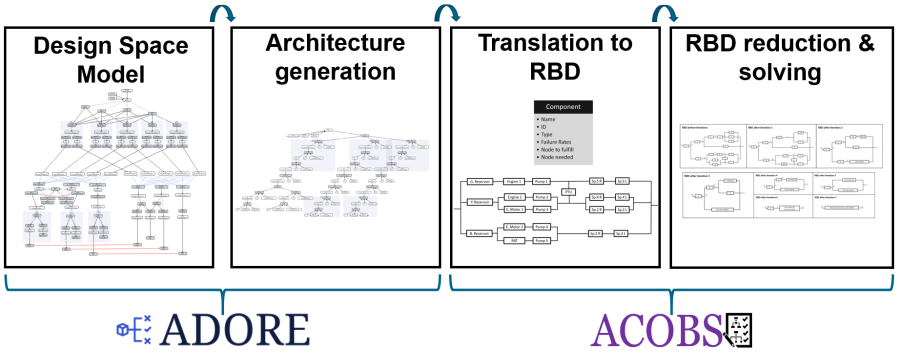


Fig. 3.35 Example overview of the process from architecture definition to RBD solving

reliability it just compares the result with a reference value provided as input. This value can be the one provided by the certification specifications or the result from a baseline architecture. The single-failure condition is evaluated directly in the RBD, this condition is easy to assess with some algorithms that are developed specifically for this purpose. They simply look for components without any type of redundancy and decide if the architecture passes this condition or not. For the back-up systems condition evaluation more input is needed. This is assessed thanks to the information provided in ADORE, each power distribution system needs to have a quantity of interest called "power source" which can be of type "engine" or "back-up". One rule loops through components and checks that at least one "back-up" component is found in the architecture and provides a 1 in affirmative case and a 0 in negative case. This is later linked with the corresponding constraints and provided to the optimization algorithms.

Evaluating more than one top function per system

This section explains how to handle the whole aircraft in which several systems are present with multiple functions each. Figure 3.36 shows this idea in a schema. Each OBS (or subsystem) needs to be addressed in a different and separated design space model. This means that each OBS is evaluated in a different framework, at least in terms of certification. One OBS can have different top functions (e.g., the FCS controls roll, pitch and yaw). This results in one design space model with several top functions. For each architecture generation, some components are assigned and this forces one architecture that fulfills all those top functions. Afterwards, each top function needs to be separated from the others, having three different "sub-

architectures" (one per top function) that can be given to ACOBS. The result is one RBD for each top function, having results for probability of failure per function (not per architecture). This idea is not implemented in the application case for limitations in the software, all the examples are shown for just one top function.

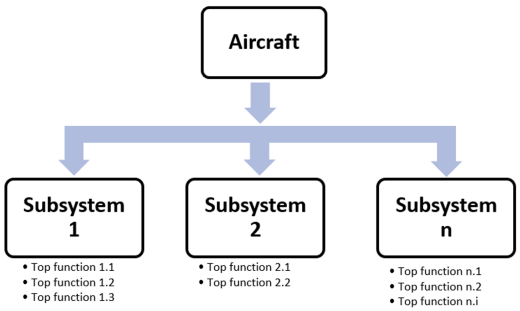


Fig. 3.36 Systems with more than one top function

3.4 Performance: Architectures Evaluation

This section explains with more details the part of the methodology regarding the architectures evaluation part, specifically the performance evaluation. It corresponds to the third block of the schema in figure 3.1 and to Blocks 3.0, 3.1, 3.2 and 3.3 in the XDSM diagram of figure 3.2.

As seen in figure 3.2, the performance evaluation framework has two main parts (i.e., OBS and aircraft evaluation) that are looped until convergence is found. This convergence loop accounts for the snowball effect that was previously explained in chapter 2.3.1 and allows to reach a better fidelity on the results. This kind of convergence loop has already been used in literature before [24]. The main input of this methodology part is an architecture of the OBS, which changes on every iteration; and some aircraft TLARs, which are constant among iterations. Each OBS architecture has certain components that have an impact on the weight and power consumption. It is essential to properly estimate these parameters in order to assess performance (i.e., impact in mass and fuel consumption). For this, a tool for OBS sizing is needed. Such tool needs to be able to provide results for all the OBS. One example is ASTRID [70], tool used in this analysis and explained in the following subsection. Another example of a commercial tool that could be used for this analysis is PACE, preliminary aircraft and systems design [148]. After the OBS have been

properly sized, they can be installed inside an aircraft baseline. The mass, bleeding and power off-take from the systems has an impact on the overall aircraft design. Assessing this effect is key to properly evaluate the impact of an OBS architecture. Concluding the analysis at a subsystem level (i.e., not measuring the impact in the aircraft) leads to non-accurate and non-representative conclusions. This error has already been reported in some studies [24], which estimated a deviation of a 40% error in the results when the mass and power off-takes effects where not considered at an aircraft level. A tool for overall aircraft design is hence needed. OpenAD [149] is used for this analysis but other available tools could be used such as FAST-OAD [150].

The huge number of possible architectures creates a new problem that is not usual in this kind of frameworks. An example is that one of [3], here only four different OBS architectures are evaluated. The OBS sizing and OAD tools can be connected directly without much technical issues. However for an application case with for instance a million different architectures it is quite difficult to successfully connect the sizing tools with every possible architecture. As a result, the available tools and software cannot typically evaluate all the architectures that are generated from the design space. This is due to the tools design itself, which does not usually prevent unconventional or non-feasible architectures to be evaluated. For example, the design space allows to create electrical systems with only one electric line. This cannot be installed in a real aircraft for safety reasons since back-ups are always needed. Also, having one system for only 1 line is quite sub-optimal. For this, tools that evaluate electrical system characteristic do not allow an input of just one electric line, since it is not worth it to evaluate an architecture that is known a priori to be non-feasible. However, the design space model does not know this and can generate such architectures. As a result it is possible to have an architecture generated from the design space that cannot be sized later by the evaluation framework. Some adjustments need to be done to the workflow to adapt and avoid these cases and are shown in chapters 4 and 5 directly with each of the application cases.

Some things can be done to the framework in order to avoid the issues commented in the previous paragraph. One is to just assume that an architecture is not feasible if it does not run successfully inside a tool. This solution although simple can hide other errors inside the tools and hence it is not recommended. Another solution is to create surrogate models for the tools. A surrogate model is a simplified approximation of a more complex and higher-order model [151]. They are usually used to link

input data to outputs when the actual relationship between the two is unknown or computationally expensive to evaluate. In this context they can ensure robustness in the results (e.g., a tool might crash if an unexpected architecture is given as input, but a surrogate model is usually ready to provide results for such extreme cases).

Another simplification of the framework is the removal of the convergence loop. However, this can only be done in certain conditions and needs to be tested and checked for each case in particular. Such simplification is done in both application cases of this Ph.D., this is possible since only one power consuming system changes from one architecture to another (i.e., only FCS). If the aircraft baseline is properly sized, only one power consuming system changes (and is a low-power utilizer) and the architectural changes do not significantly modify the aircraft structure, then the convergence loop can generally be removed. This comes from the effect that the architectural changes under such conditions can typically have an impact of around a 1% of the MTOM, or less. This makes the convergence loop reach convergence in only one iteration, which makes it possible to simply remove further iterations. It must be highlighted that the aircraft baseline initially provided must be similar to the final aircraft, and that these results must be tested. When these conditions are met, another last simplification can be done. This simplification consists of separating the sizing of the system of interest and the rest of the OBS. If the architectural decisions only affects one power consuming system it is not needed to size the others on every iteration. The application cases of this Ph.D. focus on the FCS, as a result a separated tool can be built to properly assess the FCS architectures and later use a more general software for the rest of OBS. This allows to reach a higher level of detail in the system of interest without losing the information on the rest of the OBS. Distribution systems need to be taken care of accordingly. An example would be developing a Level-2 (explained in section 2.3.1) tool for the system of interest, keeping a Level-1 for the distribution systems and a Level-0 for the rest of the OBS. The separation of systems and removal of convergence loop also helps to alleviate computational times, providing more than one advantage.

A new XDSM is obtained if all the simplifications previously commented are applied to the XDSM from figure 3.2. The simplified version can be seen in figure 3.37, showing only the evaluation framework part which is the one that changes. The certification and maintenance domains remain untouched, but the performance part is different. The diagram is particularized for the analysis done in the application cases, in which the FCS is the main system of interest. If other system is analyzed

the methodology remains the same and the dedicated tool needs to be done for that system. The simplified performance evaluation framework consists on four blocks that are now described:

1. Aircraft Sizing Initialization: this block is in charge of creating a proper aircraft baseline in which the OBS architectures will later be installed. An OAD tool is used for this purpose. The input consist of some TLARs (e.g., passengers, cruise mach, range) that are used to preliminary size the aircraft. This baseline provides as output some geometrical data, masses, efficiencies and characteristics of the aircraft (e.g., wing span, MTOM, specific fuel consumption, cabin length), which build the required baseline.

2. Flight Control System Sizing: this block sizes each of the architectures with certain level of detail. A Level-2 or at least Level-1 of detail is recommended. The main input is the TLARs and architecture, and the main output is the summary of mass and power required by the system. A dedicated tool can be used although any OBS sizing tool is also feasible as long as it can reach the level of detail of the design space.

3. On-Board Systems Sizing: this block sizes the rest of the OBS. Distribution systems must have a feedback or connection with the results from the previous tool so that architectural changes are properly caught. As input the tool takes the aircraft baseline and provides as output the mass, power and power off-takes needed by the OBS.

4. Aircraft Sizing Synthesis: this block re-sizes the aircraft from the baseline with more detailed information about the OBS. This allows the aircraft sizing to be done accounting for the OBS architectural changes (e.g., an aircraft without hydraulic system is generally lighter). As a result this provides a more precise and higher-fidelity aircraft baseline. Values such as the MTOM or fuel burn are now more meaningful and sensitive to the architectures. The previously commented snowball effect is also considered with this methodology.

This diagram is further explained on each of the application cases and the used tools are now commented under the implementation section.

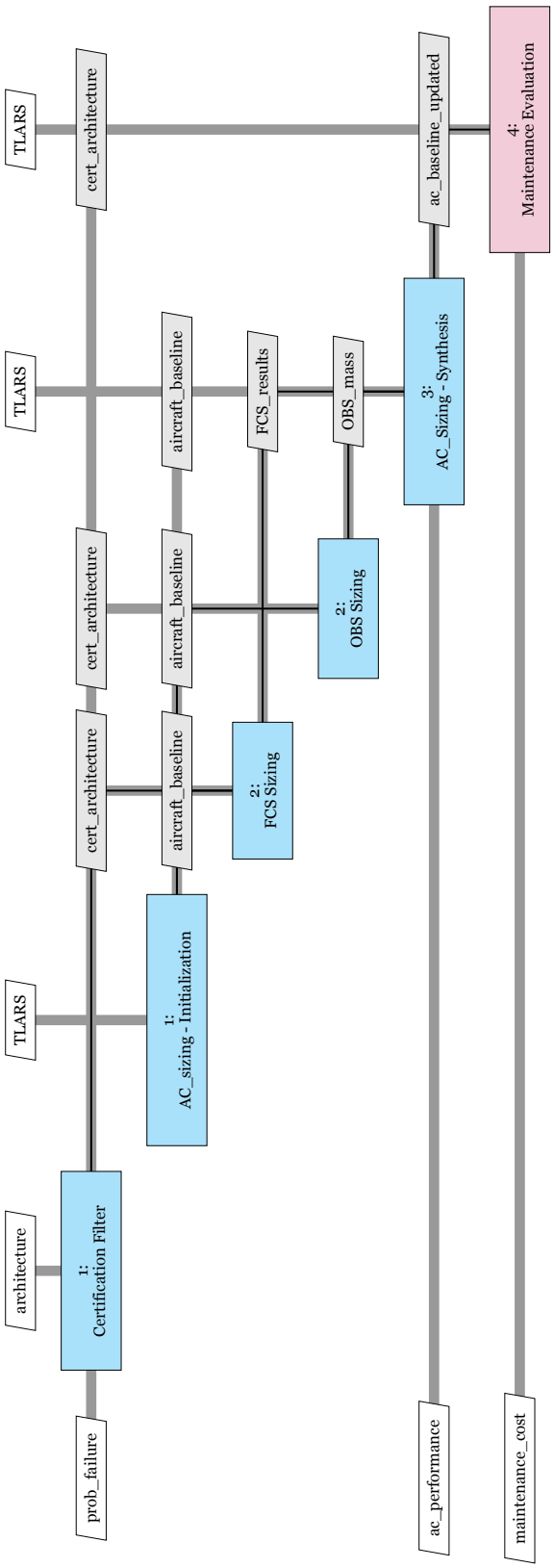


Fig. 3.37 XDSM of the evaluation framework, simplified version without convergence loop

3.4.1 Implementation

As mentioned before, three tools are used for this analysis. One for overall aircraft design, one for on-board systems sizing and one dedicated for flight control system sizing.

Overall Aircraft Design - OpenAD

OpenAD [149] is a DLR in-house tool used for overall aircraft design that has been extensively used in several projects. As input it takes a CPACS file [152], an xml file that has a list of input data with the TLARs (e.g., mach number, passengers, range, take-off length, etc...). These values can be set as fixed or initial estimations, which help with computational times (e.g., providing an initial MTOM, wing load, or geometrical input such as the wing dihedral angle or the aspect ratio). This flexibility allows to use the tool for the generation of the initial baseline and for the final sizing of the aircraft using the same file. The first time using some initial estimations for the baseline generation, later fixing such results with the values obtained by the other tools. As an example, the OBS mass is initially provided as an initial estimation. After the exact value is obtained with the OBS sizing tool this value is updated in the CPACS file and the attribute of the parameter is switched from estimation to fixed value. Following this procedure the second run of the file does not need long times to converge since it does not differ much from the initial estimation. The success of this depends on the quality of the first estimations, however this process can be iterative in order to create a better initial baseline that makes posterior runs converge easily. As an output OpenAD provides a detailed model of the aircraft with different results for different disciplines. A detailed aerodynamic map is provided, the geometry is fully defined (i.e., fuselage, wing, tail, systems, etc...), the mass breakdown is provided with details per system and per aircraft part, the fuel burn is estimated, etc... OpenAD is based on well-known and well-understood handbook methods (i.e., Torenbeek, Raymer, Rosskam, Jenkinson). The reader is referred to [149] for further details. Here there is more information about the masses estimation and center of gravity, geometry definition, aerodynamic modeling, engine performance modeling and mission fuel calculation. The average run time depends entirely on the correctness of the initial file and initial estimations. If this is done properly the software can produce results quite fast. For this analysis, an A320neo baseline is

generated. After several runs and estimations a proper baseline is built that takes on average around a minute to converge per OBS architecture.

On-Board Systems Sizing - ASTRID

ASTRID [70] is an on-board systems sizing tool developed at Politecnico di Torino. It has been used in several projects and allows to properly size the OBS with different levels of electrification. It allows to select the level of detail (i.e., Level-0, Level-1 or Level-2, as explained in 2.3.1) for each OBS separately. Although not all systems have all the levels available. As input it takes a series or requirements and data from a file. This file can be a CPACS file, which simplifies the data exchange with OpenAD. However an extra internal file is needed that contains information about components such as mechanical efficiencies, material properties, etc. Some necessary input examples are the MTOM, range, wing shape, fuselage dimensions, tail size, number of passengers, the engine characteristics, etc. As output, it provides an estimation of the mass, off-take and bleed required per system. The design method follows the classic approach starting by the power consuming systems, then the distribution systems and lastly the power generation systems. The power consuming systems are sized from the requirements previously mentioned (e.g., the number of passengers and cabin dimensions affect the sizing of the ECS). Once all users are sized in terms of mass and power, the power distribution systems can be designed according to such power demands. Lastly, the power previously estimated needs to be provided by the power generation systems, this sizes the systems such as the RAT and APU, but also calculates the final off-takes and bleed that need to be extracted from the engine. The tool is sensitive to more-electric architectures, which improves the fidelity of the results for the studied application cases. For more detailed information on the equations, models and methods used in ASTRID, the reader is referred to [70, 95, 153]. These references present the base methods behind the tool. Other systems have been updated in later publications such as the environmental control system [77]. These updates provide more refined models that are needed for more-electric version of some OBS. Apart from these, the tool has been used and calibrated in other further studies. Some examples of different MEA and AEA architectures are found for a 19-passenger in [24], for a 90-passenger in [154], or for a a business jet in [155].

Flight Control System Sizing - Dedicated Tool

This tool for FCS sizing is developed to provide a higher level of detail compared to the one provided by ASTRID. Some architectures that are generated in the design space cannot be sized by ASTRID, and some technologies are not included. Also, the huge amount of architectures creates non-conventional configurations that cannot be easily linked with ASTRID's input. For these reasons a dedicated FCS tool is developed and explained in this section.

The tool follows the methodology proposed in [156], since this can provide component-level estimations with only high-level input. This input is variable giving two different options to the user. The first one is to provide directly the hinge moments and forces on each of the FCS surfaces. This option is quite powerful for analysis in which experimental data is available, or in which these forces are already known or estimated with higher-fidelity analysis such as CFD. The second option is to use the proposed surrogate model [156] to estimate the hinge moments and forces from some aircraft TLARs. For this the required input are only four parameters: MTOM, cruise mach, wing surface and vertical tail surface. With only these parameters the model is able to estimate the hinge forces and moments. This methodology can only be used for commercial aircraft. After this estimation the rest of the input must be provided. This input consists of the architecture of the FCS. The required parameters are: number of ailerons, elevators, rudders, spoilers, flaps and slats, number of actuators per surface previously defined, and type of actuator for each. The actuator type can be HSA, EHA or EMA for ailerons, rudder, elevator and spoilers. The actuators type is ballscrew or EMA for flaps and slats, but the PDU can be selected as hydraulic or electric in the ballscrew case. It can be noticed how the input is quite generic, allowing non-conventional and non-typical architectures. The TLARs needed for the hinge moments estimation can be extracted from the general TLARS and the CPACS file generated by the first iteration of OpenAD. The FCS architecture is extracted directly from the architectural decisions. All the input is available and does not require a lot of knowledge. As output this tool provides a detailed description of the components of the architecture with the mass of each of them. It also provides the overall maximum an average power required by the system. An example is found in table 4.4 in chapter 4 once the application cases are assessed.

The methodology is validated with the hinge moments presented in [67] and the actuator masses from [157], providing results in line with the expectations. Further results are shown at the end of this section. However, the methodology [156] lacks the mass estimation of some new components needed in the application cases. EMAs can be sized but need some extra calibration and additional information in order to be usable for some innovative concepts. One study from the 1990s [47] provides some data for an EMA, stating that a stall load of 133 kN requires an EMA of 36.3 kg. Introducing this stall load in the equations provided in [156] the result is an EMA of 39 kg. The minor difference of 3 kg between the model and the real actuator is considered negligible and validates the equation usage for the EMAs, since these kind of actuators were not validated originally in [156]. The electronic devices that are added to the high-lift devices based on EMAs (chapter 4 for reference) must also be accounted for. The study in [44] shows a mass breakdown of an EMA, giving the the power electronic components a 12% of the mass of the whole actuator mass. Such electronic devices are included in the actuator mass, however the actuator control electronics are not generally included. An extra 10% increase in mass is added to the total mass of the EMAs for high-lift devices to account for such devices. The tool is developed in python following the proposed methodology with the mentioned enhancements. This methodology and tool has also been used in [158, 110], where further validation and analysis are provided.

Three different aircraft that are evaluated with the tool are now shown for validation purposes. The architecture description of each is now presented while the input data of the surrogate model and the final estimated value of the FCS mass are shown in table 3.1. The Airbus A320 has two ailerons with two HSAs each, two elevators with two HSAs each, one rudder with three HSAs and ten spoilers with one HSA each. For the high-lift devices there are two inboard and two outboard flaps with ballscrew actuators and a hydraulic PDU and ten slats with ballscrew actuators and a hydraulic PDU. The Airbus A330 has four ailerons with two HSAs each, two elevators with two HSAs each, one rudder with three HSAs and twelve spoilers with one HSA each. The high-lift devices consist of two inboard and two outboard flaps with ballscrew actuators and a hydraulic PDU and fourteen slats with ballscrew actuators and a hydraulic PDU. Lastly, the ERJ190 has two ailerons with two HSAs each, two elevators with two HSAs each, one rudder with two HSAs and ten spoilers with one HSA each. For the high-lift devices it has two inboard and two outboard

flaps with ballscrew actuators and an electric PDU and eight slats with ballscrew actuators with an electric PDU.

Table 3.1 Dedicated FCS tool: input and mass results for three different test cases

Aircraft	MTOM [kg]	Cruise Mach	Wing Surface [m]	Fin Surface [m]	FCS Mass [kg]
A320	73500	0.78	124	21.5	826
A330	240000	0.8	363	48	2380
ERJ190	52000	0.78	92.5	16.2	602

The difference in the A320 FCS mass value (826 kg) compared to the one calculated in [156] (756 kg) is noticeable. This discrepancy is expected and can be explained. In [156], precise hinge moments and forces for the A320 are used, derived and taken from various sources. The paper explains that hinge forces differ significantly between Airbus and Boeing models due to structural and geometric differences (e.g., slight variations in chord ratios). To avoid biasing the result toward one model, using an average between the two is recommended. However, the final results use specific values for the Airbus model. Boeing models exhibit higher forces and moments, so using Airbus values leads to a lower FCS mass, while using Boeing values results in a higher mass. Since this analysis does not aim to estimate the hinge models, the surrogate model for hinge moment estimation proposed in [156] is applied. This allows the subsystem to be sized using only a few TLARs, without needing specific hinge moment values. Consequently, this leads to a slight overestimation of the hinge moments and forces, resulting in a higher FCS mass, which fully accounts for the observed difference in results. The other results are also in line with the expectations. Further validation of this tool is shown in [158, 110].

3.5 Maintenance: Architectures Evaluation

This section explains with more details the part of the methodology regarding the architectures evaluation part, specifically the maintenance evaluation. It corresponds to the fourth block of the schema in figure 3.1 and to Block 4 in the XDSM diagram of figure 3.2.

This part of the methodology is addressed after the aircraft and OBS are sized. As a result the aircraft characteristics are known (e.g., MTOM, fuel consumption, geometry, thrust, or cabin length). Further information is needed, such as the fleet

size, daily utilization of the aircraft or seat configuration (TLARs that were not used before for the sizing). For this framework the aircraft is sized and fixed, as well as the further input. This means that these added parameters are constant for the analysis, hence analyzing the impact of the OBS architectures for a determined value of utilization, fleet and flight hours per flight cycle. These inputs are specified for each application case in their corresponding chapters directly. However, as proved in [131], these parameters have an impact that is not too relevant for similar aircraft (i.e., small differences do not cause big changes in the results). Assuming some values and making them constant is hence advised to simplify the analysis.

A method for maintenance cost estimation is required, as presented in the state of the art review (2.3.3). This method shall reach component level and it is limited to the DMC estimation. The result of this part of the methodology is the maintenance man hours (MMH) that an OBS architecture needs with respect to a conventional one. For instance, if a certain commercial aircraft is evaluated, the result would be the increase or decrease in MMH that a certain new OBS technology achieves when compared to the baseline one. The methodology here presented is the result of two previous analyses carried out in the context of this Ph.D. [33, 131]. Although some updates and small modifications are needed. The main structure and concept of the methodology are here explained, however the reader is forwarded to the two references if more details are needed. A surrogate model is created, which is able to estimate the MMH when changing the aircraft OBS. For the methodology behind the surrogate model, the reader is referenced to [33], while for the creation of the model itself the reader is referenced to [131].

The methodology starts by performing the so called "MPD analysis", as explained in [33]. The MPD of the aircraft is needed as a result, although the MPD of a similar commercial aircraft can be used if such document cannot be obtained. The MPD serves as a baseline for the maintenance tasks and maintenance times for each task. This analysis consist of going through all the scheduled tasks that are reported in the MPD, and remove or modify those that change if a new OBS architecture is installed. Some tasks disappear from one concept to another, other tasks are substituted by others, others are added. For instance, one example is if the FCS concept is changed from a conventional one with HSAs to a more-electric one with EHAs. Those tasks related to the actuators change, since the actuators are still present but the maintenance times and frequency changes for them. The tasks related to the distribution systems also change. The ones for the hydraulic system might get

reduced or directly disappear, since they are not needed anymore for the new concept. Nevertheless, new tasks associated to the electrical distribution system appear for the same effect. When a task disappears, the corresponding MMH can be directly removed since that task is not needed anymore. When a task changes, the MMH change following a certain ratio. This ratio is suggested to be the ratio between the failure rates of the old and new component (assuming always constant failure rates). This idea simplifies the addition of new tasks by only knowing the failure rate ratio. At the end, the sum of all the MMH from the different tasks can be done, considering the contribution through the different maintenance checks (i.e., A-checks, B-checks, C-checks and D-checks), and taking into account their correspondent intervals. This provides the total amount of MMHs with respect to the baseline architecture.

The analysis done in [131] created a surrogate model for the MPD of the A320 for different more-electric and all-electric OBS architectures. This simplifies the methodology for those who do not have access to the MPD of the Airbus A320 but want to estimate the DMC of innovative OBS architectures. The result of the study is a list of equations that can be used directly to assess changes in the DMC when certain components are switched from conventional to innovative ones, while covering the most relevant OBS. Two methods are mixed in order to obtain such equations: MPD analysis and expert interviews. The MPD analysis is the one commented in the previous paragraph without changes. The expert interviews consist of a series of questions and insight given by experts in maintenance and OBS architectures that were used to mitigate the results. A Monte-Carlo simulation is performed with the results from the interviews [131]. The final results are the already mentioned equations, which are mitigated and provide more realistic results. Each OBS is represented by one equation, making it possible to assess the impact of each subsystem separately. This provides flexibility to the model and allows to isolate the impact per system. Some small adjustments to the equations are needed for the analysis performed in this Ph.D. However they are shown directly in each of the application case chapters since they are particular for each case. These adjustments are needed to provide a higher sensitivity to the results.

3.5.1 Implementation

No specific tools were used for the implementation of this part of the methodology. This consists just on the use of the main equations provided by the surrogate model

created in [131], which are connected automatically to the architectural choices from the design space. The corresponding methodology adjustments are implemented after such equations are used, providing the extra level of sensitivity required. For more reference the reader can see the application cases chapters that provide some specific examples. The process works as follows: the architecture, TLARs and results from the performance analysis are given to the surrogate model. This model provides results per OBS for the MMH change with respect to the conventional aircraft. Some application-case-specific adjustments are applied after the results from the surrogate model are obtained, generating the definitive results. The main outcome comes from the sum of the MMH from all the OBS, providing a total MMH increase or reduction for the evaluated architecture. All these steps are implemented with a simple python code.

3.6 Optimization Process: Metrics Evaluation

This section explains with more details the part of the methodology regarding the optimization process. It corresponds to the lower part of the schema in figure 3.1 and to Blocks 0 and 5 in the XDSM diagram of figure 3.2.

The output for each iteration of the process is an architecture from the design space. The architecture is selected through the architectural choices that can be discrete (i.e., a selection from various options) or continuous (i.e., a value between certain margins such as a mach number). These choices are mapped to the design variables of the optimizer. Having discrete variables makes the optimization problem itself also discrete, as a result genetic algorithms are needed. Gradient optimization methods are not feasible for such discrete problem since the gradients cannot be properly provided. Constraints are linked to the certification parameters. Optimization objectives are linked correspondingly to each of the metrics obtained in the evaluation framework. A summary of the optimization problems is now provided:

- Design variables: architectural choices
- Constraints:
    1. Single Failure
    2. Back-up Systems

3. Minimum Reliability

- Optimization objectives (from different disciplines or domains):

1. MTOM: minimize - Performance domain
2. Probability of failure: minimize - Certification domain
3. Maintenance Man Hours (MMH) decrease: maximize - Maintenance domain

The whole process works as follows. The optimization algorithms select some design variables based on the results from the previous population (i.e., previous runs). These design variables are mapped to the architectural choices and define a new architecture for the system of interest, that comes from the given design space. The architecture goes into the certification filter, where the different certification checks are done. If the architecture passes all the constraints (i.e., single failure, back-up systems and minimum reliability) it is sent to the rest of the evaluation framework. If the architecture fails one, or more, of the constraints then the rest of the evaluation is skipped and the architecture is considered as non-feasible. When an architecture is feasible it is further evaluated in the performance and maintenance blocks. At the end of the process the architecture has three optimization metrics assigned: MTOM, probability of failure and MMH. These metrics are analyzed by the genetic algorithms and another architecture is chosen for evaluation. The process continues until the algorithms stop based on certain conditions, after this the optimum points are selected. Since the optimization problem is multi-objective the result is a Pareto Front (three objectives create a 3-dimensional Pareto Front). Only after this whole process is finished the trade-off analysis can be performed as the last step.

3.6.1 Implementation

For the implementation ADORE is also used [8, 144]. As commented before in section 3.2, ADORE includes some optimization algorithms and allows to automatically map the results from the optimization objectives to the design variables. The algorithm used is the NSGA-II [159], but this is further explained directly for each of the application cases. ADORE is connected to the rest of the framework and

performs the optimization process from the design space directly, with the selected algorithms.

3.7 Trade-Off Analysis: Metrics Synthesis

This section explains with more details the part of the methodology regarding the trade-off analysis. It corresponds to the fifth and last block of the schema in figure 3.1 and it is not represented in the XDSM diagram of figure 3.2 since it is performed after the optimization process, once the most promising architectures have been found.

The optimum architectures are provided as input. The output is a trade-off analysis with the most promising architectures according to certain parameters from different scenarios. Three different metrics from three different disciplines are traded, it can be a challenging task to decide which discipline or metric is more or less important. A methodology that allows to perform multi-criteria decision-making is needed in order to properly perform such trade-off. The Multi Attribute Utility (MAU) theory [160] is used for this analysis. This methodology is suggested by [161] for the trade-off of manufacturing and supply chain of aeronautical products. It allows to aggregate multiple criteria from different domains (i.e., disciplines) into one single result, called "value". This value provides a quantitative estimation of the satisfaction of a decision maker towards a certain solution, it is the weighted aggregation of the satisfaction towards multiple metrics or indicators. This manages to quantify somehow the preferences of the designer and stakeholders while aiming at selecting the best solution from the trade space. In general the MAU theory is recommended when the number of disciplines is at least three [162], which is precisely the number that is had later in the application case. However, the idea of this study is to keep adding disciplines and life cycle stages to the analysis in the future (e.g., manufacturing and disposal). This makes the MAU really suitable since it allows easy and fast addition of new parameters, as shown later in the equations.

The equation to calculate the value is shown in equation 3.18, and the explanation of each element of the equation is provided in the following paragraph:

$$Value = \sum_{i=1}^{N} \lambda_i \cdot U(X_i) \tag{3.18}$$

Where:

- N is the total number of attributes
- λ_i is the weight associated to attribute i
- $U(X_i)$ is the single attribute utility value function for attribute i

The Value is a quantity with no dimensions that goes between 0 and 1. The closer to 1, the better the solution meets the expectations from the designer. Attributes are the different metrics used to evaluate the different disciplines, they represent how well an decision objective is met [163]. Weights represent the level of relative importance of the attributes, λ_i is a value from 0 to 1, the sum of all weights must be 1. As a result, λ_i can be seen as the percentage importance of an attribute among all the attributes. Weights are defined by the decision maker based on experience and knowledge. As an example two attributes can have a weight of 0.5 each, meaning that each is traded with a 50% of importance. Or they could have a 0.2 and 0.8, meaning that one is much more important than the other. Utility functions model the stakeholder's expectations with regards to an attribute, in practice they translate qualitative preferences into quantitative functions. They can be linear or not and are typically constrained between the maximum desirable and least acceptable results for that attribute. Examples of attributes and utility functions are shown directly in the application case chapter 5, section 5.4.

One practice is to aggregate all the different results of an analysis into the value leaving the cost aside [161]. This results in a 2-dimensional trade-off analysis result representing the cost in one axis and the value in the other. The value represents all the metrics apart from cost. This creates an interesting Pareto Front that can support the decision making process. One example is shown in [161] where the value aggregates different production and supply chain metrics (i.e., time, risk and quality). However, in the analysis here presented the cost is still not estimated. For this reasons a one-dimensional trade-off analysis is suggested in which everything is aggregated into the value. In the future, once more disciplines are evaluated these can be included easily into the value, and once the cost of an architecture is assessed it can be represented as the second dimension of the solution trade space.

3.7.1 Implementation

Valorise is used for the implementation of this part of the methodology. It is a DLR in-house software that allows to model the utility curves with a user interface. It also allows to import the results from an analysis and to define their attributes and weights. It shows the solution trade space and assists the designer in the decision making process.

Chapter 4

Application Case 1: High-Lift Devices

This chapters shows the first application case. It focuses on the high-lift devices, more specifically in comparing new innovative architectures to conventional ones. This application case does not focus on the system architecting part of the methodology in which a huge design space is present. Here a small number of architectures are compared, leaving the next application case (in chapter 5) deal with this problem. Consequently this application case focuses on executing the methodology step-by-step for a predefined number of architectures, showing the details of each of them. There is no optimization process as a result and the objective of this analysis is to evaluate each of the three disciplines (i.e., performance, maintenance and certification). Some parts and results of this application case were already disseminated in [158].

Section 4.1 explains the main reason and purpose of this application case, and why it can be interesting to perform such analysis. It provides the necessary context and shows previous relevant studies on the topic. Section 4.2 focuses on the adjustments, extra explanations and modifications done to the original methodology so that the reader can perform the study in the same way. It has one subsection per discipline. Lastly, section 4.3 contains the results.

4.1 Motivation & Context

The main motivation of this application case is to apply the methodology to a specific innovative architecture and show the potential of executing it. The design space in

this case is small, with just some conventional and some innovative options. The next application case in chapter 5 shows how to deal with big design spaces. In this application case the system of interest are the high-lift devices for commercial aircraft.

State of the art high-lift devices consist of flaps and slats. Both have analogous architectures, typically built with the same components. This conventional architecture is described in figure 4.1, showing an example with the flaps. Power is provided by a central power drive unit (PDU) that provides torque to a main shaft through a series of gearboxes [156]. This shaft is mechanically connected through more gearboxes to linear ballscrew actuators [164], which are the components in charge of moving the control surfaces [165, 166]. The connection from the PDU to the actuators is purely mechanical, through several gearboxes and shafts. This connection ensures a synchronous and symmetrical deployment of the flap and slat surfaces, since all the actuators are connected to the same central shaft. The PDU can be hydraulic or electrical depending on the source of power [167] and can be built as single or dual, depending on the number of motors and redundancies involved [164]. As seen in the figure, each flap surface is connected to two redundant actuators. Each actuator is connected to the main shaft through one gearbox. Two extra gearboxes, called corner gearboxes, are usually needed per wing in order to redirect the shaft from the actuators to the PDU through the wing and fuselage. One torque limiter is also present per wing, its main purpose is to ensure symmetry in case of failure but this is explained later with more details. State of the art slats have the same structure. Two actuators are used per slat surface, all of them connected to one gearbox that is powered by the main shaft. The main difference with the flaps is the number of surfaces. Flaps are typically composed of an inboard and an outboard flap per wing (i.e., a total of four flap surfaces), while slats are divided into more surfaces (e.g., five smaller surfaces that make a total of ten slat surfaces). The number of corner gearboxes can also differ depending on the specific aircraft geometry. All the rest of components remain as for the flap case, having one main PDU, two ballscrew actuators per surface and two torque limiters.

The certification regulations make it obligatory to neutralize asymmetries in flight [164] since these can cause a catastrophic failure. As a result maintaining a synchronous and symmetric deployment of the flap and slat surfaces is key to ensure the safety of such systems. This is guaranteed for the conventional architecture previously mentioned. All actuators are mechanically connected and deploy

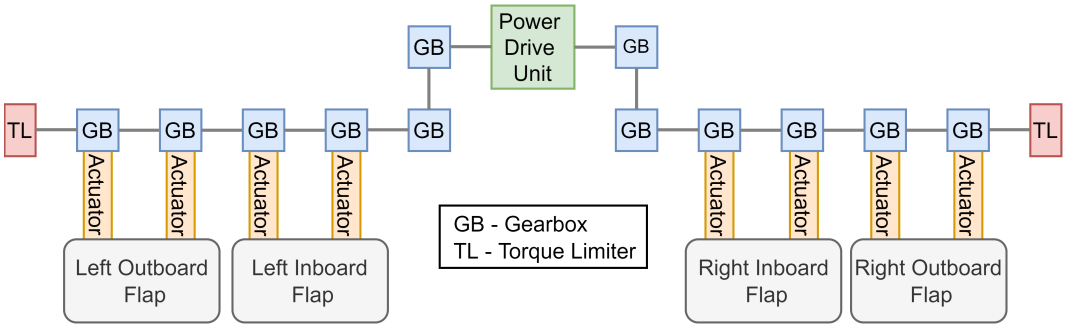


Fig. 4.1 Scheme of a conventional flap architecture for a commercial aircraft, from [158]

synchronously as a result. However if the main shaft breaks, symmetry can get compromised. An example is given to the reader. If for instance the shaft breaks between the left inboard and outboard flaps, this causes the left outboard flap to be mechanically disconnected from the chain. Three flap surfaces (i.e., the two right flap plus the left inboard flap) can be actuated by the PDU, but the left outboard flap can now move freely and uncontrolled. This effect creates the unwanted asymmetries and torque limiters are used to mitigate them. Such torque limiters are used as wing tip brakes, they activate when the mechanical transmission breaks [164], fixing the now disconnected shaft into place and not letting it spin freely and in an uncontrolled way. These limiters are built with friction disks that are activated by the monitoring system in case of the previously commented failure. These disks provide enough torque to brake the disconnected shaft, solving the asymmetry problem. This state of the art solution guarantees the safety of the system but at the same time involves a high number of components, mechanical connections and design effort [165].

The presented baseline architecture is currently being used in most commercial aircraft. Research regarding innovative flap geometries and architectures is currently being done since they could manage to reduce the aircraft MTOM, and as a consequence fuel burn [168]. New actuators are enabling innovative design concepts with promising potential. Electromechanical actuators (EMAs) operate with their own servo motor. This independence removes the need for central devices to supply torque and eliminates the need for a central hydraulic line, as opposed to traditional hydraulic actuators [46]. Used for high-lift devices the EMAs eliminate the need for a central power unit, main shaft and distribution gearboxes. Linear EMAs can directly replace linear ballscrew actuators, at the cost of introducing a new challenge. Torque limiters (i.e., wing tip brakes), used to prevent asymmetry, cannot be applied for the EMAs because there is no central mechanical shaft in this system. To address this,

electronic devices need to be incorporated to deactivate the corresponding motors if a malfunction is detected, as proposed in [165]. A fully independent flap actuation system has been developed in literature [165], with an EMA at each drive station, eliminating the need for transmission shaft linkages between stations. Synchronous deployment is ensured by actuator control electronics (ACE) and slat/flap control computers, which coordinate the movement of both symmetric actuators on each flap panel to prevent skewing or twisting. This architecture was further explored in [169], where a more detailed design was presented. The EMA is powered by an internal permanent magnet brushless DC motor, which directly converts electrical power into motion, using a ball-nut assembly. The authors highlight that this system meets fault-tolerant requirements and is suitable for high-reliability flap applications. A brief mention of this architecture can also be found in [167], where alternative strategies for central power drive units in conventional designs are discussed. This innovative flap architecture is shown in figure 4.2, where all the involved components are displayed. The central components, such as the main shaft and PDU, are now eliminated, resulting in each flap surface operating independently. While each surface still has two actuators, these are now EMAs. The ballscrew actuator, motor, and power control electronics are integrated into a single unit. Each flap surface contains two of these units, with each pair requiring the electronic components needed for the precise and symmetrical deployment of such surfaces (i.e., ACE). These units are connected to the flap/slat control computers, which facilitate communication between all ACE units to ensure asymmetry is effectively neutralized. In this architecture, slats are designed similarly to flaps, with the primary difference being the number of surfaces. Flaps consist of four surfaces (two on each side), while slats typically have ten surfaces (five on each side). Each slat surface also uses two EMAs, with each connected to its corresponding PCE unit, and both linked to the corresponding ACE. A separate image for slats is omitted as the only difference with figure 4.2 is the number of surfaces (four for flaps versus ten for slats).

The main differences between the innovative architecture found in literature (figure 4.2) and the conventional one (figure 4.1) are quite straightforward to point out. There is a clear transition from mechanical components and linkages to electric ones, following the more-electric aircraft (MEA) trend commented in chapter 2. Typically the hydraulic and pneumatic components are replaced by electric ones, in this case since there are only mechanical components these are the ones that get substituted. The addition of electric components comes at a cost since new challenges

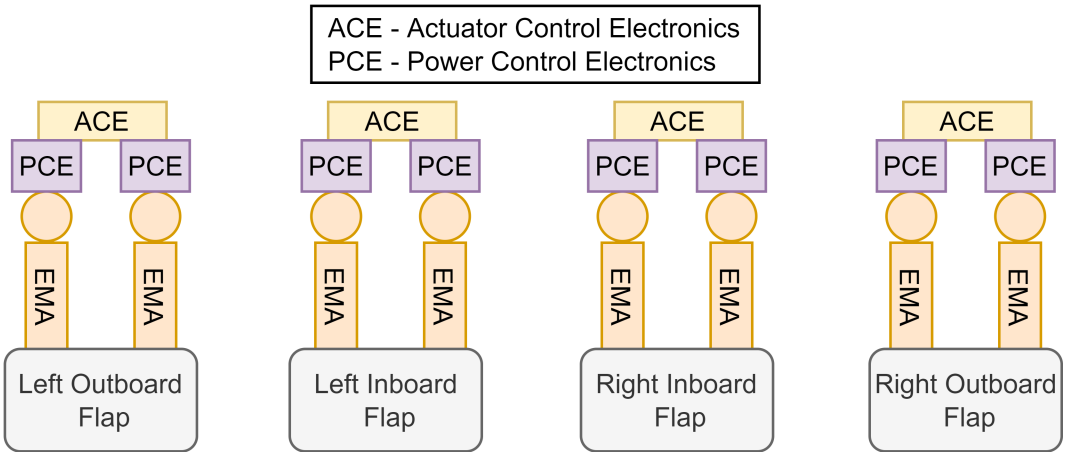


Fig. 4.2 Scheme of the proposed innovative flap architecture, from [158]

appear, such as increased complexity, new types of failures or performance in harsh environment [170]. The main advantage of this innovative architecture is that it can potentially reduce the high-lift devices mass. Unfortunately, on their current state, EMAs are not being utilized for this purpose owing to safety issues. This is caused by the jamming, one of their failure modes. When this failure occurs, it blocks the actuator into position. As a consequence, the control surface gets fixed in place and cannot be moved even if the redundant actuator is active and working well, not being able to successfully deploy the high-lift device surface. As a summary it can be said that this failure mode cancels the actuator in redundancy as well, massively reducing the system reliability. Currently this is the main disadvantage of the EMAs. According to [46], mechanical jamming remains a key challenge for EMAs and likely represents the largest barrier to their use in safety-critical applications. This has also been highlighted in other studies [47]. As a result, EMAs are not currently employed in primary flight surfaces but may be suitable for lower power applications [44]. Although studies show that jamming is not a frequent failure mode [46, 171], it still compromises control surfaces, requiring further investigation before EMAs are widely adopted in aircraft systems. Recent studies are following this line of research developing jamming-free actuators [172], which could be the solution and answer for the implementation of the proposed innovative high-lift architecture.

The key question is: can EMAs improve reliability somehow? The answer lies in the elimination of the central mechanical shaft. In conventional systems, if the main shaft, PDU, or a gearbox fails, the entire system must be stopped to prevent asymmetries, with the wing tip brakes being used for this purpose. In

contrast, the new EMA architecture removes these central components, mechanically disconnecting the surfaces among them. Using the flap system as an example, EMAs introduce a capability previously unavailable: the independent deployment of either the inboard or outboard flap surfaces, while still ensuring symmetrical operation. For instance, if one inboard flap fails due to a jammed actuator, the electronic system can stop the corresponding inboard flap on the opposite wing at the same position, while still allowing the outboard flaps to be fully deployed. This functionality is not possible in current architectures, where a single actuator or gearbox failure would immobilize the entire transmission shaft, causing a total flap system failure. As a result, the EMA concept has the potential to improve the system reliability for this specific application case. Previous studies have already suggested that the independence of all flap panels enables the aircraft to operate with just one pair of flaps, which would "increase operational reliability" [165]. However, a critical consideration is that the wing must be designed with this capability in mind, as retrofitting the system into existing designs would not be feasible. Therefore, this system must be integrated from the early stages of design to fully leverage its potential, which would influence the overall wing design. The importance of incorporating such disruptive technologies in preliminary design phases to achieve meaningful aviation advancements has been emphasized in [173]. Redesigning the wing may seem like a significant change, but these type of changes might become a necessity in future aircraft designs.

The difference in failure rates between conventional and innovative architectures is explained now, since it has a remarkable effect of the system reliability. To understand this difference, a comparison with shaft failure in conventional systems is useful. Experimental data from one study revealed that the primary failure modes in slat systems involve transmission shaft fractures and motion seizure [174]. From 1956 to 2004, over 40 accidents were linked to slat mechanism failures caused by these issues. This suggests that shaft fractures in conventional high-lift devices may be more common than jamming in EMAs, leading to positive expectations for the reliability of the innovative EMA architecture. For the innovative architecture, jamming is not the only failure mode introduced by EMAs, though it is the most critical. These failure modes differ significantly from those encountered in conventional actuators. The design and development of EMAs present new challenges, such as the high integration of motors and bearings, lubrication in harsh environments, optimized thermal management, standards for software and hardware, compact embedded

power electronics, and health monitoring of critical components [170]. Despite these challenges, other studies highlight the advantages of EMAs, such as generally lower failure rates and improved performance in adverse environmental conditions [166]. Regarding maintenance, not much information is available. However, the reduction in the number of component and the change from mechanical to electrical components has potential in terms of maintenance time reduction.

Other topic that needs to be addressed is the maturity level of the EMAs. They are not currently being used in safety-critical devices, however EMAs have already been implemented in aircraft for different purposes. For example, the B787 uses EMAs for horizontal stabilizer trim actuation and some spoilers [47]. They are also utilized by the X-33 or F16 [47]. The A380 and A350 also use these actuators during landing to activate the thrust reversal [44].

With some level of speculation, the adoption of EMAs in high-lift devices could soon become a reality, provided that advancements in anti-jamming technologies are successful. This includes effective jamming mitigation techniques [175] or anti-jamming EMAs capable of functioning despite mechanical interference [172]. Implementing this new technology could offer several benefits. By eliminating central mechanical components, inefficiencies are reduced [47], allowing for more optimal sizing of each actuator. The decrease in components could also lower the overall weight of the system, leaving further space for other systems [176]. Additionally, this could lead to fewer maintenance tasks [176] and potentially reduced manufacturing effort [165], thus shortening production time [47]. While the integration of additional electronic devices must be considered, the overall impact is expected to be a reduction. Some studies are even working towards developing more reliable EMAs [177].

Concluding, the objective of this application case is to compare the innovative and conventional architectures in terms of performance, certification and maintenance. This can lead to interesting results, specially since safety is a current limiting factor for EMAs. Even if these results have a low technology readiness level (TRL), they could finally provide quantitative results to measure the difference between both architectures. Furthermore they can also set a strong starting point for further higher-fidelity studies.

4.2 Methodology Adjustments

The first step is to define the aircraft to be analyzed. The main inputs and assumptions of this aircraft are summarized in table 4.1.

Table 4.1 Assumed input parameters for the reference aircraft; Airbus A320neo

Parameter	Units	Value	Parameter	Units	Value
Max. Take-Off Mass	kg	78981	Engine Model	-	PW1133
Fuselage Length	m	37.57	Static Thrust (ISA)	kN	147.3
Wing Area	m^2	124.78	Seats	-	180
Wing Span	m	35.8	Design Cruise Mach	-	0.78
Vertical Tail Area	m^2	22.49	Design Range	nm	2935
Max. Lift Coefficient, Landing	-	2.9	Max. Lift Coefficient, Take-Off	-	2.55

The failure rates of the present components can be seen in table 4.2, this information is needed to solve the RBDs.

Table 4.2 Failure rates values for the components of the high-lift device system, taken from the Quanterion Automated Databook (NPRD-2016)

Component name (generic; from database)	Failure rates (per hour)	Quality	Environment
Ballscrew Actuator; Ballscrew Assembly	1.274×10^{-5}	Military	Airborne Cargo
EMA Actuator; Actuator Electromechanical Linear	2.335×10^{-5}	Military	Airborne Inhabited Attack
Gearbox; Gearbox Assembly	2.35×10^{-7}	Military	Airborne Cargo
Corner Gearbox; Gearbox Assembly	2.35×10^{-7}	Military	Airborne Cargo
Shaft; Shaft Assembly, Flap Drive Torque	5.81×10^{-8}	Military	Airborne Cargo
Electric Motor; Motor AC	2.39×10^{-6}	Military	Airborne Cargo
Electronic Devices; Computer Flight Control	2.79×10^{-6}	Military	Airborne Cargo

After the aircraft is selected the methodology can be executed as suggested in figure 3.2. However, for this application case not all the steps are needed. There are a few number of architectures to be studied, as a result the system architecting part is not required, as well as the optimization block. Only four different architectures are evaluated, and are explained in the following section. This reduces the diagram into 3 main blocks: certification filter, performance loop and maintenance evaluation. Each of the disciplines are executed sequentially for this particular application case. Each discipline needs small modifications that are now explained in the following subsections.

Certification discipline adjustments

Only one small adjustment is needed here. The methodology can be executed as explained in section 3.3. However there are two separated cases for this application case. When the innovative architecture is evaluated, two different RBDs are created, one corresponding to the case in which there is no jamming and one in which the actuator suffers jamming every time it fails. The RBDs are shown for extra clarity in the results. Figure 4.3 shows the resulting RBD for a conventional flap architecture. It can be seen how all the components and actuator couples are in series, all connected to the PDU. The conventional slat RBD is the same with the only difference being the number of actuators couples (from four to ten) and the number of corner gearboxes (from four to two). A separated figure is not displayed since it does not add any further relevant info.

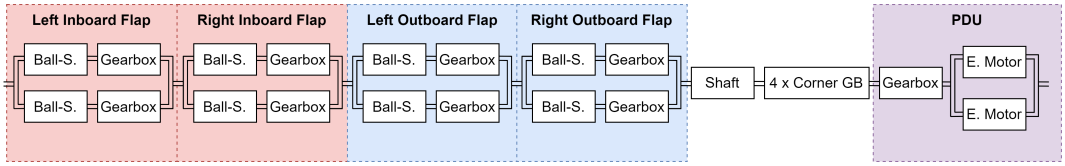
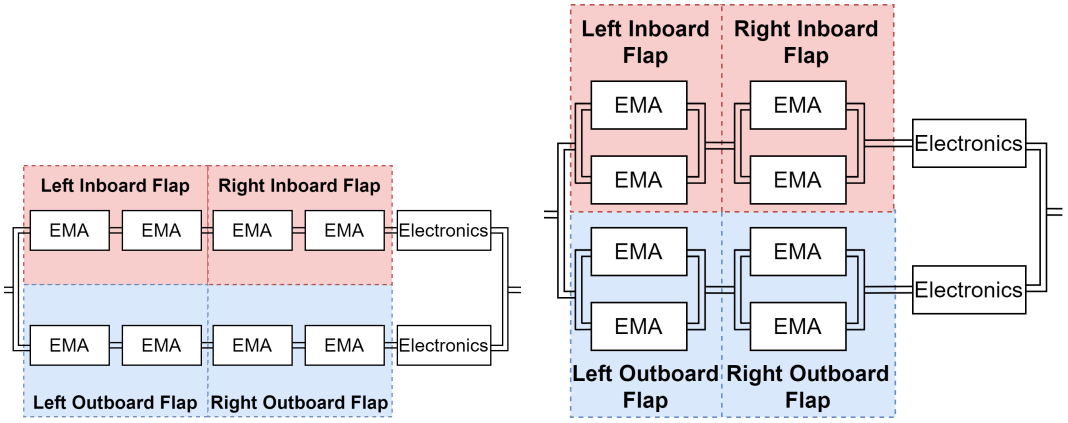


Fig. 4.3 RBD scheme for the conventional flap architecture, from [158]

The case for the innovative flap architecture changes from the case with and without jamming. The two different resulting RBDs are shown in figure 4.4. It is noticeable how the main shaft and PDU components totally disappear for this architecture. The main difference between the two RBDs resides in how the actuators are connected among them inside the same flap surface. In the case without jamming (figure 4.4b) the actuators are connected in parallel, since the failure of one does not affect the other. However, in the case with jamming (figure 4.4a), both actuators are in series because if just one of them fails the whole chain fails. As a result, two different results are obtained for the innovative architecture and are shown in the following section 4.3.

Two extreme cases have been considered. One in which there is no jamming, and one in which the actuator always jams. This is sufficient for a preliminary study but can be too conservative. There is an intermediate and more realistic solution that is here proposed but not used due to a lack of available data. This solution considers the use of jamming-tolerant devices, which are used in case of jamming to unblock the actuator (e.g., disengagement clutch). This approach separates the EMA in two



(a) Case with jamming. (b) Case without jamming.

Fig. 4.4 RBD scheme for the innovative flap architecture, from [158].

parts. One part considers the reliability of all functionalities except the jamming related ones, the other part considers the reliability related to jamming. Both are in series and together form the EMA. The jamming tolerant devices are added in parallel to the jamming related reliability to finalize the concept. This solution is represented in figure 4.5.

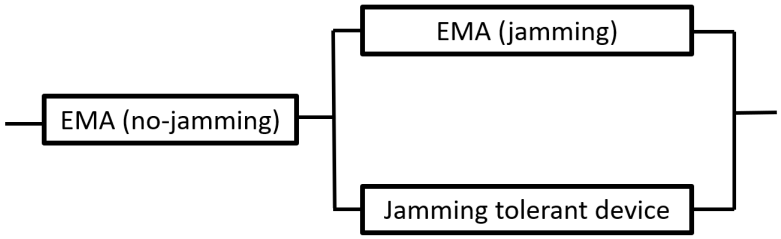


Fig. 4.5 RBD scheme for the case considering jamming-tolerant strategies. This case is added for discussion but has not been assessed in the analysis.

Lastly, conventional slats can be solved with the RBD in figure 4.3, as mentioned before. Regarding the innovative architecture for slats two cases are also presented here, in an analogous way to the flaps. Both cases are condensed in figure 4.6. As an example to the reader, the component "Left Slat 1" would be represented by two EMAs in series for the case with jamming, and by two EMAs in parallel for the case without jamming. All the different actuator couples are connected among them following a r-out-of-n RBD structure (4-out-of-5 in particular). This means that one slat surface can fail, but no more than one. This case is selected to stay conservative since the failure of two out of five slats might be risky to certify.

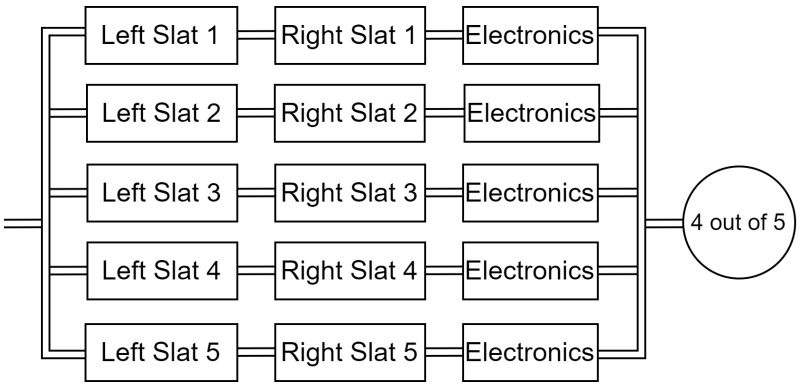


Fig. 4.6 RBD scheme for the innovative slat architecture, from [158]

Performance discipline adjustments

The performance sizing loop starts with the reference aircraft definition. The data in table 4.1 is used to generate the A320neo with OpenAD. This reference aircraft, together with the OBS architecture, are provided to ASTRID, which sizes the OBS and later provides the results back to OpenAD to re-size the aircraft with the higher-fidelity results. This starts an iterative loop that can be simplified for this application case. The first simplification is running ASTRID only to create the OBS baseline and then using the dedicated FCS tool for the sizing of that system, then merging both results. This provides a better result since ASTRID is not sensitive to the innovative flap architecture. Another effect is that only the FCS is changing from one OBS architecture to another, so it can be assumed that only one loop of the convergence loop is needed to achieve convergence. This assumption was later checked and validated. As a result, the performance loop stays as follows: one aircraft baseline is created with OpenAD, two OBS baselines are created with ASTRID (one for conventional and other for all-electric architectures), the dedicated FCS tool is run for each high-lift devices architecture that needs to be evaluated, results from ASTRID and FCS are merged and forwarded back to OpenAD together with some other previous results from the initial aircraft baseline. Lastly, OpenAD is run again to catch the effect that the OBS have on the overall aircraft, this run provides the final results for fuel burn and MTOM of each of the aircraft with each architecture. It can be seen how the loop is not needed since only one run converges and provides the required results, this helps a lot with computational times. The results of both OBS baselines are shown in table 4.3. The table includes a description of each of the

systems, FCS are calculated and shown later since it is the system of interest and needs a more detailed description.

Table 4.3 Results for both OBS baselines calculated with ASTRID

	Conventional OBS Baseline		**All-Electric OBS Baseline**	
	Description	**kg**	**Description**	**kg**
Avionics	Transoceanic-capable avionics	781	Same as previous	781
FCS	Calculated later with the dedicated FCS tool	-	Same as previous	-
ICS	Pneumatic concept for wing, tail and nacelles	73	Electro-thermal blankets for wing and tail. Small local bleed for nacelles	76
ECS	3-wheel high pressure bootstrap cycle	480	Bleedless with external compresors	589
Fuel System	State of the art: electrically actuated pumps	344	Same as previous.	344
Landing Gear	Hydraulically actuated	2176	Electrically actuated (EMAs)	2320
Fire protection, lights, oxygen and water waste	Conventional as for the A320.	850	Same as previous.	850
APU	Powering penumatic and electrical system	138	Powering electrical system	122
Pneumatic System	Engine bleeding	160	Not needed	0
Hydraulic System	3 lines with engine-driven pumps, electric pumps and connection to the RAT	1022	Not needed	0
Electrical System	Generation at 115 V AC with IDG Ni-Cd batteries	1443	Generation at 230V AC with VFG Ni-Cd batteries	1366
Total (without FCS)	-	**7466**	-	**6447**

Maintenance discipline adjustments

For the maintenance discipline of this application case the surrogate model explained in chapter 3.5 cannot be used. This surrogate model does not consider changes in the high-lift devices. As a result, the methodology prior to the surrogate model shall be used. The first step, as explained in the corresponding chapter, is to examine the MPD of the aircraft and extract the relevant information. For this case, only the high-lift devices are needed. The most relevant parameters for the needed tasks from the MPD are shown in the following figures for each of the systems of interest: flaps (figure 4.7), slats (figure 4.8), and the corresponding part of electrical wiring interconnection system (EWIS) (figure 4.9). As a reference for the reader, the Airbus A320 FCS has 84 tasks in the MPD, from which 22 are linked to the high-lift devices. All these tasks are assigned accordingly to the different A-checks and C-checks. A-checks have been condensed every 750 FH, 1500 FH, 2250 FH and 3000 FH. C-checks occur every 18 months from month 18 until month 288, making a total of 16 C-checks. These checks are as suggested in [131].

These tasks presented in the paragraph before can be used directly to evaluate the conventional high-lift devices. For the innovative case the tasks need to be modified. Some tasks are completely removed since the components associated to them do

TASK NUMBER	PREPARATION	ZONE	DESCRIPTION	TASK CODE	100% INTERVAL	MEN	TASK M.H.	ACCESS M.H.	PREP M.H.
275100-01-2		210	FLAPS OPERATIONAL CHECK OF WTB/POB NOTE: FLAP SURFACES MOVE AUTOMATICALLY	OPC	108 MO OR 12000 FH	2	0.30		
275100-02-1		210	FLAPS OPERATIONAL CHECK OF FLAP INTERCONNECTING STRUT AND FLAP DISCONNECT PROXIMITY SENSORS	OPC	108 MO OR 12000 FH	1	0.10		
275100-03-1	FLAPS EXTENDED;	575 675	FLAPS CHECK OF FLAPS TRANSMISSION SHAFTING INTEGRITY, INCLUDING INSPECTION OF SEAL WITNESS DRAINS.	CHK	108 MO OR 12000 FH	1 1	0.40 0.40	0.02 0.02	0.02
275117-01-1		212	FLAP/SLAT COMMAND SENSOR UNIT REMOVE CSU FOR WORKSHOP CHECK OF FRICTION BRAKE	FNC	40 MO	1	0.30		
275144-01-1	FLAPS EXTENDED; SLATS FULLY EXTENDED;	580 680	FLAP INTERCONNECTING STRUT DETAILED INSPECTION OF INTERCONNECTING STRUT AND ATTACHMENTS	DET	108 MO OR 12000 FH	1 1	0.30 0.30		0.04
275400-01-1	FLAPS EXTENDED; SLATS FULLY EXTENDED;	570 670	FLAP TRANSMISSION ASSY DETAILED INSPECTION OF TRANSMISSION ASSY	DET	80 MO OR 24000 FH	1 1	1.00 1.00	0.02	0.04
275446-01-1	FLAPS EXTENDED; SLATS FULLY EXTENDED;	530 630	FLAP TRACKS LUBRICATE ALL CARRIAGE ROLLERS ON ALL FLAP TRACKS	LUB	10 MO OR 1800 FC	1 1	0.40 0.40		0.04
275446-02-1	FLAPS EXTENDED; SLATS FULLY EXTENDED;	530 630	FLAP TRACKS DETAILED INSPECTION OF TRACKS, ROLLERS AND SPHERICAL BEARINGS (AS FAR AS VISIBLE).	DET	80 MO OR 24000 FH	1 1	0.30 0.30		0.04
275449-06-1		575 675	ACTUATOR ASSEMBLY DRAIN AND REFILL FLAP ACTUATOR ASSEMBLIES WITH SEMI-FLUID	SVC	72 MO OR 24000 FH	1 1	4.00 4.00		
275451-01-1		147	FLAP POWER CONTROL UNIT REPLENISH PCU GEARBOX TO CONFIRM FLUID LEVEL	SVC	36 MO	1	0.10	0.02	

Fig. 4.7 MPD tasks associated to the flaps for the conventional architecture; tasks that disappear for the innovative architecture are marked with a red dot, the ones that change are marked with an orange triangle

TASK NUMBER	PREPARATION	ZONE	DESCRIPTION	TASK CODE	100% INTERVAL	MEN	TASK M.H.	ACCESS M.H.	PREP M.H.
278100-03-1		210	SLATS OPERATIONAL CHECK OF WTB/POB	OPC	48 MO OR 12000 FH	1	0.20		
278100-04-1		520 620	SLATS CHECK OF SLATS TRANSMISSION SHAFTING INTEGRITY, INCLUDING INSPECTION OF SEAL WITNESS DRAINS	CHK	48 MO OR 12000 FH	1 1	0.10 0.10	1.17 1.19	
278400-01-1		147 190 520 620	SLAT TRANSMISSION ASSY DETAILED INSPECTION OF TRANSMISSION ASSY	DET	80 MO OR 24000 FH	1 1 1 1	0.50 0.50 0.50 0.50	0.02 0.37 0.90 0.90	
278451-01-1		147	SLAT POWER CONTROL UNIT REPLENISH PCU GEARBOX TO CONFIRM FLUID LEVEL	SVC	36 MO	1	0.10	0.02	
278468-01-1		520 620	SLAT TRACKS LUBRICATE ALL TRACK ROLLERS AND PINIONS	LUB	20 MO OR 4500 FC	1 1	0.50 0.50	1.00 1.00	
278468-02-1	FLAPS EXTENDED; SLATS FULLY EXTENDED;	520 620	SLAT TRACKS DETAILED INSPECTION OF SLAT TRACKS, ROLLER BEARINGS, PINION BEARINGS AND CURVED RACK GEARS - AS VISIBLE.	DET	80 MO OR 24000 FH	1 1	1.00 1.00	1.00 1.00	0.04

Fig. 4.8 MPD tasks associated to the slats for the conventional architecture; tasks that disappear for the innovative architecture are marked with a red dot

TASK NUMBER	PREPARATION	ZONE	DESCRIPTION	TASK CODE	100% INTERVAL	MEN	TASK M.H.	ACCESS M.H.	PREP M.H.
ZL-531-01-1	FLAPS EXTENDED;	531 631	FLAP TRACK NO 2 AND FAIRING GENERAL VISUAL INSPECTION OF FLAP TRACK NO 2 AND FAIRING (EWIS)	GVI	48 MO	1 1	0.10 0.10	0.03 0.03	0.02
ZL-531-02-1	FLAPS EXTENDED;	531 631	FLAP TRACK NO 2 AND FAIRING GENERAL VISUAL INSPECTION OF FLAP TRACK NO 2 AND FAIRING (EWIS)	GVI	72 MO	1 1	0.30 0.30	0.16 0.16	0.02
ZL-532-01-1	FLAPS EXTENDED;	532 632	FLAP TRACK NO 3 AND FAIRING GENERAL VISUAL INSPECTION OF FLAP TRACK NO 3 AND FAIRING (EWIS)	GVI	48 MO	1 1	0.10 0.10	0.03 0.03	0.02
ZL-532-02-1	FLAPS EXTENDED;	532 632	FLAP TRACK NO 3 AND FAIRING GENERAL VISUAL INSPECTION OF FLAP TRACK NO 3 AND FAIRING (EWIS)	GVI	72 MO	1 1	0.30 0.30	0.16 0.16	0.02
ZL-533-01-1	FLAPS EXTENDED;	533 633	FLAP TRACK NO 4 AND FAIRING GENERAL VISUAL INSPECTION OF FLAP TRACK NO 4 AND FAIRING (EWIS)	GVI	48 MO	1 1	0.10 0.10	0.03 0.03	0.02
ZL-533-02-1	FLAPS EXTENDED;	533 633	FLAP TRACK NO 4 AND FAIRING GENERAL VISUAL INSPECTION OF FLAP TRACK NO 4 AND FAIRING (EWIS)	GVI	72 MO	1 1	0.30 0.30	0.15 0.15	0.02

Fig. 4.9 MPD tasks associated to the EWIS for high-lift devices for the conventional architecture; tasks that change for the innovative architecture are marked with an orange triangle

not exists in this architecture. These tasks are marked in the figures with a red dot, and are: 275100-01-2 and 275117-01-1 (flap wing tip brakes), 275100-03-1 and 275400-01-1 (flap transmission shaft), 275451-01-1 (flaps PDU), 278100-03-1 (slats wing tip brakes), 278100-04-1 and 278400-01-1 (slat transmission shaft) and 278451-01-1 (slat PDU). Resulting in 9 tasks, out of 22, that disappear completely. For the new components, some tasks are modified as suggested in the methodology. They are marked in the figures with an orange triangle. Task 275449-06-1 is modified since it applies to the actuators, which change from ball-screw to EMA. All the tasks associated to the EWIS are also modified to catch the effect of adding more electronic devices into the architecture (e.g., ACE). This modification is performed following the instructions from [131], in particular using the failure rates proportion. The failure rate change of the main component is from ball-screw to EMA, which failure rates are shown in table 4.2. Dividing these values a proportional failure rate change of 1.833 is obtained, this factor is used to scale the MMH of the tasks that change from one architecture to another. Summarizing, the maintenance results for the innovative architecture can be estimated summing three contributions: leaving the tasks that remain the same, subtracting the tasks that disappear and summing the new contributions of the tasks that change (i.e., old value multiplied by the 1.833 failure rate factor). With these details the complete maintenance analysis can be performed and result are shown in the following section.

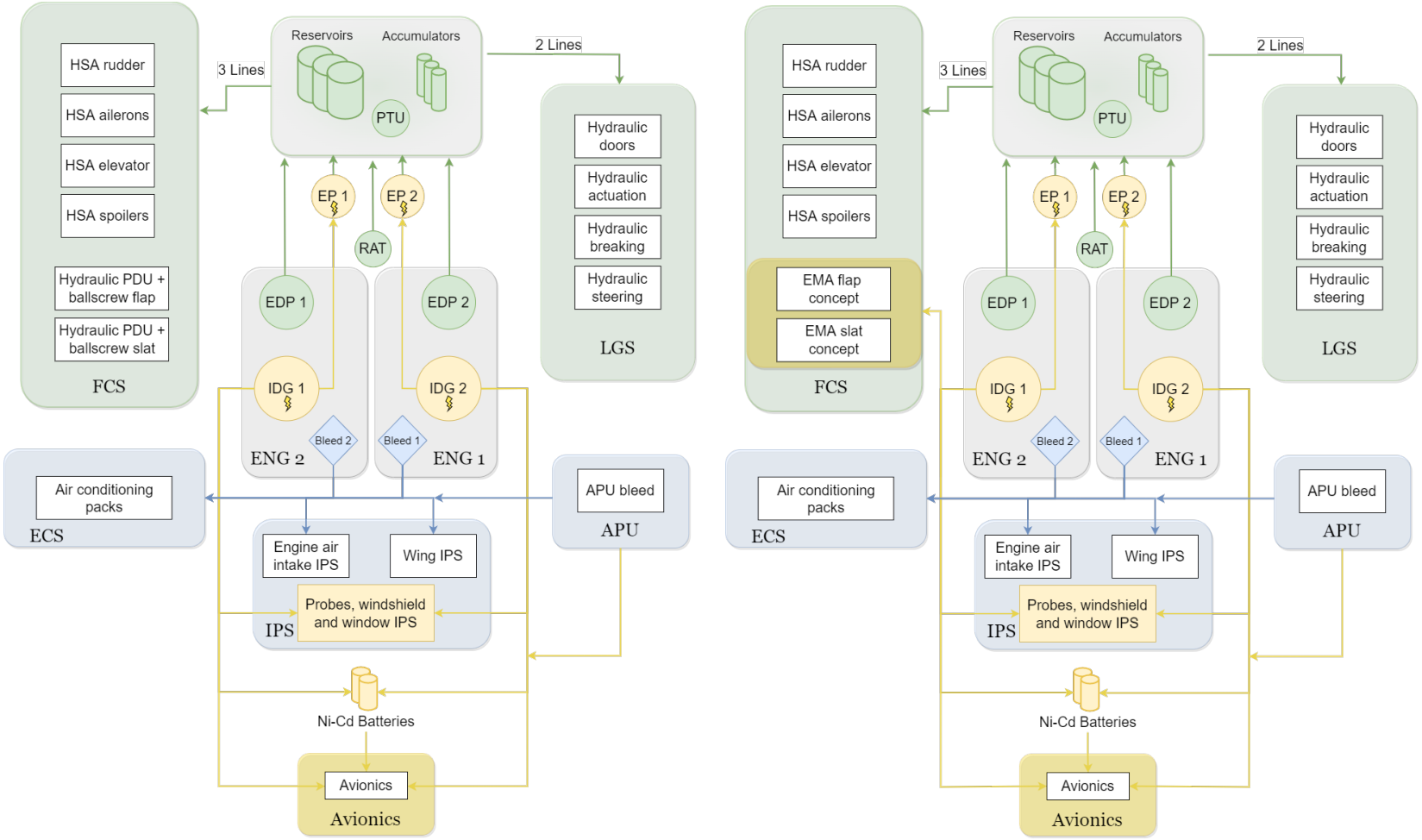
4.3 Results

A total of four high-lift devices architectures are evaluated and installed in different aircraft, they are as follow:

- **Aircraft 1:** Conventional OBS with conventional high-lift devices.
- **Aircraft 2:** Conventional OBS with innovative high-lift devices.
- **Aircraft 3:** All-electric OBS with conventional high-lift devices.
- **Aircraft 4:** All-electric OBS with innovative high-lift devices.

A detailed description of each aircraft OBS and high-lift devices architecture is provided in figures 4.10 and 4.11. Here the reader can find information about each of the power consuming systems for each of the aircraft, with their respective connections to the distribution systems and the power generation.

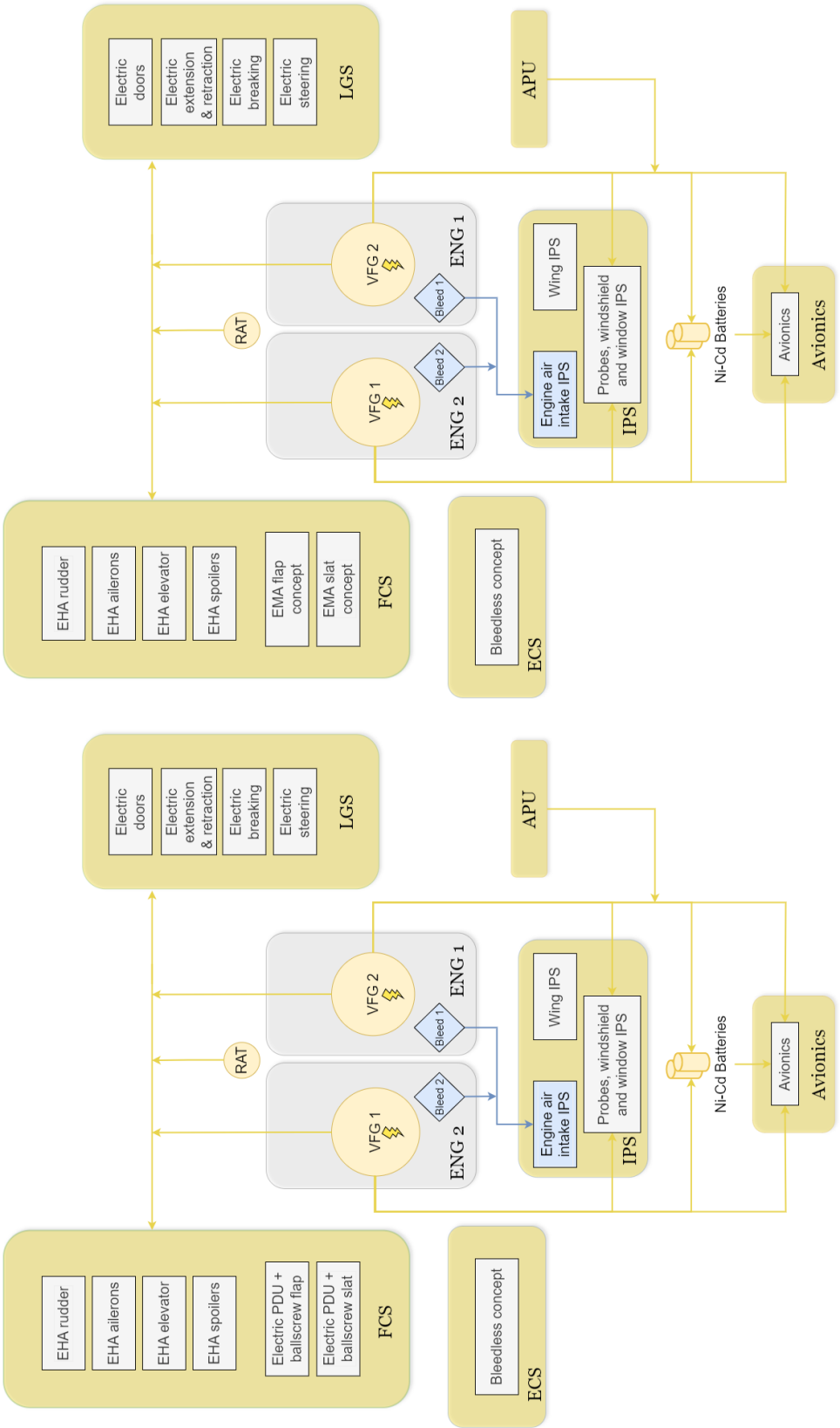
The four different aircraft are evaluated according to the three disciplines. Performance results are here commented. First, the dedicated FCS tool is run and the results are shown in table 4.4. Component level details are achieved, being able to fully size the high-lift devices and understand which components give higher and lower contributions to the total mass. Primary surfaces are also shown, seeing the difference in mass when changing from HSAs to EHAs. Looking at the total mass of primary surfaces the reader can notice how the all-electric concepts are heavier since EHAs are generally heavier than conventional hydraulic actuators, since they require their own local hydraulic circuit and pump. For flaps and slats the effect is the opposite. The innovative concept manages to reduce the weight of the system because it does not need some components such as the PDU or gearboxes. The removal of the central shaft also removes the corresponding mechanical inefficiencies. As a result there is a trade-off between primary surfaces and high-lift devices resulting in aircraft 2 having the lightest FCS architecture, and aircraft 3 the heaviest. However, adding the rest of the OBS changes the final results. Even if aircraft 2 has the lightest FCS, it still has a central hydraulic system that increases the overall mass. Summing the effect of the OBS that were evaluated in table 4.3 switches the trade-off results. The heaviest OBS architecture becomes the one of aircraft 1, and the lightest now is aircraft 4. These results emphasize the importance of assessing the OBS and the aircraft as a whole, not stopping only at subsystem level since this



(a) Aircraft 1: OBS architecture

(b) Aircraft 2: OBS architecture

Fig. 4.10 OBS architecture of the two aircraft with conventional OBS



(a) Aircraft 3: OBS architecture

(b) Aircraft 4: OBS architecture

Fig. 4.11 OBS architecture of the two aircraft with all-electric OBS

can lead to unrealistic expectations. The influence of the needed distribution systems for a given architecture can change the trade-offs of one specific architecture.

Table 4.4 Mass results of the flight control system and on-board systems of the four studied aircraft

	Aircraft 1		**Aircraft 2**		**Aircraft 3**		**Aircraft 4**	
Component	**Number of instances**	**Mass of one instance [kg]**	**Number of instances**	**Mass of one instance [kg]**	**Number of instances**	**Mass of One instance [kg]**	**Number of instances**	**Mass of one instance [kg]**
Aileron actuator	4	18.75	4	18.75	4	29.40	4	29.40
Elevator actuator	4	17.91	4	17.91	4	28.05	4	28.05
Rudder actuator	3	11.26	3	11.26	3	18.01	3	18.01
Spoiler actuator	10	9.66	10	9.66	10	15.45	10	15.45
Total, Primary Surfaces	-	*276.96*	-	*276.96*	-	*438.33*	-	*438.33*
Flap actuator	8	13.28	8	23.24	8	13.28	8	23.24
Flap gearbox	8	12.24	0	0	8	12.24	0	0
Flap corner gearbox	4	13.40	0	0	4	13.40	0	0
Flap torque limiter	2	5.51	0	0	2	5.51	0	0
Flap PDU	1	55.14	0	0	1	55.14	0	0
Flap shafts	1	4.30	0	0	1	4.30	0	0
Flap electronics	0	0	1	18.59	0	0	1	18.59
Total, Flaps	-	*328.21*	-	*204.48*	-	*328.21*	-	*204.48*
Slat actuator	20	3.36	20	6.53	20	3.36	20	6.53
Slat gearbox	20	6.21	0	0	20	6.21	0	0
Slat corner gearbox	2	6.76	0	0	2	6.76	0	0
Slat torque limiter	2	2.80	0	0	2	2.80	0	0
Slat PDU	1	27.81	0	0	1	27.81	0	0
Slat shafts	1	2.73	0	0	1	2.73	0	0
Slat electronics	0	0	1	13.06	0	0	1	13.06
Total, Slats	-	*241.17*	-	*143.69*	-	*241.17*	-	*143.69*
Total, FCS	-	**846.3**	-	**625.1**	-	**1007.7**	-	**786.5**
Total, OBS	-	**8312**	-	**8091**	-	**7455**	-	**7234**

As explained in the methodology, the OBS mass is not the only parameter that affects the aircraft sizing. Engine off-takes and bleeding must also be accounted for in order to fully catch the impact of an architecture. High-lift devices do not require bleed air, but they affect the power off-takes since they require electrical power to properly function. Table 4.5 shows the maximum electrical power that each architecture requires for flaps and slats. These are calculated with the dedicated FCS tool. ASTRID provides the bleed and OBS off-takes. The bleeding can be left untouched, but the power needs to be slightly updated with the results from the dedicated FCS tool. As a result, the last three rows show the results estimated by ASTRID but updated with the FCS power calculated by the FCS tool.

The reader can notice how aircraft 1 does not require electrical power for the high-lift devices since it has a hydraulically powered PDU. Aircraft 2 and 4 (EMA concepts) have a lower power requirement than aircraft 3 (electric PDU) since the innovative architectures are more efficient after removing the mechanical efficiencies caused by the central mechanical components (e.g., shaft, PDU, corner gearboxes). This effect is not negligible. Regarding the engine bleed, the two all-electric aircraft

Table 4.5 Engine off-takes and bleeding results of the flight control system and on-board systems of the four studied aircraft

	Aircraft 1	**Aircraft 2**	**Aircraft 3**	**Aircraft 4**
Flaps: Max. electrical power [kW]	0	2.91	5.94	2.91
Slats: Max. electrical power [kW]	0	1.83	2.99	1.83
OBS: Cruise bleed [kg/s]	0.88	0.88	0	0
OBS: Climb bleed [kg/s]	1.43	1.43	0	0
OBS: Cruise power off-takes [kW]	218	218	387	387
OBS: Climb power off-takes [kW]	170	175	245	241
OBS: Take-off power off-takes [kW]	59	64	147	143

manage to completely remove the bleed required by the OBS, this has a positive effect in the specific fuel consumption. On the other hand, the all-electric aircraft need a higher power demand to successfully supply power to the users. This can be seen in the power off-takes results.

The MTOM and fuel burn of each aircraft are evaluated as a final result for the performance discipline. Both values are calculated with OpenAD after providing the previous results from tables 4.5 and 4.4 as feedback, and re-sizing the aircraft baseline. This catches the snowball effect and increases the fidelity of the results. Both parameters are shown in table 4.6. There is a correlation of 1-to-1, which means that both MTOM and fuel burn increase and decrease accordingly from one aircraft to another. The heaviest (and most fuel consuming) aircraft is aircraft 1. This aircraft has conventional OBS and conventional high-lift devices, as a result the penalties that come from having less efficient distribution systems (i.e., hydraulic, pneumatic, and central mechanical components for high-lift devices) causes the overall mass and fuel consumption to increase. The lightest and most efficient aircraft is number 4, which uses all the new technologies. However the technology readiness of such technologies is still low. The biggest jump in mass and fuel burn reduction is found between aircraft 2 and 3, showing that the OBS as a whole have a higher impact than the FCS alone, which is completely aligned with the expectations. An interesting result is that the fuel burn reduction is higher than the MTOM reduction, this could come from the summed effect from the MTOM, off-takes and bleeding. All these results are calibrated and validated according to the results in [158].

The certification discipline is assessed now. For this application case this discipline is summarized only by the probability of failure of the architecture, having one

Table 4.6 MTOM and fuel burn results of the four studied aircraft

	Aircraft 1		**Aircraft 2**		**Aircraft 3**		**Aircraft 4**	
	Mass [kg]	Variation [%]	Mass [kg]	Variation [%]	Mass [kg]	Variation [%]	Mass [kg]	Variation [%]
MTOM	78966	0	78625	-0.43	77422	-1.96	77080	-2.39
Fuel Burn	19079	0	18921	-0.83	18488	-3.1	18332	-3.92

single index or parameter to evaluate it. There are only two architectures regarding this discipline since it is not sensitive to the changes in OBS architecture, only to changes in high-lift devices. The two architectures are the conventional high-lift devices and the innovative EMA-based architecture. However, two cases can be contemplated as explained before: the case without jamming, and the case with every failure causing jamming. Table 4.7 shows the result for flaps and slats, for the two architecture concepts, and for the two cases. For reference, each result has a pin that associates each result to their correspondent RBD. Results with (*) are calculated with the RBD from figure 4.3. The ones with (†) with figure 4.4b. Those with (§) are evaluated following the RBD in figure 4.6. Lastly, the one with (¶) corresponds to figure 4.4a.

Table 4.7 Probability of failure results of the different high-lift devices concepts

	Probability of failure [per hour]	
Concept	**Flaps**	**Slats**
Conventional with central PDU	$^{*}\ 1.2 \times 10^{-06}$	$^{*}\ 7.6 \times 10^{-07}$
Innovative with EMAs (no jamming)	$^{\dagger}\ 7.8 \times 10^{-12}$	$^{\S}\ 7.8 \times 10^{-11}$
Innovative with EMAs (every actuator failure causes jamming)	$^{\P}\ 9.2 \times 10^{-09}$	$^{\S}\ 9.2 \times 10^{-08}$

The main result is that the new high-lift device concepts massively increase reliability thanks to the implementation of the EMAs. Having the flap/slat surfaces in parallel allows the mission to be achieved even if one surface fails, this highly increases the operational reliability. However, it is important to mention that for this condition to be real, the wing must be designed for it. It is noticeable how in this case, even the case with jamming has a higher reliability than the reference one. The main conclusion is that eliminating the mechanical link among surfaces has a great overall effect in terms of reliability. As a disclaimer, the probabilities of failure for the innovative concepts are calculated with the new RBDs, which leverage the operational reliability increase that comes from the idea that one flap, or slat, surface can safely fail. Without this assumption the results are different and the RBD

would have all components in series. The result for the innovative flap would have a probability of failure of 5.6×10^{-6}, while the conventional one was 1.2×10^{-6}. For this case both results are comparable and the innovative case would not have an increase in reliability. This comparison is not the result of this analysis and should only be considered to emphasize that the main advantage of the new innovative concept relies on the operational reliability increase that comes from letting one flap, or slat, surface fail.

Lastly, the results on the maintenance discipline are here commented. High-lift devices, as explained in the previous section, cannot be evaluated following the surrogate model presented in the methodology. Because of that, the methodology prior to the surrogate model is utilized. Figures 4.7 to 4.9 contained the MPD tasks needed for the analysis. Results are shown in table 4.8 starting from the conventional architecture and showing the different contributions to reach the innovative one.

Table 4.8 MPD-MMH results for the high-lift devices concepts

Contribution	**MPD-MMH [hours]**	**Description**
Conventional	211.4	Total 22 tasks.
Tasks removed, decrease	-63.1	9 tasks removed from the conventional architecture.
Tasks changed, increase	+ 41.6	7 tasks changed from the conventional architecture.
Innovative	189.9	Total 13 tasks: 6 original, 7 changed.

It is important to highlight that these values of MMH are the ones directly extracted from the MPD. To transform them into real hours it is recommended to multiply them by a factor of three to account for inefficiencies, preparation times, human factors, etc... [131]. Finally, the four aircraft are evaluated in order to put the results into perspective. Table 4.9 shows the results in terms of MMH for each aircraft for their high-lift devices and total OBS. The sum of the OBS considers FCS, ECS, IPS, landing gear, hydraulic system, pneumatic system, electrical system and RAT. The table shows the MMH results in absolute value and the percentage reduction with regards to the conventional aircraft (i.e., aircraft 1).

Table 4.9 MPD-MMH results of the four studied aircraft

	Aircraft 1		**Aircraft 2**		**Aircraft 3**		**Aircraft 4**	
	MMH [hours]	Variation [%]	MMH [hours]	Variation [%]	MMH [hours]	Variation [%]	MMH [hours]	Variation [%]
High-lift	211.4	0	189.9	-10.2	211.4	0	189.9	-10.2
OBS	3383.4	0	3361.9	-0.64	2939.7	-13.1	2918.2	-13.7

Aircraft 1 and 2 change only the high-lift devices, the impact on this level is around a 10 percent of reduction in MMH. However, when installed in a real aircraft, this percentage gets cut to less than a 1 percent on the whole OBS. The innovative architecture fits better in a AEA aicraft, such as aircraft 3 or 4. These both aircraft achieve a MMH reduction of the OBS of more than a 13 percent. As a result, it can be concluded that also in terms of maintenance, the innovative high-lift devices architecture has potential and could be beneficial. It could reduce the number of maintenance tasks and maintenance man hours of the aircraft. In order to provide more context to the reader, these results are compared with the MMH of the whole aircraft. Aircraft Commerce provides a value for the total MMH for the A320 of 166750 hours [114 , 111]. Dividing this number by three to translate it into MPD-MMH, the result is 55583.3 MMH for the whole aircraft. With the results from table 4.9 a 13 percent reduction in the OBS MMH translates in almost a 1 percent for the aircraft as a whole.

Summarizing, the three disciplines (i.e., performance, certification and maintenance) show positive results about the new presented architecture for high-lift devices. The TRL is still low for this technology and its implementation could depend directly on the issues associated to the jamming failure mode. However this architecture could slightly improve the fuel burn and maintenance cost of the aircraft. In terms of safety it could also increase the operational reliability if the wing is designed to let one flat surface fail (or one slat surface fail). With this new operation concept the operational reliability of the system is increased even if the jamming issues of the EMAs still exist.

Chapter 5

Application Case 2: Flight Control System, Roll Control

This chapters shows the second application case. It is focused on the roll control function of the flight control system. This application case presents an example on how to execute the whole methodology when dealing with huge design spaces. This includes all the steps from the architectures generation from the design space, to the optimization of the objectives. The automated connection among all tools is now required and the use of surrogate models is also needed to alleviate computational times. Some parts and results of this application case were already disseminated in [178]. Details are shown in the following sections.

Section 5.1 contains the motivation and context of the application case, presenting and showing the system of interest. Section 5.2 deals with the design space modelling of the system of interest and the adjustments that need and can be done to the methodology to reduce its complexity and alleviate computational times. Section 5.3 shows the results, including an initial design of experiments and the posterior optimization runs. Lastly, 5.4 deals with the trade-off analysis, explaining how to achieve conclusions with results from such different disciplines.

5.1 Motivation & Context

The main motivation of this application case is to provide an example with a big design space. The flight control system fulfills this by nature. The high number of control surfaces, typology of actuators and connection to distribution systems, among other things, create a large number of possible architectures that can fulfill the system functions. This specific on-board system can be divided in three main functions: roll, pitch and yaw (assuming that high-lift devices are considered as a different system). Each of these functions has a certain probability of failure (or reliability). This differs with the previous application case, in which only one function was present. There are two ways to deal with this situation. One is to create three different design spaces and then merge them after the decisions have been taken, this solution however can be messy and unclear to implement. A better way is to create a common design space from which three different supplementary design spaces (one per function) are derived (as explained in section 3.3.2). With this solution the architectural choices are taken from only one model, leading to a better and more consistent solution. In addition to that, the number of possible architectures grows immensely when considering more that one function. As a reference, only the roll control has more than 6 billion possible architectures, as it will be shown later. Merging this with yaw and pitch creates enormous design spaces. For these reasons, the scope of this analysis was limited to just roll control. This function alone manages to successfully show the execution of the methodology and provides enough results and insight.

Therefore, the system of interest is the flight control system, in particular the roll control function. This includes ailerons and spoilers. The architecture can be built starting from the number of spoilers and ailerons, and the number of actuators per each control surface. The type of actuator used per surface is also an architectural choice and it influences the corresponding distribution systems. As explained in section 2.2, depending on the actuator choice the connection needs to be done to the hydraulic or electrical system. The posterior definition of such distribution systems, and their respective connection to the power generation systems, finishes the architectural choices.

A reference architecture is now provided. The reference aircraft is the Airbus A320neo. It must be noticed that the FCS architecture is the same as the one from the original A320, the main differences between both aircraft are the engines and winglets. The architecture of both is as a result the same, but the TLARs are taken

for the A320neo since it describes a more current and representative version of the aircraft. The whole architecture of the on-board systems can be found in the flight crew operating manual [179], and it is described in the following paragraphs.

Roll control of the A320 is achieved thanks to the ailerons and spoilers. Figure 5.1 shows a schema of the architecture. There is one aileron per wing, each of them connected to two actuators in redundancy. Both ailerons have one actuator connected to the green hydraulic line and other to the blue hydraulic line. Spoilers are simpler since they are only connected to one actuator each. There are five spoilers per wing, but only four of them are used as flight spoilers. Spoiler 1 is, as a result, a ground spoiler and is used only as a speed brake during landing. The other four spoilers (i.e., from spoiler 2 to spoiler 5) can be used as flight spoilers and/or ground spoilers. Focusing on the roll control, four spoilers act in redundancy. Spoilers 2 and 4 are connected to the yellow hydraulic line, spoiler 3 is connected to the blue line and spoiler 5 to the green hydraulic line. It can be noted how ailerons and spoilers are completely symmetric, even in the connection to the distribution lines. This helps to avoid the aerodynamic problems that can arise if the control surfaces are deployed non-symmetrically. If for instance the left spoiler 4 breaks, the right spoiler 4 is not used anymore and another spoiler pair would be used instead.

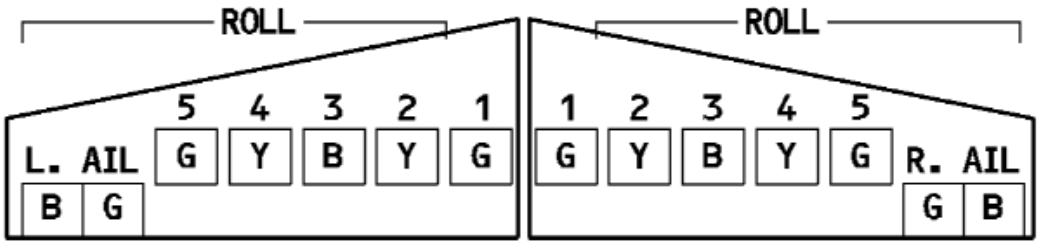
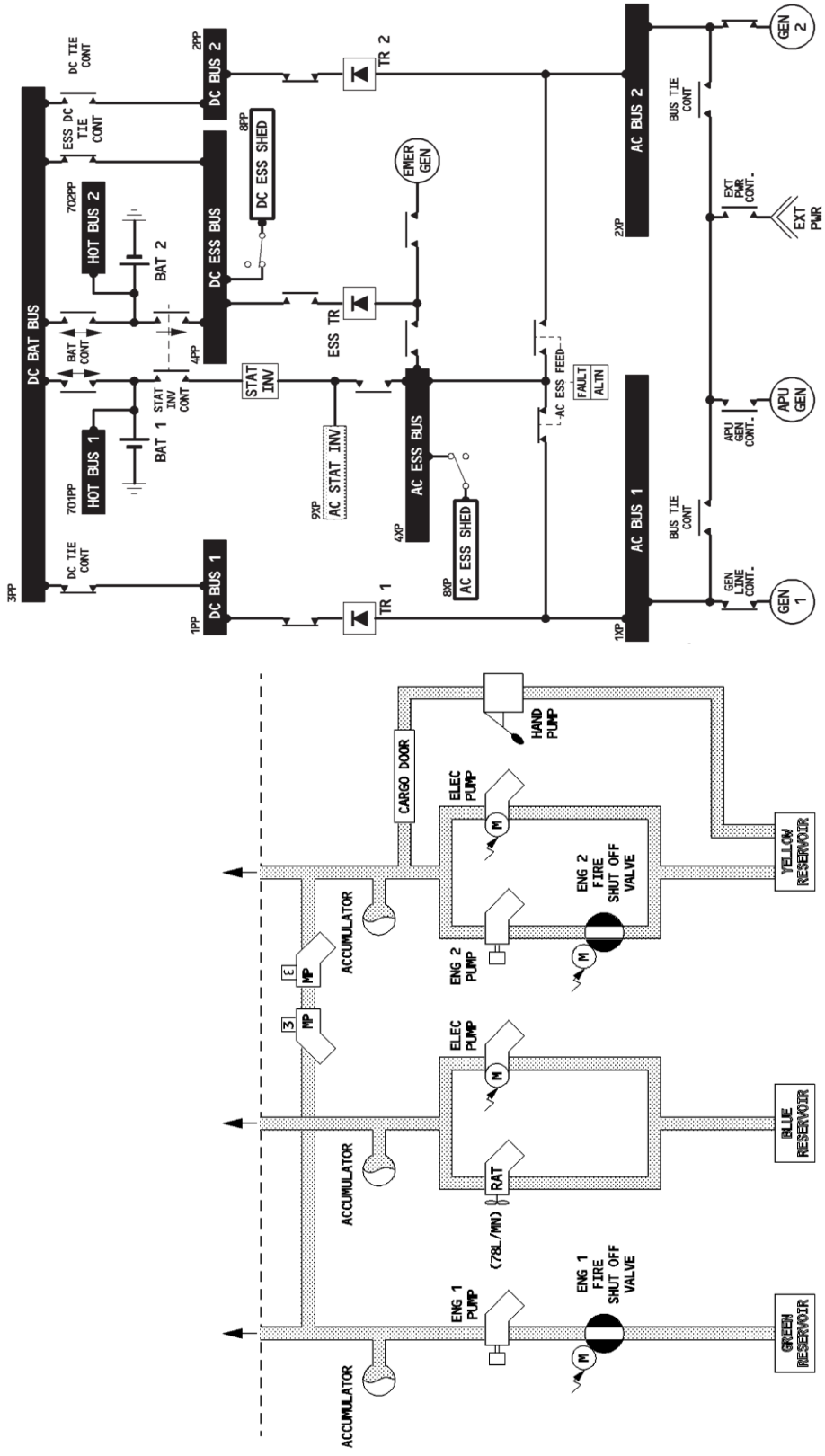


Fig. 5.1 Reference architecture for flight control system roll control; Airbus A320 ailerons and spoilers, from [179]

Several surfaces work in parallel. Redundancies come from several ways, such as different surfaces, different actuators per surface or different connections to distribution lines. All actuators from the Airbus A320 are conventional hydraulic actuators (HSA, as explained in section 2.2). This requires them to be connected to a hydraulic line from the hydraulic system, as explained in the previous paragraph. If these actuators were substituted by more-electric actuators (e.g., EHAs or EMAs) they would require the connection to a electric line from the electrical system instead. For this, it is important to explain these two distribution systems from the reference

architecture. Figure 5.2a shows the hydraulic system of the Airbus A320. Three lines can be identified, which are called green, yellow and blue. The green and yellow lines are used as main lines, while the blue one is used as a emergency one. Each line owns a reservoir to store the hydraulic fluid, and a line accumulator. The green line is powered by a hydraulic pump connected to engine 1. The yellow line is powered by two pumps in parallel, to add redundancies. One is connected to engine 2 and the other to a electric motor. These two lines (i.e., green and yellow) are connected between them though a power transfer unit, this allows one line to provide hydraulic power to the users of the other line in case of line failure. Lastly, the blue line is powered by two pumps. One is connected to other electric motor and the other to the ram air turbine. It can be seen how different levels of redundancy are achieved. Some lines can be powered by different pumps, in case of pump failure. Some lines are connected through PTUs, in case of line failure. Some components are connected to different lines (e.g., ailerons). This schema represents quite well the hydraulic system with a proper level of detail.

Figure 5.2b shows the electrical system of the Airbus A320. As explained before, this system would provide power to the actuators if these are substituted by more-electric versions of them, such as EHAs. Two main lines are present in this system (Bus 1 and Bus 2). Each line is connected to one engine-driven generator and a generator driven by the APU as a redundancy. This means that the APU can supply power in the place of either engine. If one engine fails, the APU can take over the power supply duties of that engine. These lines can also be powered by an external unit on ground, so that neither the engines nor the APU need to be on, however this does not apply during flight for obvious reasons. Generation is done in AC, more specific three-phase 115/200V at 400Hz constant frequency. Hence, the main lines provide power to some users directly in AC, but some components are added in order to also provide DC power. These are the transformer rectifiers (TR in the schema), and there is one present per line. This way, AC Bus 1 uses the transformer rectifier 1 to convert AC into DC and provide power to DC bus 1. For reference, this users work with 28V DC. The batteries are connected to both busses so that they can be charged. Another bus is present in the diagram, the essential bus (ESS). This bus is used in case of emergency to power the essential AC and DC users (e.g., essential avionics). The AC ESS bus can be connected to each of the main AC lines, but it can also receive power from an emergency generator that is connected to the RAT. This buss has also its own transformer rectifier (TR ESS) to convert AC from the AC ESS



(a) Reference architecture for the hydraulic system; Airbus A320

(b) Reference architecture for the electrical system; Airbus A320

Fig. 5.2 Reference architecture for the distribution systems; Airbus A320, from [179]. Color names of the hydraulic system correspond to the ones from figure 5.1

.

to DC in the DC ESS. The DC ESS bus can also be powered by the batteries, but only for a limited period of time. Same for the AC ESS, it can be powered by the batteries thanks to a static inverter located among the DC and AC essential busses. It is noticeable how the redundancies here start from the power generation systems, were the APU already provides extra connection opportunities to raise the reliability of the system. Also two main busses are present plus a emergency one that can receive power from several sources.

The architectures of the relevant systems have already been presented in the previous paragraphs. These show the architecture of the flight control system, hydraulic system and electrical system of the Airbus A320. The Airbus A320neo, as said before, shares the same exact architecture and OBS. The analysis of the reference architecture provides a baseline for the design space modelling, as well as some insight on how future architectures could look like. The evaluation of the specific aircraft requires some TLARs and parameters to be provided also. The main parameters and assumptions are summarized in table 5.1. These apply to the performance, certification and maintenance disciplines.

Table 5.1 Assumed input parameters for the reference aircraft; Airbus A320neo

Parameter	**Units**	**Value**	**Parameter**	**Units**	**Value**
MTOM	kg	78981	Utilization	Flight hours per year	2800
Static thrust (ISA)	kN	147.3	Design range	nm	2935
Engine	-	PW1133	Seats	-	180
Wing area	m^2	124.78	Fleet size	-	890
Fuselage length	m	37.57	Flight hours per flight cycle	-	2.59

Now the baseline aircraft has been defined. The reference architecture is also identified. The methodology can be executed step-by-step to assess the impact on the three disciplines (i.e., performance, certification and maintenance) for different OBS architectures. The first step is to define the design space model from which all the possible architectures will be generated. Then the three disciplines can be assessed following the previously commented order.

5.2 Design Space Modelling & Methodology Adjustments

The overall methodology was presented in chapter 3. However, as seen in the previous application case, this methodology can be simplified and tailored depending on the specific problem that needs to be addressed. For this application case, all the steps and disciplines are executed, however a simplification for the performance loop is suggested in order to alleviate the computational times. This section provides details on each step of the methodology. The first step, that was not present in the previous application case, is the design space modelling. Before there were only two architectures, a dedicated design space was hence not needed. For this application case the number of possible architectures is still unknown since a huge number of components and connections are available. A method supporting the modelling of such design space is required.

The design space model is created with the software ADORE. It is created following the principles of system architecting as explained in section 3.2. First of all, the top function of the system is identified. This function is "to control roll" and has different quantities of interest associated to it. This can be seen in figure 5.3, where six quantities are linked to this function. Three of them are the optimization objectives (i.e., probability of failure, MTOM and MMH reduction), the other three are the constraints (i.e., minimum reliability, single failure and back-up systems). This allows to automatically link the quantities of interest to the design space model, connecting the evaluation framework to the design variables.

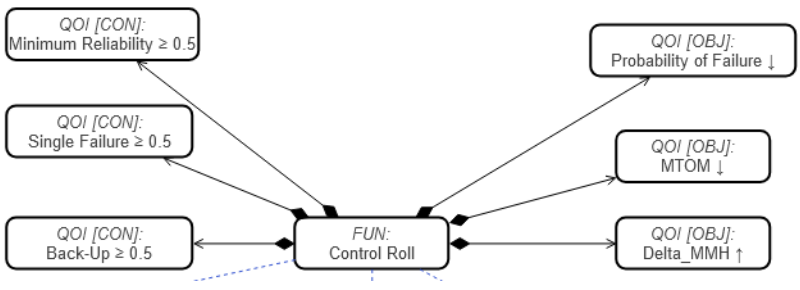


Fig. 5.3 Design space model; zoom in the top function and quantities of interest, screenshot from ADORE

The top function and its quantities of interest are now identified and the design space can be built from it. The first decision is on how many flight spoiler surfaces

are present in the architecture, this can vary from three to five. After this the number of surfaces is known since the ailerons are fixed to two. The first connection to an OBS is hence to the FCS. Figure 5.4 shows a zoom of this part of the design space. The whole design space is too big to be properly displayed in one single image, so it is shown in pieces. However it is important to notice that all images are cuts of the same unified design space, not different design spaces themselves.

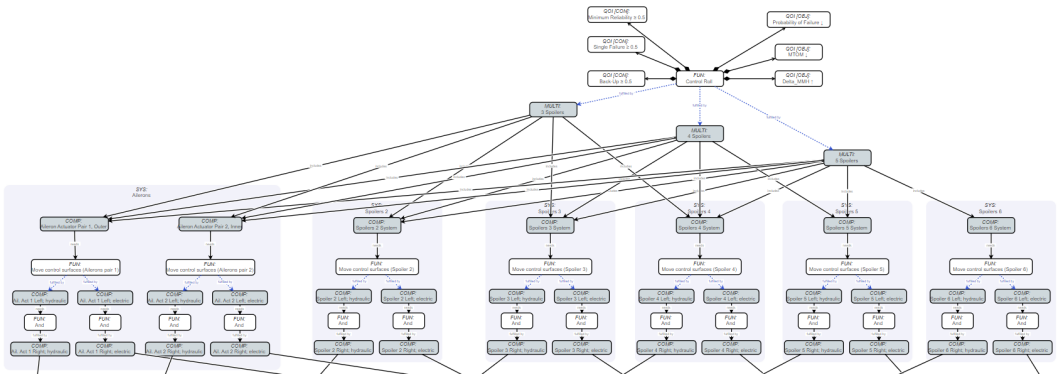


Fig. 5.4 Design space model; zoom in the FCS, screenshot from ADORE

It can be noticed how each spoiler is represented as a system. Inside this component only one decision is needed, the actuator type. The right and left spoilers for each pair are connected through a function to ensure symmetry, this means, both left and right have the same actuator type and are connected to the same distribution line. This restriction comes from the design of the system itself. Ailerons work in a similar way, however they possess two actuators in redundancy per surface, called "pairs" in the figures. Despite this, they are modelled as the spoilers. Symmetry issues also apply and the actuator types are the same ones. Regarding the actuators, hydraulic actuators correspond to the HSA model commented in section 2.2, while the electric version correlates to the EHA model. It can be noticed in the posterior figures how when the HSA is selected, this component is linked to a corresponding "provide hydraulic power to spoiler/aileron" function. When a EHA is chosen, the induced function is then "provide electric power to spoiler/aileron". These induced functions are fulfilled by components from different subsystems (i.e., hydraulic and electrical systems), highly influencing the rest of the design space. The hydraulic system is shown in figure 5.5. Three main lines are modelled, as in the reference architecture. Two main lines (green and yellow) are connected with a power transfer unit, and a emergency line (blue) is also present. The green line can be powered by an engine driven pump, a pump driven by an electric motor, or both in redundancy. The same

logic applies to the yellow line. The blue emergency line can be powered by a pump connected to the RAT, a different pump driven by an electric motor or both. It is interesting to see the modelling of the PTU since this component represents one of the unconventional configurations once transformed into a reliability block diagram (H-configuration).

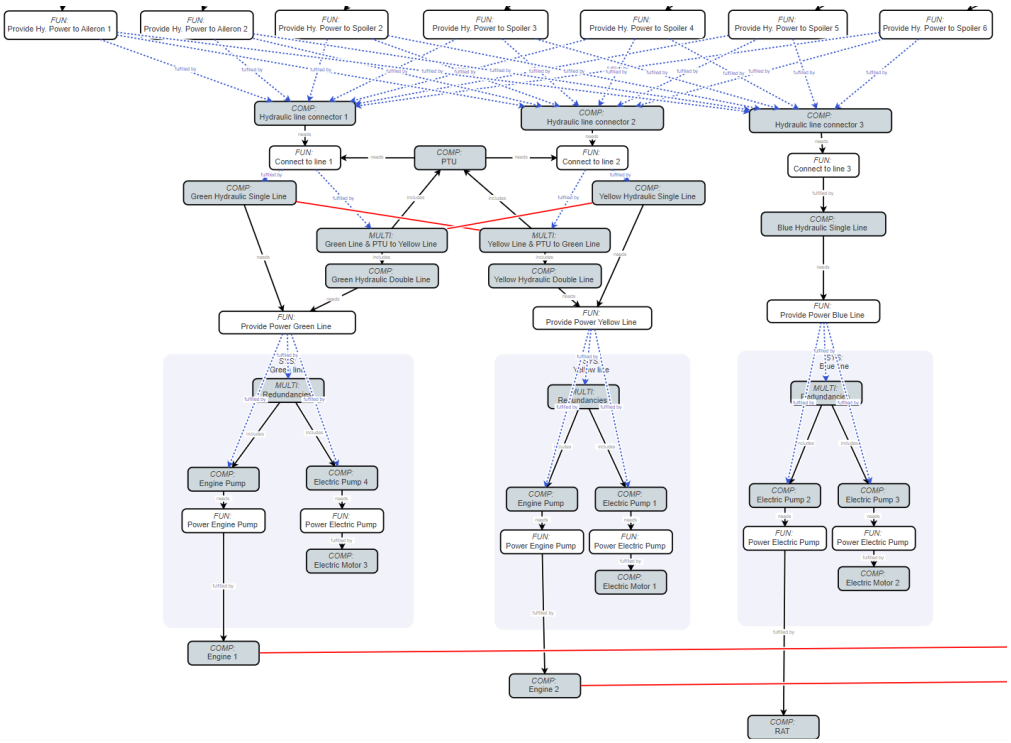


Fig. 5.5 Design space model; zoom in the hydraulic system, screenshot from ADORE

The last piece of the design space is the model of the electrical system. It is shown in figure 5.6. Three lines are present, as in the reference baseline. Lines one and two are the main ones, they can be powered by engine generators (one or two per engine) and/or by the APU generator. The emergency line can be powered by a generator connected to the RAT, from an electric generator connected to a hydraulic or pneumatic motor, or both. This motor is selected as just "motor" since a differentiation between pneumatic and hydraulic could not be done in terms of mass nor failure rates. This comes from a lack of literature in these components. However, for the purpose of this study this does not compromise the model, this "motor" component was considered as a generic motor that contributes as an extra redundancy for the electrical emergency line. Another interesting aspect is that the

APU is working here as another unconventional configuration when transformed into a RBD (V-configuration), as explained in chapter 3.

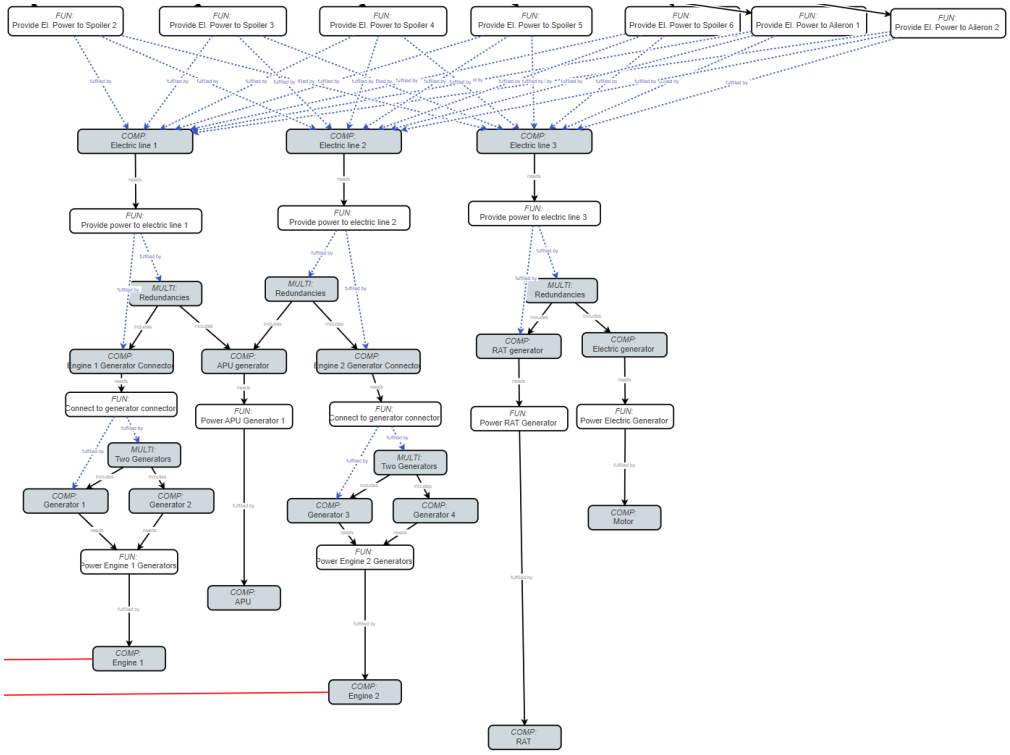


Fig. 5.6 Design space model; zoom in the electrical system, screenshot from ADORE

Once the whole design space is finished, specific information can be added. This applies to the components, functions, systems, etc... For this specific case, the information regarding the failure rates of the components is needed for the assessment of the reliability. This data can be directly provided inside of ADORE, ensuring traceability and a single source of truth for the input. Some components are now shown as an example. Figure 5.7 shows what is inside all the components representing a HSA actuator, in particular for spoiler 2 but all of them have the same structure. It can be seen how the connection to the respective functions is displayed, as well as the given failure rate value for that component. This way of providing the failure rates is repeated for all the components. All the values are summarized in table 5.2. Here the reader can find all the failure rates used for the components inside this design space. All the values are taken from the Quanterion Automated Databook, since it provides data from real component tests. Even if most of the components come from a military environment, this information is really useful since failure rates

estimation cannot be easily found in literature and it is sensitive information that companies do not generally share.

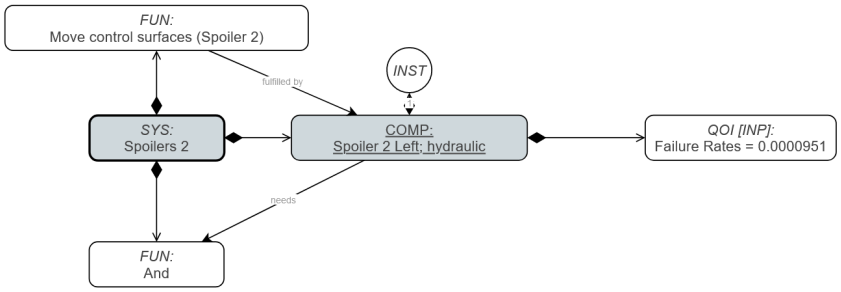


Fig. 5.7 Design space model; zoom inside the spoiler hydraulic actuator component, screenshot from ADORE

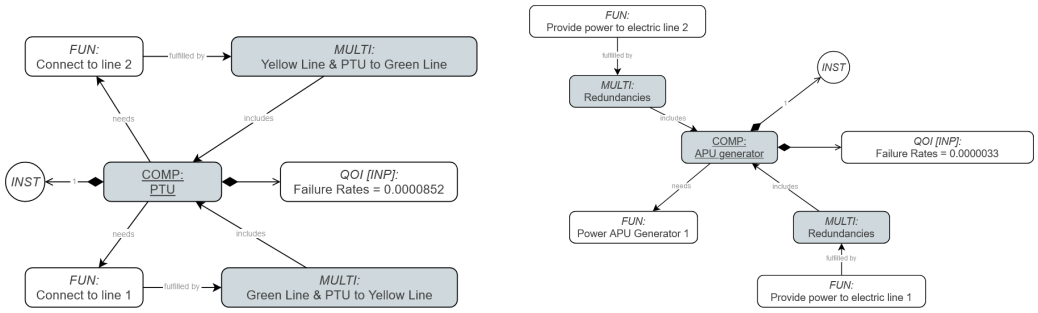
Table 5.2 Failure rates values for the components of the flight spoiler system, taken from the Quanterion Automated Databook (NPRD-2016)

Component name (in ADORE; from database)	Failure rates (per hour)	Quality	Environment
Spoiler Actuator (H); Actuator Hydraulic Linear	9.51×10^{-5}	Commercial	Airborne Uninhabited Cargo
Spoiler Actuator (E); Actuator Electric Linear	5.89×10^{-5}	Commercial	Airborne Uninhabited Cargo
Power Transfer Unit; Two series: Pump Hydraulic	8.52×10^{-5}	Commercial	Airborne Uninhabited Transport
Engine Pump; Pump Hydraulic, Engine Driven	2.19×10^{-6}	Military	Airborne Cargo
Electric Motor Pump; Pump Hydraulic, Motor Driven	1.70×10^{-5}	Military	Airborne Uninhabited Fighter
Engine; Engine, Turbofan	7.74×10^{-6}	Military	Airborne Uninhabited Attack
Electric Motor; Motor AC	2.39×10^{-6}	Military	Airborne Cargo
Ram Air Turbine; Turbine Unit Assembly	3.49×10^{-7}	Military	Airborne Cargo
Generators; Generator AC	3.30×10^{-6}	Military	Airborne Attack
Auxiliary Power Unit; Power Unit, Aircraft	5.47×10^{-6}	Military	Airborne Uninhabited Cargo

Other interesting components to show are the previously mentioned "unconventional-RBD components". In this design space these are the power transfer unit (H-configuration) and the auxiliary power unit (V-configuration). Their component details, input and connections are shown in figure 5.8a and figure 5.8b.

After the design space has been fully defined, it is interesting to check the characteristics and statistics of it. This includes for example the number of choices or architectures that can be generated from it, both are shown in table 5.3.

It is noticeable how exponentially the design space grows, just 32 decisions create more than 6 billion possible architectures. The design space grows even more if the whole FCS is modelled (i.e., including not only roll but also pitch and yaw). Some techniques are needed in order to explore such vast design spaces, hence raising the importance of including some kind of filtering in the evaluation framework, as well



(a) PTU component (b) APU component

Fig. 5.8 Design space model; zoom inside unconventional components, screenshots from ADORE

Table 5.3 Design space statistics

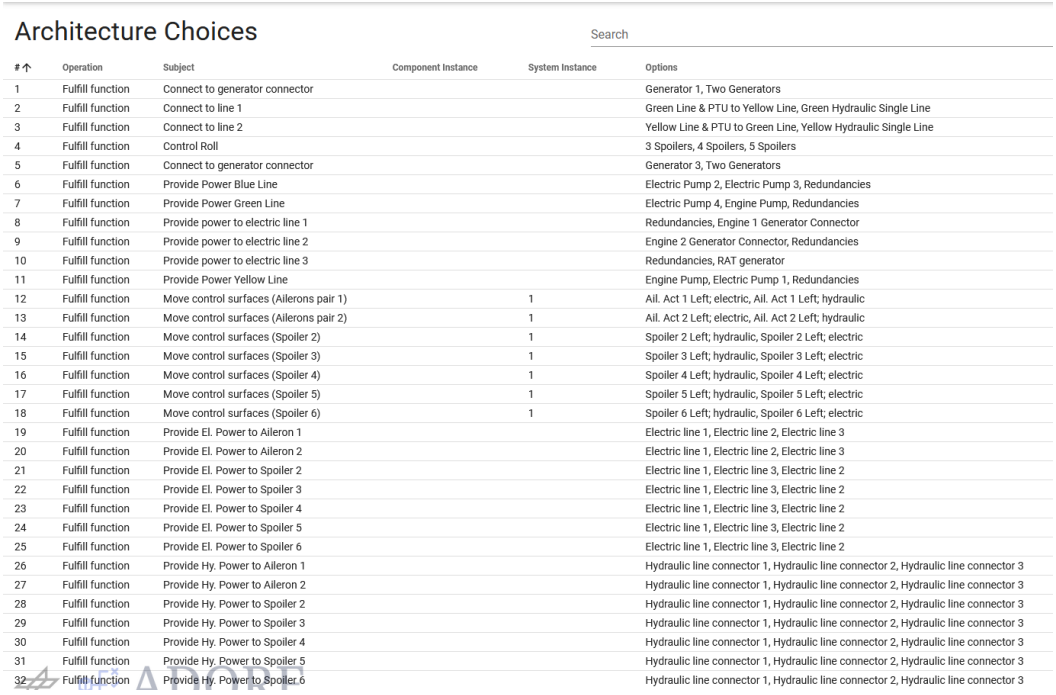
Parameter	Value	Description
Decisions	32	Total number of discrete decisions that need to be taken to fully define one architecture from the design space
Valid design space	6 347 497 291 776	Total number of possible different architectures that can be generated from the design space

as optimization algorithms. Figure 5.9 shows the list of all the different architectural choices that model the design space of this application case.

The design space has now been fully defined. The next step is to run the whole evaluation framework (i.e., certification, performance and maintenance disciplines). As explained in chapter 3, the methodology can be reduced and simplified for each particular application case to alleviate computational times and help with convergence. The running order is as shown in figure 3.2, where the XDSM shows the certification filter first, the performance convergence loop second, and the maintenance tool last. For this application case the certification discipline remains untouched, the tool is executed as explained in the methodology. However, the performance and maintenance disciplines can be simplified. Both are explained now in the following subsections.

Performance discipline adjustments

The convergence loop to evaluate the performance of an architecture can be quite challenging to achieve. Computational times can be high and convergence can be



Architecture Choices

Search

#	Operation	Subject	Component Instance	System Instance	Options
1	Fulfill function	Connect to generator connector			Generator 1, Two Generators
2	Fulfill function	Connect to line 1			Green Line & PTU to Yellow Line, Green Hydraulic Single Line
3	Fulfill function	Connect to line 2			Yellow Line & PTU to Green Line, Yellow Hydraulic Single Line
4	Fulfill function	Control Roll			3 Spoilers, 4 Spoilers, 5 Spoilers
5	Fulfill function	Connect to generator connector			Generator 3, Two Generators
6	Fulfill function	Provide Power Blue Line			Electric Pump 2, Electric Pump 3, Redundancies
7	Fulfill function	Provide Power Green Line			Electric Pump 4, Engine Pump, Redundancies
8	Fulfill function	Provide power to electric line 1			Redundancies, Engine 1 Generator Connector
9	Fulfill function	Provide power to electric line 2			Engine 2 Generator Connector, Redundancies
10	Fulfill function	Provide power to electric line 3			Redundancies, RAT generator
11	Fulfill function	Provide Power Yellow Line			Engine Pump, Electric Pump 1, Redundancies
12	Fulfill function	Move control surfaces (Ailerons pair 1)		1	Ail. Act 1 Left; electric, Ail. Act 1 Left; hydraulic
13	Fulfill function	Move control surfaces (Ailerons pair 2)		1	Ail. Act 2 Left; electric, Ail. Act 2 Left; hydraulic
14	Fulfill function	Move control surfaces (Spoiler 2)		1	Spoiler 2 Left; hydraulic, Spoiler 2 Left; electric
15	Fulfill function	Move control surfaces (Spoiler 3)		1	Spoiler 3 Left; hydraulic, Spoiler 3 Left; electric
16	Fulfill function	Move control surfaces (Spoiler 4)		1	Spoiler 4 Left; hydraulic, Spoiler 4 Left; electric
17	Fulfill function	Move control surfaces (Spoiler 5)		1	Spoiler 5 Left; hydraulic, Spoiler 5 Left; electric
18	Fulfill function	Move control surfaces (Spoiler 6)		1	Spoiler 6 Left; hydraulic, Spoiler 6 Left; electric
19	Fulfill function	Provide El. Power to Aileron 1			Electric line 1, Electric line 2, Electric line 3
20	Fulfill function	Provide El. Power to Aileron 2			Electric line 1, Electric line 2, Electric line 3
21	Fulfill function	Provide El. Power to Spoiler 2			Electric line 1, Electric line 3, Electric line 2
22	Fulfill function	Provide El. Power to Spoiler 3			Electric line 1, Electric line 3, Electric line 2
23	Fulfill function	Provide El. Power to Spoiler 4			Electric line 1, Electric line 3, Electric line 2
24	Fulfill function	Provide El. Power to Spoiler 5			Electric line 1, Electric line 3, Electric line 2
25	Fulfill function	Provide El. Power to Spoiler 6			Electric line 1, Electric line 3, Electric line 2
26	Fulfill function	Provide Hy. Power to Aileron 1			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3
27	Fulfill function	Provide Hy. Power to Aileron 2			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3
28	Fulfill function	Provide Hy. Power to Spoiler 2			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3
29	Fulfill function	Provide Hy. Power to Spoiler 3			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3
30	Fulfill function	Provide Hy. Power to Spoiler 4			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3
31	Fulfill function	Provide Hy. Power to Spoiler 5			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3
32	Fulfill function	Provide Hy. Power to Spoiler 6			Hydraulic line connector 1, Hydraulic line connector 2, Hydraulic line connector 3

Fig. 5.9 Design space model; summary of all the architectural choices, screenshot from ADORE

difficult to reach. For these reasons it is generally recommended to make small adjustments that help with these issues. The original unchanged methodology requires each architecture to be run firstly through the OBS sizing tool, then provide the results to the OAD tool and repeat the loop until convergence in MTOM is reached. However, as explained in section 3.4, this loop can be reduced. The main reason is that there is only one power consuming system (with a small power request) changing from one architecture to another. This lowers the total impact on mass and power that the architectural changes have, hence not always needing the convergence loop (i.e., converging in one single run). This simplification can be done only if the baseline aircraft is fixed, as in this case (A320neo). On the other hand, the FCS is the only power consuming system that is sensitive to the design variables, hence another simplification can be done. The OBS tool (i.e., ASTRID) does not really need to be run for every architecture since the dedicated FCS tool already evaluates them individually. The OBS sizing tool is then used only for those other systems apart from the FCS. This tool can be executed outside of the loop and provide a baseline for all those systems, not needing to be run for every single architecture. However one thing must be noticed, the OBS are not the same for a conventional

aircraft or for an all-electric one. Each of these aircraft require a different baseline for the remaining OBS, having a total of three baselines that will be explained later. The final workflow was shown in figure 3.37. It is noticeable how the convergence loop disappears and the performance discipline remains as follows:

1. A preliminary certified architecture is provided by the certification filter.
2. The dedicated FCS tool runs this architecture and evaluates the mass and power required by it.
3. The proper OBS baseline is selected for the rest of the systems depending on the FCS architecture. The correspondent previously-calculated file with the results for such systems is given and merged with the FCS results.
4. A small surrogate model is executed for the relevant distribution systems (i.e., hydraulic and electrical systems). This allows to account for changes in the distribution systems that the FCS architecture can have. This includes only small changes made by single components, the general distribution system architectures are already described and estimated in the OBS baseline files.
5. All the previous results are merged and given to the OAD tool (i.e., OpenAD). The toolchain provides the aircraft MTOM and fuel consumption as results.

The convergence loop is not needed anymore and the OBS is not run for every iterations. This highly reduces the computational time per architecture and avoids convergence issues. As a reference, ASTRID's run time can vary from some seconds to some minutes depending on the architecture, saving some minutes per architecture makes a huge difference for a design space such as the one presented, the computational time savings are not negligible.

Regarding the OBS simplification, three baselines are needed. One with a conventional architecture, one with a more-electric and one with an all-electric. The conventional baseline is selected when all the FCS actuators are conventional hydraulic servo actuators (HSA). On the other hand, the all-electric baseline is used when all the FCS actuators are electrified (with EHA). If there are actuators from both typologies, the more-electric baseline applies. The mass results for each of these baselines are shown in table 5.4, this table displays the results run with ASTRID plus

Table 5.4 ASTRID results for the three OBS baseline architectures

	Conventional [kg]	**More-Electric [kg]**	**All-Electric [kg]**
Avionics	781	781	781
*Flight Control System (FCS)**	846	908	1008
Ice Protection System (IPS)	73	73	105
Environmental Control System (ECS)	480	480	589
Fuel System	344	344	344
Landing Gear	2176	2176	2320
Fire Protection	95	95	95
Lights	341	341	341
Oxygen	112	112	112
Water Waste	302	302	302
APU	138	138	122
Pneumatic System	169	169	0
Hydraulic System	1022	649	0
Electrical System	1443	1539	1395
Total	**8322**	**8107**	**7513**

* Calculated with the dedicated FCS tool, not with ASTRID

the FCS from the dedicated tool. The exact architecture of each baseline is explained in the following paragraphs together with the results of each.

The results from table 5.4 are given per OBS as provided per ASTRID. Each of the baselines is now commented with the results:

- The conventional architecture is as the one of the A320. All the FCS and landing gear actuators are HSA. The high-lift devices are moved by ball-screw actuators connected to a hydraulic central power drive unit. The ECS is conventional based on engine bleeding, as well as the IPS. The hydraulic system has three lines, two on them powered by the engines and connected through a PTU, and one emergency line connected to a RAT. The electrical system is powered by two integrated drive generators connected to the engines, the APU and batteries and a conventional pneumatic system is also installed. It generates power at 115V AC.

- The all-electric baseline is built in the case is that all the systems are fully electrified. Flight controls are moved by EHAs and high-lift devices by electric power drive units. The ECS is based on a bleed-less technology, having two extra external compressors that condition the air before providing it to the air packs. IPS utilizes thermal blankets for the wing surfaces and anti-ice valves connected to the engine flow for the nacelles. All actuators for the landing gear are electrified. Starter generators are used for the engine start. The electrical system remains similar but generates power at 230 Volts in alternate current.

The pneumatic and hydraulic systems fully disappear since they are not needed anymore.

- The more-electric baseline is based on the A350, representing an intermediate point between the previous ones. In this case flight controls are mixed since they have HSAs and EHAs. The rest of the power consuming systems remain as in the conventional architecture: high-lift devices, ECS, IPS and landing gear. Also the pneumatic and electrical systems remain unchanged. The hydraulic system however is based on two lines instead of three, being lighter as a result.

It can be seen how the main changes from one baseline to another affect the FCS and distribution systems, as expected. FCS mass increases when using electric actuators, since these are generally heavier. However, the removal or reduction of the pneumatic and hydraulic systems compensates this increase, making the total OBS mass decrease. The electrical system is lighter for the all-electric baseline, which might come as as surprise since this system now needs to provide higher power demands. This comes from the technology advance of generating at a higher voltage which manages to reduce the size of some components (e.g., transformers, cables, power distribution units, rectifiers...). As a result, the total mass of the electrical system is lightly reduced even if the required power is higher compared to the conventional architecture.

All the OBS are now sized and defined. However one issue arises for this implementation, ASTRID cannot handle all the architectures that are generated. This means, the tool is not sensible to all the design variables. The main reason is that the design space allows to generate quite random architectures while ASTRID is design to be used for real aircraft. The result is that there might be some non-conventional and non-typical architectures that cannot be properly sized. For this a small surrogate model is added right after running ASTRID. This surrogate model interprets the design variables and checks if any modification to ASTRID results is needed. It is only done for the power distribution systems (i.e., electrical and hydraulic systems in this case). The flight control system is already fully defined by the dedicated FCS tool, and the rest of the OBS are properly sized by ASTRID since the design variables do not affect them with a high level of detail. As a result this surrogate reads the design variables and determines whether or not the distribution systems sized by ASTRID need small adjustments in their results. Some key components

are now analyzed for each distribution system. For the hydraulic system, only some determined components are affected by the design variables, these are the power transfer unit, extra engine-driven pumps and electric-motor-driven pumps. For instance, it should be possible to assess the same architecture with or without PTU, or with electric pumps instead of engine-driven ones. Regarding the electrical system, the number of generators is a design variable, as well as the possibility to have an extra conjunct of motor plus generator in the emergency line. Such motor can be hydraulic or pneumatic, the choice is not really relevant for the design variables since both are estimated with the same results since no further information about these components could be retrieved.

The result is a surrogate model for distribution systems mass. This model adds or subtracts mass to the results provided by ASTRID for such systems, but only in case a non-typical architecture is given to the performance loop. This architectures are characterized by some key components that are listed in the following list. The values should not be taken as exact, but can be used as a proper initial estimation that manages to make the performance loop sensible to all the design variables. Furthermore, these estimations can only be used for the proposed application case (i.e., A320neo). The following values show the estimated extra mass that must be added/subtracted in that case that an extra instance of that specific component is added/removed from the baseline architecture. Results are as it follows:

- Engine driven pump: 23 kg
- Electric motor pump: 13 kg
- Power transfer unit: 37 kg
- Motor in the electrical emergency line: 23 kg
- Engine generator (all-electric): 67 kg + 12% more for the GCU
- Engine generator (conventional): 61 kg + 12% more for the GCU

Maintenance discipline adjustments

The same issue affects the maintenance discipline. The maintenance tool is not sensible to all the design variables and a small surrogate model needs to be added. The

surrogate model explained in the methodology in chapter 3 is executed and then these small contributions are added. The baseline OBS architecture is assessed obtaining a reference value for the MMH reduction. Depending on the design variables, some additions/subtractions are performed. These estimations are calculated by checking the correspondent tasks in the A320 MPD and extracting the interesting data. As for the previous surrogate, the results can be directly added or removed from the total after the maintenance model has been run. The results show the estimated extra MMH for a specific extra component:

- Integrated drive generator: 26 hours
- Variable frequency generator: 24 hours
- Extra redundancy in the electrical emergency line: 32 hours
- HSA spoiler: 2.4 hours
- EHA spoiler: 1.5 hours

5.3 Results

This sections presents the results of the execution of the whole framework for the application case. Firstly, a design of experiments is performed. Then, several optimization processes are presented.

Initial Design of Experiments

A design of experiments is performed before running any optimizations. This allows to run some interesting architectures first and get some insight about the possible results. For this run the certification filter was disabled, this means that the certification discipline is evaluated and provides results, but if the architecture comes as "non-certifiable" it is not discarded. The objective of this is to see the results of different interesting architectures in the three domains to have some preliminary conclusions before launching the optimization processes. Eleven relevant architectures are evaluated for this design of experiments. They are explained in a summarized way in table 5.5.

Table 5.5 Summary of the architectures analyzed in the initial design of experiments

#	Name	Description
1	CONV-baseline	Baseline OBS architecture from the A320. Four flight spoilers per wing, three hydraulic lines and three electrical ones.
2	AEA-baseline	All actuators are electrified to EHAs. The hydraulic system disappears and the electrical system is left as it is in the CONV-baseline
3	MEA-1	Similar to the A350 architecture. Only two hydraulic lines remain (green and yellow with PTU). The actuators that were connected to the blue line are now electrified and connected to one main electrical line.
4	MEA-2	Similar to the B787 architecture. Three hydraulic lines without PTU. Only two spoilers are electrified.
5	AEA-enhanced	As architecture 2 but adding one extra generator per engine and one extra redundancy in the emergency electrical line.
6	AEA-extra-gen	As architecture 5 but adding only the extra generator per engine.
7	AEA-extra-eline	As architecture 5 but adding only the extra redundancy in the emergency electrical line.
8	CONV-3spo	As architecture 1 but with only 3 flight spoilers (instead of four).
9	CONV-5spo	As architecture 1 but with 5 flight spoilers (instead of four).
10	AEA-3spo	As architecture 5 but with only 3 flight spoilers (instead of four).
11	AEA-5spo	As architecture 2 but with 5 flight spoilers (instead of four).

It can be noticed how these architectures represent single changes from the baseline architectures. Architecture 2 represents a change to all-electric without any adjustment on the electrical system. Architecture 5 represents an enhanced version of architecture 2. Architectures 3 and 4 represent more-electric variants based on other vehicles. Architectures 8 to 11 aim at checking the impact of changing the number of spoilers for a conventional and an all-electric version. All these cases were evaluated following the methodology and the results are displayed in 5.6. This table shows the results on the three different domains. The FCS mass is calculated with the dedicated FCS tool, the hydraulic and electrical systems mass with ASTRID plus the surrogate models modifications, the MTOM and fuel burn with OpenAD, the probability of failure and the three constraints with ACOBS, and lastly the MMH reduction with the maintenance tool plus the surrogate adjustments. It should be noticed how the architectures that have one of the constraints equal to zero would be filtered. However they were not discarded for this initial design of experiments.

Table 5.6 shows the results of the design of experiment architectures. These architectures are manually created with ADORE and automatically evaluated with the proposed methodology. As a reminder to the reader, the displayed value for the probability of failure is a non-dimensional value that comes from dividing the result by the reference probability of failure (i.e., the one of the A320). As an example, architecture 2 fails 39 times more than the reference one, and hence would not be certifiable. Any value between 1 and 0 is acceptable, more than 1 is non-certifiable.

Table 5.6 Results for the 11 architectures of the design of experiments. The reference probability of failure* is 1.2e-22. The reference MMH† for the A320 are 166.750 hours

Architecture & Name	FCS mass [kg]	Hydraulic system mass [kg]	Electrical system mass [kg]	MTOM [kg]	Fuel burn [kg]	Probability of failure ratio [-]*	Back-up constraint [-]	Single failure constraint [-]	Minimum reliability constraint [-]	Reduction of MMH [hours]†
1) CONV-baseline	846	1022	1443	78967	19080	1	1	1	1	0
2) AEA-baseline	1008	0	1395	77511	18529	39	1	1	0	1502
3) MEA-1	908	663	1539	78656	18935	0.6	1	1	1	335
4) MEA-2	897	952	1539	79085	19134	91	1	1	0	245
5) AEA-enhanced	1008	0	1636	77883	18700	0.085	1	1	1	1422
6) AEA-extra-gen	1008	0	1546	77744	18636	22	1	1	0	1454
7) AEA-extra-eline	1008	0	1485	77651	18593	0.15	1	1	1	1471
8) CONV-3spo	827	1022	1443	78938	19066	2171	1	1	0	5
9) CONV-5spo	866	1022	1443	78997	19094	0.008	1	1	1	-5
10) AEA-3spo	977	0	1636	77836	18678	193	1	1	0	1425
11) AEA-5spo	1039	0	1395	77559	18551	33	1	1	0	1499

The other three constraints are also shown (back-up systems, single failure and minimum reliability). If only one of those is equal to zero the architecture would not be certifiable. The MMH represent the total hours reduced with respect to the baseline architecture. Some of the masses from table 5.6 are displayed in figure 5.10 for visual support.

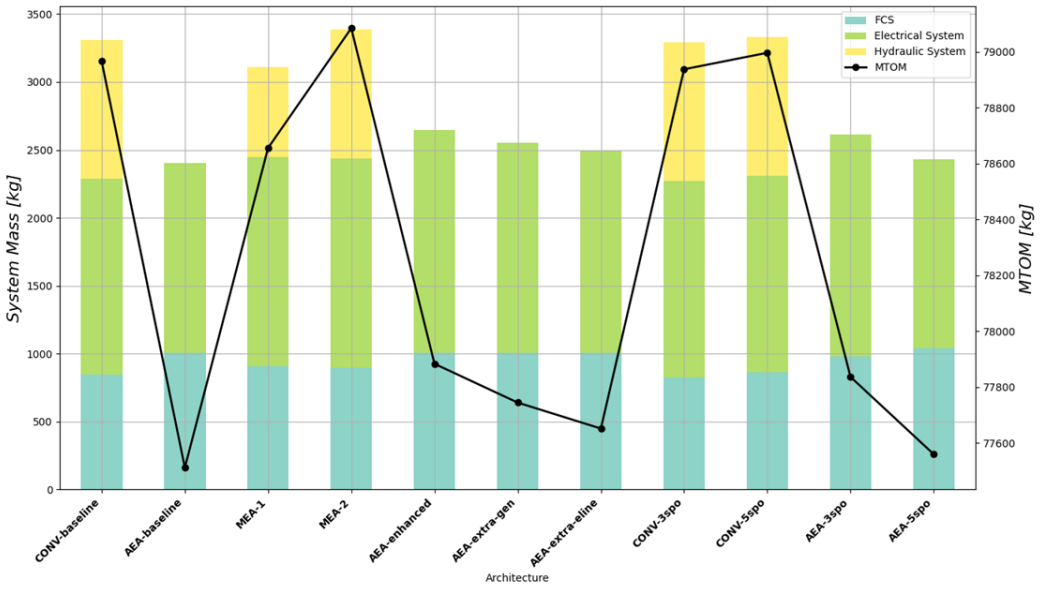


Fig. 5.10 System mass breakdown by architecture; with MTOM

Regarding the performance results, there are three main regions for the FCS mass. In general the conventional architectures are the lightest, since the HSAs are generally lighter than the more-electric versions. The all-electric architectures result as the heaviest, for the same reason, while the more-electric ones stay in the middle between the previous. It can be seen how there is a slight mass variation

for each architecture when the number of spoilers change. On the other hand, the hydraulic system mass is zero for the AEA and close to a ton for the conventional case. The MEA architectures result in a middle point that depends on the number of lines and components selected. These variation in the results come as a result of adding the surrogate model for the distribution systems, and adds more fidelity to the results. The electrical system mass is also affected by the distribution systems surrogate model. As a result, architecture 5 is the heaviest since it has more lines and components than the reference one (extra generators and emergency line). The MTOM varies accordingly depending on the mass of the OBS. The bleed penalties also have an impact, although lower. Initially two optimization objectives were set for the performance discipline, which were MTOM and fuel burn. However, both results are quite linear-dependent in this specific case. This can be seen in figure 5.11. The marked linearity comes from the effect of having small changes from a common aircraft baseline. For this reason only MTOM was left as an objective and fuel burn was discarded.

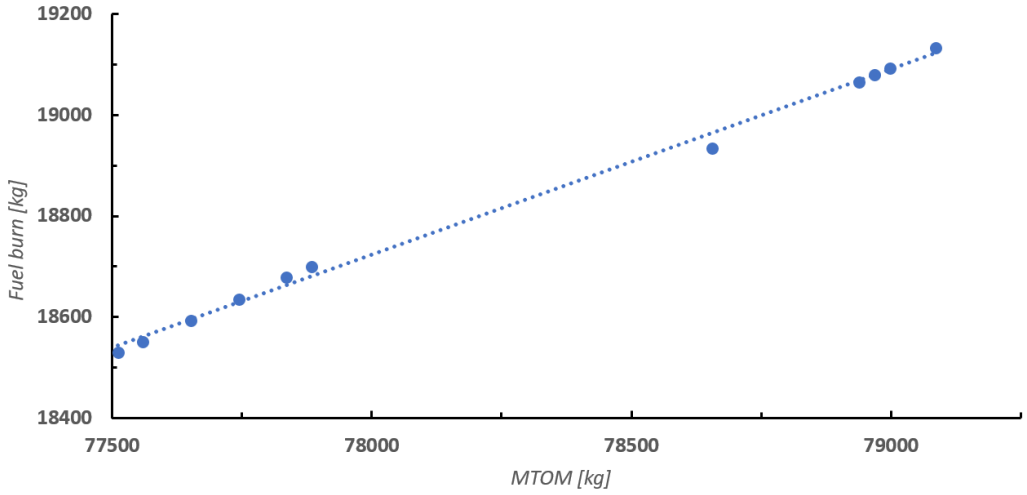
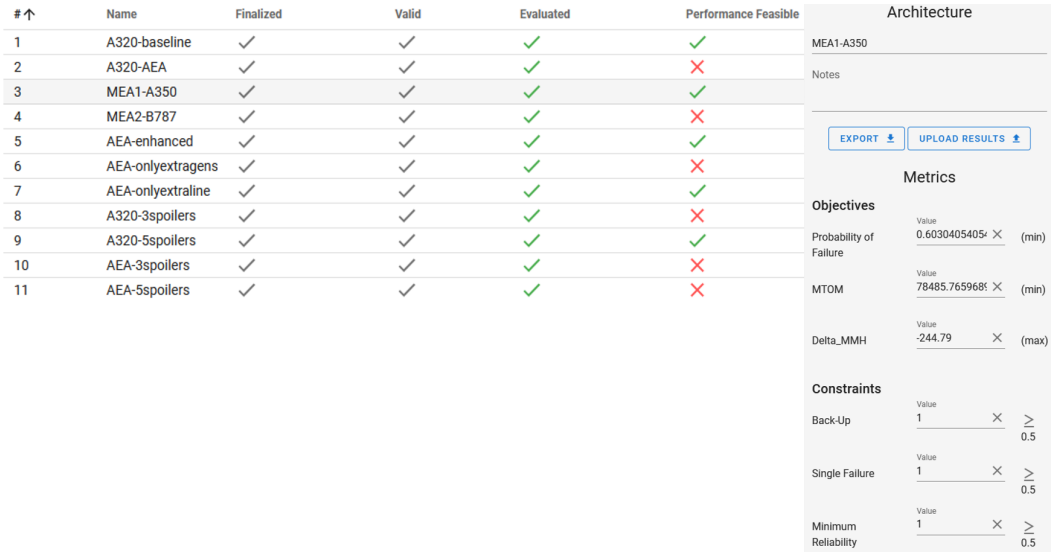


Fig. 5.11 Linear dependency between MTOM and fuel burn for the specific application case

Regarding the certification discipline it is noticeable how only five out of eleven architectures are potentially certifiable. This intensifies the need for assessing certification related parameters during early design phases. Results can easily be seen in ADORE by opening the file with the results as shown in figure 5.12. The software provides instant visual feedback for each of the architectures telling the user which ones have not met the constraints. It also shows the results for the different optimization objectives. It can be seen how architecture 2 would not be certifiable.

This architecture replicates the electrical system of the baseline aircraft and uses it directly to power the new added electric actuators. The result is that changing actuators from HSA to EHA and connecting them to the existing electrical lines might not be enough for an architecture, since here it is shown that for this case this would not be certifiable. The main explanation is that for the baseline architecture with hydraulic actuators there are different levels of redundancy. Three main hydraulic lines are present. Two main lines with a PTU are present, and an emergency one with two different power sources (i.e., RAT and electric pump). When switching to the AEA architecture the electrical system is done similarly but with differences. There are also two main lines connected to the engine generators and APU. And there is one emergency line connected to the RAT and batteries. However, the batteries cannot be used for such function as powering the flight controls because they would be too big for this functions. This leaves the emergency line without an extra redundancy and as a result the reliability drops. In this case this architecture fails 39 times more than the reference one and hence is not certifiable. Architectures five, six and seven are variations of architecture two in which the electrical system has more redundancies. These are implemented to check how can this reliability issue be fixed for the AEA. Architecture 6 is the same as architecture two but adding two generators per engine. This does increase the reliability, but not enough to be certifiable (fails 22 times more than the reference). Architecture seven is the same as architecture two but adding one extra redundancy in the emergency line. This is done by adding an extra generator connected to a hydraulic/pneumatic motor. For this solution the reliability increases massively, instantly fixing the issue. For this case the non-dimensional probability of failure is 0.15, which means that the architecture fails only a 15% of the baseline value. Just by adding these extra components the architecture is now safe, although slightly heavier. It is also noticeable how changing components from hydraulic to electric, and adjusting the redundancies results in a safer system in general. This comes from the effect of having electrical components that are generally safer (i.e., they have lower failure rates) when compared to hydraulic ones. This conclusion can be extrapolated to other FCS functions or even systems, the probability of failure needs to be checked when moving from conventional to all-electric architectures.

Lastly, the results for the maintenance discipline are easier to read. AEA architectures achieve a bigger reduction in MMH, which does not come as a surprise since both the pneumatic and hydraulic systems are removed. It is true that the electrical system MMH are now higher, but the trade-off between these two effects comes



#	Name	Finalized	Valid	Evaluated	Performance Feasible
1	A320-baseline	✓	✓	✓	✓
2	A320-AEA	✓	✓	✓	✗
3	MEA1-A350	✓	✓	✓	✓
4	MEA2-B787	✓	✓	✓	✗
5	AEA-enhanced	✓	✓	✓	✓
6	AEA-onlyextragens	✓	✓	✓	✗
7	AEA-onlyextraline	✓	✓	✓	✓
8	A320-3spoilers	✓	✓	✓	✗
9	A320-5spoilers	✓	✓	✓	✓
10	AEA-3spoilers	✓	✓	✓	✗
11	AEA-5spoilers	✓	✓	✓	✗

Fig. 5.12 Visualization of the results, screenshot from ADORE

as favorable. MEA architectures only manage to reduce around a quarter part of what AEA architectures achieve. This is normal since the hydraulic system is just reduced, not removed. The OBS results are now compared with the whole aircraft in order to give a better perspective to the reader. Aircraft Commerce provides a value for the total MMH for the A320 of 166.750 hours [33, 131]. This is obtained by summing all the different hours from A-checks and C-checks. The reduction achieved by the conversion from a conventional to an all-electric architecture is of less than 1% of the total MMH. This can seem low but is actually not negligible. As general and preliminary conclusions from this design of experiments it can be said that MEA architectures are generally heavier that AEA ones, but also more reliable because they mix two distribution systems for the same power consuming system. If AEA architectures are enhanced to reach the required reliability they have the potential to also reduce the MTOM and MMH. Adding or removing spoilers does not have a good impact since changing the distribution lines has a bigger impact in the probability of failure.

Optimization with NSGA-II Algorithm

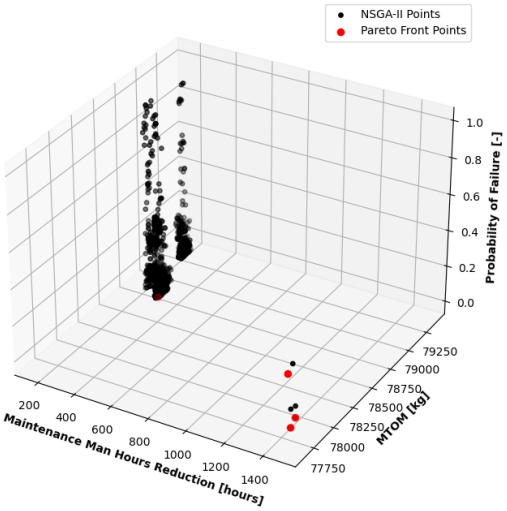
An optimization process is performed for this design problem. The architectural choices are selected as design variables. The optimization objectives are the MTOM, MMH reduction and probability of failure of the architecture (divided by the ref-

erence probability of failure). The design variables are mostly discrete, and some of them even hierarchical. This discards gradient-based optimization as a feasible option, leaving genetic algorithms as the best choice for the optimization algorithms. In particular, the multi-objective genetic algorithm NSGA-II [159] is chosen for this analysis. The primary reason is its ability to find multiple Pareto-optimal solutions in one single run, since it is an evolutionary algorithm [159]. Furthermore, this algorithm has been used extensively in literature and it is already implemented inside ADORE [144]. Diverse runs were launched with different number of generations and population size. As a general rule of thumb it can be stated that the population size should be around ten times the number of design variables. However, this statement is not taken from literature and comes from experience with these kind of analysis. As a result the final run is performed with a population size of 320 and 20 generations. This results in more than 6.000 points and took almost three days of computational time to be run. An interesting outcome is that more than seventy percent of the generated architectures were discarded by the certification filter, which takes about a second to evaluate one architecture. If these architectures had not been discarded, they would have needed to be assessed in the performance loop which takes a couple of minutes of time to be evaluated. Computational times without the certification filter would have been much higher. The most relevant results, from a run points perspective, are shown in table 5.7. It is also noticeable by looking at the results how initially most of the architectures are discarded by the certification filter, but as the run progresses the algorithms learn to avoid these architectures and start "guessing" more feasible points.

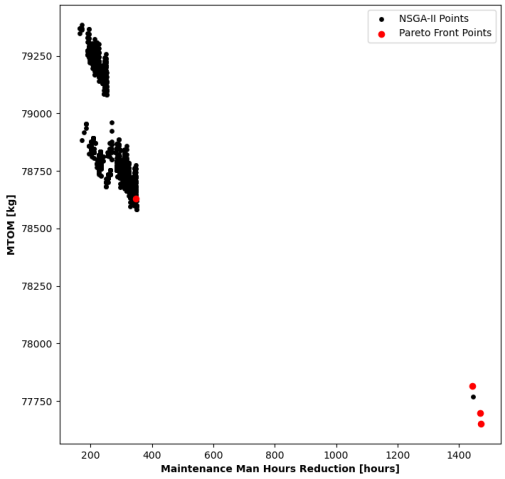
Table 5.7 Optimization results analysis

Parameter	Value	Description
Points (Theoretical)	6400	Total number of points that should be run in theory. It comes from multiplying the population size (320) by the number of generations (20).
Points (Run)	6381	Total number of run points. Sometimes it can happen that some architectures are not run or fail, this does not stop the algorithm and runs the next architecture. Only 29 architectures failed the execution on this analysis, which can be considered as a very low number (less than 0.5%).
Points (Performance Feasible)	1891	Total number of architectures that passed the certification filter and were evaluated in terms of performance and maintenance. They represent around a 30% of the total number of architectures.

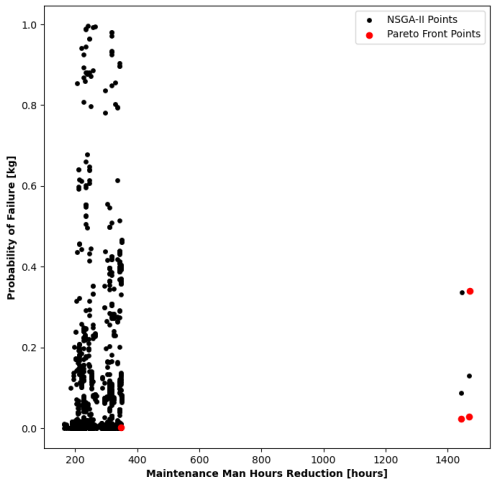
The result of the optimization process is a 3-dimensional Pareto front, each dimension corresponding to one of the optimization objectives. Results are shown in figure 5.13. This figure displays such Pareto front and the corresponding 3 projections on each of the 2D planes. The 3D visualization can be found in figure 5.13a. Here it is noticeable how there are two very separated regions, one of them



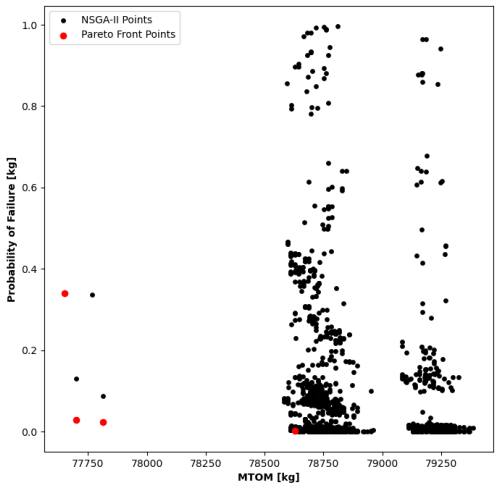
(a) 3D Pareto Front



(b) 2D projection of the Pareto Front



(c) 2D projection of the Pareto Front



(d) 2D projection of the Pareto Front

Fig. 5.13 Visualization of results for the optimization run with NSGA-II genetic algorithm

with a low number of points and another with the majority of them. It is better to check the projections of the Pareto in order to reach some conclusions. Figure 5.13d shows the 2D projection of the 3D Pareto Front on the plane where the MTOM and probability of failure are represented, leaving the maintenance results in the other two projections. Regarding this figure, the two very separated regions are again found. Those points, or architectures, that go to MTOM values below 78 tons correspond to all-electric architectures that successfully passed the certification filter. It is very noticeable how only six architectures are present here. This phenomenon is

explained and commented now. It was seen before how AEA need extra redundancies in the electrical system in order to be certifiable (this was a result from the design of experiments in table 5.6). This means that AEA architectures need extra care in order to pass such filter, and most of them do not make it. Random AEA architectures are generated from the design space, but it is difficult for them to pass the filter and be evaluated. Another effect is that the total number of possible AEA architectures is quite low when compared to the total amount. This comes purely from the design space. Even if only one actuator is hydraulic it will make the architecture a MEA, not an AEA. Summing the effect of having a low chance of creating an AEA architecture plus the difficulty for them to pass the certification filter results in having a really low number of AEA points in the final graph, as seen in the results. This is quite interesting since only the most interesting AEA points are now considered, needing to interpret and compare less results. This narrows down the solutions quite a lot from the huge amount of architectures that were envisioned initially. From these six points there are three that belong to the Pareto Front, but these results are explained later. Moving now to the right hand side region of architectures in figure 5.13d, where they correspond to aircraft that are heavier than 78500 kg. These points represent conventional and more-electric solutions, and it is noticeable that their quantity is much higher that in the previous case. This effect is explained again because these architectures are more common in the design space and are also easier to be certified. As a result much more solutions pass the certification filter and are evaluated and displayed. One interesting effect that is found is the verticality of such results. Some points have the same, or very close, MTOM but completely different result regarding the probability of failure. This effect exists because two or more architectures can have the same components but different connections among them. For example an architecture with four actuators (A, B, C and D) and two lines (1 and 2) can connect their components in different manners. The probability of failure would be low if they are connected like this: A2, B1, C1, D1; because if line 1 and actuator A fail the whole system fails. The probability of failure would be higher in the case of a more-optimized connection, such as: A2, B2, C1, D1; since now the system is not lost when line 1 and actuator A fail. Both examples had the same components, and as result would have the same result in terms of mass. However their probability of failure is quite different depending on how well optimized the connections among components are. This fully accounts for the verticality effect commented before. Continuing still with figure 5.13d, a slight trade-off between the probability of failure

and the MTOM is seen. This means that, in general, the resulting aircraft with a lower probability of failure architecture are heavier as a consequence. This does not come as a surprise since a higher number of components would generally mean that there are more redundancies available, which raises the reliability levels but increases the mass as well. As an instance, having more spoilers or more generators would increase reliability but at the same time penalize the mass. MEA architectures can reach really low values of probability of failure because they allow a mix of distribution lines (electrical and hydraulic lines). A possible MEA solution could have three hydraulic lines and three electrical ones connected to the spoilers. This solution reaches a high level of redundancy but highly penalizes the mass. Figure 5.13c also shows this verticality effect as a consequence. Here the MMH reduction is displayed together with the probability of failure. The AEA points shift now to the right hand side of the figure. As before, the same number of components can conform different architectures depending on how the connections are managed. This provides diverse probabilities of failure but the same MMH reduction. An analogous trade-off to the previous figure is here found. Incrementing the number of components lowers the probability of failure but increases the required maintenance as a consequence, which penalizes the MMH reduction. Lastly, figure 5.13b shows the projection of the Pareto Front in the plane of MTOM and MMH reduction. In this case there is not a trade-off between the variables. Moving from a MEA to an AEA solution removes two distribution systems (i.e., hydraulic and pneumatic), as commented in previous chapters. As a result, and as it was seen in the previous analyses, this derives in a more-maintainable and lighter aircraft. There is a clear optimum point for this projection so no trade-off is present. One interesting general effect is the low number of conventional architectures that were generated. These solutions have no improvement in any of the three disciplines, hence the algorithms understand quite fast that this region of the design space is not an area of interest and do not explore around it.

The four points that build the Pareto Front are shown in table 5.8. It is quite surprising that only four, out of 6400 points, lean on this region. The explanation to this is now provided and comes as a result of filtering most of the AEA architectures. Figure 5.13b showed that in this projection there is a clear optimum, leaving the trade-off only with the probability of failure. MEA architectures have difficulties competing against AEA ones. On the other hand, the number of certifiable AEA solutions is quite low, as explained before. The optimization algorithms together

with the certification filter manage to highly narrow down the design space to a small number of interesting solutions. Such solutions shall be further evaluated with higher fidelity analysis. However one main purpose of this dissertation is here achieved, managing to successfully remove the majority of non-feasible and non-interesting architectures from the huge initial design space.

Table 5.8 Pareto Front points from the optimization run with NSGA-II genetic algorithm

	MTOM [kg]	**Probability of Failure [-]**	**MMH [hours reduced]**
Pareto Point 1	77699	0.0279	1468
Pareto Point 2	77651	0.3395	1471
Pareto Point 3	78630	0.0017	349
Pareto Point 4	77815	0.0228	1444

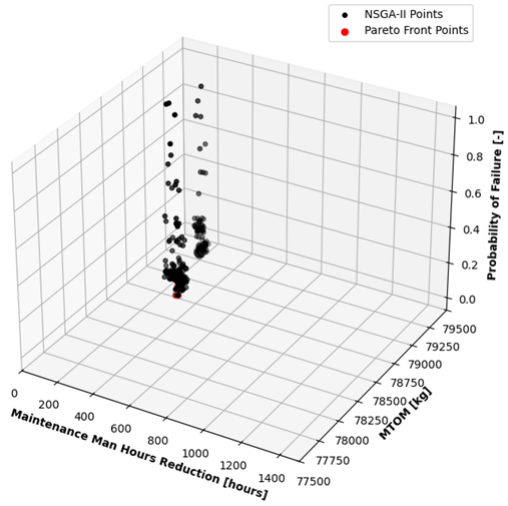
The four optimal architectures that form the Pareto Front are commented in this paragraph. Before seeing the specifics it is important to highlight that the objective of this dissertation is not to find the best architecture for the shown flight spoilers system. The focus is on the methodology that allows to filter the design space and supports the decision making process by showing interesting trade-offs and trends. This helps the designers during the design process by providing relevant and useful preliminary conclusions. The final decision of which architecture is best in a real design would also need to consider aspects from manufacturing and/or other disciplines as well. The Pareto Front points are now briefly commented.

- **Pareto Point 1:** AEA architecture with five flight spoilers. All actuators are EHAs (i.e., spoilers and ailerons). This solution has a classic electrical system architecture. There are two main lines, each of them powered by each of the engine generators, and both connected to the APU as redundancy. There is also an emergency line powered by the RAT and a redundant motor linked to a generator. The two ailerons and two of the spoilers are connected to each of the main lines, respectively. The three remaining spoilers are powered by the emergency line.
- **Pareto Point 2:** AEA architecture with four flight spoilers. The rest of the architecture is practically the same as for the previous one. Since there is now one less spoiler the MTOM suffers a small decrease, the MMH reduction sees a slight improvement but the probability of failure increases massively. Nevertheless, this architecture is yet more reliable than the baseline one.

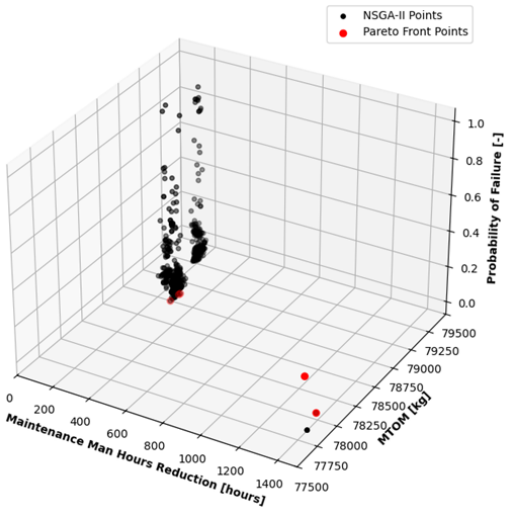
- **Pareto Point 3:** MEA architecture with five flight spoilers. Both ailerons are HSAs and the hydraulic system is built by only two lines. They are connected respectively one to one. These lines are redundant with electric-motor-driven pumps. The spoilers are all electric, two of them connected to one main electrical line, two connected to the other main line, and the last spoiler to the emergency one. The main electrical lines are powered by their respective engine-driven generators, and the APU as common redundancy. The emergency line is only connected to the RAT since now the extra redundancy is not needed.

- **Pareto Point 4:** Same architecture as point 1 but with the addition of one extra generator in engine one. This modestly penalizes the MTOM and MMH but favors the mission reliability. This architecture is not really of interest since it breaks the symmetry of the aircraft but nevertheless shows an interesting effect and trade-off.

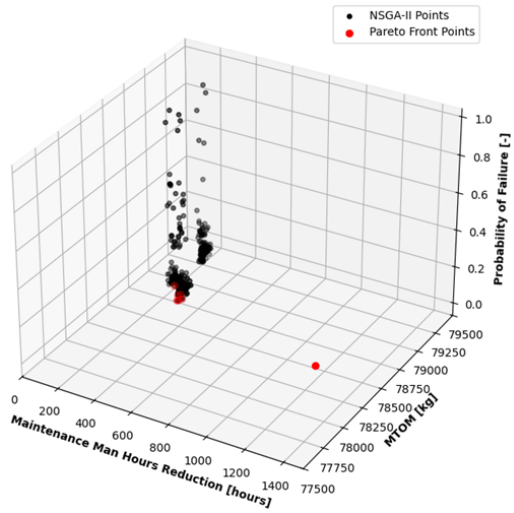
As a side note, these results were obtained as a result of a run of the optimization problem with the NSGA-II algorithm with a population size of 320 and 20 generations. As a general rule of thumb that comes from experience, the population size should generally be at least ten times the number of design variables (i.e., decision for this case) and a number of at least ten generations. This is to let the algorithms enough time to learn the patterns and properly analyze the design space. However, this rule is not compulsory and serves only as a reference. Further runs must be carried out to assess the sensitivity of these parameters in the results. Preliminary to this run other different runs were performed in order to see such sensitivity and impact on the results. These results are shown in figure 5.14. The most noticeable effect is seen in figure 5.14a where the algorithms did not manage to find any all-electric architecture yet. This is a clear example of a low number of generations in which a fake Pareto Front is displayed from a lack of representative points. Figure 5.14b solves this issue by just adding 5 more generations. This gives the algorithms more chances to properly analyze the AEA architectures and manage to make them pass the certification filter. Increasing the generations from 10 to 20 does not change the Pareto Front shape substantially, it just adds one more point that is not really representative (i.e., Pareto point 4 which breaks aircraft symmetry). For this main reason it is assumed that 20 generations is a sufficient number. Looking at figure 5.14c it can be notice how only 1 AEA architecture has been found. This does



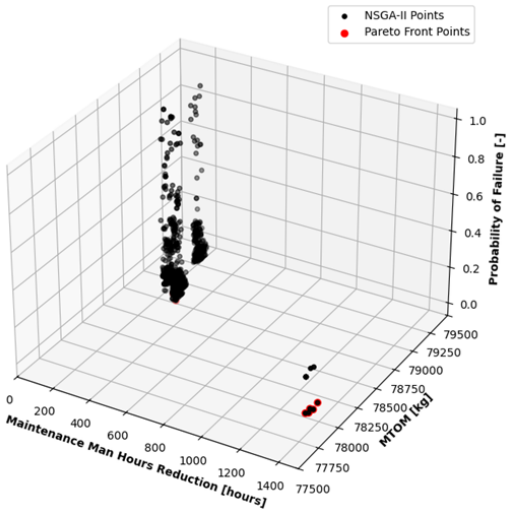
(a) 3D Pareto Front with a population size of 320 architectures and 5 generations



(b) 3D Pareto Front with a population size of 320 architectures and 10 generations



(c) 3D Pareto Front with a population size of 160 architectures and 10 generations



(d) 3D Pareto Front with a population size of 480 architectures and 10 generations

Fig. 5.14 Sensitivity analysis of the NSGA-II genetic algorithm parameters for the application case

not come as a surprise since providing a lower population size gives the algorithm less maneuverability to choose architectures. The Pareto Front moves to the MEA region due to a lack of AEA architectures. The last analysis consist of increasing the population size, as seen in figure 5.14d. In this case the number of AEA architectures is higher. However most of them are small variants of Pareto Points 1 and 2 (i.e., adding one generator or adding/removing one spoiler) and they do not provide more

information than the one found previously in the run with a population size of 320 and 20 generations.

Lastly, another interesting effect seen in figure 5.14 is the verticality of the results. This effect was already seen in the first run and is maintained during the rest, reassuring it. Also, the low number of AEA architectures is still observed on every run. In some of them the algorithms do not even manage to find feasible AEA solutions. This enhances the importance of running the optimization process with different algorithm settings and perform some kind of sensitivity analysis. This has an even higher impact if there is a filter in the process since the algorithms might not have enough points to know how to pass it. In this analysis the effect of the certification filter is quite interesting and provides useful insight to the designer, but it needs to be accounted for the optimization process or there is a risk of not achieving meaningful results (e.g., stopping the analysis in figure 5.14a not realizing there is another region in the solutions that has not yet been found).

Some posterior analysis to this one would include running the whole optimization framework with other algorithms. Since the design space is quite huge, this could also include surrogate-based optimization such as the one presented in [180]. This could potentially find other optimal solutions without needing to run such long runs again with different algorithms, just exploring further the design space.

5.4 Trade-off Analysis

This sections answers the question of how to trade different results that come from different disciplines. The designer, and decision maker in this case, needs a method to quantitatively evaluate the different solutions based on the results that come from completely different domains that might not have clear priority among them. The first step is to decide which are the most interesting points that need to be evaluated. This comes from the results that were obtained in the optimization process with small modifications. Table 5.9 shows such points. First point is the first point from the Pareto front. The second point is related to the point 2 of the Pareto front, however there is a more optimum point than this one, point 7 from the DOE has the same results in terms of MTOM and MMH, but a lower probability of failure. For this reason, point 2 from the Pareto front is substituted for point 7 from the DOE. These two points represent AEA architectures. Point 3 corresponds to a MEA architecture

which comes from point 3 of the Pareto front. Lastly, point 4 of the Pareto front is discarded, since it represents a non-symmetric architecture, as explained before. An extra point is added to the trade-off points, this represents the conventional baseline and serves as a reference to the other points.

Table 5.9 Trade-off points

Trade-off Point (#)	Concept	MTOM [kg]	Probability of Failure Ratio [-]	Reduction of MMH [hours]	Point Associated
1	AEA	77699	0.0279	1468	Pareto Point 1
2	AEA	77651	0.15	1471	DOE Point 7
3	MEA	78630	0.0017	349	Pareto Point 3
4	CONV	78967	1	0	DOE Point 1

The methodology for decision making support presented in chapter 3.7 is now utilized. For that the different attributes must be selected, which are the MTOM, MMH reduction and probability of failure of the architecture. The three attributes are condensed into one single metric, called value. Usually, two metrics are left for the decision making, one being the cost and the other the value. For this application case no cost is estimated, leaving only the value as a result. This does not present a problem since a 1-dimensional result can also express the wanted results.

The attributes are now defined and their corresponding weights and utility functions can be modelled. Usually, the decision maker would create different cases based on certain business models. However, no industry feedback was provided for this analysis and as a result, the different scenarios are directly modelled by the author. Just four different cases are presented in order to provide an example to the reader, but in reality a huge number of cases and analyses shall be done after the results are obtained.

The boundaries of the attributes are selected according to certain expectations. The MMH reduction goes between zero (for a conventional architecture) to 1502 (found for point 2 of the DOE in table 5.6). The probability of failure can vary from 0 (ideal extreme case in which an architecture never fails) and 1 (fails the same as the reference architecture of the conventional case). The maximum MTOM is found in point 4 of the DOE, showing 79085 kg, while the minimum comes from point 2 of such DOE, with 77511 kg.

Four different cases, or scenarios, are presented and come from the combination of two cases for the utility functions and other two for the weights. Regarding the

weights, the first combination of them is chosen to be one third of the total weight per attribute. This represents a simple and standard case in which all the different attributes have the same importance. Opposed to this one, another list of weights is suggested. This other option rates the MTOM the highest, with a fifty percent of the weight, and the MMH reduction as the second highest, with a forty percent. This leaves the probability of failure with the remaining ten percent. The reasoning behind these weights is that MTOM directly affects fuel burn and cost, so it should be prioritized. Maintenance also affects the direct operating cost, hence should have a lower but close weight to the MTOM. The probability of failure is rated less since the architecture is already certifiable and increasing the reliability can have benefits but without a so important impact as the two previous ones. The utility functions are build according to the boundaries previously presented. Two sets of functions are presented. The first one is just the simple case in which all the utilities are just linear functions from one boundary to the other one. The other set of utility functions shows different models that can be used. For MTOM and MMH reduction, a function composed by two different linear parts is used. The inflection point in both cases is the result that corresponds to trade-off point number 3 (MEA aircraft), giving it a 0.5 of the utility content for that point. This model, translated to general terms means that the decision maker is satisfied with half the rating when achieving the most promising MEA solution, with respect to the best result possible for that attribute. The probability of failure is modelled as an ellipse and can be understood as follows. If the probability of failure is almost zero, the decision maker is fully satisfied giving a rating of 1. At the same time, as long as the probability of failure is noticeably higher than the reference one (e.g., content between 0.1 and 0.8) the decision maker is almost as satisfied as in the other case. The rating only drops for contents that are too close to the reference one, meaning that an increase in reliability is not achieved. The comparison of such utility functions per each attribute is shown in figures 5.15, 5.16 and 5.17. This is just an example of utilities and the reader should not interpret that these utility functions are the only correct ones for the application case.

The combination of the two options of weights and utility functions results in four different cases. Each case provides one result of the value per each trade-off point. Case 1 and 2: use the linear utility functions (left side of figures 5.15, 5.16 and 5.17) and different weights. Case 3 and 4: use the more complex utility functions (right side of figures 5.15, 5.16 and 5.17) and different weights. Table 5.10 shows the results of the trade-off analysis using the value model. The trade-off points inside

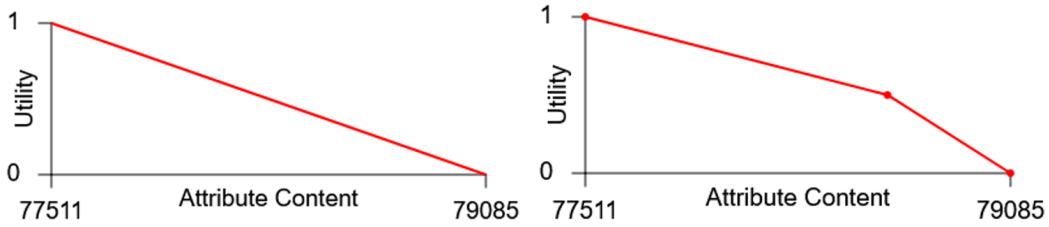


Fig. 5.15 Utility functions for the MTOM attribute

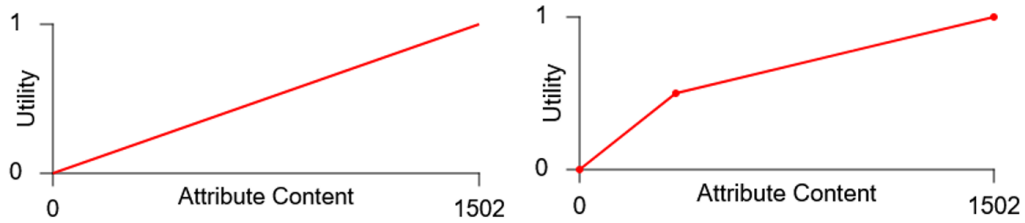


Fig. 5.16 Utility functions for the MMH attribute

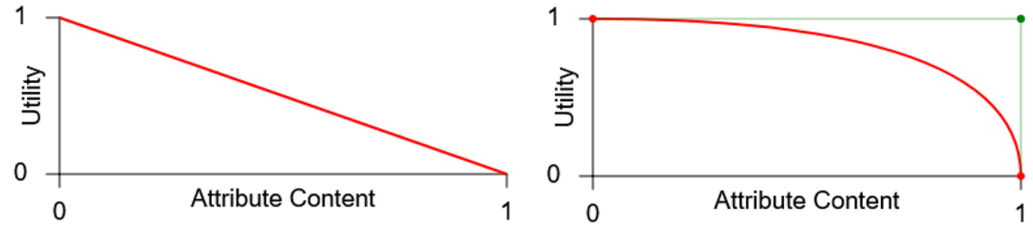


Fig. 5.17 Utility functions for the probability of failure attribute

each case are ordered by the value from the highest to the lowest. It can be seen how point 1 is the optimal solution for case 1, however the rest of cases switch this to case 2. Case 4 is always at last as expected.

Table 5.10 Trade-off results for the four different cases

Case 1		**MTOM 33.3%**		**MMH 33.3%**		**Prob. Failure 33.3%**	
#	Value	Content	Utility	Content	Utility	Content	Utility
1	0.9434	77699	0.88	1468	0.98	0.0279	0.97
2	0.9134	77651	0.91	1471	0.98	0.15	0.85
3	0.5066	78630	0.29	349	0.23	0.0017	$\sim$1
4	0.0250	78967	0.07	0	0	1	0
Case 2		**MTOM 50%**		**MMH 40%**		**Prob. Failure 10%**	
#	Value	Content	Utility	Content	Utility	Content	Utility
2	0.9322	77651	0.91	1471	0.98	0.15	0.85
1	0.9285	77699	0.88	1468	0.98	0.0279	0.97
3	0.3373	78630	0.29	349	0.23	0.0017	$\sim$1
4	0.0375	78967	0.07	0	0	1	0
Case 3		**MTOM 33.3%**		**MMH 33.3%**		**Prob. Failure 33.3%**	
#	Value	Content	Utility	Content	Utility	Content	Utility
2	0.9726	77651	0.94	1471	0.99	0.15	0.99
1	0.9670	77699	0.92	1468	0.99	0.0279	$\sim$1
3	0.6665	78630	0.50	349	0.5	0.0017	$\sim$1
4	0.0432	78967	0.13	0	0	1	0
Case 4		**MTOM 50%**		**MMH 40%**		**Prob. Failure 10%**	
#	Value	Content	Utility	Content	Utility	Content	Utility
2	0.9627	77651	0.94	1471	0.99	0.15	0.99
1	0.9521	77699	0.92	1468	0.99	0.0279	$\sim$1
3	0.5498	78630	0.50	349	0.5	0.0017	$\sim$1
4	0.0648	78967	0.13	0	0	1	0

It is interesting to see how the different value results change based on the weights and utilities that the designer modelled. It does not come as a surprise that the two AEA aircraft have the highest value ratings among all cases, and that the conventional case always has the lowest. Figure 5.18 shows the value result per case and per trade-off point visually. It is noticeable how point 4 has always a low value regardless of the case, since the results are not promising when compared with the other technologies. Point 3 represents a MEA architecture, this point is interesting since it has a lot of variability depending on the priorities and preferences of the decision maker. As a result this point can vary quite a lot based on the case. Both AEA architectures (point 1 and 2) are always at the top of the value score and do not have a lot of variability. These two points always perform well when compared to the rest. The most interesting effect is that point 1 and 2 switch places as the one with highest value when moving from case 1 to case 2. As a result, the final decision on which point is more optimum depends fully on the designer perspective and there is not one true and only best point.

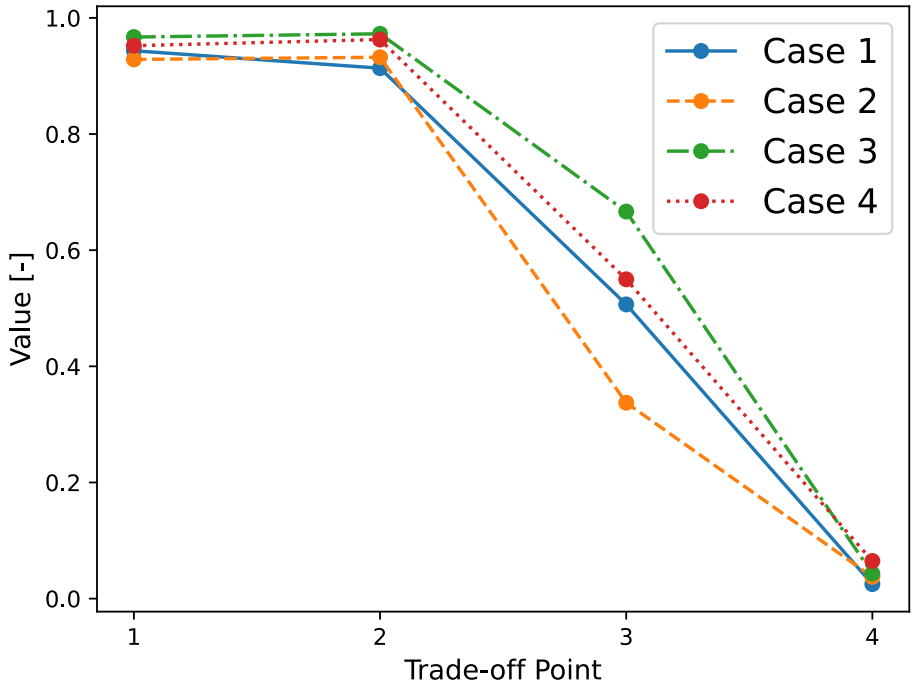


Fig. 5.18 Value result for each trade-off point for each of the proposed cases

This example, although simple for only four points, shows the potential of the methodology. The larger the number of points, the more useful it gets since it allows to filter some points (e.g., discard everything below a certain value margin). Furthermore, in later stages of design when there is more information available, additions can be made. More attributes can be added per discipline, instead of having just one. This would allow a more complex trade-off analysis with a higher number of points, but the simplicity of interpreting the result would be maintained. Lastly, other disciplines or life cycles stages could be added after to the analysis. Some examples could be manufacturing, supply change, risk... Adding these disciplines could for instance raise the value of the MEA architectures, since they are cheaper and less risky to manufacture. The trade-off allows the designer to identify the most interesting solutions and see how they change based on the different cases.

Chapter 6

Conclusions

On-board systems are crucial enablers for future aircraft, and their evaluation is essential for understanding the viability of innovative concepts. Performance is usually the first discipline to be assessed when determining the potential impact of a new concept. However, it's also important to simultaneously consider other disciplines from different life cycle stages of the aircraft (such as maintenance and manufacturing) during concept development, gaining a comprehensive understanding of the overall impact of new technologies and driving the architecture design of the system. This analysis focuses on integrating certification and maintenance into the evaluation process, leaving other life cycle stages for future studies.

A significant challenge in on-board system architecture design is the large number of potential solutions that can fulfill a specific function, leading to an extensive design space that cannot be fully explored. To manage this complexity, filtering the architectures and using optimization algorithms is necessary to narrow down the design space to the most viable options. This study proposes a methodology for assessing performance, certification, and maintenance aspects of on-board system architectures during the preliminary design phase. Performance is key to evaluate the potential of a new architecture, while maintenance considerations provide important insights into the maintainability of innovative concepts that may not have been previously assessed. Certification plays a crucial role in eliminating architectures that do not meet regulatory requirements, serving as a filter that saves computational resources by allowing only potentially certifiable architectures to be evaluated. Automation is essential to effectively explore the design space and link these analyses

to optimization algorithms, helping to identify the most promising solutions and highlight key trade-offs. This methodology aims to assist and support decision-making in the multi-disciplinary and multi-objective design process of on-board system architectures.

An application case is shown in this dissertation, applying the proposed methodology to new high-lift devices of a commercial transport aircraft. The results demonstrate that a new architecture based on Electro-Mechanical Actuators (EMAs) offers significant potential when compared to the state-of-the-art one. This is mainly due to the fact that the new concept eliminates mechanical connections among actuators, allowing each surface to be deployed independently. This leads to a lower number of components and less number of mechanical inefficiencies, which increases the performance and lowers the mass of the system. Reliability is also improved since the surfaces can now operate in parallel instead of in series, ensuring that the mission can still be completed even if one surface fails. This capability could significantly increase the operational reliability. However this improvement is dependent on the assumption that jamming (i.e., one failure mode of EMAs) issues are mitigated and are not critical. The new architecture also shows benefits in terms of maintenance, achieving a reduction in the total maintenance man hours. This methodology finally allows to have quantitative estimations for this innovative architecture that were not yet achieved in literature, establishing a proper foundation for future higher-fidelity studies.

Another application case is presented to demonstrate another capability of the methodology, which is the filtering of architectures from a huge design space based on rules extracted from the certification specifications. The application focuses on the roll control function of the flight control system of the Airbus A320neo. The design space is created using a system architecting approach, resulting in billions of possible architectures, emphasizing the need for automation and filtering. This design space is linked to a framework that automatically filters out architectures that do not meet the minimum certification requirements. The remaining architectures are then evaluated for performance using various tools. First, a specialized tool for flight control system sizing estimates the mass and power requirements of each architecture. Then, the rest of the on-board systems are sized and integrated into an aircraft baseline. Finally, maintenance considerations are assessed by analyzing the reduction in Maintenance Man-Hours (MMH) compared to the conventional A320neo. The results are then fed into optimization algorithms, which explore the most promising areas of the design

space, identifying the most viable architectures without the need to evaluate every single one. The results reveal notable trade-offs. In general, all-electric architectures (AEA) show significant potential in terms of performance and maintenance but require additional redundancies to meet certification standards compared to the baseline aircraft. Most AEA designs do not pass the certification filter; however, those that do exhibit considerable improvements and potential. MEA (More Electric Aircraft) architectures show slight benefits over the conventional design, serving as a practical middle ground between conventional and all-electric concepts. It is shown that transitioning to fully all-electric designs results in substantial performance enhancements. Additionally, the certification filter plays a crucial role in eliminating the majority of the architectures, leaving only a reduced number of promising solutions. This greatly helps the decision-making process by narrowing down feasible solutions and presenting valuable trade-offs and conclusions for the designer.

The research questions posed at the beginning of this manuscript in section 2.4 are now answered:

How can new on-board system architectures be identified by considering simultaneously performance, certification and maintenance aspects during early stages of design?

Considering simultaneously performance, certification and maintenance aspects is important since it can provide a better understanding of the overall impact of a new architecture. Using the methodology here proposed allows to account for these three disciplines at the same time. This leads to interesting results and trade-offs that are not available if only performance is assessed. This question is furthered answered in the following more-specific questions.

How to determine whether innovative and automatically-generated on-board system architectures are not certifiable?

This can be achieved by automatically generating the reliability block diagram (RBD) of the architectures following the proposed methodology. A good practice is to evaluate first the reliability of a baseline architecture that comes from a conventional case and later compare the rest of innovative architectures to this one. This approach allows to filter out all the architectures that have a lower reliability than the state of the art one, ensuring that only potentially certifiable architectures are evaluated.

Other certification filters can be applied such as checking if there are back-up systems providing power to the users in case of failure of all the engines. Another one is checking the single failure condition that says that the single failure of a component can never make the whole system fail. All these filtering conditions are extracted from the certification specifications.

How to reduce and filter the huge design spaces that characterize the on-board system architectures design?

This issue is solved together with the previous question. The reliability of an architecture can determine if that solution has potential in terms of certification or not. The other two certification checks (i.e., single failure and back-up systems) also affect it. All the architectures that do not reach a reliability equal or higher than the baseline one can be filtered, as well as all those who do not pass the back-up and/or single failure checks. This avoids the execution and evaluation of these architectures, which saves computational times and reduces the possible solutions. The application case shown in this study presents and example with a run of 6400 points, from which approximately a 70% are filtered out. This shows the strong potential of this approach. Optimization algorithms are used in conjunction with this filter in order to converge faster to the optimum solutions of the design space.

How to select the best architectures and perform trade-off analysis with results that come from different life cycle stages (i.e., performance, maintenance and certification)?

The best solutions can be traded after the optimum ones have been found in the Pareto Front after running the multi-objective optimization problem. These architectures represent the best possible solutions regarding the three disciplines. The Multi Attribute Utility (MAU) theory is used in order to select the best architecture according to certain scenarios. These scenarios are created by the decision maker (i.e., designer and stakeholders) depending on their perspective, opinion and expectations. Different scenarios lead to diverse results which manage to finally select the best architecture (or best architectures) from the optimum ones found with the optimization algorithms.

To conclude, the main outcomes and original contributions of this manuscript include the identification of the RBD technique as a feasible and standardized method to successfully assess certification aspects of conventional and innovative on-board system architectures during conceptual design, when other methods cannot still be used due to a lack of information. Some typical architectural configurations of aircraft systems have been identified and their RBDs have been shown and solved, providing the exact equations and simplifications to estimate the probability of failure of such. The RBD method has been automated through the development of new algorithms, this allows for the automation of the process achieving a huge filtering of OBS architectures. Another important point includes the identification of the requirements contained in the certification specifications that have a high impact in the OBS at an architectural level, these are listed and presented. All the previous points have led to the development of ACOBS, a tool that handles the automated preliminary certification of OBS architectures. Improvements have also been made in the maintenance discipline, providing a method that combines different techniques into a surrogate model that allows to assess the impact at a component level. The combination of the previous with performance results lead to interesting trade-off analyses, for which a methodology is also suggested and presented.

References

[1] Aldo Boglietti, Andrea Cavagnino, Alberto Tenconi, and Silvio Vaschetto. The safety critical electric machines and drives in the more electric aircraft: A survey. In *2009 35th Annual Conference of IEEE Industrial Electronics*, pages 2587–2594. IEEE, 2009.

[2] Jasper Faber, Julius Király, David Lee, Bethan Owen, and Aoife O'Leary. Potential for reducing aviation non-co2 emissions through cleaner jet fuel. *CE-Del*, 22:022, 2022.

[3] Marco Fioriti, Carlos Cabaleiro De La Hoz, Thierry Lefebvre, Pierluigi Della Vecchia, Massimo Mandorino, Susan Liscouet-Hanke, Andrew K Jeyaraj, Giuseppa Donelli, and Aidan Jungo. Multidisciplinary design of a more electric regional aircraft including certification constraints. In *AIAA AVIATION 2022 Forum*, page 3932, 2022.

[4] Oliver Bertram, Florian Jäger, Viola Voth, and Jan Rosenberg. Onboard system design challenges of evtols, 2021.

[5] Hannu Jaakkola and Bernhard Thalheim. Architecture-driven modelling methodologies. In *Information Modelling and Knowledge Bases XXII*, pages 97–116. IOS Press, 2011.

[6] Carlos Cabaleiro, Jasper H Bussemaker, Marco Fioriti, Luca Boggero, Pier Davide Ciampa, and Bjoern Nagel. Environmental and flight control system architecture optimization from a family concept design perspective. In *AIAA Aviation 2020 Forum*, page 3113, 2020.

[7] Edward Crawley, Bruce Cameron, and Daniel Selva. *System architecture: strategy and product development for complex systems*. Prentice Hall Press, 2015.

[8] Jasper H Bussemaker, Pier Davide Ciampa, and Bjoern Nagel. System architecture design space exploration: An approach to modeling and optimization. In *AIAA Aviation 2020 Forum*, page 3172, 2020.

[9] Jaroslaw Sobieszczanski-Sobieski, Alan Morris, and Michel Van Tooren. *Multidisciplinary design optimization supported by knowledge based engineering*. John Wiley & Sons, 2015.

[10] INCOSE. *INCOSE systems engineering handbook*. John Wiley & Sons, 2023.

[11] David D Walden et al. Systems engineering handbook: A guide for system life cycle processes and activities. *(No Title)*, 2015.

[12] International Organization for Standardization. Iso 14040:2006 - environmental management, life cycle assessment, principles and framework, 2006.

[13] D Yu Strelets, SA Serebryansky, and MV Shkurin. Concept of creation of a digital twin in the uniform information environment of product life cycle. In *2020 13th International Conference" Management of large-scale system development"(MLSD)*, pages 1–4. IEEE, 2020.

[14] Roland Rosen, Georg Von Wichert, George Lo, and Kurt D Bettenhausen. About the importance of autonomy and digital twins for the future of manufacturing. *Ifac-papersonline*, 48(3):567–572, 2015.

[15] Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58:346–361, 2021.

[16] Thierry Pardessus. Concurrent engineering development and practices for aircraft design at airbus, 2004.

[17] Biren Prasad. *Concurrent engineering fundamentals*, volume 1. Prentice Hall PTR NJ, 1996.

[18] Kent Richard McCord. *Managing the integration problem in concurrent engineering*. PhD thesis, Massachusetts Institute of Technology, 1993.

[19] Ian Moir and Allan Seabridge. *Aircraft Systems: Mechanical, electrical, and avionics subsystems integration*, volume 52. John Wiley & Sons, 2011.

[20] Imon Chakraborty. *Subsystem architecture sizing and analysis for aircraft conceptual design*. PhD thesis, Georgia Institute of Technology, 2015.

[21] DI Smagin, RS Savelev, and AA Satin. Methods for the design of modern on-board systems of advanced aircraft. In *2019 IEEE 10th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pages 97–101. IEEE, 2019.

[22] Luca Boggero, Marco Fioriti, Sabrina Corpino, and Pier Davide Ciampa. On-board systems preliminary sizing in an overall aircraft design environment. In *17th AIAA aviation technology, integration, and operations conference*, page 3065, 2017.

[23] Marco Fioriti, Luca Boggero, Sabrina Corpino, Alik Isyanov, Artur Mirzoyan, Riccardo Lombardi, and Roberto D'Ippolito. Automated selection of the optimal on-board systems architecture within mdo collaborative environment. In *18th AIAA/ISSMO Multidisciplinary analysis and optimization conference*, page 3150, 2017.

[24] Marco Fioriti, Pierluigi Della Vecchia, and Giuseppa Donelli. Effect of progressive integration of on-board systems design discipline in an mda framework for aircraft design with different level of systems electrification. *Aerospace*, 9(3):161, 2022.

[25] DA Prosvirin and VP Kharchenko. Model-based solution and software engineering environment for uav critical onboard applications. In *2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, pages 312–315. IEEE, 2015.

[26] Susan Liscouet-Hanke. *A model-based methodology for integrated preliminary sizing and analysis of aircraft power system architectures*. PhD thesis, Institut National des Sciences Appliquées de Toulouse, 2008.

[27] Nicole Viola, Roberta Fusaro, Bayindir Saracoglu, Christophe Schram, Volker Grewe, Jan Martinez, Marco Marini, Santiago Hernandez, Karel Lammers, Axel Vincent, et al. Main challenges and goals of the h2020 stratofly project. *Aerotecnica Missili & Spazio*, 100(2):95–110, 2021.

[28] Mark Davies. *Standard handbook for aeronautical and astronautical engineers*. McGraw-Hill Education, 2003.

[29] Taewoo Nam, Danielle Soban, and Dimitri Mavris. Power based sizing method for aircraft consuming unconventional energy. In *43rd AIAA aerospace sciences meeting and exhibit*, page 818, 2005.

[30] KA Friedrich, Josef Kallo, Johannes Schirmer, and Gerrit Schmitthals. Fuel cell systems for aircraft application. *ECS transactions*, 25(1):193, 2009.

[31] Jiawei Chen, Chengjun Wang, and Jie Chen. Investigation on the selection of electric power system architecture for future more electric aircraft. *IEEE Transactions on Transportation Electrification*, 4(2):563–576, 2018.

[32] Kai Ni, Yongjiang Liu, Zhanbo Mei, Tianhao Wu, Yihua Hu, Huiqing Wen, and Yangang Wang. Electrical and electronic technologies in more-electric aircraft: A review. *IEEE Access*, 7:76145–76166, 2019.

[33] Simone Dell'Anna. Concept development for maintenance cost estimation of more-electric aircraft on-board system architectures. Master's thesis, Politecnico di Torino, Turin, Italy, 2023.

[34] Marco Fioriti, Flavio Di Fede, et al. A design model for electric environmental control system in aircraft conceptual and preliminary design. *INTERNATIONAL REVIEW OF AEROSPACE ENGINEERING*, 2023.

[35] Christian Müller and Dieter Scholz. The vapor compression cycle in aircraft air-conditioning systems, 2004.

[36] Roman Savelev, Denis Smagin, and Tatiana Zyazeva. Methodology for analyzing competing options for air management systems at the stage of conceptual design of civil aircraft's complex of onboard systems. In *MATEC Web of Conferences*, volume 304, page 04015. EDP Sciences, 2019.

[37] Qinglin Ma. *Aircraft icing and thermo-mechanical expulsion de-icing technology*. PhD thesis, Cranfield University, School of Engineering, 2010.

[38] Stefan Kazula and Klaus Höschler. Ice detection and protection systems for circular variable nacelle inlet concepts. *CEAS Aeronautical Journal*, 11(1):229–248, 2020.

[39] Brigitte Boden, Matheus Henrique Padilha, Tim Burschyk, Carlos Cabaleiro de la Hoz, Erwin Moerland, and Marco Fioriti. Automating the verification of geometric requirements for aircraft fuel systems using knowledge-based engineering. In *34th Congress of the International Council of the Aeronautical Sciences Florence, Italy*, 2024.

[40] Mark W Maier. *The art of systems architecting*. CRC press, 2009.

[41] Paul C Clements. A survey of architecture description languages. In *Proceedings of the 8th international workshop on software specification and design*, pages 16–25. IEEE, 1996.

[42] Eberhardt Rechtin. The art of systems architecting. *IEEE Spectrum*, 29(10):66–69, 1992.

[43] Marco Fioriti, Luca Boggero, Sabrina Corpino, Prajwal Shiva Prakasha, Pier Davide Ciampa, and Björn Nagel. The effect of sub-systems design parameters on preliminary aircraft design in a multidisciplinary design environment. *Transportation Research Procedia*, 29:135–145, 2018.

[44] Guan Qiao, Geng Liu, Zhenghong Shi, Yawen Wang, Shangjun Ma, and Teik C Lim. A review of electromechanical actuators for more/all electric aircraft systems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 232(22):4128–4151, 2018.

[45] Xu Han, Tatiana Minav, Mingkang Wang, Yongling Fu, and Matti Pietola. Thermal coupling simulation of electro-hydrostatic actuator subjected to critical temperature conditions. *International Journal of Fluid Power*, 23(3):379–394, 2022.

[46] Mirko Mazzoleni, Gianpietro Di Rito, Fabio Previdi, et al. *Electro-Mechanical Actuators for the More Electric Aircraft*, volume 2021. Springer, 2021.

[47] Jianming Li, Zhiyuan Yu, Yuping Huang, and Zhiguo Li. A review of electromechanical actuation system for more electric aircraft. In *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, pages 490–497. IEEE, 2016.

[48] M Budinger, J Liscouët, S Orieux, and J-Ch Maré. Automated preliminary sizing of electromechanical actuator architectures. *variations*, 3(2), 2008.

[49] REJ Quigley. More electric aircraft. In *Proceedings Eighth Annual Applied Power Electronics Conference and Exposition,*, pages 906–911. IEEE, 1993.

[50] Pat Wheeler and Sergei Bozhko. The more electric aircraft: Technology and challenges. *IEEE Electrification Magazine*, 2(4):6–12, 2014.

[51] RT Naayagi. A review of more electric aircraft technology. In *2013 international conference on energy efficient technologies for sustainability*, pages 750–753. IEEE, 2013.

[52] Babak Rahrovi and Mehrdad Ehsani. A review of the more electric aircraft power electronics. In *2019 IEEE Texas Power and Energy Conference (TPEC)*, pages 1–6. IEEE, 2019.

[53] Pat Wheeler. Technology for the more and all electric aircraft of the future. In *2016 IEEE International Conference on Automatica (ICA-ACCA)*, pages 1–5. IEEE, 2016.

[54] QI Haitao, FU Yongling, QI Xiaoye, and LANG Yan. Architecture optimization of more electric aircraft actuation system. *Chinese Journal of Aeronautics*, 24(4):506–513. Elsevier, 2011.

[55] David Wayne Jackson. *Robust aircraft subsystem conceptual architecting*. PhD thesis, Georgia Institute of Technology, 2013.

[56] JA Rosero, JA Ortega, E Aldabas, and LARL Romeral. Moving towards a more electric aircraft. *IEEE Aerospace and Electronic Systems Magazine*, 22(3):3–9, 2007.

[57] Bulent Sarlioglu and Casey T Morris. More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft. *IEEE transactions on Transportation Electrification*, 1(1):54–64, 2015.

[58] RI Jones. The more electric aircraft—assessing the benefits. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 216(5):259–269, 2002.

[59] Joseph A Weimer. Electrical power technology for the more electric aircraft. In *[1993 Proceedings] AIAA/IEEE Digital Avionics Systems Conference*, pages 445–450. IEEE, 2005.

[60] Eugenio Brusa. Digital twin: Towards the integration between system design and rams assessment through the model-based systems engineering. *IEEE Systems Journal*, 2020.

[61] Nils Kuelper, Jasmin Broehan, Thimo Bielsky, and Frank Thielecke. Systems architecting assistant (sara)-enabling a seamless process chain from requirements to overall systems design. In *33rd Congress of the International Council of the Aeronautical Sciences, Stockholm, Sweden*, 2022.

[62] David Manuel Judt and Craig Lawson. Development of an automated aircraft subsystem architecture generation and analysis tool. *Engineering Computations*, 2016.

[63] Imon Chakraborty, Dimitri N Mavris, Mathias Emeneth, and Alexander Schneegans. An integrated approach to vehicle and subsystem sizing and analysis for novel subsystem architectures. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(3):496–514, 2016.

[64] Johann Bals, Yang Ji, Martin R Kuhn, and Christian Schallert. Model based design and integration of more electric aircraft systems using modelica. In *Moet forum at European power electronics conference and exhibition*, pages 1–12, 2009.

[65] Michael James Armstrong. *A process for function based architecture definition and modeling*. PhD thesis, Georgia Institute of Technology, 2008.

[66] Cyril De Tenorio. *Methods for collaborative conceptual design of aircraft power architectures*. Georgia Institute of Technology, 2010.

[67] Imon Chakraborty, Dimitri N Mavris, Mathias Emeneth, and Alexander Schneegans. A methodology for vehicle and mission level comparison of more electric aircraft subsystem solutions: Application to the flight control actuation system. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 229(6):1088–1102, 2015.

[68] AGILE 4.0 H2020 research project. Towards cyber-physical collaborative aircraft development. https://www.agile4.eu/, 2023.

[69] Marc Jünemann, Frank Thielecke, F Peter, Mirko Hornung, F Schültke, and Eike Stumpf. *Methodology for design and evaluation of more electric aircraft systems architectures within the AVACON project*. Deutsche Gesellschaft fuer Luft-und Raumfahrt-Lilienthal-Oberth eV, 2019.

[70] Sergio Chiesa, Giovanni Antonio Di Meo, Marco Fioriti, Giovanni Medici, and Nicole Viola. Astrid–aircraft on board systems sizing and trade-off analysis in initial design. *Research and Education in Aircraft Design–READ 2012*, 2012.

[71] Douglas P Wells, Bryce L Horvath, and Linwood A McCullers. The flight optimization system weights estimation method. Technical report, NASA, 2017.

[72] Egbert Torenbeek. Airplane design, 1982.

[73] Jan Roskam. Airplane design, part i: Preliminary sizing of airplanes, design. *Analysis and Research Corporation, Lawrence, Kansas*, 1997.

[74] Daniel Raymer. *Aircraft design: a conceptual approach*. American Institute of Aeronautics and Astronautics, Inc., 2012.

[75] Imon Chakraborty and Dimitri N Mavris. Integrated assessment of aircraft and novel subsystem architectures in early design. *Journal of Aircraft*, 54(4):1268–1282, 2017.

[76] Imon Chakraborty, David R Trawick, Dimitri N Mavris, Mathias Emeneth, and Alexander Schneegans. A requirements-driven methodology for integrating subsystem architecture sizing and analysis into the conceptual aircraft design phase. In *14th AIAA Aviation Technology, Integration, and Operations Conference*, page 3012, 2014.

[77] Marco Fioriti, Flavio Di Fede, et al. A design model for electric environmental control system in aircraft conceptual and preliminary design. *INTERNATIONAL REVIEW OF AEROSPACE ENGINEERING*, 2023.

[78] Mavris DN, LL Phan, and E Garcia. Formulation and implementation of an aircraft–system–subsystem interrelationship model for technology evaluation. *Environment*, 21(30):35–36, 2006.

[79] Hamdi Ercan and Mustafa Akın. Performance analysis of electrical flight control actuation system in a commercial transport aircraft. *Aircraft Engineering and Aerospace Technology*, 94(10):1671–1683, 2022.

[80] CS25 EASA. Certification specifications for large aeroplanes, 2009.

[81] CS23 EASA. Certification specifications for normal, utility, aerobatic, and commuter category aeroplanes, 2009.

[82] SAE Aerospace. Arp4754: Certification considerations for highly-integrated or complex aircraft systems, sae systems integration requirements task group as-1c, asd., rev. *A, Society of Automotive Engineers, Inc*, 2010.

[83] SAE Aerospace. Arp4761. guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. *SAE International, Warrendale*, 1996.

[84] Francesco Bruno. *A Model-Based RAMS Estimation Methodology for Innovative Aircraft on-board Systems developed in a MDO Environment*. MSc Thesis, Politecnico di Torino, 2020.

[85] Chenling Li, KaiCheng Li, Tao Tang, JiDong Lv, and Ling Huang. Model-based generation of safety test-cases for onboard systems. In *2013 IEEE International Conference on Intelligent Rail Transportation Proceedings*, pages 191–196. IEEE, 2013.

[86] Felipe Augusto Sviaghin Ferri, Leonardo Ramos Rodrigues, João Paulo Pordeus Gomes, Ivo Paixão de Medeiros, Roberto Kawakami Harrop Galvão, and Cairo Lúcio Nascimento. Combining phm information and system architecture to support aircraft maintenance planning. In *2013 IEEE International Systems Conference (SysCon)*, pages 60–65. IEEE, 2013.

[87] Li Jun and Xu Huibin. Reliability analysis of aircraft equipment based on fmeca method. *Physics Procedia*, 25:1816–1822, 2012.

[88] Matheus Henrique Padilha. Enabling fuel system component placement on the basis of geometric and certification requirements within a semantic knowledge-based engineering framework. Master's thesis, Politecnico di Torino, 2023.

[89] Mark S Saglimbene. Reliability analysis techniques: How they relate to aircraft certification. In *2009 Annual Reliability and Maintainability Symposium*, pages 218–222. IEEE, 2009.

[90] Michael James Armstrong. *Identification of emergent off-nominal operational requirements during conceptual architecting of the more electric aircraft*. Georgia Institute of Technology, 2011.

[91] Andrew K Jeyaraj and Susan Liscouët-Hanke. A safety-focused system architecting framework for the conceptual design of aircraft systems. *Aerospace*, 9(12):791, 2022.

[92] Jeremy Ross. Complex system reliability analysis using a model-based shared systems simulation. In *INCOSE International Symposium*, volume 33 (1), pages 212–228. Wiley Online Library, 2023.

[93] Nataliya Yakymets, Matthieu Perin, and Agnes Lanusse. Model-driven multi-level safety analysis of critical systems. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings*, pages 570–577. IEEE, 2015.

[94] PC Berri, MDL Dalla Vedova, and P Maggiore. Enhanced hybrid prognostic approach applied to aircraft on-board electromechanical actuators affected by progressive faults. In *Safety and Reliability–Safe Societies in a Changing World*, pages 1077–1083. CRC Press, 2018.

[95] Sergio Chiesa. *Affidabilità, sicurezza e manutenzione nel progetto dei sistemi*. CLUT, 2008.

[96] Saideep Nannapaneni, Abhishek Dubey, Sherif Abdelwahed, Sankaran Mahadevan, Sandeep Neema, and Ted Bapty. Mission-based reliability prediction in component-based systems. *International Journal of Prognostics and Health Management*, 7(1), 2016.

[97] Marvin Rausand and Arnljot Hoyland. *System reliability theory: models, statistical methods, and applications*, volume 396. John Wiley & Sons, 2003.

[98] Jean-Pierre Signoret and Alain Leroy. *Reliability Assessment of Safety and Production Systems: Analysis, Modelling, Calculations and Case Studies*. Springer Nature, 2021.

[99] IEC. 61078, analysis techniques for dependability-reliability block diagram and boolean methods. *Geneva: IEC*, 2006.

[100] Paul A Tobias and David Trindade. *Applied reliability*. CRC Press, 2011.

[101] Timothy Coyle, Robert G Arno, and Peyton S Hale. Application of the minimal cut set reliability analysis methodology to the gold book standard network. In *IEEE Technical Conference Industrial and Commerical Power Systems*, pages 82–93. IEEE, 2002.

[102] Christian Schallert. *Integrated safety and reliability analysis methods for aircraft system development using multi-domain object-oriented models*. PhD thesis, Technischen Universität Berlin, 2016.

[103] Riko Bornholdt, Tobias Kreitz, and Frank Thielecke. Function-driven design and evaluation of innovative flight controls and power system architectures. Technical report, SAE Technical Paper, 2015.

[104] Yang Jianzhong and Zhang Julian. Application research of markov in flight control system safety analysis. *Procedia Engineering*, 17:515–520, 2011.

[105] Xiao Liu, Zili Wang, Yi Ren, and Linlin Liu. Modeling method of sysml-based reliability block diagram. In *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, pages 206–209. IEEE, 2013.

[106] Philipp Limbourg and Hans-Dieter Kochs. Multi-objective optimization of generalized reliability design problems using feature models—a concept for early design stages. *Reliability Engineering & System Safety*, 93(6):815–828, 2008.

[107] Xiao Xiong and Ping Zhang. Reliability analysis of flight control system for large civil aircraft with imperfect fault coverage model. In *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing)*, pages 1–5. IEEE, 2012.

[108] Chanseok Park. A note on the existence of the location parameter estimate of the three-parameter weibull model using the weibull plot. *Mathematical problems in engineering*, 2018(1):6056975, 2018.

[109] Albert F Lowas III and Frank W Ciarallo. Reliability and operations: Keys to lumpy aircraft spare parts demands. *Journal of Air Transport Management*, 50:30–40, 2016.

[110] Feliciano Scarcelli. Rams-and mass-based optimization of aircraft on-board system architectures during preliminary aircraft design. Master's thesis, Politecnico di Torino, Turin, Italy, 2023.

[111] Kouroush Jenab and K Rashidi. Operational reliability assessment of an aircraft environmental control system. *Reliability Engineering & System Safety*, 94(2):456–462, 2009.

[112] M Sghairi, A De Bonneval, Y Crouzet, J-J Aubert, and P Brot. Challenges in building fault-tolerant flight control system for a civil aircraft. *IAENG International Journal of Computer Science*, 35(4), 2008.

[113] Manju Nanda and Shrisha Rao. A formal method approach to analyze the design of aircraft flight control systems. In *2009 3rd Annual IEEE Systems Conference*, pages 64–69. IEEE, 2009.

[114] L Xue. Actuation technology for flight control system on civil aircraft. Master's thesis, Cranfield University, 2009.

[115] Fredrik Roos and Sture Lindah. Distribution system component failure rates and repair times–an overview. In *Nordic distribution and asset management conference*, pages 23–24. Citeseer, 2004.

[116] Qianwen Xu, Yan Xu, Pengfei Tu, Tianyang Zhao, and Peng Wang. Systematic reliability modeling and evaluation for on-board power systems of more electric aircrafts. *IEEE Transactions on Power Systems*, 34(4):3264–3273, 2019.

[117] Marcantonio Catelani, Lorenzo Ciani, and Matteo Venzi. Rbd model-based approach for reliability assessment in complex systems. *IEEE Systems Journal*, 13(3):2089–2097, 2018.

[118] Leonard MacLean, Alex Richman, and Mark Hudak. Failure rates for aging aircraft. *Safety*, 4(1):7, 2018.

[119] Andrew K. Jeyaraj, Jasper H. Bussemaker, Susan Liscouët-Hanke, and Luca Boggero. Systems architecting: a practical example of design space modeling and safety-based filtering within the agile 4.0 project. In *the 33rd congress of the international council of the aeronautical sciences. ICAS*, 2022.

[120] CH Friend. *Aircraft maintenance management*. Longman Publishing Group, 1992.

[121] Randy Heisey. 717-200: low maintenance costs and high dispatch reliability. *Aero Mag*, 19:18–29, 2002.

[122] Huawei Wang, Jun Gao, and Haiqiao Wu. Direct maintenance cost prediction of civil aircraft. *Aircraft Engineering and Aerospace Technology: An International Journal*, 86(5):406–414, 2014.

[123] Helen L Lockett and K Arvanitopoulos-Darginis. An automated maintainability prediction tool integrated with computer aided design. *Procedia Cirp*, 60:440–445, 2017.

[124] Kossi Tiassou, Karama Kanoun, Mohamed Kaâniche, Christel Seguin, and Chris Papadopoulos. Aircraft operational reliability—a model-based approach and a case study. *Reliability Engineering & System Safety*, 120:163–176, 2013.

[125] MIL-HDBK-472. Military standardization handbook: Maintainability prediction, 1966.

[126] MIL-HDBK-470A. Military standardization handbook: Designing and developing maintainable products and systems, 1997.

[127] Joseph A Caroli and George W Lyne. A testability-dependent maintainability-prediction technique. In *Annual Reliability and Maintainability Symposium 1992 Proceedings*, pages 223–227. IEEE, 1992.

[128] Kossi Tiassou. *Aircraft operational reliability-A Model-based approach and case studies*. PhD thesis, INSA de Toulouse, 2013.

[129] Jerry F Lipa. A modern maintainability prediction technique. *IEEE Transactions on Reliability*, 30(3):218–221, 1981.

[130] Geoff Burrows, Christine A Brown, Trevor W Thom, John MC King, and John Frearson. Real-time cost management of aircraft operations. *Management Accounting Research*, 12(3):281–298, 2001.

[131] Niccolo Ninotta. Development of a maintenance man hours estimating relationship during preliminary design of more and all electric aircraft. Master's thesis, Politecnico di Torino, Turin, Italy, 2024.

[132] Edy Suwondo. Lcc-ops: Life cycle cost application in aircraft operations, 2007.

[133] Tseko Mofokeng, Paul T Mativenga, and Annlizé Marnewick. Analysis of aircraft maintenance processes and cost. *Procedia CIRP*, 90:467–472, 2020.

[134] Adnan Niazi, Jian S Dai, Stavroula Balabani, and Lakmal Seneviratne. Product cost estimation: Technique classification and methodology review, 2006.

[135] Robert H Liebeck, Donald A Andrastek, Johnny Chau, Raquel Girvin, Roger Lyon, Blaine K Rawdon, Paul W Scott, and Robert A Wright. Advanced subsonic airplane design and economic studies. Technical report, NASA, 1995.

[136] Bin Ren, Zihang Liu, and Haiyan Zhu. The study of entire aircraft level direct maintenance cost analysis method for civil aircraft. In *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)*, pages 1–5. IEEE, 2021.

[137] Marco Fioriti, Valeria Vercella, and Nicole Viola. Cost-estimating model for aircraft maintenance. *Journal of Aircraft*, 55(4):1564–1575, 2018.

[138] Valeria Vercella, Marco Fioriti, and Nicole Viola. Towards a methodology for new technologies assessment in aircraft operating cost. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 235(8):879–892, 2021.

[139] Carlos Cabaleiro, Marco Fioriti, et al. Assessment of new technologies in a multi-disciplinary design analysis and optimization environment including rams and cost disciplines. In *32nd Congress of the International Council of the Aeronautical Sciences, ICAS 2021*. International Council of the Aeronautical Sciences, 2021.

[140] Jeff A Estefan et al. Survey of model-based systems engineering (mbse) methodologies. *Incose MBSE Focus Group*, 25(8):1–12, 2007.

[141] James N Martin. *Systems engineering guidebook: A process for developing systems and products*. CRC press, 2020.

[142] Andreas Page Risueño, Jasper Bussemaker, Pier Davide Ciampa, and Bjoern Nagel. Mdax: Agile generation of collaborative mdao workflows for complex systems. In *AIAA Aviation 2020 Forum*, page 3133, 2020.

[143] Sparsh Garg, Jasper Bussemaker, Luca Boggero, and Björn Nagel. Mdax: Enhancements in a collaborative mdao workflow formulation tool. In *ICAS Conference, Florence, Italy*, 2024.

[144] Jasper H Bussemaker, Luca Boggero, and Bjoern Nagel. System architecture design space exploration: Integration with computational environments and efficient optimization. In *AIAA Aviation 2024 Forum*, 2024.

[145] Carlos Cabaleiro, Marco Fioriti, Luca Boggero, et al. Methodology for the automated preliminary certification of on-board systems architectures through requirements analysis. In *Proceedings of the 33rd Congress of the International Council of the Aeronautical Sciences, Stockholm, Sweden*, pages 4–9, 2022.

[146] Carlos Cabaleiro, Marco Fioriti, and Luca Boggero. Automated generation of aircraft on-board system architectures and filtering through certification specification requirements. In *Journal of Physics: Conference Series*, volume 2716 (1), page 012044. IOP Publishing, 2024.

[147] David S Velasco, Thomas Kuhn, and Sören Kemmann. Reliability analysis in model-driven development of embedded systems. In *2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7. IEEE, 2013.

[148] PACE Preliminary Aircraft and Systems Design. https://pace.txtgroup.com/products/preliminary-design, 2024.

[149] Sebastian Woehler, Georgi Atanasov, Daniel Silberhorn, Benjamin Fröhler, and Thomas Zill. Preliminary aircraft design within a multidisciplinary and multifidelity design environment. In *Aerospace Europe Conference 2020*, 2020.

[150] FAST-OAD: An open source framework for rapid Overall Aircraft Design. https://github.com/fast-aircraft-design/fast-oad, 2024.

[151] Bianca Williams and Selen Cremaschi. Novel tool for selecting surrogate modeling techniques for surface approximation. In *Computer aided chemical engineering*, volume 50, pages 451–456. Elsevier, 2021.

[152] Marko Alder, Erwin Moerland, Jonas Jepsen, and Björn Nagel. Recent advances in establishing a common language for aircraft design with cpacs. In *Aerospace Europe Conference 2020, Bordeaux, Frankreich*, 2020.

[153] Luca Boggero. Design techniques to support aircraft systems development in a collaborative mdo environment. *Turin (IT): Doctoral dissertation*, 2018.

[154] Massimo Mandorino, Pierluigi Della Vecchia, Salvatore Corcione, Fabrizio Nicolosi, Vittorio Trifari, Giovanni Cerino, Marco Fioriti, Carlos Cabaleiro De La Hoz, Thierry Lefebvre, Dominique Charbonnier, et al. Multidisciplinary design and optimization of regional jet retrofitting activity. In *AIAA AVIATION 2022 Forum*, page 3933, 2022.

[155] Jasper H Bussemaker, Pier Davide Ciampa, Jasveer Singh, Marco Fioriti, Carlos Cabaleiro De La Hoz, Zhijun Wang, Daniël Peeters, Philipp Hansmann, Pierluigi Della Vecchia, and Massimo Mandorino. Collaborative design of a business jet family using the agile 4.0 mbse environment. In *AIAA Aviation 2022 Forum*, page 3934, 2022.

[156] Carlos Cabaleiro de la Hoz and Marco Fioriti. New methodology for flight control system sizing and hinge moment estimation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 236(12):2375–2390, 2022.

[157] Sergey Vladimirov and Scott Forde. Demonstration program to design, manufacture and test an autonomous electro-hydrostatic actuator to gimbal large booster-class engines. In *42nd AIAA/ASME/SAE/ASEE joint propulsion conference & exhibit*, page 4364, 2006.

[158] Carlos Cabaleiro de la Hoz, Marco Fioriti, and Luca Boggero. Performance and reliability evaluation of innovative high-lift devices for aircraft using electromechanical actuators. *Aerospace*, 11(6):468, 2024.

[159] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[160] Ralph L Keeney. *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge university press, 1993.

[161] Giuseppa Donelli, Pier Davide Ciampa, Björn Nagel, Gléverson Lemos, João Mello, Ana Paula C Cuco, and Ton van der Laan. A model-based approach to trade-space evaluation coupling design-manufacturing–supply chain in the early stages of aircraft development. In *AIAA AVIATION 2021 FORUM*, page 3057, 2021.

[162] A Ross and D Rhodes. Value-driven tradespace exploration ofr system design, lecture 5: Basics of applied utility theory. *System Engineering Advancement Research Initiative (SEAri), MIT*, 2010.

[163] Giuseppa Donelli, João MGD Mello, Felipe IK Odaguil, Ton van der Laan, Thierry Lefebvre, Nathalie Bartoli, Luca Boggero, and Nagel Björn. Value-driven systems engineering approach addressing manufacturing, supply-chain and aircraft design in the decision-making process. In *INCOSE International Symposium*, volume 33 (1), pages 463–481. Wiley Online Library, 2023.

[164] Dario Belmonte, Matteo Davide Lorenzo Dalla Vedova, and Gaetano Quattrocchi. A new active asymmetry monitoring and control technique applied to critical aircraft flap control system failures. In *MATEC Web of Conferences*, volume 304, page 04011. EDP Sciences, 2019.

[165] Martin Recksiek. Advanced high lift system architecture with distributed electrical flap actuation. In *Proceedings of the 2nd International Workshop on Aircraft System Technologies*, pages 49–59. Shaker Verlag Hamburg, 2009.

[166] P Janker, F Claeyssen, B Grohmann, M Christmann, T Lorkowski, R LeLetty, O Sosniki, and A Pages. New actuators for aircraft and space applications. In *Proceedings of the 11th International Conference on New Actuators, Bremen, Germany*, pages 9–11, 2008.

[167] Maamar Benarous and Isabella Panella. Flap system power drive unit (pdu) architecture optimisation. *The Journal of Engineering*, 2019(17):3500–3504, 2019.

[168] LIN Tianlong, Rosario Pecora, Danilo Ciliberti, XIA Wei, and HU Shuling. Aerodynamic optimization of an adaptive flap for next-generation green aircraft. *Chinese Journal of Aeronautics*, 37(2):100–122, 2024.

[169] L Castellini, M D'Andrea, and N Borgarelli. Analysis and design of a linear electro-mechanical actuator for a high lift system. In *2014 international symposium on power electronics, electrical drives, automation and motion*, pages 243–247. IEEE, 2014.

[170] Jean-Claude Derrien and Sagem Défense Sécurité. Electromechanical actuator (ema) advanced technologies for flight controls. In *International congress of the aeronautical sciences*, pages 1–10, 2012.

[171] Mirko Mazzoleni, Fabio Previdi, Matteo Scandella, and Giulio Pispola. Experimental development of a health monitoring method for electro-mechanical actuators of flight control primary surfaces in more electric aircrafts. *IEEE Access*, 7:153618–153634, 2019.

[172] Gianpietro Di Rito, Benedetto Luciano, Nicola Borgarelli, and Marco Nardeschi. Model-based condition-monitoring and jamming-tolerant control of an electro-mechanical flight actuator with differential ball screws. In *Actuators*, volume 10, (9), page 230. MDPI, 2021.

[173] Danilo Ciliberti, Pierluigi Della Vecchia, Vittorio Memmolo, Fabrizio Nicolosi, Guido Wortmann, and Fabrizio Ricci. The enabling technologies for a quasi-zero emissions commuter aircraft. *Aerospace*, 9(6):319, 2022.

[174] Huan Pang, Tianxiang Yu, and Bifeng Song. Failure mechanism analysis and reliability assessment of an aircraft slat. *Engineering Failure Analysis*, 60:261–279, 2016.

[175] Yameen M Hussain, Stephen Burrow, Leigh Henson, and Patrick Keogh. A review of techniques to mitigate jamming in electromechanical actuators for safety critical applications. *International Journal of Prognostics and Health Management*, 9(3), 2018.

[176] A Garcia, I Cusido, JA Rosero, JA Ortega, and L Romeral. Reliable electro-mechanical actuators in aircraft. *IEEE Aerospace and Electronic Systems Magazine*, 23(8):19–25, 2008.

[177] Mirko Mazzoleni, Yamuna Maccarana, Fabio Previdi, Giulio Pispola, M Nardi, F Perni, and S Toro. Development of a reliable electro-mechanical actuator for primary control surfaces in small aircrafts. In *2017 IEEE international conference on advanced intelligent mechatronics (AIM)*, pages 1142–1147. IEEE, 2017.

[178] Carlos Cabaleiro, Marco Fioriti, and Luca Boggero. Automated evaluation of performance, certification and maintenance aspects of aircraft on-board system architectures during preliminary design stages. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 4324, 2024.

[179] AIRBUS. Airbus training - a320 flight crew operating manual, 2017.

[180] Jasper H Bussemaker, Paul Saves, Nathalie Bartoli, Thierry Lefebvre, and Bjoern Nagel. Surrogate-based optimization of system architectures subject to hidden constraints. In *AIAA Aviation 2024 Forum*, 2024.