POLITECNICO DI TORINO

MASTER's Degree in AEROSPACE ENGINEERING

MASTER's Degree Thesis

Environmetal Control System and Flight Control System Architecture Optimization from a Family Concept Design Perspective

Supervisors

Prof. Marco Fioriti
Jasper Bussemaker (DLR)
Luca Boggero (DLR)

Candidate

Carlos Cabaleiro de la Hoz

July 2020

[This page was intentionally left blank]

Acknowledgements

This thesis would not have been possible without the guidance and help of many people. First of all I would like to thank DLR (German Aerospace Center) for giving me the opportunity to develop this thesis and to Polytechnic University of Turin for considering me as a candidate to do it.

I would like to thank both of my supervisors at DLR, Jasper Bussemaker and Luca Boggero, for all the time, advice and guidance provided. I would also like to thank Prof. Marco Fioriti, my supervisor from Polytechnic University of Turin, for all the help and support given through all these nine months of work.

Also, many thanks to all DLR employees and students that have helped and worked with me in Hamburg. This period has been really rewarding for me and I have learned a lot of new useful things.

Finally I would like to thank my family for all the support given when I moved to Hamburg.

[This page was intentionally left blank]

Index

List of Figures

List of Tables

Abbreviations

ACS	Air Conditioning System
APU	Auxiliary Power Unit
AAD	Average Absolute Deviation
BAS	Bleed Air System
CBV	Cross Bleed Valve
CHL	Cabin Heat Load
CPCS	Cabin Pressure Control System
CPCV	Cabin Pressure Control Valve
DOC	Direct Operating Cost
DSM	Design Structure Matrix
EBHA	Electric Backup Hydraulic Actuator
ECS	Environmental Control System
EHA	Electro-Hydrostatic Actuator
EMA	Electro-Mechanical Actuator
FCS	Flight Control System
FCV	Flow Control Valve
GA	Genetic Algorithms
HSA	Conventional Hydraulic Servo Actuator
MDO	Multi-disciplinary Design Optimization
MEA	More Electric Aircraft
MPL	Maximum Payload
MTOW	Maximum Take-Off Weight
OEM	Aerospace Manufacturing Company
PCI	Product Line Commonality Index
PFPF	Product Family Penalty Function
SFC	Specific Fuel Consumption
TAV	Trim Air Valve
THSA	Trimmable Horizontal Stabilizer Actuator
TLARs	Top Level Aircraft Requirements
WAIV	Wing Anti-Icing Valve
XDSM	Extended Design Structure Matrix

1. Introduction

1.1. Thesis overview

The scope of this thesis is to do the optimization of an aircraft's environmental control system (ECS) and flight control system (FCS) from a systems architecture perspective and from a family concept design problem. From a systems architecture perspective because the optimization goal is about the overall configuration of the system, not of one single component or function. And family concept design because its goal is to optimize several aircraft at the same time. This means, generally speaking, that the optimization should find the optimum ECS and FCS architecture configurations for each member of a new aircraft family. This problem requires a Multi-disciplinary Design Optimization (MDO) approach since it involves more than one discipline (thermodynamics, costs, aircraft performance...). And will be part of a bigger optimization problem (On-Board Systems Optimization). One of the main problems with these kinds of optimizations is that architectures are discrete variables, and the quantification of the decision of choosing that configuration among the others is not clear. For this reason, optimization based on gradient methods is discarded and the main tools used are genetic algorithms (heuristic methods).

Now, the main concepts are going to be explained more in depth and after the main design variables, parameters and optimization objectives are shown.

1.2. Thesis' main concepts

1.2.1. Family concept design

Family concept design is a way of covering a certain market area with several products which are designed at the same time. The advantage of it is that the development and production cost (and hence the acquisition cost) can be reduced if some parts, pieces or processes are shared among the different products. This is called commonality among members [1–4] and it is wanted to be maximized in order to reduce the costs for the producing company. Some new concepts are introduced here like flexibility and platform design, which are related to the commonality previously mentioned.

What family concept design is about is trying to find a way of designing several products which can use common components or manufacturing processes. So modularity can be applied, which consists on producing different products by changing some parts. Also scale-based product families can be designed, which consist on enlarging or shortening one reference product [2, 3].

In the aviation sector a close to scaled approach is used. When designing a certain aircraft family, the main changes among members are removing fuselage rings or adding more, changing the engines and then modifying the needed subsystems, like the tail or control surfaces in order to fulfill the regulations. Usually, the typical aircraft family consists on a regular plane, and some short range or long range

versions. With these family members a certain market region is covered and hence the customer's requirements are fulfilled.

The market region in terms of aeronautics is given as payload-range diagram points. But it is known that one plane model always performs more than one single route. This means that the overall functioning can be represented with some points in the diagram with some associated frequencies that the aircraft will have to cover [5–7]. And the family optimization will consist on where to position each member in the PL-R diagram (Figure 1) and the degree of commonality among the members.

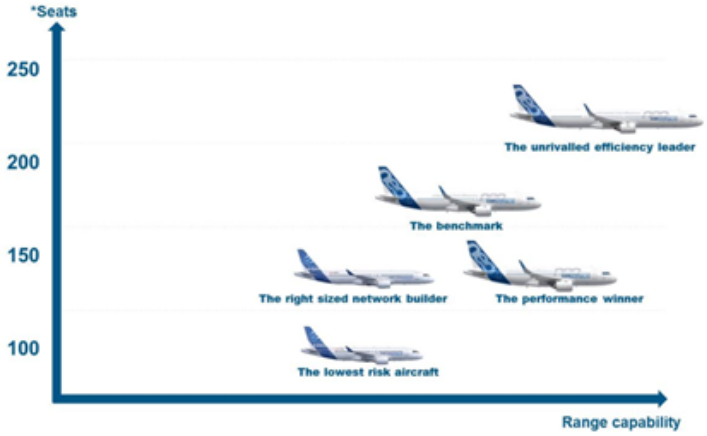


Figure 1: Aircraft family in PL.R diagram example

Commonality is an abstract concept, and it is difficult to evaluate. Some methods are implemented in the industry and selecting one or another depends on the problem itself. One method is the Product Family Penalty Function (PFPF). This method evaluates common measurements among components between family members. If a component is shared its penalty is zero, but if it is not its penalty will depend on the degree of non-similarity that it has [8]. But detailed information about components is needed in order to be able to use this method. One metric that requires less amount of details is the Product Line Commonality Index (PCI). This method just considers whether a component is shared or not and scales its size with one sizing factor [9]. In chapter 4 this metric will be developed more in depth.

1.2.2. Systems' architecture

Systems architecture design means that the objective is defining an architecture for the system. So the goal is defining the conceptual model that defines the structure and behavior of a system. This means that all the different components and their positions will be properly defined after this analysis. This part of the conceptual design is the one that has a bigger impact on the system's performance, while the posterior detailed design will just modify small percentages of the overall performance for that configuration. So choosing the appropriate architecture is important and will highly influence the system's final result.

In general, architectures are deeply studied and the best ones are known for almost every case. But currently, new methods are arising in order to optimize and discover new architectures for more complicated and integrated systems that are now appearing in engineering [10]. One way of doing this is thinking about components and functions instead of thinking in configurations as a whole. For example a compressor fulfills a function which is compressing air, but needs to be moved by an external component, which can be a turbine. Thinking like this and abstracting the systems as functions to be fulfilled can lead to new configurations. This is the main goal of systems engineers.

In the early design stage (conceptual design), the system architecture is defined. In the past, this has been done based on experience, or sometimes by considering only a small amount of all the possible architectures and giving them some trade-off based on semi-quantitative metrics, which were mostly also based on experience. But these metrics do not provide with detailed information about the real performance of the system with that architecture. This is known as the knowledge paradox: in the early design stage much freedom is available to modify the existing design, but not much knowledge of the system is available [10, 11].

The need to analyze more architectures in a more detailed way has deal to the apparition of new methods. In some cases the huge number of alternatives makes it impossible to analyse all of them so optimization methods should be used. Multi-disciplinary design optimization (MDO) makes this analysis feasible.

1.2.3. Multi-disciplinary design optimization

MDO is a field of engineering that uses optimization methods in design problems that involve several disciplines. The objective is to analyze all disciplines simultaneously because the optimum point when evaluating all the disciplines together can be different to the optimum point on each separated field [12]. The problem is that this adds a lot of complexity to the problem. It is really used in the aerospace field since it involves a lot of different disciplines that must be integrated in order to minimize costs and weight but accomplishing all the different requirements that every single discipline needs (Aerodynamics, aeroelasticity, structures, propulsion, flight control, on-board systems, mission analysis...).

One of the main problems for the companies is that these methods require a lot of time for formulation and verification, which means that the information has to go through different departments and every decision influences the others. So in order to make changes, propose new configurations or check a different value of certain parameters a lot of time is needed. This is caused by the different programing languages used, the different methods used among disciplines, the diverse objectives...

All this leads to the apparition of collaborative MDO. Which is used when a single user cannot do the analysis in all the disciplines involved. Each of the partners have then their own area of expertise and the main problem now is tool integration. In order to reduce all the output/input interfaces a common language for all the tools

is highly recommended.

This thesis has been developed in DLR, in the context of AGILE project [13, 14]. This project involves several entities of different European and non-European countries. Some of them include DLR, Leonardo, Bombardier, Fokker, Politecnico di Torino, TU Delf, Airbus Defence and Space, Onera... One of the main objectives of this project is integrating all the different disciplines so that the formulation times are reduced and more time can be spent in the study and analysis of different configurations. So, at the end more exploration possibilities are allowed and hence new and more efficient aircraft subsystems configurations can be discovered. One powerful tool to make this is CPACS [15, 16], which consists in a common language for all the disciplines. Like this each discipline has to be able to translate its data and read only one common language, not all the different ones of each discipline. Hence, the number of interactions among different disciplines is reduced. If for instance, one discipline changes one parameter it should translate that into CPACS language and the others will directly read this from CPACS, there is no need of communicating the change to each of them individually. The benefits of this method are noticeable now, for example, doing a parameter analysis (changing one parameter and checking the impact on all the other disciplines) is easy and much faster, so as a result more analysis can be done and more configurations can be studied. A visual example of the difference between the usual approach and how CPACS works is now shown in figures 2 and 3.

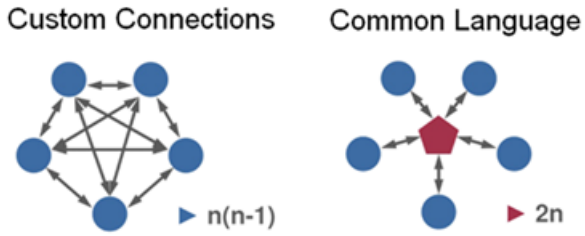


Figure 2: Difference between common and uncommon language

This method also allows to do conceptual and detailed design and analysis at once. In system architecting (conceptual part) the decisions have a lot of impact in the final result, there is a large design freedom but the details are not known or specified. On the other hand, in design optimization the architecture is fixed, so the high-impact decisions have been made and the remaining ones are the detailed choices on how to approach certain problems. The result is that on the early design phases the configuration is decided and later the detailed design engineers encounter problems that were not prevented, but as the architecture is fixed and cannot be changed these problems are solved without reaching the optimum solution.

With this new way of architecture optimization in a multidisciplinary field big and small decisions can be made in early and later phases of the project. Because now there is more time to test configurations and change consequently what is needed to reach the optimum points.

So by connecting system architecting with collaborative MDO, the previously

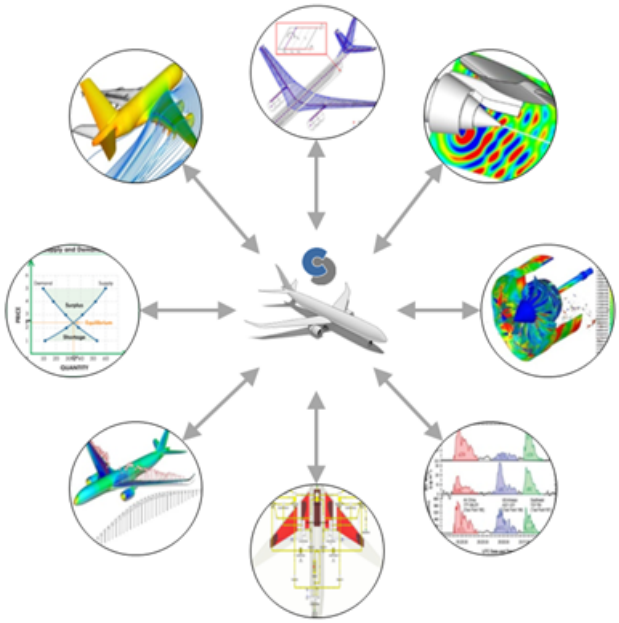


Figure 3: CPACS language

commented "knowledge paradox" is mitigated. Now there is design freedom but systems' performance can still be precisely analysed with high-fidelity models on each of the disciplines.

As mentioned before, this does not come without drawbacks. The main one is the complexity of integrating all these systems and decisions. One powerful tool to approach the problem is the XDSM (Extended Design Structure Matrix), which is the extended version of the DSM [17, 18]. It is used to show all the links, data flow, inputs and outputs of each discipline. It properly presents the convergence loops if needed, or the optimization loops and clearly shows the different disciplines involved and how are they related to the other ones. In the following chapters the XDSM of this specific problem presented in the thesis is shown. It was done with an early version of MDAx. A difficult task involved in this thesis was linking MDO with the family concept design [19, 20]. The way of approaching this is developed later in further chapters.

1.2.4. Genetic Algorithms

As a final element for this optimization introduction, it is necessary to mention the optimization algorithms that are going to be used. For this particular problem gradient methods are not usable, or at least not the pure ones, maybe a modified version of them could be implemented but in general for problems in which the main design variables are not continuous genetic algorithms (GA) are recommended [21, 22]. Genetic algorithms are also known as heuristic methods. One main design variable is which configuration to choose, this is clearly a discrete variable and gradient methods have problems when evaluating this since they cannot estimate well the values in boundary conditions or limits (because there is no following value

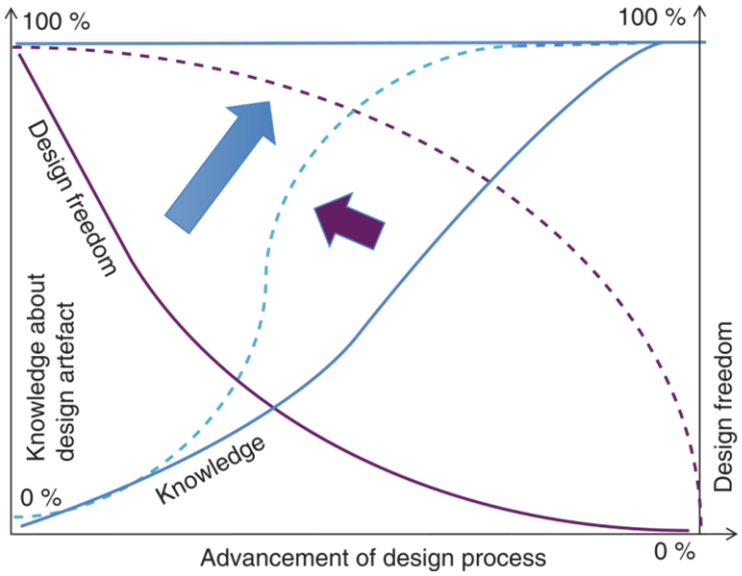


Figure 4: Mitigating the knowledge paradox, from [11]

to estimate the gradient). Once GA have been selected, their use is really useful in terms of computational time. They are commonly used to generate solutions to optimization problems by doing mutations, crossovers and selections from the past calculations, and hence, reaching the optimum points with less iterations.

1.3. Thesis design problem's approach

The last step of this introduction is explaining the main design problem and how it is going to be approached in general terms. A more in depth development with XDSM and the list of different tools used is shown in later chapters.

The optimization objectives that are wanted are the costs. But there are several costs involved in the design of a new aircraft (design costs, production costs, aircraft price, operating costs, maintenance costs, certification...) and there are a lot of different variables that have a big influence on them (fuel consumption, reliability, complexity, materials chosen...). So a more precise study of these costs should be made.

Supposing a certain market region requirement given by an airline, the producing company (OEM) should design an aircraft family to satisfy this Payload-Range diagram conditions. Usually, the more money the production company needs to invest in order to produce the aircraft, the less benefits it will have but, on the other hand, the better the product performs, the less costs it is going to induce for the airline while operating it. So as a result the airline wants the products to be as much efficient as possible to reduce their operating cost, but in order to do this the manufacturing company should spend more in order to make every family member work on its optimum point. Here is where commonality appears. The bigger

the commonality among members, the cheaper it will be to produce (less different production plants, less design effort, less component design...) but the members will operate more time out of their design points and the overall performance will be penalized, so this trade-off between costs is the main optimization problem. A division of costs is now done.

The main costs for the operator (airline) are now summarized. These costs are divided in several parts which are: fuel and oil, airport handling, crew salaries, airframe maintenance cost, engine maintenance, passenger service, administration costs, taxes, revenue management, aircraft rent or price, depreciation, amortization... In this thesis the ECS is being studied, so just those terms that are directly affected by it are going to be analyzed. So in our simplified model the operation cost is as follows:

$$Operation\ Cost = f(fuel\ consumption,\ maintenance\ costs) = f(performance) \tag{1}$$

The fuel price is directly dependent on the specific fuel consumption (SFC), the weight and the routes that the plane is going to do. The maintenance cost is dependent on the reliability of the different components and the number of components and it is directly related with the architectures. The main costs for the manufacturer (OEM) are:

$$Acquisition\ cost = f(design,\ manufacturing,\ certification) = f(commonality) \tag{2}$$

Basically this means that if there is a bigger number of common components the costs of producing, testing and certificating will be less. So commonality has a direct impact on the manufacturing and acquisition costs.

Since maintenance models were not available at the moment, this analisys focuses on comparing the commonality (as something related to the acquisition costs) and fuel burn (related to performance which is related to operation costs). At the end it can be seen that there is a trade-off between commonality and performance, and this trade-off can be translated in terms of costs. This effect is what is going to be studied in this thesis, the final result that is expected is a Pareto front among commonality vs performance. Both costs are wanted to be minimized and in order to do this the performance and commonality should be maximized. The shape of this expected Pareto front is represented in figure 5.

In order to do this we need to specify which ones are the inputs, parameters and design variables of the optimization problem.

1.3.1. Inputs

The inputs are basically all the initial information given to the optimization process that it will have to consider to get the required results. In this case the input is the market segment, as said before. So the main information given will be a Payload-Range diagram that needs a new aircraft family to be operated. As it can

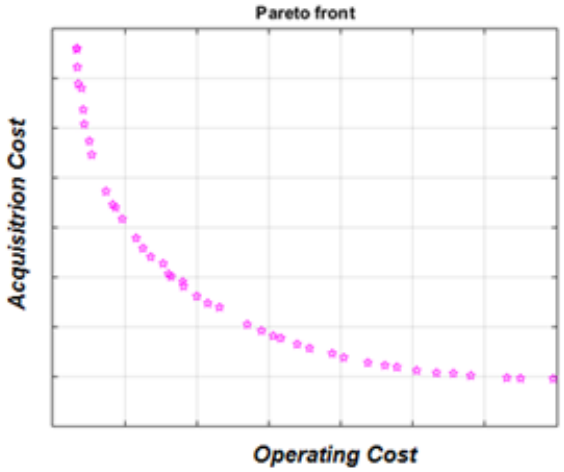


Figure 5: Expected result from the Pareto front

be seen in figure 6, planes operate several routes, so this will be taken into account. The model of this input will be some points of the diagram with their corresponding frequencies. So one simplified example would be: the aircraft should fly 1000 hours on a 3000 km route carrying 120 passengers, 300 hours on a 1700 km route with 110 passengers and 50 hours on a 500 km route with 80 passengers, all this in a determined utilization time or cycle.

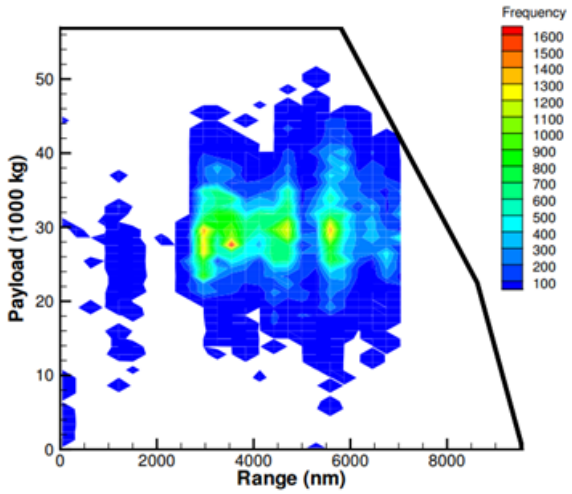


Figure 6: Typical PL-R histogram

One variable that can be used as input or as design variable is the number of family members. So it can be specified how many planes are going to be implemented (for example 3: standard, short-range version and long-range version). Or it could be asked to the optimization process how many planes are needed to fly these routes consuming the less fuel possible with the biggest commonality between them.

1.3.2. Design variables

The design variables are those parameters that the GA are going to change in order to find the optimum points and obtain the Pareto front. These variables shall reflect the different ECS and FCS architectures and the family concept. So in order to do this the variables that have been chosen are:

- ECS architecture: bleedless or conventional, number of ACU wheels, high or low pressure cycle...
- FCS architecture: electric or hydraulic actuators.
- Sharing subsystems: if the ECS and FCS is going to be shared among members or not

1.3.3. Optimization objectives

These will be both parameters previously discussed:

- Fuel burn (related to performance).
- Commonality index.

1.3.4. Model validation

The model should be sensitive to certain parameters and some simple expected results shall be checked once the problem is solved in order to be sure that the results have enough reliability. For example, the number of family members should influence the biggest achievable commonality. The more aircraft there are, the less probable it is for them to share components. The sharing variables should have a noticeable impact on the results. Finally, the design points and the off-design points should have a clear impact on the fuel burn consumption.

These previous analysis were done in the results chapter and will be properly explained and developed later.

2. ECS State of Art

2.1. ECS introduction

The environmental control system ensures that the cabin conditions are adequate, which means that the physiological needs of the passengers and crew are correctly achieved with some extra comfort demands. The functions include humidity, temperature and pressure control. The minimum and maximum values are fixed by legislation. For instance, in FAR 25 a minimum of fresh air mass flow of 4.16 grams per passenger and second are required, but companies usually exceed this minimum. The requirements specified in FAR Part 25.841 [23] and CS-25 [24] for safety and comfort are now summarized:

- The cabin temperature must be between 18°C and 25°C, with and optimum value of 21°C.
- The ECS must guaranty a minimum pressure level corresponding to the pressure at 8000 feet high by the ISA atmosphere model. If an aircraft does not overpass that altitude, it does not need cabin pressure control.
- The cabin humidity should be in a range between 10% and 20%. This requirement is usually done in order to maintain low humidity in the avionics room to avoid corrosion problems.
- The minimum airflow that must be delivered to the cabin is of 250 grams per minute per person. This condition ensures that the cabin air is breathable and has enough oxygen for all the persons in it.

Apart from these requirements the ECS has a lot of different possibilities and operating conditions. So each company has its own requirements about this system. For instance, the temperature at which air is delivered to the cabin. But all this information is usually confidential so all the exact values of mass flows, temperatures are unknown for the public. Despite that, the general configuration of an ECS is almost fixed. It is shown in figure 7:

2.2. ECS functioning and subsystems

The functioning of the environmental control system can be divided into more subsystems. The air is extracted by the Bleed Air System and then delivered to the Air Conditioning System, which regulates the air conditions and delivers the air to the cabin. To explain the overall functioning, a subsystem functioning explanations is being made in the following paragraphs [25–29].

2.2.1. Bleed Air System

The functions of this subsystem are taking the air directly from the atmosphere and providing it at high pressure and temperature to the Air Conditioning System. In general this is done by doing a bleeding to the engine and then delivering it

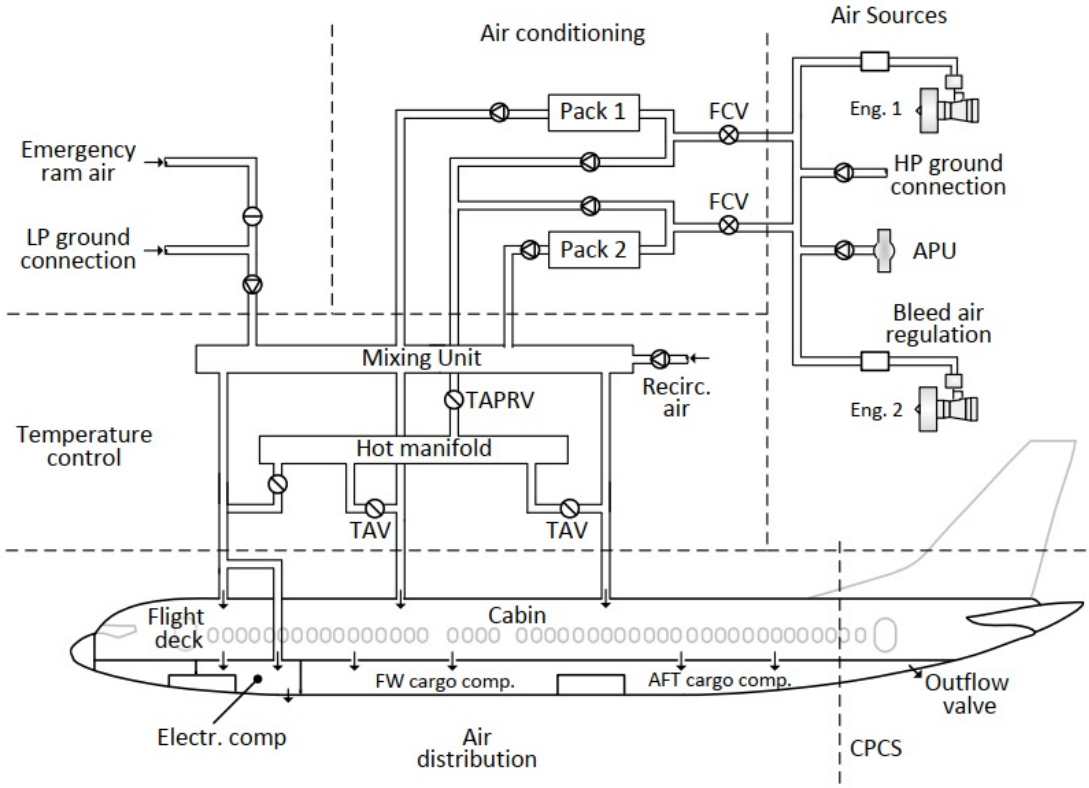


Figure 7: ECS general components' configuration, from [13]

through pipes and valves. In case of failure the air can be provided by the Auxiliary Power Unit (APU). When the aircraft is on ground with the engines off, this air is provided by a high pressure ground connection. New tendencies show that it is possible to omit the bleeding and take the air form inlets and compress it with electric fans in which is called bleed-less concept, but this idea will be developed later. A general configuration of a Bleed Air System is shown in figure 8.

Without going too much into detail, the air pressure is regulated by the valves and delivered to the Air Conditioning System. But a small part of the bleeding, in case of need, will be delivered to some pipes located in the wing in order to remove the ice. This is controlled by the Wing Anti-Icing Valve (WAIV) and the whole system is the Wing Anti Ice System. This system is important to consider in this analysis because, as it will be seen later, if a bleedless concept is chosen there won't be bleeding and this system should be redesign.

There are no design choices about the nacelles anti-ice system since it is usually done by bleeding some high temperature air from the engine, recirculate it through the nacelle so ice will not be formed and deliver this air again to the engine. Another important element is the Cross-Bleed Valve. Its function is communicating the bleeding of both engines, so in case of failure of one of them all the ECS can be supplied by just one of the engines by opening this valve. The rest of the time it remains closed. All this ducts and valves connections are shown in figure 9.

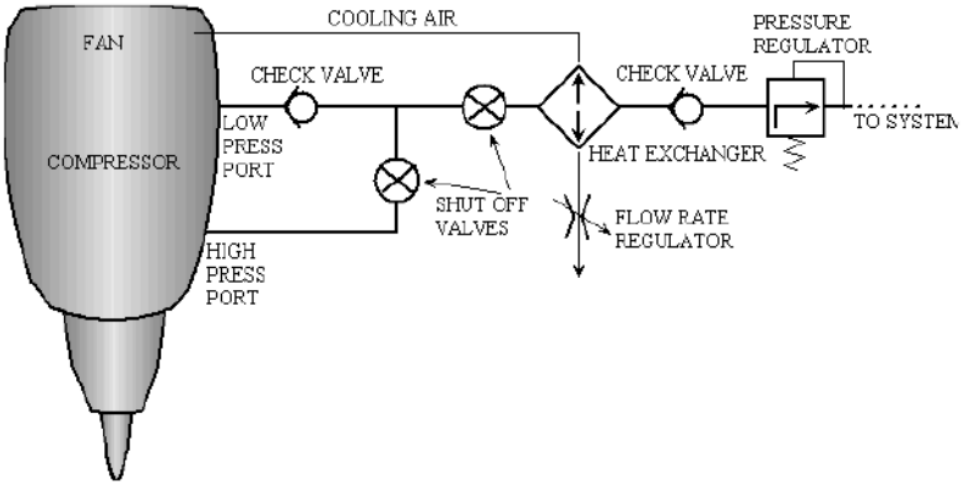


Figure 8: Bleed Air System classic configuration, from [30]

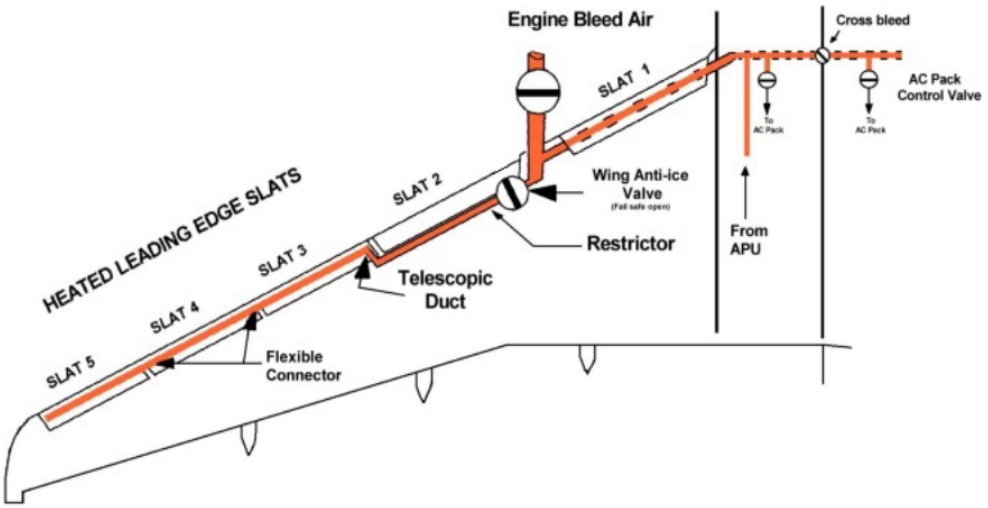


Figure 9: Wing Anti Ice System, from [A320]

2.2.2. Air Conditioning System. Pack types

This subsystem receives hot pressurized air, conditions it and then delivers it to the cabin. In order to do that, it can be split in five main elements: Air Conditioning Packs, Mix Manifold, Recirculation system, Air Distribution System and Ram Air System.

As it can be seen in figure 10, there are usually two Air Conditioning Packs for safety reasons, when one fails the other one can provide all the air flow needed working at 180% of the nominal flow rate [29], if both fail the aircraft will proceed the descent and land in the closest airport while regulating the temperature directly with the ram air. They receive the air from the BAS for any of the engines through a valve called Flow Control Valve (FCV). There the air is treated in order to reduce its temperature and pressure to an adequate level. This subsystem is one of the most important of the ECS and it has a lot of different ways of functioning, for this reason it is going to be explained more in depth at the end of this chapter.

To cool the air in the packs, heats exchangers are used. They use cold air from the outside provided by the Ram Air System, which consists in two ram inlets (one for each pack) that take the outside air to refrigerate and then deliver it again to the atmosphere through two ram outlets. On ground this air is provided by external fans which can be driven electrically (like in the B727) or pneumatically (B737) [25].

The air that has been conditioned in the packs goes to a chamber called Mix Manifold. The air of this chamber comes from two places actually, the packs and the recirculation system. Here, the total temperature will be the proportional sum of the parts, considering the corresponding mass flows. From this chamber the air is directly delivered to the cabin. But before arriving to the cabin, the temperature can be increased heating it a bit with air that comes directly from the bleeding, controlled with some Trim Air Valves (TAV).

Now the air is properly controlled, pressurized and tempered. So the Air Distribution System delivers it appropriately to the cabin. This is done by a ducts and inlets system that make sure that the recirculation in the cabin is enough to make the air flow. This means that no big recirculation bubbles are created, the noise is not too high and that there are not cold currents flowing through the interior. The cargo bay is also controlled according to the requirements (animals or just luggage).

Focusing on the recirculation system, it was not used in the past but its use now is essential. It allows reaching proper levels of cabin mass flow without doing a big bleeding. Flow conservation says that the mass flow delivered to the cabin is equal to the one done in through the bleeding, but with this recirculation it is possible to deliver a bigger mass flow by taking air from the own cabin and recirculating it. In conclusion, in order to deliver the same mass flow, a less bleeding from the engine is needed, which increases the engine performance.

Two recirculation fans are always implemented, one for each duct. The air is now filtered in order to remove the bacteria and small particles that might come from the cabin. As a result, two inlets take air from the cabin with recirculation

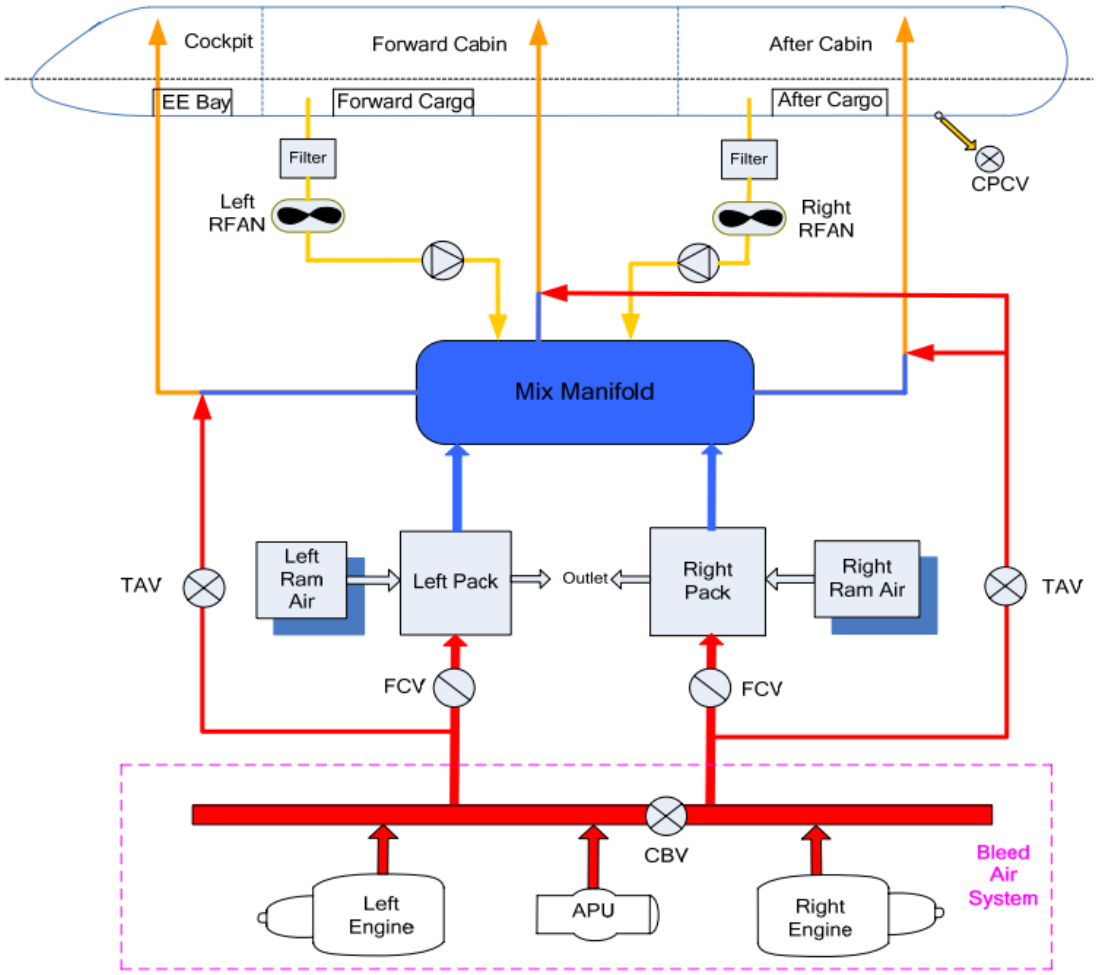


Figure 10: Air Conditioning System, from [29]

fans, then; the air is filtered and delivered to the mix manifold where it will be again taken to the cabin. Usually the recirculated mass flow is around a 40% and a 60% of the total mass flow delivered to the cabin.

Regarding the pack types. The Air Conditioning Packs receive the air at high temperature for the cabin at around 200°C. In order to do that, as said before, heat exchangers are used. These exchangers use ram cold air to refrigerate but just with these elements it won't be enough to reach the requirement, so a turbine is also added. There are many ways to do this process but just some of them are worth mentioning. Now, the main types of packs are going to be explained with their advantages and disadvantages and with a component description of each of them.

The first decision is on how to make the Air Cycle Machine (ACM). This means, how to link the compressor and the turbine. The compressor is used in order to make the cooling process more efficient. In figure 11, a first approach to the problem is shown. Just cooling the air and then delivering it to the turbine is a very inefficient process called "Simple Air Cycle" and it has been used just once in the Fokker 100 model.

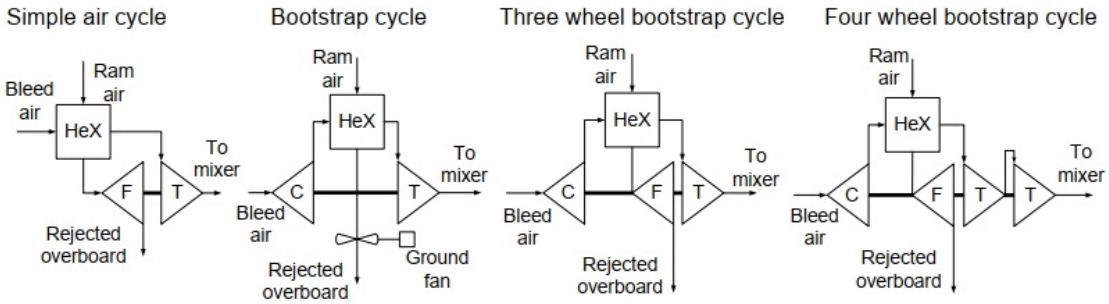


Figure 11: Air pack types, from [31]

If the air goes from the compressor to the heat exchanger, and then to the turbine, this is called bootstrap cycle (reversed classic Brayton cycle). The Two Wheel Bootstrap Cycle consists just in a compressor and a turbine, but it needs a "ground fan" to generate ram air in on-ground conditions. The Three Wheel BC links this ground fan to the turbine in the same shaft. Now the system is self-contained although the turbine is a bit more inefficient. The Four Wheel BC is like the Three Wheel but with two turbines instead of just one. One linked to the compressor and another one to the ram fan. This improves the efficiency but on exchange the weight, mechanical complexity and number of components increases.

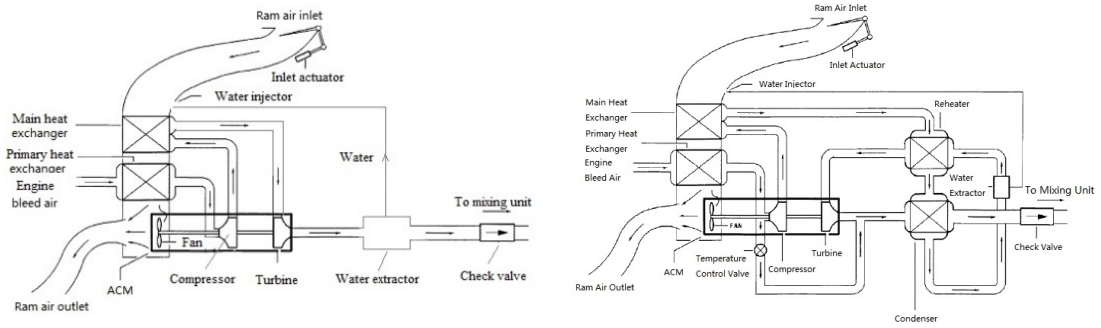
One problem that might appear is that ice could be formed in the turbine under certain conditions. This is due to the fact that the atmospheric air can have high percentage of humidity and since the turbine lowers the temperature, the water contained in the air could freeze. This situation lowers a lot the turbine's performance and should be avoided. Two concepts for dealing with this problem are now introduced.

- Low pressure bootstrap cycle (or non-subfreezing): a water extractor is posi-

tioned after the turbine, so that water won't come out in the cabin. But the minimum temperature in the turbine is restricted to 2 or 3 degrees. Like this the ice problem is solved but the turbine's performance has been highly restricted.

- High pressure bootstrap cycle (or subfreezing): the water extractor is located before the turbine, which can now reach lower temperatures and be more efficient as a result. However, in order to do this some heat exchanges should be done in order to guarantee the proper performance of the system and some extra components are needed.

A low pressure example is shown in figure 12a. It is shown how the cooling is done with two heat exchangers, one before the compressor and another one (the bigger one) after it. It can also be seen how the water that is extracted is later injected in the ram to make a more efficient refrigeration. In figure 12b a high pressure cycle can be seen. In this case after the air passes through the main heat exchanger it goes to a reheater, which cools it a bit before entering the condenser, where water is condensed in order to extract it in the water extractor. Later it passes again by the reheater (this time on the cold side), and goes to the turbine without water on it. Then the air from the turbine is used as the cold side of the condenser.



(a) Low pressure **(b)** High pressure

Figure 12: Three Wheel Bootstrap Cycle examples, from [32]

2.2.3. Cabin Pressure Control System

This system makes sure that the cabin air pressure is adequate at any moment. As said before, the pressure must not be lower than the one at 8000 feet height. Usually companies maintain this pressure at 6000 feet, in order to improve the cabin comfort.

The main element of this subsystem is the Cabin Pressure Control Valve (CPCV). Some big aircrafts have two but in general with one it is enough. Changing the opening angle of this valve the outlet airflow rate is controlled, and like this the pressure is modulated. It is located in the back down part of the fuselage. There are also two more pressure valves that are used to prevent pressure differences that could break the fuselage structure.

2.3. Considered ECS architectures

The scope of this thesis is finding the optimum architectures. This means that it might not be necessary to calculate all the variables (pressures, temperatures...) of the cycles, just those ones that are needed and that represent better the overall performance of each system. The most important parameters that could be considered for one configuration to another could: reliability, weight, fuel consumption, innovation (design effort), off-takes and ram air mass flow needed. Considering all these effects and the degree of behavior on each of them, the configuration will be totally defined and could be evaluated by the genetic algorithms. Maybe not all of these variables are going to be taken into account, but they are the main ones.

There are some subsystems that have no impact on these parameters, like for instance the Cabin Pressure Control System. This element´s architecture does not represent any design choice since the subsystem is fixed and functions in order to make the pressure increase of decrease accordingly to the legislation, but there is not a direct impact of any of the variables previously defined. On the other hand, the Bleed Air System and the Air Conditioning Pack configurations highly change all those parameters, so they are considered design choices. They are now developed in more depth.

2.3.1. Bleedless or conventional

The first design choice for the ECS is if the Bleed Air System is going to be a conventional one or if a more electric approach is being done with a bleedless configuration. While the conventional configuration takes the air for the Air Conditioning Unit from the engines, the bleedless one takes it from fuselage inlets and compresses it with electrically driven compressors.

Currently there is just one commercial aircraft that uses a bleedless architecture, the Boeing 787 (first flight in 2009, introduction in the market in 2011). The main reason is that in the conceptual design phase of the project Boeing decided to make a more electric plane and several systems where redesigned accordingly. This was said by Boeing itself in all the publicity they made for their new model [33, 34]. The main changes that are of interest for this thesis are the Bleedless ECS and the Wing Anti-Ice devices. The main ideas behind this more electrical concept are:

- This architecture is more reliable as a whole, since hydraulic fluid is reduced, so the maintenance costs and the number of interruptions will decrease.
- There is no need to bleed air from the engine for the ECS, so the fuel consumption will be less penalized than with the conventional approach, the overall specific fuel consumption will decrease.

But there are also some drawbacks that should be taken into consideration, the main ones are now shown:

- Electric configuration has not been used until currently, so it might have some design and production problems that in the conventional one are currently

solved since it has been under production for a lot of years. This will lead to a higher design and manufacturing cost in order to properly certificate this new configuration.

- Two new fuselage inlets are now needed in order to take the external air. This will consequently penalize the total drag. The two ram inlets are the same as they were in the conventional configuration.
- There is no bleeding, but the air is now compressed with electric fans. These fans are dedicate electric-compressors that need a source of power. The power comes from electric generators that extract energy from the engines' shafts. So more components are needed and the weight increases, but also the fuel consumption because the shaft is now providing more energy for the electric system than before.

It can be seen that the final performance is not clear since a lot of different effects are happening at the same time. Boeing trusted in this new configuration strongly believing that the overall fuel consumption will highly decrease. There are also some other studies that estimated the impact that this system would have in the Direct Operational Cost (DOC) [35], which at the end summarizes almost all these effects previously commented. All the studies and estimations previewed a decrease in the fuel consumption and in DOC; some of them with values around 7% and others with more modest ones. At the end, when the Boeing 787 started flying it was seen that this reduction was not as big as expected. It is true that the Specific Fuel Consumption (SFC) is slightly lower but there were some effects, not considered by Boeing, which could have interfered with the final results. There is not an exact value on this reduction, or it hasn't been found for this thesis. Although there hasn't been a big improvement, all the new technology invested in this project could lead to future developments for more efficient aircrafts.

The exact Air Pack type that was used for the Boeing 787 is not known, since that information is not public. But the objective of the pack is the same as in the conventional designs, reducing the temperature to a precise and controlled value. The only thing that changes is where the air is coming from, and the values of its corresponding pressure and temperature. So for this approach it is going to be supposed that the possible pack types can be the same ones as in the conventional configuration. Then, the genetic algorithms will determine if those air conditioning packs are appropriate or not from a bleedless perspective. Maybe some of them can't perform properly with the bleedless concept, or it could happen that just one type fits with it. As this information is not known, the optimization method will determine it. An image of the B787 ECS is shown in figure 13. It can be seen that it also has two packs, with their corresponding ram inlets and heat exchangers, so these assumptions are completely reasonable.

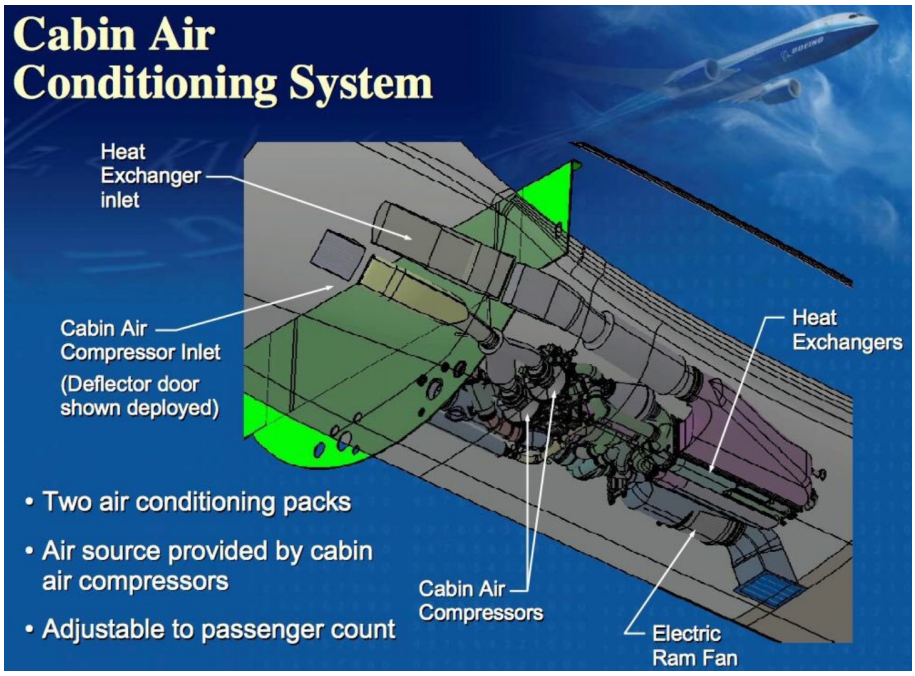


Figure 13: Bleedless ACU, from [B787]

2.3.2. Air Pack Type

The other main design choice for an ECS is which kind of pack will be implemented. As it was said before, the simple air cycle has been used just once in the industry (Fokker 100, first flight in 1986), and has never been implemented again owing to its low performance. So there is not a lot of information about this pack type and it won't be considered as a design choice in this thesis' project.

Regarding the bootstrap cycles, the following are the possible combinations, and hence, design choices:

1. Two Wheel Low Pressure Bootstrap Cycle
2. Three Wheel Low Pressure Bootstrap Cycle
3. Four Wheel Low Pressure Bootstrap Cycle (not used)
4. Two Wheel High Pressure Bootstrap Cycle
5. Three Wheel High Pressure Bootstrap Cycle
6. Four Wheel High Pressure Bootstrap Cycle

The main differences have been explained before, but from a design perspective for this thesis a few more comments are now being added.

- For the two wheeled configurations, the external source of power should be considered, and consequently its weight increase and reliability penalty.
- The three and four wheeled models are self-contained but they have a higher number of components, which increases the weight and reduces the reliability.

- The high pressure architectures are able to reach lower temperatures and hence they need to deliver less air mass flow to the cabin. Knowing this, the fuel consumption will be less penalized than with a low pressure model since the bleeding is lower. In case of a bleedless the same effect happens because there is less pressure and temperature conditions required for the fans and hence they will need to compress the air less needing less power and so, penalizing less the power extraction from the shaft. But on the other hand, high pressure configurations have much more number of elements and the overall weight increases.

Doing a state of art analysis is interesting to see what the current tendencies are and how the different configurations have been used in the past years [29]. It can be seen how the five configurations have been tried and discarded or continued, hence this information can provide with some relevant data about how well a configuration performs. A table, showing some aircraft models is now provided. The first flight year is given as a time reference for all the aircraft. All this information can be seen in table 1.

There has not been found any kind of correlation between the air pack type and the MTOW of the aircraft or with the number of engines. It seems to be a pure design choice regardless the plane's weight or shape. There are some noticeable tendencies that are now being commented:

- Low water separation configurations (Low pressure systems) were used in the past but it seems that the tendency is changing to a high pressure concept.
- The two wheeled model is not being implemented in new aircraft for the last 25 years.
- The new versions of past planes still use the three wheel system while the new models are starting to use four wheeled ones, at least on the bigger aircraft.
- Airbus stopped using low pressure configurations, and Boeing seems to be doing the same.
- Two Wheel High Pressure Air Pack does not seem to be a good choice. It has just been implemented once and in the posterior versions it was not used anymore.

Almost all the literature about ECS control and optimization is done with the three wheel high pressure system. It has been the most used and it is expected to still be used in the next years. The four wheeled is worth mentioning since it seems to be the next step in the ECS design but also considering the bleedless configuration. The other models seem to be kind of obsolete but they will be considered for the optimization because they can still have some advantages.

2.4. ECS components analysis

A list of the main components of all the configurations is now shown. This is important for this thesis since the number of components and their different

		Low Pressure		High Pressure		
Aircraft	First Flight Year	2 Wheel BC	3 Wheel BC	2 Wheel BC	3 Wheel BC	4 Wheel BC
B727	1963	x				
DC 10	1970		x			
A300	1972		x			
MD 80	1979	x				
A310	1982		x			
B767	1981				x	
B757	1982				x	
B737-300	1984	x				
A320	1987				x	
B747-400	1988		x			
MD 11	1990	x				
B737-400	1984			x		
B737-500	1984	x				
A340	1991				x	
A330	1992				x	
MD 90	1993	x				
B777	1994					x
MD 95	1998		x			
B737-700	1997		x			
B737-800	1997				x	
B737-600	1997		x			
B737-900	1997				x	
A380	2005					x
B787	2009					*Bleedless*
B747-800	2010				x	
A350	2013					x
C919	2017				x	
	Total	6	7	1	9	3

Table 1: ECS pack types state of art, from [29]

characteristics will stringy influence the system weight, performance and reliability.

Regarding the conventional configuration compared with the bleedless concept. In the conventional, in order to have a proper Bleed Air System, the only elements needed are some pipes and valves (previously detailed), a small preheater and the engine. When changing to a bleedless approach it has been seen that two more inlets have to be opened in the fuselage, with typically four fans to compress the air (two for each) but the valves and pipes system disappears and its reduced to two main conducts. The corresponding electric generators and cables are also noticeable.

While analyzing the air packs it can be seen that the number of different components increases. It has been stated that some configurations have more components than another ones, but this has not been quantified. In the following table 2 there is a list of all components involved on each of them.

Component	2W - LP	3W - LP	2W - HP	3W - HP	4W - HP	Comments
Primary Heat Exchanger (PHE)	1	1	1	1	1	Cools a bit before the ACM
Main Heat Exchanger (MHE)	1	1	1	1	1	Main cooling process
ACM Compressor	1	1	1	1	1	Axial compressor
ACM Turbine	1	1	1	1	2	Axial turbine
ACM Fan	0	1	0	1	1	Generates ram air on ground, landing and low altitude
Ground Fan	1	0	1	0	0	Generates ram air on ground, landing and low altitude
ACM Shaft	1	1	1	1	2	Bi-shaft or mono-shaft
Reheater + Condenser	0	0	1	1	1	Increases turbine efficiency
Water Extractor	1	1	1	1	1	Extracts water
Plenum	1	1	1	1	1	Stores air from the heat exchangers

Table 2: Pack components

Each component will be sized depending on their requirements for each configuration and mass flow requested. For example, a heat exchanger's size depends on the temperature gradient that is wanted and on the corresponding flow on each of the sizes. The ACM is always compound by axial compressors and turbines. The shaft can be single or double in case there are two turbines (high pressure shaft and low pressure shaft). In case there is not a fan, a ground fan is needed and it will be considered as a drawback for that system. The reheater and the condenser are considered together since its functioning depends on each other. Some elements as the water extractor are always in the system but can change its position. Others like the plenum are there without any design choice available. Most of these elements have been taken from [36].

The ECS analysis was done with ASTRID, a software developed by the Polythecnic University of Turin. At the begining of this thesis, the ECS package was unfinished so some research had to be done in order to model some of the components, since the use of it was uncertain. Here are some used references about how to model some components [29, 37–46] or requirements [47, 48].

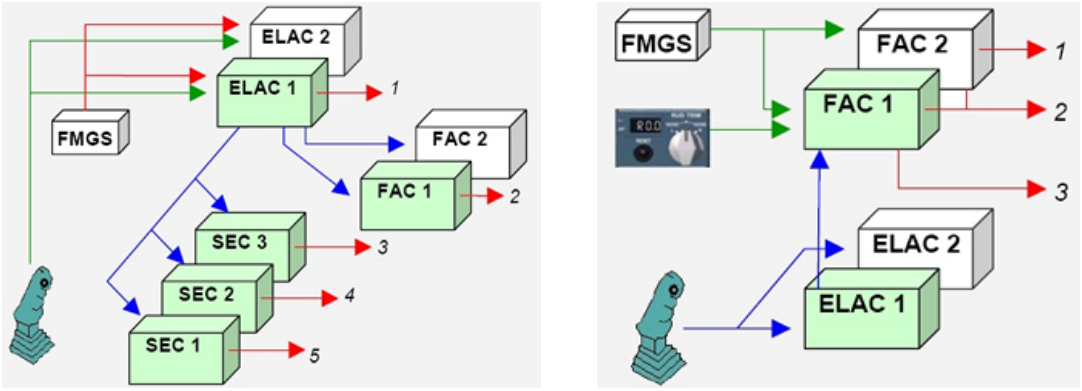
3. FCS State of Art

3.1. FCS introduction

In this thesis the objective is to change the FCS architecture of a conventional passenger aircraft. Hence all non-conventional architectures are discarded (supersonic, hypersonic, canard...). The main architecting choice will be about the actuators and the power supplies for each of them, not on how and where the surfaces are positioned. A conventional fixed-wing aircraft flight control system consists of flight control surfaces, the respective cockpit controls, connecting linkages, and the necessary operating mechanisms to control the plane. These systems include: flight data acquisition, flight data computation, flight control surfaces management, pilot information devices, autopilots, etc... In this analysis all the electronic devices, sensors, cockpit panel devices and computers are considered as part of the avionics and will not be taken into account as architecting decisions [49].

The FCS, as well as all the subsystems and components, need to have redundancies in order to ensure safe operation [24]. This redundancies apply to the actuators that move the surfaces as well as the computers and supply lines. For instance the A320 has 3 hydraulic lines and the A380 or A350 combine electric and hydraulic backups. In the following chapters this concept would be explained more in depth.

As said before, the flight control computers also need redundancies. They are applied to the three flight control computers (ELAC, FAC, SEC) in the case of the A320. This is not an objective or competence of this thesis so it will not be analysed. However, one example is now given to the reader in figure 14.



(a) Roll **(b)** Yaw

Figure 14: A320 roll and yaw computers redundancies

3.2. FCS functioning and subsystems

The aircraft is controlled with the stick and pedals, so there is a need to connect the control surfaces with the pilot commands. The first engineering solutions to link

the stick with the control surfaces was with a mechanic command. This included rigid bars or cables and pulleys, as seen in figure 15.

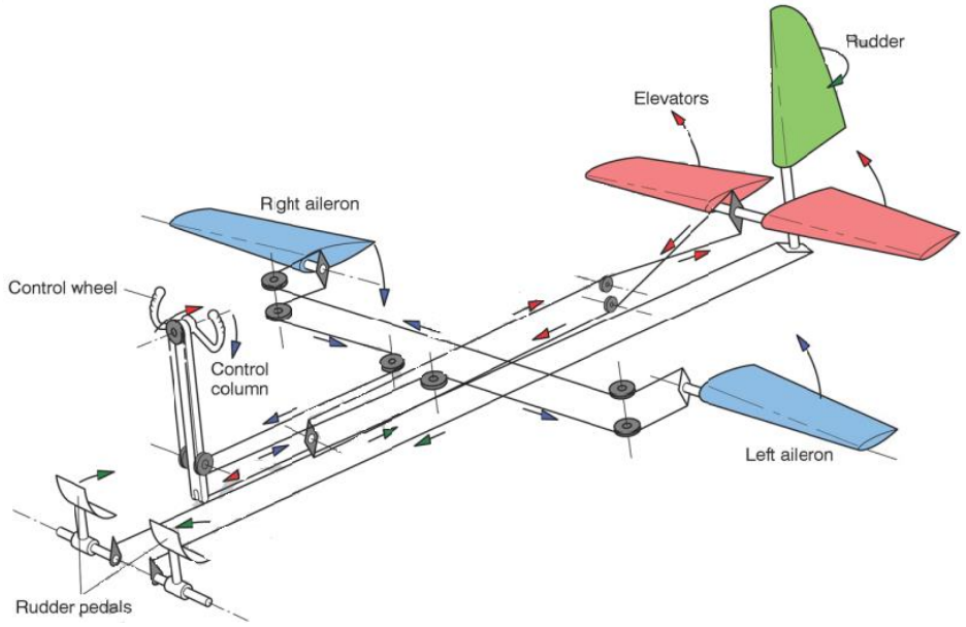


Figure 15: Pure mechanic commands

The next step was increasing the pilot action by linking the surface to an hydraulic actuator. Like this the pilot force didn't need to be high since the actuator would generate the required one, figure 16. This actuator was mechanically activated as the previous solution. This means using hydraulic power to assist the pilot. It made it possible to develop larger aircraft such as the Boeing 707. This hydraulic system allowed for larger control surfaces to be moved, and larger aircraft to be flown. In this case the pilot still feels a manual force on the stick, as with a mechanical system, but the pulleys an bars are now moved with the assistance of a hydraulic system [50].

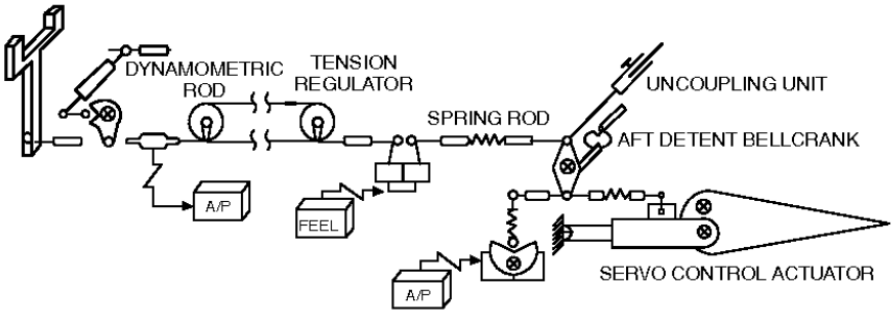


Figure 16: Mechanic command with actuator back-up

The next generation of flight control systems came with the fly-by-wire. This architecture consists on substituting the mechanical lines for electric ones. The main idea is communicating the actuators with the pilot through an electric signal. The

actuator now can be hydraulically or electrically moved. There is a need to give the pilot feedback in the stick so an artificial fell unit is also included, figure 17. The actuator input can now be modified by the on-board computers to increase safety and performance by giving feedback to it with sensors [51].

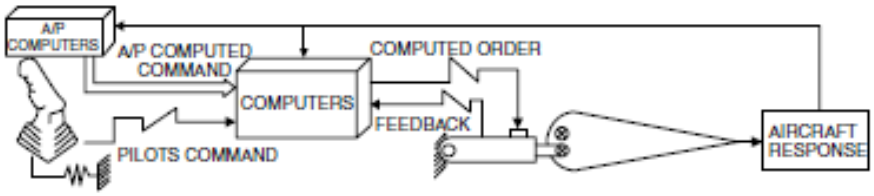


Figure 17: Fly-by-wire concept

The state of art nowadays is in general: mechanic commands for small aircraft (like gliders), hydraulic or fly-by-wire for medium and big aircraft and in some military planes fly-by-light.

Actuators can also be linear or rotary depending on their objective. In general the elevator, ailerons and rudder are moved by linear actuators while flaps and slats need rotary ones.

Another important concept to consider is redundancies. Every surface needs to have redundancies in order to fulfill the safety requirements imposed by the legislation. This can include redundant actuators per control surface or redundant power supply lines for the actuators. Which means than more than one hydraulic or electric line is needed, usually both are combined. All these features are now explained in the following sub-chapters.

3.2.1. Control surfaces

The aircraft flight control surfaces are aerodynamic devices that allow the pilot to adjust and control the aircraft's flight attitude and position. They are divided into primary and secondary depending on their function. A general schema of all the classic surfaces with their positioning and nomenclature can now be seen in figure 18.

3.2.1.1. Primary

Primary surfaces are used to control the attitude of the aircraft around the three control axes: pitch, roll and yaw. They are constantly active during flight. In general the elevator is fixed in a position during cruise and the ailerons and ruder are in zero position but their acting times should be fast enough in order to compensate gusts or fast manoeuvres. This means that they are active during the whole mission profile even if they are not being used in that specific moment. These devices use linear actuators.

Rudder

The rudder is used to control the yaw angle. It is usually mounted on the

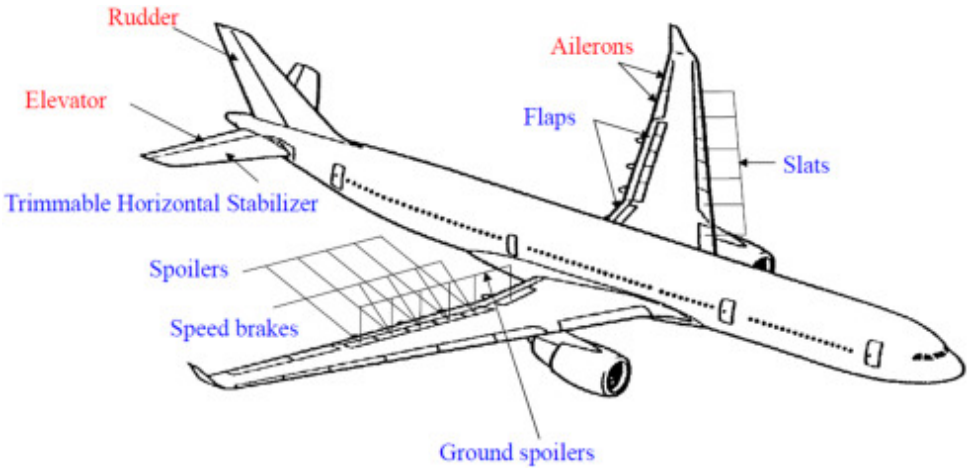


Figure 18: FCS schema

trailing edge of the vertical stabilizer, in the vertical empennage. It is controlled by the pedals. If the pilot pushes the left pedal, the rudder deflects left and the opposite happens with the right one. Deflecting the rudder right pushes the tail left and causes the nose to yaw to the right. Centering the rudder pedals returns the rudder to neutral and stops the yaw, as visually explained in figure 19.

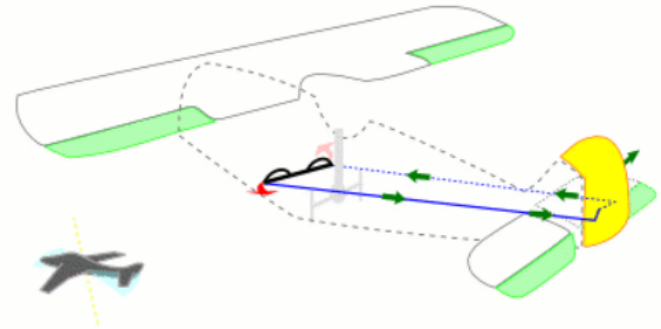


Figure 19: Ruder-pedal interaction

Ailerons

Ailerons control the roll speed, and hence the roll angle. They are typically mounted on the trailing edge of each wing near the wingtips and move asymmetrically when used. If the pilot moves the stick left, the left aileron goes up and the right aileron goes down. A raised aileron reduces lift on that wing and a lowered one increases lift, so moving the stick left causes the left wing to drop and the right wing to rise. This causes the aircraft to roll to the left and begin to turn to the left. Centering the stick returns the ailerons to neutral maintaining the bank angle. This behaviour is represented in figure 20

Elevator

The elevator is a movable surface located in the horizontal stabilizer, which can be also be movable itself. The elevators move up and down together symmetrically.

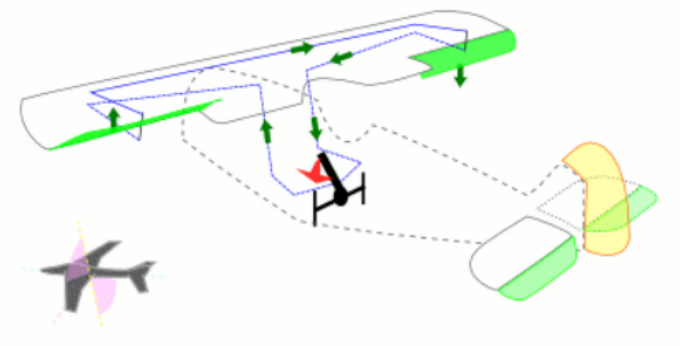


Figure 20: Ailerons-stick interaction

When the pilot pulls the stick backward, the elevator goes up. Pushing the stick forward causes the elevators to go down. Raised elevators push down on the tail and cause the nose to pitch up, as shown in figure 21. This makes the wings fly at a higher angle of attack. Centering the stick returns the elevators to neutral and stops the change of pitch. The opposite happens in canard configurations.

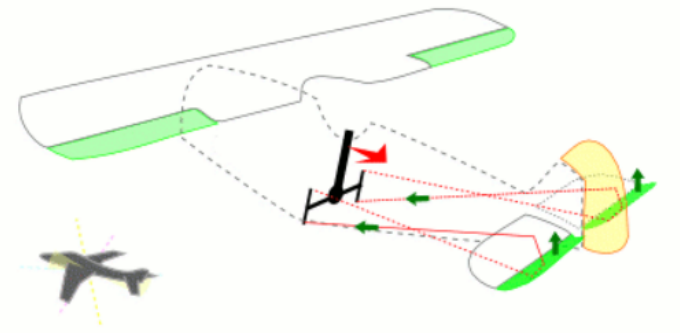


Figure 21: Elevator-stick interaction

3.2.1.2. Secondary

The secondary control surfaces are only used in some specific segments of the mission profile (approach, landing, take-off...). Their main function is to modify the wing geometry to increase lift and/or drag during that specific segment.

Flaps and Slats

Flaps an slats are high-lift devices used to increase the lift during take-off, and drag during descend, approach and landing. This devices are different to the previous ones since they usually have rotary actuators instead of linear ones, as it can be seen in figure 22. They usually have the Flap Power Control Unit inside the fuselage and this component moves the shafts transferring power to the respective devices [52, 53]. For this reason this actuators and supply lines have to be considered different from the ones of the other FCS subsystems.

Flaps are mounted on the trailing edge of each wing. They are deflected down to increase the effective curvature of the wing. Flaps raise the maximum lift coefficient of the aircraft and therefore reduce its stalling speed. Slats are positioned in the

leading edges, also for lift augmentation, and are intended to reduce the stalling speed by altering the airflow over the wing. There are multiple types for each of them depending on the maximum lift coefficient that the aircraft needs. For instance flaps can be plain, split, slotted, flap Fowler, flap Gurney... And slats can be nose flap, Krueger slat, leading edge drop... These devices are always used symmetrically (in both wings at the same time).

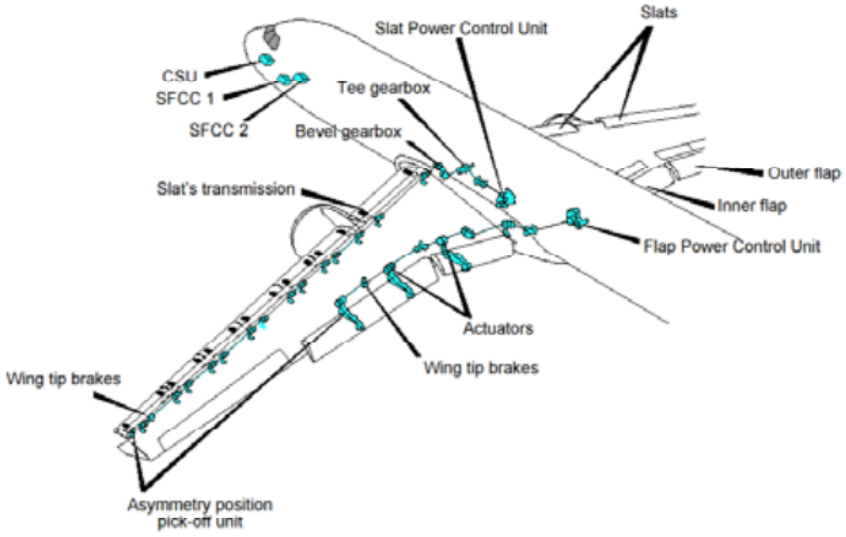


Figure 22: A320 high lift actuation system, from [54]

Spoilers

Spoilers are used to disrupt airflow over the wing and greatly reduce lift and increase drag, that is why they are also known as lift dumpers. During landing they are symmetrically activated as soon as the aircraft touches ground to destroy lift and stick the plane to the floor avoiding return to air. They can also be used asymmetrically to roll during flight since they increase drag and reduce lift in that wing. Using spoilers instead of ailerons can be used to avoid some adverse yaw problems, usually a combination of spoilers and ailerons is used for roll turns.

This devices use linear actuators, as well as ailerons for instance, as it can be seen in figure 23. The most external ones are used for this since the roll moment they generate is bigger, they are usually called "flight spoilers" while the inner ones are only used during landing and they are called "ground spoilers", even though all of them are used during landing.

3.2.2. Actuators

The objective in this chapter is to explain the aircraft-used actuators that are used nowadays and that could be considered in the FCS architecture modeling. The most used one is the conventional hydraulic servo actuator (HSA), used for example in the A320. The new technological concepts started moving forward to more electric aircraft and a new concept appeared, the power-by-wire (PBW). This lead to

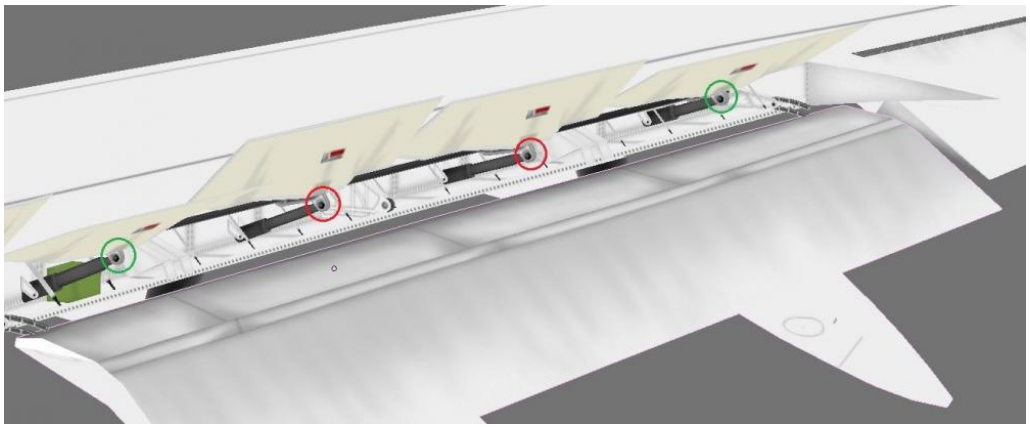


Figure 23: Spoiler actuators example, from [55]

a tendency of eliminating hydraulic systems from the aircraft and substitute them with more electrical systems (lighter and more reliable). With them new concepts came, like the Electro-Hydrostatic actuator (EHA) or the Electro-Mechanical actuator (EMA). Supplying with hydraulic pipes and using electrical cables for energy transmission, essentially increases the reliability and energy efficiency of an actuator system [56]. The last actuator, called electric back up hydraulic actuator (EBHA) is a hybrid between both concepts.

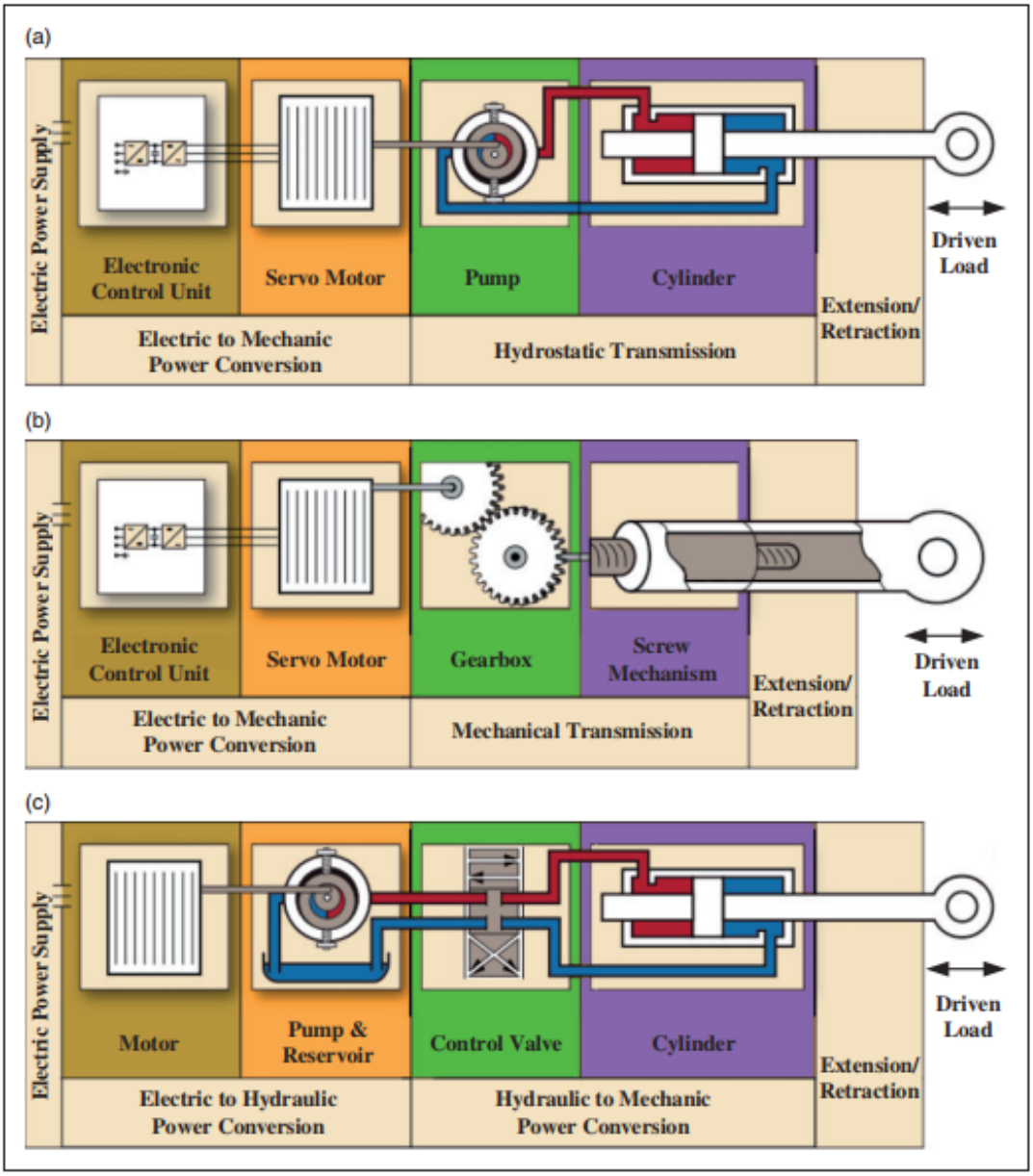
3.2.2.1. Conventional hydraulic servo actuator (HSA)

This system has been used in several commercial aircraft, like the A320. It consist on having more than one hydraulic lines to supply the aircraft systems (FCS, landing gear...). Airbus and Boeing used three lines on their planes, like this every system is given hydraulic power from at least two of them. In case of failure of one all the systems can continue working. The main problem is that the hydraulic lines are heavy and add lots of subsystems from the pumps to the actuators and rest of users. Removing these lines and substituting them by electric cables is the objective of the new aircraft concepts, like A350 or B787.

3.2.2.2. Electro-Hydrostatic actuator (EHA)

The electro-hydrostatic actuator (EHA) is one kind of PBW actuators. It also moves the actuator hydraulically but the difference now is on the power supply. In the HSA the hydraulic power came from the central hydraulic system. Now in the EHA each actuator has its own local hydraulic circuit that is given power by an electrically driven motor [57]. This source is usually a three-phase AC power which drives a variable speed electric motor.

With this concept the central hydraulic unit is substituted by a local hydraulic system for the actuator. Hence all the hydraulic lines and big and heavy hydraulic pumps and systems are removed and substituted by batteries and electric systems to power the electrical motors. Figure 24 shows the difference between this two actuators and anticipates the next one.



. Power-by-wire actuators and HSA composition. (a) EHA, (b) EMA, and (c) HSA.

Figure 24: Actuators comparison, from [58]

3.2.2.3. Electro-Mechanical actuator (EMA)

The electro-mechanical actuator replaces the electro-hydraulic powering from the EHA with a electric motor and a gearbox assembly. It is also considered a PBW actuator but while the EHA replaces the linear actuators in the more-electric aircraft, the EMA is the more-electric version of the screw-jack actuators. These ones are slower but resist bigger loads which are required in some cases like the THS (Trimmable Horizontal Stabilizer).

3.2.2.4. Electric backup hydraulic actuator (EBHA)

The last actuator is the electric back up hydraulic actuator (EBHA). It is a hybrid between EHA and HSA. Basically it consists in a HSA actuator with a second hydraulic supply line like in the EHA. In normal operation the central hydraulic line is moving the actuator and in case of failure (backup mode) the local unit drives the hydraulics. It could be said that it is a HSA that works as a EHA in case of central hydraulic unit failure. The advantage is that while in the conventional actuator at least two hydraulic lines are needed (to use the second one in case of failure), now just one is required. Figure 25 represents the actuator concept.

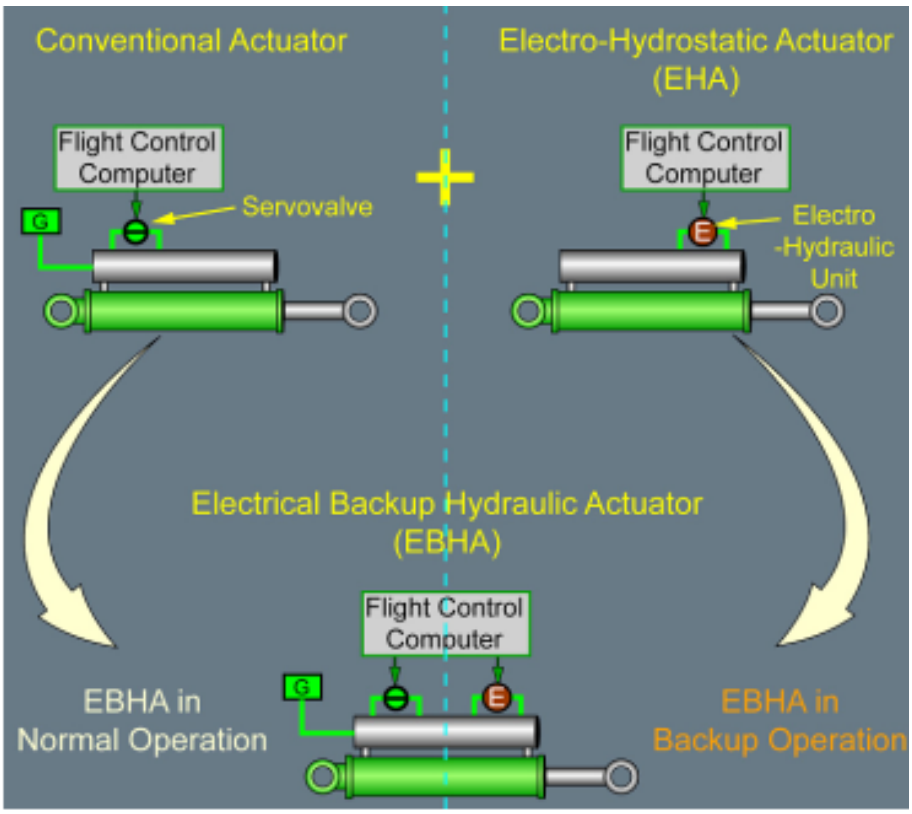


Figure 25: EBHA actuator concept

3.3. FCS Architecture Examples

Three examples are now shown for a better understanding of the system. In these aircraft the redundancies in actuators and supply lines have been solved in different ways, some of them using the more-electric aircraft concept. All the pictures and information have been taken from Airbus public database. The first example is the A320, with full hydraulic FCS and the oldest of the three. Then, the A380 shows how Airbus started removing hydraulic lines and replacing actuators for more-electric ones. And finally the A350, which is the newest model and also shows the same tendency as the A380. The last two examples are big aircraft which need more control surfaces than the A320, all these aspects will now be commented.

3.3.1. A320

The A320, which started operating in the late 80's is the oldest of the three examples. This aircraft had a full hydraulic flight control system since during these years the electric actuators and systems where not yet implemented. Hydraulic actuators have been used on civil aviation for a long time since they are very reliable and provide high actuation forces and speeds [59]. So all A320 FCS actuators are electrically-controlled and hydraulically-activated (HSA servo actuators). The FCS architecture can be seen in figure 26, the main characteristics are now being commented.

The control surface architecture consist on five spoilers and one aileron per wing, one elevator (one surface on each side) and one ruder. Three hydraulic lines (blue, green and yellow) are present in order to fulfill the required redundancies. As it can be seen each aileron has two actuators, so in case of failure of one of the hydraulic lines, the surface can still be moved with the redundant actuator which is powered by the redundant line. The same reasoning goes for each of the elevators. The ruder has three servos, each of them linked to one of the hydraulic lines. The flaps and slats also have two actuators driven by two different lines. Spoilers work differently, the don't have actuator redundancy since if they fail the are still 4 more spoilers in that side of the wing. So the redundancy relies on the number of surfaces rather than on the number of actuators.

The green and yellow circuits are both powered by an engine driven pump while normal operation. The blue one has a pump and a ram air turbine pump as backup for emergency situations. The actuators are distributed so that the aircraft can be controlled in the three axes by each of the hydraulic circuits independently. In case of failure of one, or even two hydraulic lines, the plane can still be operated safely. The computer redundancy (ELAC, SEC, FAC) can also be appreciated in the diagram. These hydraulic circuits also supply other subsystems as the landing gear actuators.

3.3.2. A380

The A380 is the biggest Airbus civil aircraft. This aircraft started flying in the middle 2000's. The main change in the FCS architecture is that the blue hydraulic

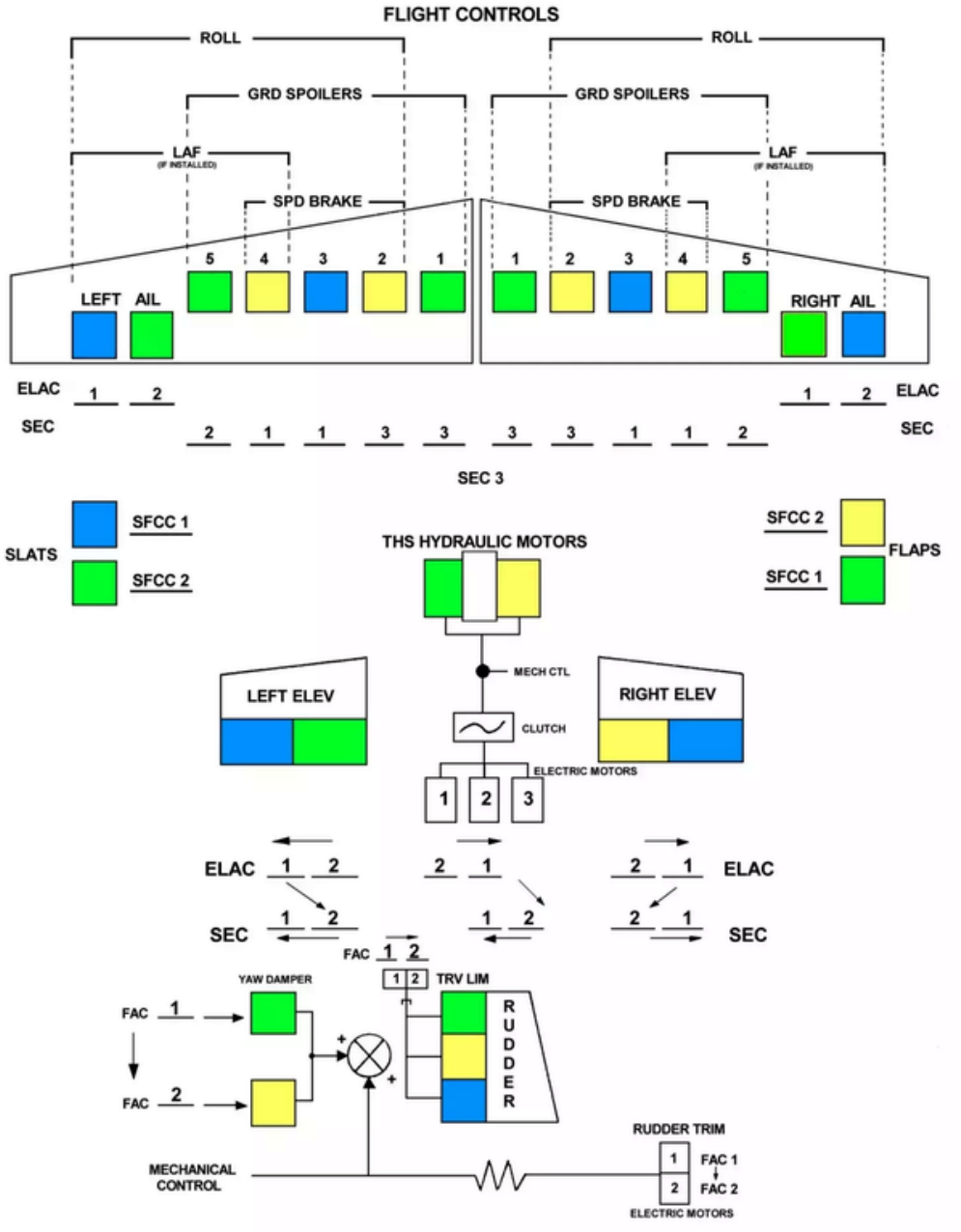


Figure 26: A320 FCS architecture, from [60]

circuit was removed and substituted by electric power supply lines. Due to its size it needs more control surfaces than the previous example. The surface architecture consist on three ailerons, two elevators and eight spoilers per side and two ruder surfaces.

The final result is that due to the more-electric tendency the main architecture consists in two hydraulic lines (green and yellow), and three electric lines (two driven by the engines and one emergency one driven by a RAT generator).

Figure 27 shows the type of actuators that were used. In normal mode all the surfaces are hydraulically moved by the yellow or green line. In case of failure there are different redundancies that can move the servos instead, these are:

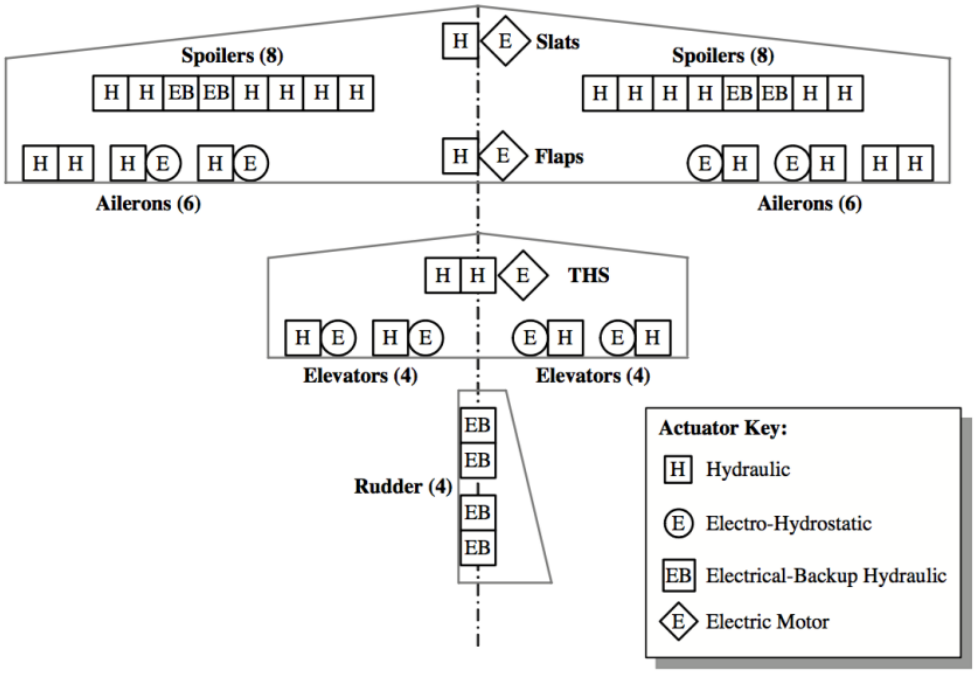


Figure 27: A380 FCS architecture, from [61]

- EHA: represented in the figure as an E in a circle. They are used as redundant servos for the elevators and some ailerons. If the main HSA fails due to a failure in the hydraulic line then they start operating.
- EBHA: represented by a EB in a square. They are used in some spoilers and in all the ruder servos. The operate as HSA but in case of failure in one of the hydraulic circuits the start operating as EHAs.
- EMA: represented as an E inside a diamond. They are used as backup for the screw-jack actuators which in this case are the flaps, slats and trimmable horizontal surface.

All the electric servos are powered by one of the two normal-mode electric lines and have a backup mode powered by the third RAT-electric line.

3.3.3. A350

A350 is the newest Airbus civil aircraft. Analysing this model will allow us to better understand which are the current state of art regarding the flight control system and comparing it with the previous ones will give us an insight about how the tendencies are evolving.

Figure 28 shows the architecture of the A350. It can be seen that, as well as the A380, two hydraulic lines are still present. But as a difference with its predecessor, now two electric lines are for emergency and just one is used under normal conditions.

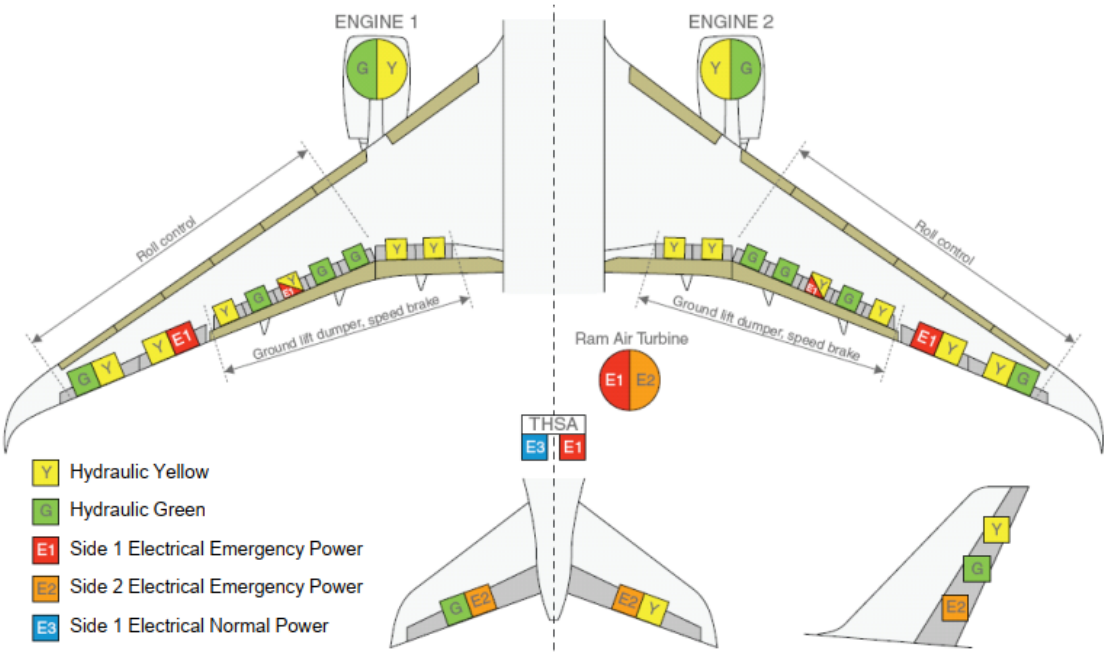


Figure 28: A350 FCS architecture, from [62]

The surface architecture consist on seven spoilers, two ailerons and one elevator per aircraft side, and one ruder surface. As before, each aileron and elevator has two servos, the ruder has three and each spoiler just one. Some ailerons are provided by the two hydraulic circuits while others have EHA servos as backup. The ruder and elevator also have EHA actuators as the A380. All the spoilers, except two, use HSA servos. The two spare ones have EBHA actuators.

In this architecture the THSA has completely lost the hydraulic components and two EMA servos are used. As before, the two hydraulic lines are powered by electric compressors driven by the engines. The normal electric circuit is powered by electric generators and the two emergency ones are supplied by ram air turbines. As a summary, the aircraft can be controlled just with one of the normal lines and the two emergency ones.

3.4. Considered FCS architectures

The FCS architectures considered will depend on the software limitations. In this thesis, the sofware used to size the on-board systems was ASTRID, developed by Politecnico di Torino.

First of all, the control surfaces architecture that was chosen is the one from A320, since the analysis considers a family with similar characteristics. This means that the model will consider 17 surfaces. The THSA and thrust reversals are not considered under this analysis

- 2 ailerons, with two actuators each
- 2 elevators, with two actuators each
- 1 rudder, with three servos
- 10 spoilers, with one actuators each
- 1 flaps central power control unit
- 1 slats central power control unit

Once the number of surfaces and actuators have been defined, the architecture choice now is hydraulic or electrical actuators. If the hydraulic option is chosen all the servos will be conventional hydraulic actuators (HSA) while if the electric option is selected two other options appear. If the actuator is linear (spoilers, ailerons, elevator and rudder) the actuator would be a electro-hydrostatic one (EHA). On the other hand, if the actuator is rotary (flaps and slats) the modeled actuator would be a electro-mechanic one (EMA). Electric backup hydraulic actuators (EBHA) are not being taken into consideration.

Conventional and electro-hydrostatic actuators have a lot of common components since the main difference is were the energy for the hydraulics is coming from. While the difference with electro-mechanic ones is quite big. All these aspects should be reflected in the commonality tool.

4. Tools

Several tools will be used in this analysis, since it is a MDO problem. The main disciplines used include a tool for Overall Aircraft Design (OAD) called OpenAD, a tool to size and calculate the performance of the aircraft's on-board systems called ASTRID, a tool to estimate the fuel burn of an aircraft family given certain routes and another one to model the aircraft family commonality. These last two tools were specifically designed for this thesis.

Disciplinary tools are integrated in a PIDO environment (Process Integration and Design Optimization), a tool to include all the others into the same workflow. This tool is called RCE and it has already some other useful tools included like optimizers, mergers, switches or convergers. RCE uses CPACS as common language in this case, as explained before in chapter 1.

4.1. OpenAD

OpenAD (previously called VAMPzero) is an Overall Aircraft Design tool developed in DLR [63, 64]. It is based on handbook methods taken from literature and its main objective is to preliminary size an aircraft from its TLARs [65]. This TLARs are given as an input writen as a CPACS file, and the output is another CPACS file with the most important data about the aircraft and its geometry.

The input file can highly vary and depending of the values given convergence will or will not be reached. One useful input value is the engine performance switch, if it is activated then OpenAD will also calculate engine performance giving more precise values for the SFC. In this thesis this switch was activated in the second OpenAD run giving the power off-takes and bleeding parameters in the input after having sized the on-board systems.

4.2. ASTRID

ASTRID (Aircraft on Board Systems Sizing and Trade-Off Analysis in Inital Design) is a tool developed in the Polytechnical University of Turin [66]. It is used for aircraft on-board systems sizing. Its main input is the conceptual design of a specific aircraft, the mission profile and some geometric and aerodynamic parameters, as well as some specific data for each of the subsystems.

Its functioning works as follows: receives the aircraft TLARs, sizes the subsystems, sizes the electric and hydraulic system and gives the results. The subsystems taken into consideration are: avionics, flight control system, landing gear, anti-ice devices, environmental control system, bleed systems, fuel system and it can also consider the furnishing systems and auxiliary power system.

The output is a detailed list of characteristics per system. The mass of each of them as well as the total on-board systems mass is the most noteworthy. But it also provides with data about the engine bleeding, pneumatic pressures, and the power

budget of the aircraft. This last one is given per subsystem and per mission profile segment. For instance, one value that can be found would be the power needed by the flight control system during descent, taxi-in or during cruise. Another example would be the value of the total power off-takes (average and/or maximum) needed by the aircraft on each segment.

4.3. RCE

Disciplinary tools are integrated in the RCE environment (Remote Component Environment), a PIDO (Process Integration and Design Optimization) developed by the DLR [67]. Hence RCE offers a graphical user interface to connect all the analysis modules and run different stages of the design process [64]. It includes several useful features by default and also allows to implement your own tools into the workflow. As an example, some of the default tools that have been used in this thesis are:

- XML mergers: useful to merge xml files (CPACS files). They need an xsl file and perform the specified mapping in it with the two files that are given to be merged.
- Converger: it automatically compares parameter values on each run and gives the appropriate file to the next tool once the convergence is reached. The interface allow to compare more than one criteria and to filter several files at the same time.
- Optimizer: it gives the new design variables on each run after having analysed the results of the previous ones. Any kind of optimization algorithms can be used and but their code should be given externally.

4.4. BRICS

Due to intellectual property issues ASTRID cannot be run from DLR. For this reason it has to be runned from polito but at the same time included in an automated way into the workflow. One way of doing this is through BRICS [68], a tool developed by NLR (Netherlands Aerospace Centre).

This tool can be implemented into RCE and will substitute ASTRID in the exact same positions as ASTRID would be. It works as follows:

1. The input file for ASTRID is given to BRICS inside the workflow.
2. BRICS creates an online folder with the input file and sends an email with a key word to the other user, in this case Polytechnic University of Turin.
3. The person who received the email runs ASTRID and BRICS in another RCE workflow and gives the address for the input and the key word.
4. The BRICS tool gives the input to ASTRID, waits and receives the output. Then it writes it in the online shared folder.
5. BRICS, in the first workflow, reads ASTRID's output and gives it to the next

tool

Like this ASTRID is fully implemented into the workflow without actually having the source code, just giving an input and receiving the output. BRICS can specified to be runned into a convergence loop with a "multi-task" feature, so it will automatically create multiple shared folders, each of them with one of the files of each iteration

4.5. Fuel tool

The objective of this analysis is to calculate the fuel consumption of that aircraft family. In this case there are three aircraft that will fly certain routes. Depending on the performance of each of them and on which aircraft flies which routes, the fuel consumption will be higher or lower. The three family members have already been sized in the previous calculations, so the performance is an input. Hence the code will decide which plane flies each route once the routes are given and then calculate Routes the fuel needed to fly each of them. Summing all this fuel tons and dividing the total by the total amount of flights results in an average fuel consumption per route. This value represents how well the aircraft family performs in terms of fuel consumption and operating costs.

4.5.1. Routes

The most important part of this analysis is getting a realistic market segment for the aircraft family. The market segment can be represented by several routes that will be flown with certain payload and range and a certain amount of times. This information can be taken from the USA Bureau of Transportation Statistic that publishes all this information to the public [69]. The data extracted for this study corresponded to all the available United-States-tracked civil flights that were done during 2018, national and international. Takin the whole year will minimize the seasonal impact on the routes. Also, only the A320-like models were considered since the aircraft model that will be studied corresponds to it. The cargo versions were also removed, so just passenger civil routes are extracted. The aircraft models considered are: A318, A319, A320-100, A320-200, A320-200n, A321, A321-200n, B737-100, B737-200, B737-300, B737-400, B737-500, B737-600, B737-800 and B737-900, making a total of 15 planes.

The relevant result is a list of routes with the payload and range (converted to kilograms and kilometers) and the times that each route was flown. This list is one of the inputs for the python tool implemented into the workflow. The total number of routes considered is 139028 and the total number of flights is 3028069. A histogram can be made in order to better understand and visualize this information [70]. This consists on dividing the payload-range diagram into cells and then counting the number of flights on each of them. Then different colors are assigned depending on the total amount of flights per cell, the next figure 29 is obtained:

The most flown routes are now seen as well as all the "off-design" points that

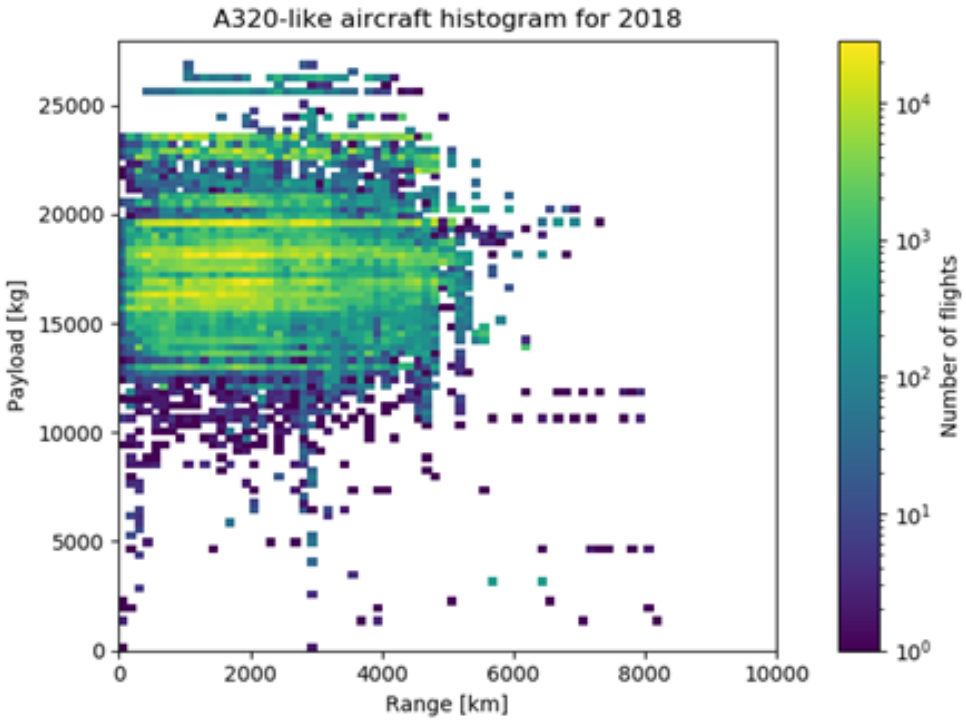


Figure 29: A320-like aircraft histogram, 2018

the aircraft should fly. This payload-range region shall be carried out by the three aircraft of the family.

Now, the average fuel consumption shall be calculated in order to have an index to represent how well and efficiently this market region is fulfilled. The first step is calculating the fuel tons needed for each of the routes, which is done in the following sub-chapter. Then summing all this fuel mass contributions and dividing by the total amount of flights will give as a result the average fuel consumption per aircraft and per flight.

4.5.2. Fuel consumption model

The total amount of flights and routes is huge, so an analytical method is wanted in order to have a fast calculation. The best option is the well-known Breguet equation, which can be used with only a couple of assumptions. The generic form of the equation is:

$$R = \int_{t_1}^{t_2} \frac{dR}{dt} dt = \int_{m_2}^{m_1} \frac{V}{f_{cons}} dm \tag{3}$$

In which the range can be written as a function of time. Then it can be transformed into a function of the speed and fuel consumption. The engines used belong to the jet engine category so the fuel consumption can be expressed as the thrust multiplied by the non-dimensional specific fuel consumption.

$$Jet\ Engine : f_{\text{cons}} = T \cdot SFC \tag{4}$$

If the whole flight is represented as a cruise from the origin airport to the destiny, all at the same height the expressions can be transformed:

$$\text{Cruise: } T \cdot SFC = D \cdot SFC = W \cdot \frac{C_D}{C_L} \cdot SFC \tag{5}$$

By also assuming that the polar is quadratic with three terms which are the parasite drag, wave drag and induced drag:

$$\text{Quadratic Polar: } C_D = C_{D0} + C_{Dw} + k \cdot C_L^2 = C_{D\,\min} + k \cdot C_L^2 \tag{6}$$

And finally, introducing the next expressions the integration is much simpler:

$$E_{\max} = \left(\frac{C_L}{C_D}\right)_{\max} = \frac{1}{2\sqrt{k \cdot C_{D\,\min}}} \tag{7}$$

$$C_{Lopt} = C_L\left(E_{\max}\right) = \sqrt{\frac{C_{D\,\min}}{k}} \tag{8}$$

The maximum efficiency is a constant that represents how aerodynamic the aircraft is, it is basically the maximum efficiency achievable under certain conditions, which in this case are a cruise at constant height and speed. The optimum lift coefficient is the lift coefficient needed to fly with maximum efficiency. They are both constants used to make the equation simpler and easier to integrate. So this means that the aircraft will not be flying at optimum lift coefficient, it is just a mathematical tool. The final equation is:

$$R(km) = \frac{2VE_{\max}}{gSFC}\left[\operatorname{atan}\left(\frac{C_{Li}}{C_{Lopt}}\right) - \operatorname{atan}\left(\frac{C_{Lf}}{C_{Lopt}}\right)\right] \cdot 1000 \tag{9}$$

This equation represents the range for a cruise in which the speed and the altitude are constant and hence the lift coefficient varies during the mission [71]. The lift coefficients can be expressed as follows:

$$C_{Lx} = \frac{g \cdot mass}{0.5 \cdot \rho \cdot S_{wing} \cdot V^2} \tag{10}$$

In which all is known except for the mass, since the density is a function of the altitude. The final mass can be easily estimated with the landing mass plus the reserve fuel and inside the initial mass the fuel mass is contained:

$$\text{mass}_{\text{final}} = LW = \frac{OEW + PL}{1 - \alpha} \tag{11}$$

$$mass_{\text{initial}} = LW + fuel \tag{12}$$

So the fuel weight can be now calculated knowing the payload, contingency and operational empty mass. The summary of the assumptions that were made is:

- Mission profile represented as a cruise all the time
- Constant cruise speed and altitude
- Jet engine
- Quadratic polar with wave drag included

4.5.3. Python tool

The tool takes two inputs, which are the routes and the input CPACS file. In the routes, as it was said before, all the payloads and ranges are contained. The CPACS input has all the information about the aircraft family stored. So the first step of the program is reading all the parameters needed for the equation. Some of they were not in the CPACS file exported by OpenAD, so another step had to be done. This consisted on reading the internal csv file internally calculated in the VampZero code and exporting it in RCE as an output. Like this the necessary values could be read and merged into a "CPACS toolspecific" address. The variables needed are the following:

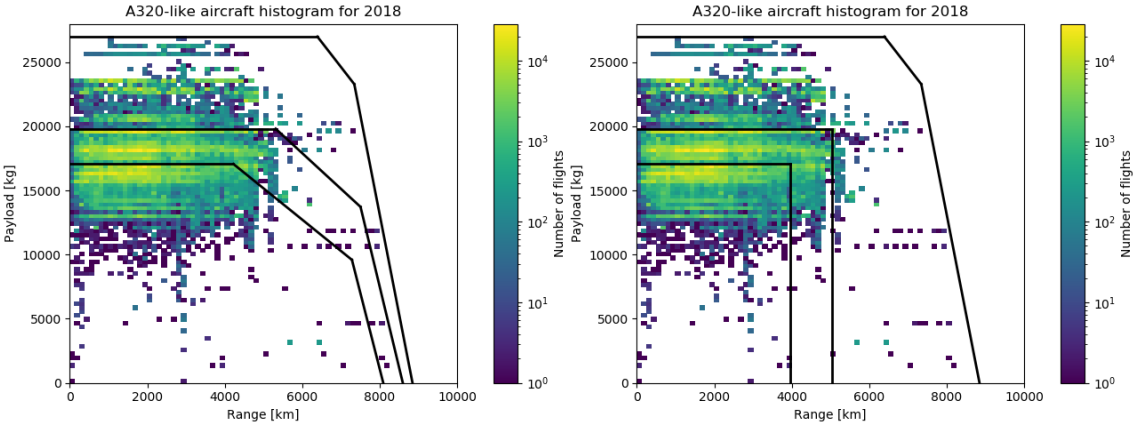
- **MTOM:** from the CPACS input file, constant for each aircraft
- **Max Payload:** from the CPACS input file, constant for each aircraft
- **Range for Max Payload:** from the csv file, constant for each aircraft
- **Contingency:** from the csv file, constant for each aircraft
- **OEM:** from the CPACS input file, constant for each aircraft
- **Wing surface:** from the CPACS input file, constant for each aircraft
- **Altitude:** from the csv file, constant for each aircraft
- **Speed:** from the csv file, constant for each aircraft
- **SFC:** from the csv file, constant for each aircraft
- **Cd minimum:** from the csv file, constant for each aircraft
- **k (induced drag factor):** from the csv file, constant for each aircraft
- **Payload for that route:** from the routes file, variable for each aircraft and route
- **Range for that route:** from the routes file, variable for each aircraft and route

With all these inputs the code shall assign each of the routes to one of the aircraft in the family and then calculate the fuel for that route. Then multiplies it by the

times that route is flown and later sums all of them. Once the total fuel tons are calculated the result is divided by the total number of flights to get the "average fuel burn per route and per aircraft". This value is written into a new "CPACS toolspecific" address and the final output is a copy of the input CPACS file merged with this new part.

For validation reasons another output was created. This consist in a csv file as the input one, with all the routes and payloads, but with two new lines, one with the value of the fuel mass for that route and another with the aircraft that will fly it (1 for the bigger, 2 for the second bigger and so on).

Now the selection criteria for each route will be explained. This means how to select which of the three aircraft is going to fly each route. The code was made so that more than three aircraft could be analyzed, as well as two or even one. The payload-range diagrams of the three members are represented in the next figure 30a. The selection is done so that every plane will fly the routes which are in the region below its maximum payload and its design range, not doing the ones that have already been flown by another aircraft, as figure 30b shows. The remaining points are carried out by the bigger aircraft. This requires a previous analysis to be sure that the bigger aircraft can really fly all the routes, as it can be seen in figure 30.



(a) PL-R diagrams for each family member **(b)** Section flown by each family member

Figure 30: Histograms

4.6. Commonality tool

The commonality among members should be sensitive to three main effects, which are:

- Aircraft with the same subsystems. This means the the same exact subsystems that one aircraft uses is also installed in other family member.
- Aircraft with the same architectures but different subsystems. Which means that the aircraft have the same architecture but each of them has their own optimized subsystem with that architecture.

- Aircraft with different subsystems' architectures and hence different subsystems.

Two design variables were defined to be able to share subsystems. Now the commonality tool should be able to represent the other two effects. The model proposed is the Product Line Commonality Index (PCI), which is a metric for evaluating design commonality in product families [9].

4.6.1. Product Line Commonality Index

This method allows to estimate and establish a commonality value for each aircraft family design. This metric differentiates the architectures by checking which components are shared and if they are exactly the same component or a scaled version of each other.

The index is defined as follows in equation 13:

$$\mathrm{PCI} = \frac{\sum_{i=1}^{P} \mathrm{CCI}_i - \sum_{i=1}^{\mathrm{P}} \mathrm{Min\,CCI}_i}{\sum_{i=1}^{\mathrm{P}} \mathrm{Max\,CCI}_i - \sum_{i=1}^{\mathrm{P}} \mathrm{Min\,CCI}_i} \times 100 \tag{13}$$

Where CCI means, Component Commonality Index. The index "i" refers to the components and goes from 1 to the total amount of components which is "P".

It can be noticed that the metric is based on setting a maximum and minimum possible commonality values for each component inside the product family and then comparing it with the commonality of that design in particular. Introducing the models on each of the terms equation 13 can be transformed into equation 14.

$$\mathrm{PCI} = \frac{\sum_{i=1}^{P} n_i \times f_i - \sum_{i=1}^{P} \frac{1}{n_i^2}}{(P \times N) - \sum_{i=1}^{P} \frac{1}{n_i^2}} \times 100 \tag{14}$$

Where "N" is the number of products in the product family, which is three in this analysis (three aircraft). The parameter "ni" is the number of products in the product family that have component "i". And "fi" is a size factor for component "i".

It can be seen that the maximum reachable commonality is modeled as the number of components multiplied by the number of products. While the actual commonality only contemplates the common components and uses the size factor to penalize when one component is common but different sized. Like this the previously commented effect of measuring common architectures with different sizes is taken into account. The minimum commonality is just a reference for both values that is set in order to be able to create the index.

The model is selected but we still need to choose which components will be considered, how to model the size factor and include all inside the tool. These steps are explained in the next sub-chapter.

4.6.2. Python Tool

This tool was specifically developed for this thesis and might or not be used for further analysis. The inputs are the architecture of the ECS and FCS of each of the aircraft of the family and the converged output files from ASTRID for each of them. The PCI is now used for both subsystems but with one difference. There is more information about the ECS than about the FCS. This is due to the fact that there are 3 design variables for the ECS while for the FCS there is just one. Both subsystems are evaluated separately and the final commonality result will be the average of both of them.

Environmental Control System Commonality

After several small analyses it was determined that the bigger the amount of components, the more difficult it was to differ the ECS architectures since ASTRID does not give components details, only information about the subsystem itself. So the number of components should not be very high but on the other hand, it has to be big enough to represent the architectures. The final list of components considered eight of them which are: primary heat exchanger plus main heat exchanger, compressor plus turbine, second turbine, ACU fan (fan joined to the turbine), ground fan, re-heater plus condenser, electric fan and fuselage inlets [72].

The compressor, turbine and heat exchangers are common to all the architectures. The ACU fan is present in 3 Wheel BC and 4 Wheel BC, while in the 2 Wheel BC the ground fan substitutes this component. The second turbine is only used for the 4 Wheel BC. The re-heater and condenser are components which are used in sub-freezing (also known as high pressure) architectures. And finally the electric fan and fuselage inlets are specific for the bleed-less concept architectures since they need this extra inlet plus fan to substitute the engine bleeding [72].

The code reads the architecture and assigns the corresponding components to each of the aircraft respectively. Once this is done the only parameter left is the size factor. The data from ASTRID is limited so 4 of the components do not have this factor (factor equals one) since it cannot be properly estimated. The other ones can be sized as follows:

- Turbines' size can be scaled with the mass flow since their size depends on the power requirement and this one depends on the mass flow.

$$\text{Turbine Power } = \dot{m} \cdot c_p \cdot \Delta T \cdot \eta_{\text{mec}} \tag{15}$$

- Compressors have the same behaviour as turbines.

$$\text{Compressor Power } = \frac{\dot{m} c_p \Delta T}{\eta_{mec}} \tag{16}$$

- Heat exchangers' size depends on their heat load and this one depends on the amount of air passing through it in both sides.

$$\dot{Q} = \dot{m}_{\text{hot side}} \cdot c_p \cdot \Delta T_{\text{hotside}} \tag{17}$$

So the conclusion is that the turbine, compressor, second turbine, re-heater and condenser factors will be created comparing the mass flows that go through each of their air conditioning units, since this is a representative value of their size.

This value was statistically calculated with the average absolute deviation. The values compared are the mass flow through the ACU and was read form ASTRID's output file. It is important to consider that sometimes one of the component can be shared between the first and second aircraft, or between the second and the third, or just between the first and the third or between all of them. So on each case the factor should be done only among the mass flows regarding the components in that specific case. Which means that the average absolute deviation has to be done between the mass flows of the corresponding aircraft in that moment. The model is the following.

$$\text{Average Absolute Deviation (AAD)} = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - m(X)\right| \tag{18}$$

Being "xi" the individual values and "m(x)" the average one. But in order to reduce this expression to a factor the AAD has to be represented as a percentage. Simply dividing by the average calculated in equation 18 and resting 1 to it we have the required factor. The final formula that has been used is represented in equation 19. Hence this value can vary from 0 to 1, being 1 if the mass flows are all the exact same and zero the limit in which the AAD equals the average (hypothetical case).

$$\text{Sizing factor} = 1 - \frac{AAD}{m(X)} \tag{19}$$

Flight Control System Commonality

In the case of the Flight Control System it has to be considered that there are only two possible architectures for three aircraft. All electric or all hydraulic actuators. This means that at least two aircraft will always have the same architecture and the commonality should not reach very low values.

It is important to also differentiate when two aircraft share the subsystem from when they just share the architecture. The PCI model without sizing factor an a low amount of components is used and the results are that when two aircraft share the FCS then the commonality is 1, if they just have the same architecture the value is 0.75 and if they don not share architectures the value will be 0.325 [72]. The problem is then reduced to five cases:

- **Case 1:** The three aircraft share subsystems
- **Case 2:** Two aircraft share subsystems and the other one has the same architecture
- **Case 3:** Two aircraft share subsystems and the other one has a different architecture

- **Case 4:** Aircraft do not share subsystems but the three have the same architecture
- **Case 5:** Aircraft do not share subsystems and just two have the same architecture

With the previous values in mind, the results for all the different cases are now summarized in table 3:

Description	**Commonality Value**
Case 1	1
Case 2	0.834
Case 3	0.55
Case 4	0.75
Case 5	0.467

Table 3: FCS possible commonality values with PCI

5. Methodology

The first step is defining the architectures for each of the subsystems. In order to do that a preliminary study was done, it was not implemented in the optimizer yet but it should be implemented in following studies to be able to evaluate more architectures. Then the disciplines and their connections are represented in a diagram to better understand the problem. With this in mind they are then implemented into a real workflow to proceed with its execution.

5.1. Modeling the Architecture Design Space

The first step is to model the systems' architectures. The possible options for the ECS and FCS architectures have been reviewed in chapters 2 and 3, now the exact design space shall be modeled. A novel methodology that enables modeling the design space from a functional perspective is used [73]. This methodology models the architecture design space using the Architecture Design Space Graph (ADSG), which maps functions to components.

Figure 31 shows the environmental control system function breakdown modeled with the ADSG. This is shown as an example to the reader to better understand the method. It is done from a subsystems level. First, the neutral function of the system shall be specified, which is delivering air to the cabin and making it suitable to the physiologic passengers' conditions. The ECS is divided into three main subsystems, as it was shown in chapter 2. The Bleed Air System extracts the air from the atmosphere and has an attribute that represents if this is done with a conventional bleeding or from a more electric plane perspective. Then the Air Conditioning System has two attributes that represent the type of Air Conditioning Unit (ACU) and the type of water extractor. The Cabin Pressure Control System just controls the cabin pressure and has no design decisions associated.

Figure 32 shows the flight control system function breakdown modeled with the ADSG. In this case the decision tree is more simple since there are only two options. These options are if the actuators will be hydraulic or electric. Each of them affects which other subsystems will be sized afterwards to provide the electric or hydraulic power.

The actual system's model goes more in depth into a component level. The functions specified are divided into smaller ones which can be fulfilled with different components. For instance, one function is "provide ram air on ground" and can be fulfilled by two different components which are an external ground fan or a fan joined to the ACU. If the first one is chosen then the final architecture is what we call two-wheel bootstrap cycle. But, if the second one is selected then a new function appears, which is: "move fan". Depending on the component that fulfills this new function the final result can be a three-wheel or a four-wheel cycle. Once the architecture design space has been modelled the MDO analysis can be done.

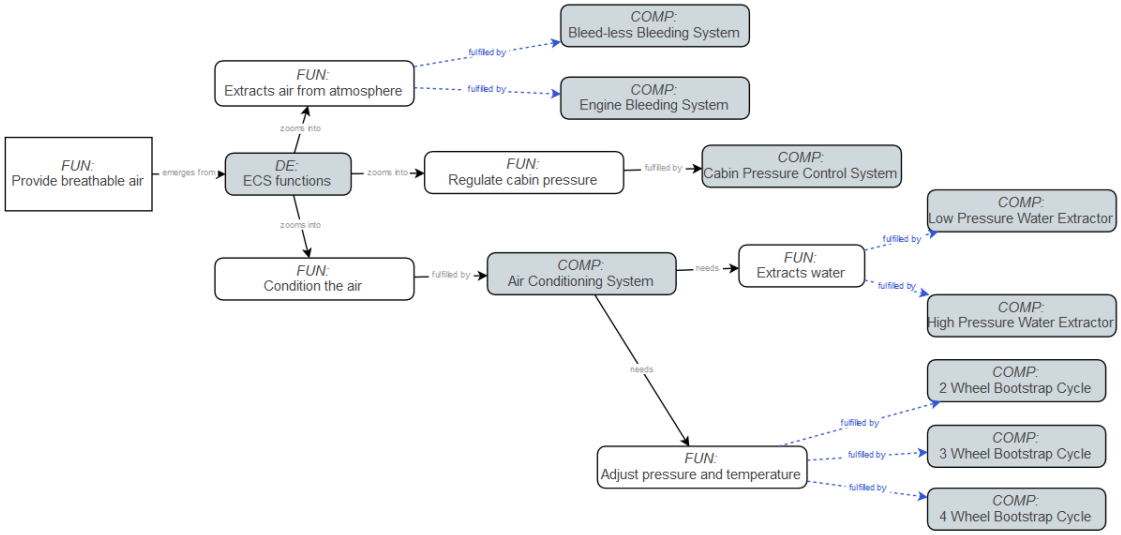


Figure 31: Architecture Design space Graph (ADSG) of the Environmental Control System

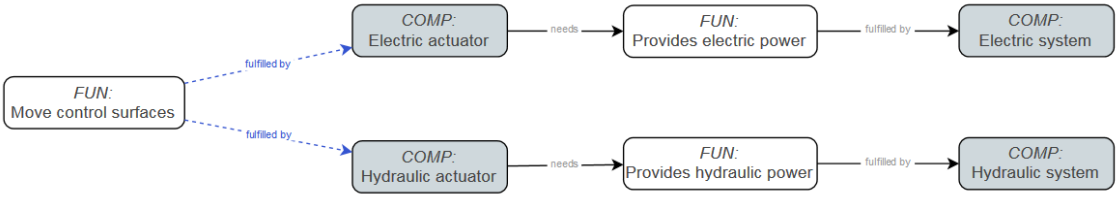


Figure 32: Architecture Design space Graph (ADSG) of the Flight Control System

5.2. XDSM

The XDSM (Extended Design Structure Matrix), is a powerful method that is used to represent workflows with their tools, inputs, outputs and connections. It is the extended version of a previous method called DSM [17]. It shows the data flow of each discipline. It also represents the convergence loops, or the optimization loops and clearly shows the different disciplines involved and how are they linked.

Two XDSM diagrams were made for this analysis. One is the reduced version that shows the general idea of the workflow. The extended version shows how the workflow remains for our specific analysis with three aircraft. They were made with MDAx, a tool developed by DLR which is used for MDO [74].

The reduced version is now shown in figure 33:

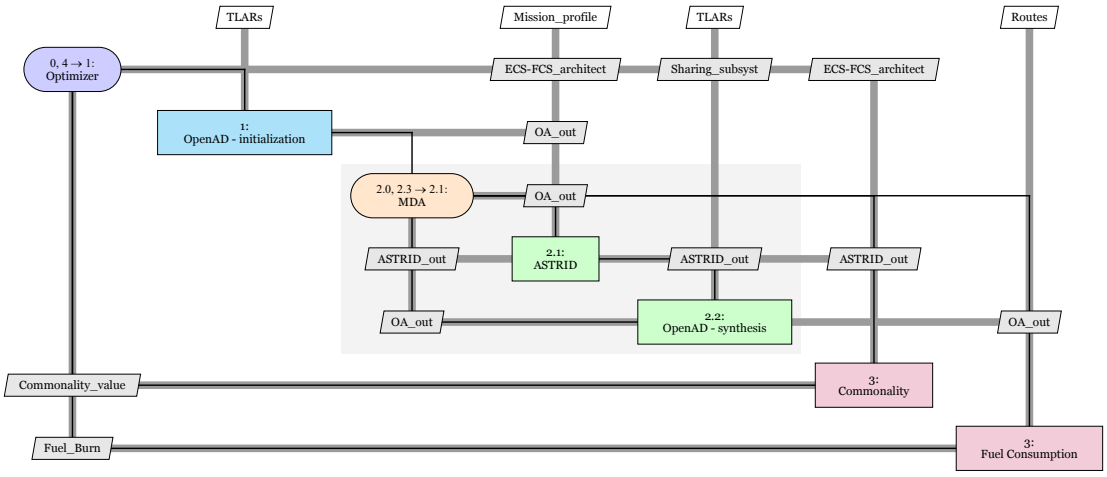


Figure 33: Reduced version of the XDSM

The diagram shall be read as follows. The tool's inputs are in the vertical lines while the outputs are in the horizontal ones. If the inputs are above that means that they come from a previous tool analysis, if the come from below, it means that they come from a loop. When an output is on the right side, it means that it will be used by another tool afterwards, while if an output is on the left it means that it will be used for a loop. The grey lines show data transference while the black line shows the order in which the tools are run. Now the tools and the order in which they are run will be explained. OpenAD, ASTRID, the commonality estimation tool and the fuel consumption tool were explained already in the previous chapter.

The order in this case is:

1. The optimizer creates the new design variables for that iteration, which are the architectures and sharing variables.
2. OpenAD reads the TLARs and creates the first file with the initial values for each of the aircraft.
3. Mass convergence loop (MDA). Runs ASTRID and the OpenAD until conver-

gence is reached and then all the masses and characteristics for each aircraft is defined.

4. Commonality and fuel consumption tools are run in parallel.
5. The optimizer receives the results of the fuel and commonality value and creates new design variables with this information.

Now all the steps are explained more in depth:

5.2.1. Optimizer

The selected optimizer is a multi-objective genetic algorithm. Multi-objective since there are two optimization variables to analyse and genetic since the design variables are discrete and these kind of algorithms can manage them. For instance gradient methods are not recommended since they have problems handling discrete variables in general since there is not a continuous front.

The algorithm will start the analysis with a given initial point and it will create the new design variables on each iteration after reading the two outputs (commonality and fuel burn).

The design variables are:

- **ECS Bleedless:** 0 for classic ECS architecture and 1 for the bleed-less concept. There are three of these variables, one for each aircraft.
- **Num Wheels:** 2 for 2 Wheel BC, 3 for 3 Wheel BC and 4 for 4 Wheel BC. These variable is present also once per aircraft.
- **Sub freezing:** 0 for non-subfreezing ACU (Low pressure) and 1 for subfreezing ACU (High pressure). There are also three of these variables.
- **FCS Power Supply:** 0 for electric actuators and 1 for hydraulic ones. There are three of these variables as in the previous ones.
- **Sharing 1 and 2:** 1 in case that the second aircraft will use the exact same subsystems as the first aircraft. Zero in case it will not.
- **Sharing 2 and 3:** 1 in case that the third aircraft will use the exact same subsystems as the second aircraft. Zero in case it will not.

There is a merger afterwards that changes the variables in case that the sharing variables are active. For example, if the "sharing 1 and 2" variable is active then this merger will remove the variables of the second aircraft and substitute them with a copy of the ones of the first family member. The same happens with "sharing 2 and 3". This makes the process slower cause the optimizer should understand that every time these sharing variables are active the result is the same even if it changes the architectures of the second and third model. But there is not a way of avoiding these extra runs.

5.2.2. External inputs

There are three external inputs in this workflow. These means that these inputs are constant and fixed during all the process. They are:

- **TLARs:** Top Level Aircraft Requirements for each of the three aircraft. They are written all together in the same CPACS file and they have the information for the first estimation with OpenAD. The main parameters that are in this file are the initial masses for the loop, geometric values for the wing and tail like sweep, tapper ratio or dihedral, range, passengers, take off lengths, engine values like turbine temperature or overall pressure ratio, fuel characteristics like density, etc... Most of the values were taken from the A320.
- **Mission profile:** The mission profile contains in a CPACS file the information of the segments in which the mission is divided and the altitude and Mach number for each of them. The segments are: pre-flight check, engine start-up, taxi out, taxi out flaps down, take off run, take off manoeuvre, take of landing gear up, take off flaps up, climb, cruise, descent, descent flaps down, approach landing gear down, approach, landing manoeuvre, landing run, taxi in flaps up, taxi in, engine shutdown and emergency. Making a total of twenty segments. ASTRID holds information about which subsystems are needed during each segment in order to estimate the power budget on each of them.
- **Routes:** Previously explained in chapter 4 fuel tool. It is a csv file that contains information about the routes. It has the range, payload and number of flights of each route and it is used for making the fuel burn estimation.

5.2.3. Mass convergence

The MDA is a convergence loop for the aircraft MTOM. There are three of them in the workflow, one for each family member. Once the first OpenAD is run, ASTRID will size the subsystems. With this information we can run again OpenAD but this time with the exact value of the systems' masses, engine bleeding and off-takes. Running again OpenAD with this new values gives a better estimation of the actual aircraft, and running again ASTRID makes the values more precise. The loop is run until convergence is reached. The converging criteria of this analysis was set in a 1% of the MTOM.

Two effects are catched with this convergence loop. One is the impact the subsystems cause on the aircraft. For instance a bleedless ECS is heavier than a conventional one since more components are needed but the bleeding penalty is much lower so the engine performance increases, so the total aircraft mass can decrease as a result. The other effect that is taken into account is the snowball effect. This effect is explained with one example. If the mass of a subsystem increases, the MTOM increases in a bigger quantity. This occurs because more elements will be added or reinforced in orther to support the loads. The same happens in the other way, reductions in subsystems' masses result in bigger reductions in the total mass. For instance, in the cited analysis they concluded (for their specific case) that a reduction of 1 kilogram implied another extra reduction of around 600 grams [75].

5.2.4. Toolchain

Now the actual XDSM diagram is shown in figure 34:

It can be seen how the three loops for each of the aircraft is separated. They cannot be run in parallel since the second loop has to wait for the first one to finish in order to use (or not) the same ASTRID output in case the sharing variables are activated. The same happens with the third converger. The three initial OpenADs are run in parallel as well as the fuel burn tool and the commonality tool.

So the tools run in the following order:

1. The optimizer creates the design variables.
2. The three initial OpenAD are run in parallel.
3. The first loop runs until convergence is reached.
4. The second loop runs. It takes the values from the previous ASTRID loop if the "sharing 1 and 2" variable is active.
5. The third loop runs. It takes the values from the previous ASTRID loop if the "sharing 2 and 3" variable is active.
6. Commonality and fuel burn tools are run in parallel and the values are saved and given to the converger.

The main problem with this workflow set-up is that it is slow. The main cause of it is the sharing variables cause these make the loops unrunnable at the same time in parallel.

5.3. RCE workflow

Now the real RCE workflow is shown. It has all the mergers, intermediate tools and scripts that were needed in order to properly communicate the tools among them.

The information is stored in CPACS files. It is based on xml language in which the main aspect is that data is stored under certain addresses (or xpaths). The root of all of them is cpacs. As an example, all the information about each aircraft is read "cpacs/vehicles/aircraft/model" and the the different disciplines are written inside (like wing, fuselage, reference...). As we have three aircraft, the adresses are the same but in this case an attribute is given to each of them. For instance, to access to the vertical tail of the second aircraft the path is: "cpacs/vehicles/aircraft/model[@uID="AircraftModel2"] /wings/wing[@uID="vtp"]".

Like this the data is properly organized and can be easily merged with more analysis without mixing it. All the different tools are stores under an address called "toolspecific". For example, "cpacs/missions/mission" is the xpath for all the mission segments. So every time a new analysis is done, the information will be stores in the toolspecific (ASTRID, commonality tool, fuel burn tool...).

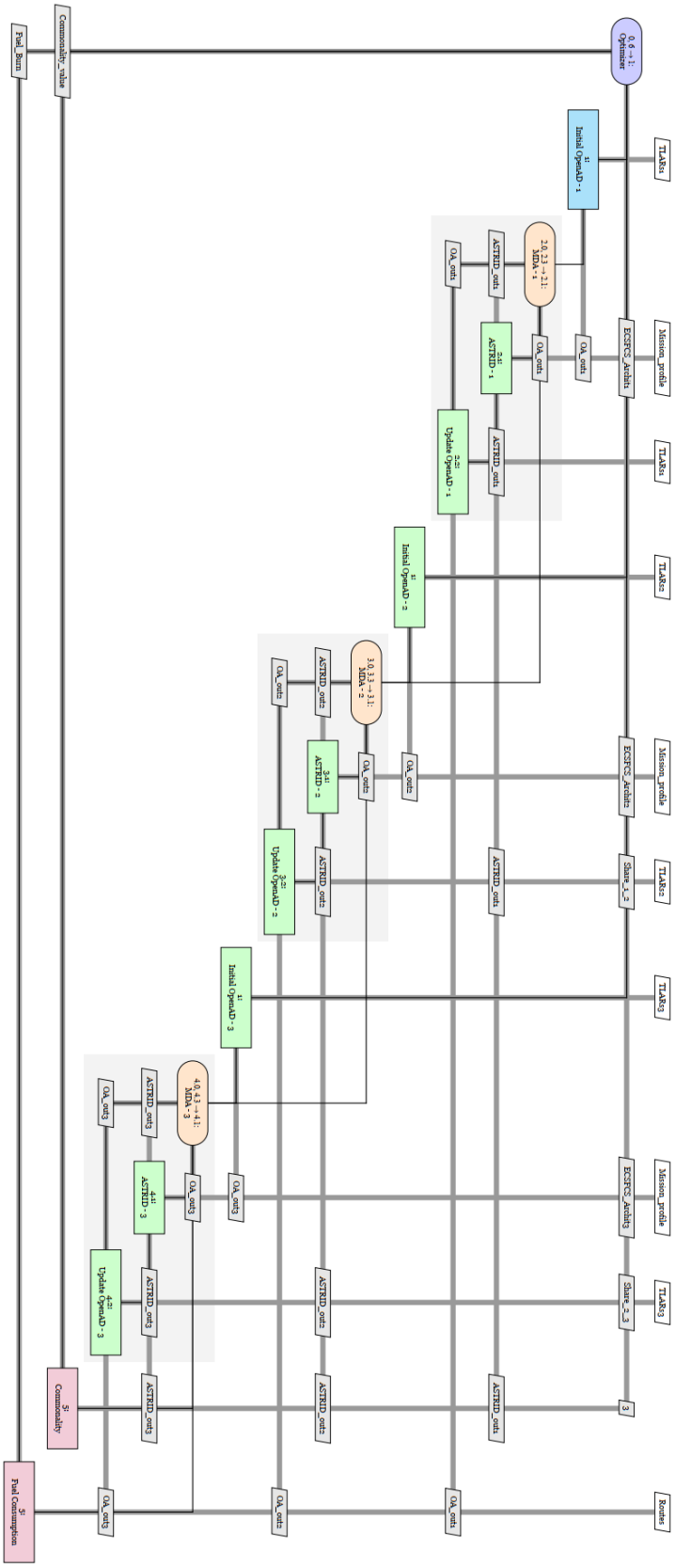


Figure 34: XDSM

An image of the RCE workflow is now shown in figure 35:

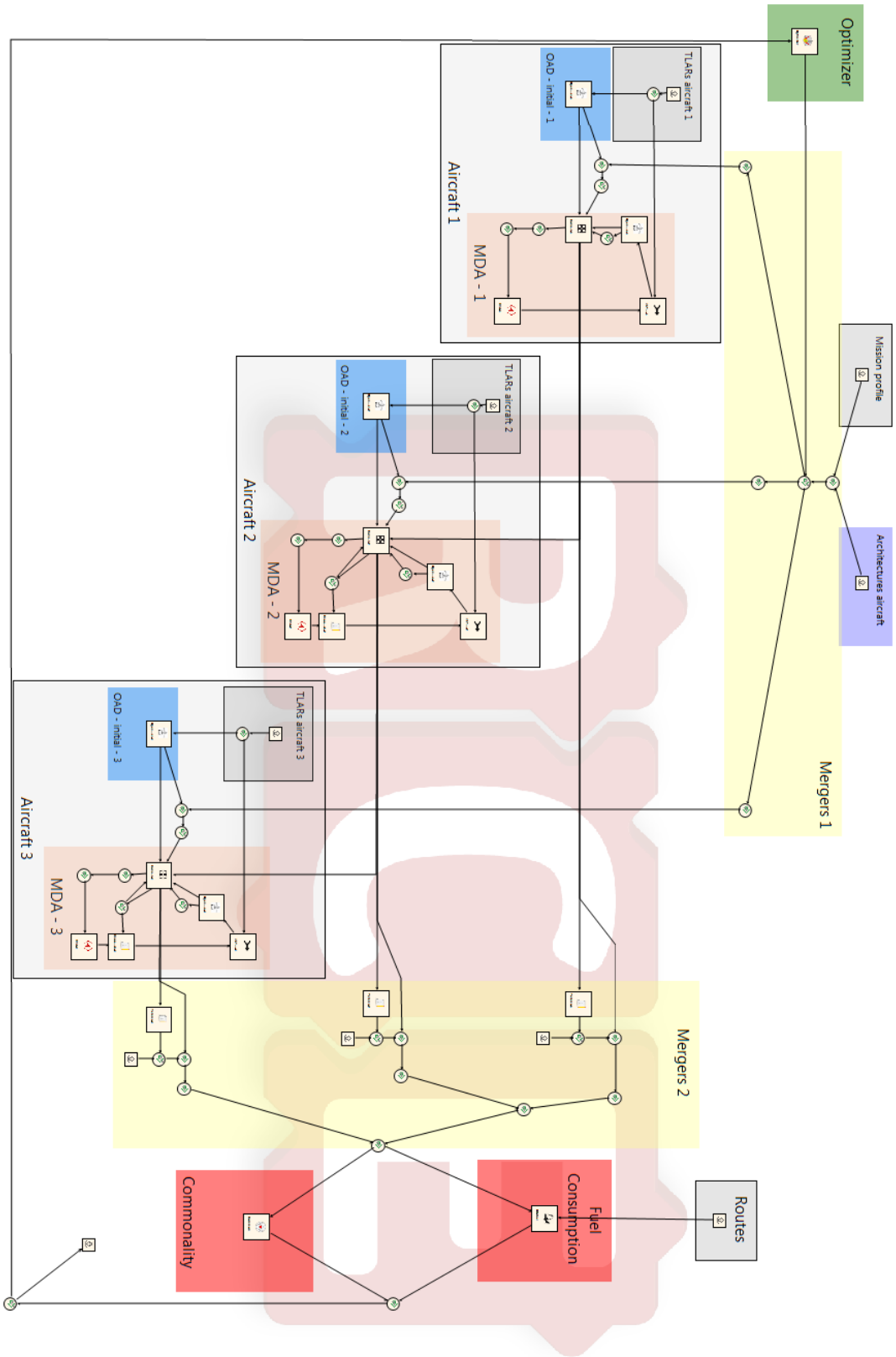


Figure 35: RCE workflow

The tool order has been explained before, now the important thing is explaining the function of the mergers and how the data is shared through the workflow.

Mergers 1

The optimizer creates the design variables, which are stored into the xpath "cpacs/ toospecific/ASTRID/architectures" and merged with the mission profile. Then the file is split into three files, each of them with the architectures of one of the aircraft and a copy of the mission profile. These files are respectivelly given to each aircraft.

Aircrafts

This subsection refers to the initial OpenAD plus the convergence loop for each aircraft. The first aircraft is simpler than the other ones. A clearer image is now shown to better explain how each of them was solved. Figure 36 shows the case for aircraft one:

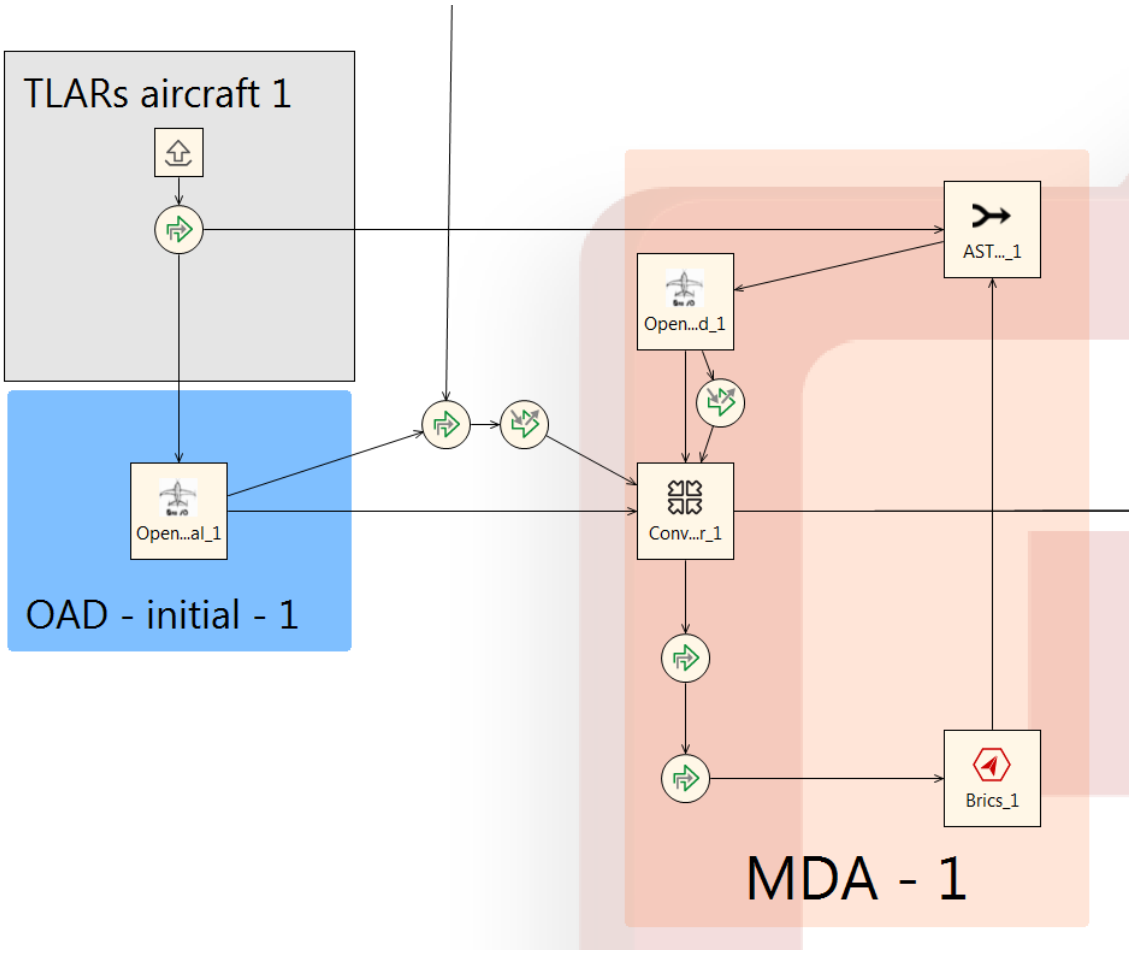


Figure 36: RCE workflow, detail of the first aircraft

The input file "TLARs aircraft" contains the three aircraft in one file, so a merger is needed to remove the second and third family members before running OpenAD. Then the result is merged with the mission profile and architecture for aircraft 1,

previously done in the component "mergers 1". After this, the file and the MTOM value are given to the Converger.

The converger works as follows. It gives the CPACS file with the results form OpenAD, mission profile and architectures. The first merger reorganizes some CPACS addresses in order to make them compatible with ASTRID, the second one removes the previous results form ASTRID (useful for the next iterations), and then the file is given to BRICS. This tool sends the file to ASTRID and takes the output back.

BRICS then returns the output from ASTRID so that the values can be updated with OpenAD. But this cannot be directly done. Another intermediate tool is needed. This tool was called "AST_OAD" and was programmed in python. The tool takes two inputs, ASTRID output and OpenAD input and what it does is copying OpenAD input and adding new addresses with the values obtained from ASTRID's output. Then this resulting file is given to another OpenAD and the CPACS file and MTOM are given to the converger. The process is repeated until the converge is reached.

The second and third aircraft are more complex since they can share subsystems with the previous aircraft, so the previous converged CPACS file has to be given also as an input since it contains the results from ASTRID for the other family member. Figure 37 shows how the approach was done for the second and third members.

The difference here is that before the tool "AST_OAD" there is now a script that reads the sharing variables and decides which file is given. If the variable is cero then the result from BRICS_2 is given cause the subsystems are not shared. If the variable is one then the given file is the one from the previous aircraft that comes through the converger. The third aircraft is done the exact same way.

Mergers 2

These mergers are needed since not all the information required was stored in the CPACS created by OpenAD. This problem concerns only CPACS 2.0 which was used in this thesis, it is solved in the version CPACS 3.0. So the result from OpenAD is not only a CPACS file but also a csv file with some needed values for the fuel consumption tool. These both files are exported by the optimizer once it converges. Figure 38 show how it was done. The csv file is given to the script called "params" which reads the values and rewrites them with the proper units. It also calculates other ones that can be estimated from parameters form this file, but that are not directly written (all of them are specified in chapter 4.5.3). Afterwards, these values are stored in a cpacs address "cpacs/toolspecific/parameters[@uID="Aircraft1"]". The uID is different for each aircraft. And finally it is merged with OpenAD's output so all is in the same file. There is another merger later that writes new uIDs to the aircraft to be able to merge them all into one single file. This merging is done later joining first the first and second members and then merging both with the third aircraft. The final result after all these mergers is one file with 3 aircraft, 3 ASTRID results, 3 "aircraft parameters" from the csv file and 1 common mission profile. This file is given to the last two tools and run in parallel.

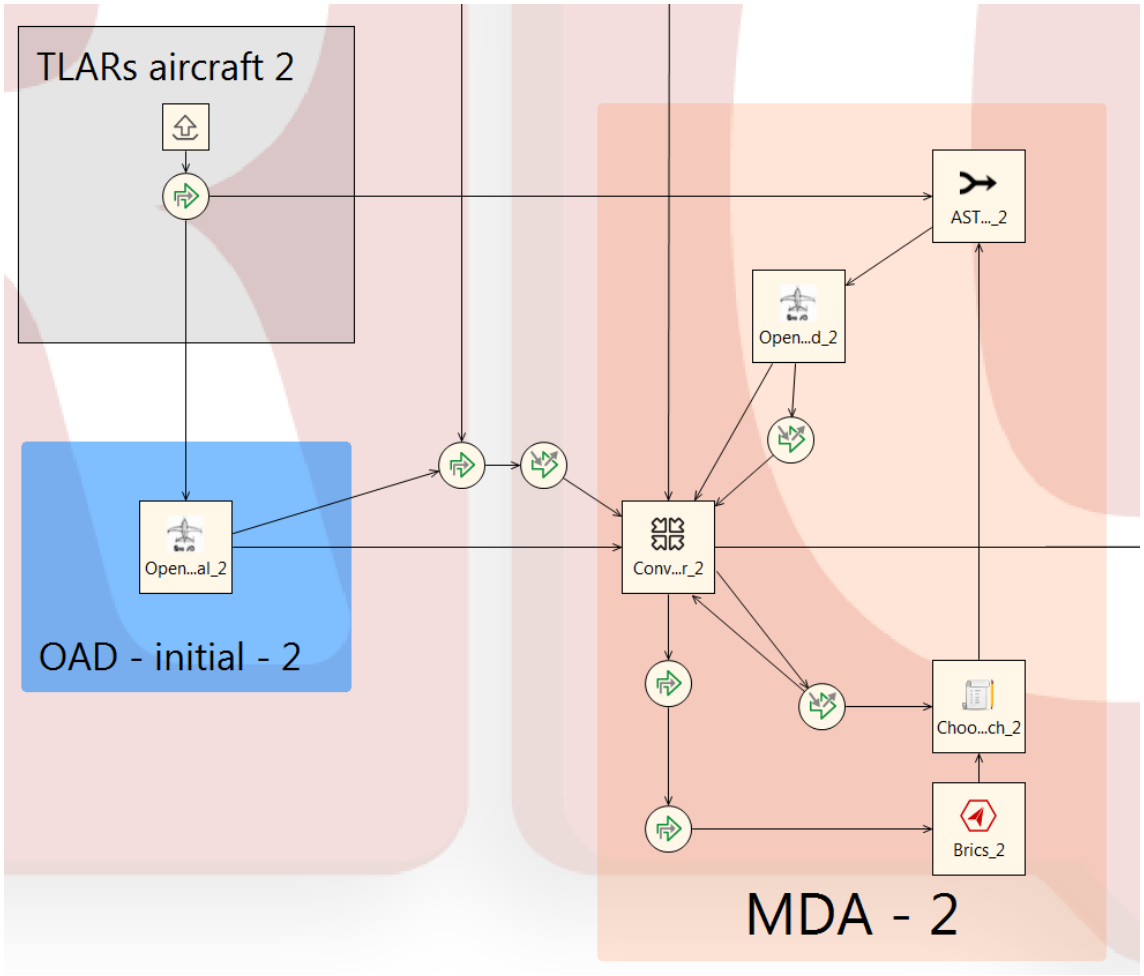


Figure 37: RCE workflow, detail of the second and third aircraft

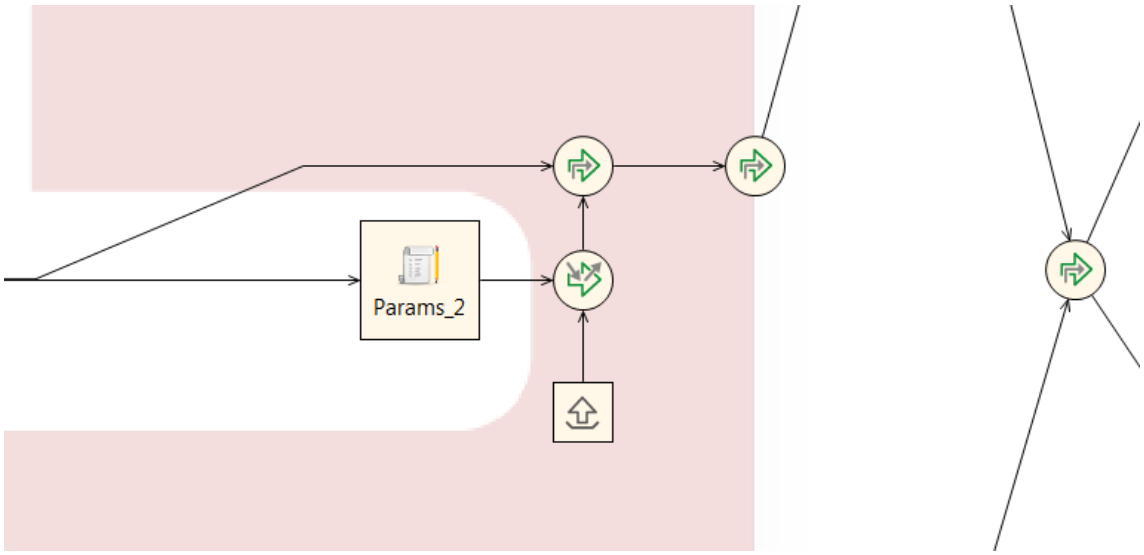


Figure 38: RCE workflow, detail of the mergers

After running the commonality and fuel burn tool, both results are stored in new toolspecific addresses and the final file is saved. This file contains all the results from the workflow. Then the two optimization objective values are read and sent back to the optimizer.

6. Results

A preliminary analysis of each of the aircraft's TLARs was done in this section. They should be optimized before changing the architectures in order to have an adequate reference point. Then a sensitivity analysis for the sharing variables was performed in order to be sure that this variable is important enough to be included. Afterwards, A DOE (Design of experiments) was done in order to find the best initial point to start the optimization and in order to have some pre-liminar results about the architectures. Finally, the results of the whole run are shown as well as the final pareto front.

6.1. Initial analysis

The objective of this first part is to find the best possible initial points for the optimization, and to fix the payload-range points for each of the members. In order to do this analysis the fuel tool was used. The main idea is running an optimization to find the better characteristics for each family member using the low fidelity model. Like this the initial aircraft will be the ones that minimize the fuel burn when the on-board systems have not been analyzed yet. This is the closes to the actual optimum point that can be estimated before doing the whole analysis.

Given the routes, the three members are sized to minimize the fuel. The biggest member is sized so it would be able to fly all the routes. The other two members will be optimized by the algorithms. The design variables are the maximum payload and maximum range (with maximum payload) of these two smaller members.

All the mergers are used to separate and merge the initial and final files from three aircraft to one, so OpenAD can run it (figure 39). The ones in the middle with the scripts are used to take some needed variables from OpenAD that are not written in CPACS, as said before in the fuel tool chapter.

The element "Script (1)" is introduced to achieve a faster convergence with OpenAD. The function of this block is filling the input file for OpenAD with better estimations for some of the parameters. This script receives the values of the maximum payload and range of the second and third aircraft and returns the approximated wing load and take-off length. These approximations were taken from the interpolation with values from similar aircraft, in particular the A318, A319, A319 and A320.

The wing loading was found to be quite linear, the resulting function can be seen in figure 40:

The take-off length was not as linear as the wing load, but this analysis is just done to give a better initial estimation, so the exact value is not important. The result and used function can be seen in the next figure 41:

Now regarding the results. The first step is to determine appropriate boundaries for the design variables. Doing some "Design of Experiments" runs these values

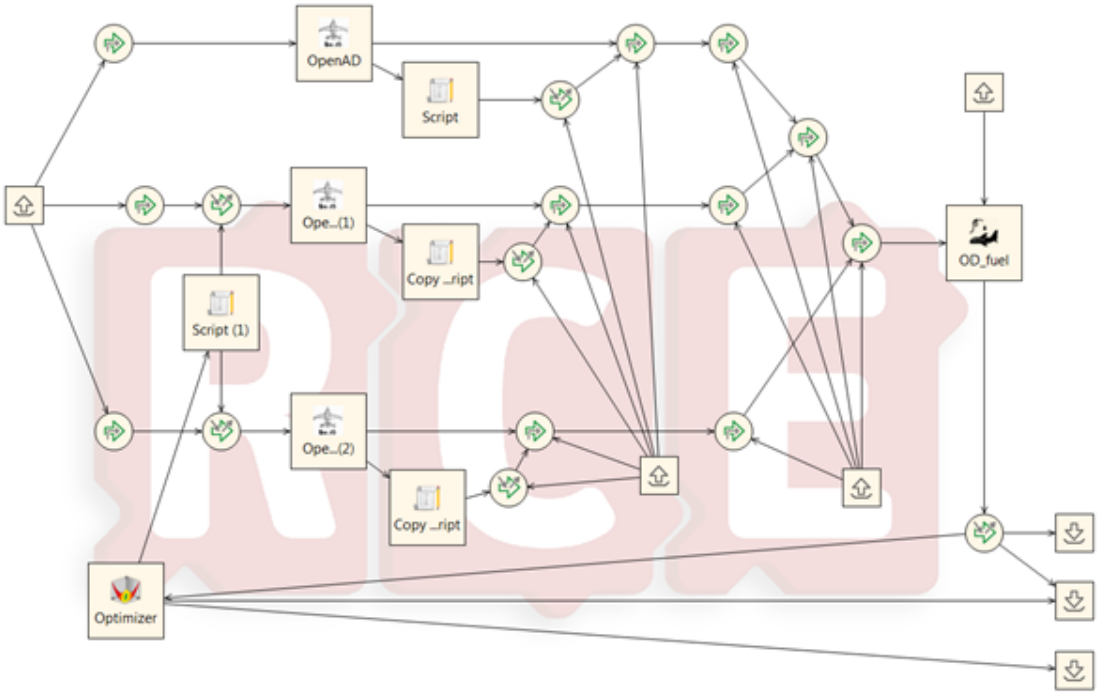


Figure 39: RCE workflow for the initial analysis

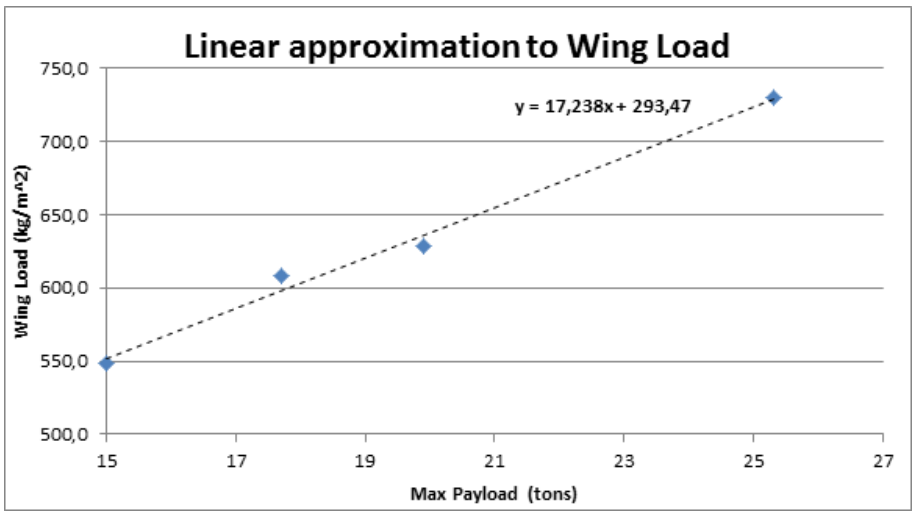


Figure 40: Approximation for the wing load

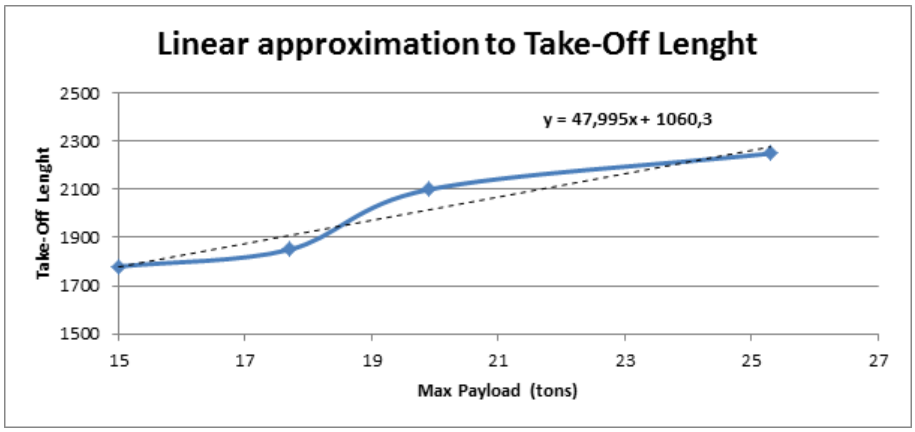


Figure 41: Approximation for the take off lenght

were fixed to:

- Maximum payload aircraft 2, between 18 and 25 tons
- Maximum payload aircraft 3, between 14 and 18 tons
- Range for MPL aircraft 2, between 4500 and 7000 kilometers
- Range for MPL aircraft 3, between 3000 and 4500 kilometers

The optimization algorithms stopped at 431 iterations with the following optimum points for minimum fuel burn:

MPL2 (kg)	MPL3 (kg)	Range2 (km)	Range3 (km)	Fuel (kg)
19797	17050	5083	3474	6665
19797	17079	5083	3592	6665
19797	17050	5083	3933	6661
19797	17079	5083	3893	6660

Table 4: Initial analysis first results

It can be noticed that both payloads and the second member's range are quite fixed their optimum point, while the third aircraft's range has small variations between 3400 and 4000 km. The most optimum one resulted in the number of flights for each aircraft shown in table 5. It can be seen that the middle aircraft would be the one with higher utilization ratio, and the biggest member would have the lowest number of flights.

	Total flights	% flights
Aircraft 1	501383	17
Aircraft 2	1443602	48
Aircraft 3	1083084	36
All	3028069	100

Table 5: First results; number of flights

The aircraft characteristics can be obtained by just running again OpenAD with the optimum values. The main results are the following:

	MTOM	MPL	OEM	Range	Wing Surface
Aircraft 1	140400	27000	80000	6500	187
Aircraft 2	94000	19800	54200	5000	148
Aircraft 3	77100	17100	46000	3900	131

Table 6: First results; masses

The first aircraft has similar characteristics to the A310-200. This model is designed to carry 240 passengers, which makes them quite comparable. The second member is more or less similar to the A321-neo and the Tu-204. The third one resembles more to the B737-800. The PL-R diagrams for each are now shown and can be seen in figure 42:

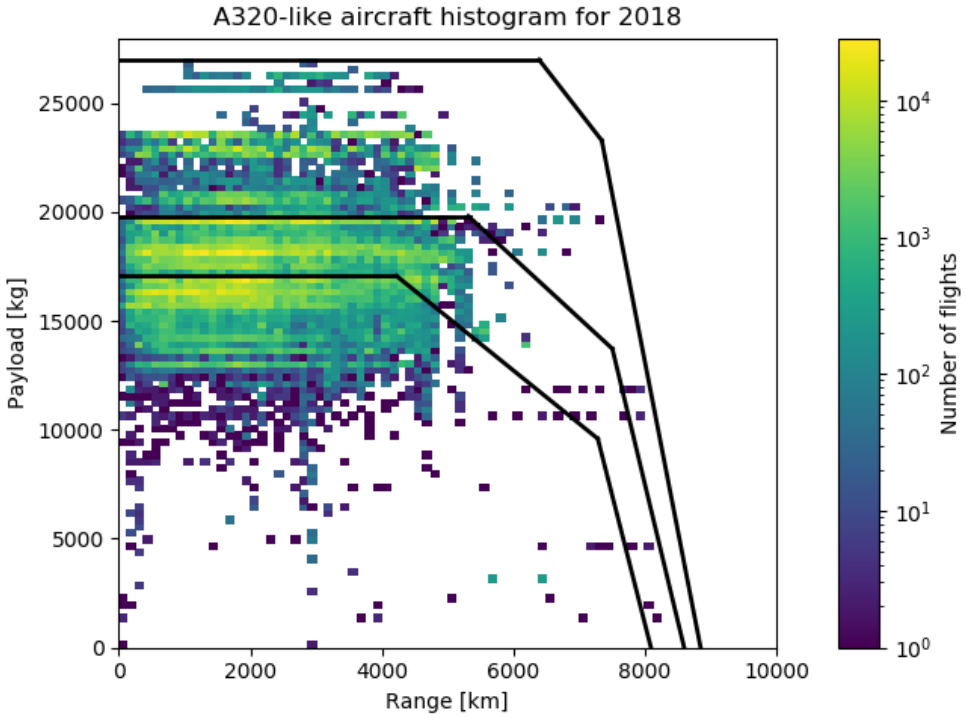


Figure 42: PL-R results for the optimum case

Now the worst result obtained is run again to double check the analysis. This corresponds with a case with 8100 tons of average fuel burn. And the resulting diagram is the following figure 43. It can be seen how this result makes the second aircraft fly almost all the routes, which makes the family quite inefficient.

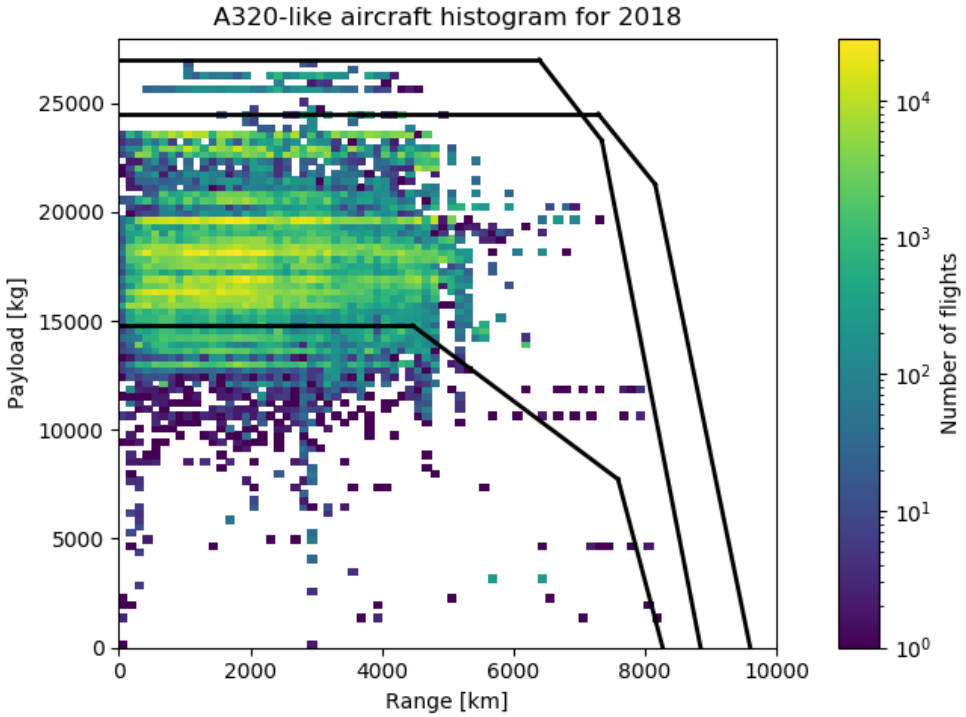


Figure 43: PL-R results for the worst case

6.2. Subsystems' sharing sensitivity analysis

In general it is a good idea to test the sensitivity of the tools to some of the variables before all the parameters are present. Like this one can test if the model is sensitive as expected to some of the assumptions or not. In this small analysis the objective is to check that the fuel model is indeed capable of sensing variations on the on-board systems changes. In order to do this a workflow with the following characteristics is set:

- Design of experiments to change the commonality design variables (share subsystems or not).
- MDA loops with the sharing capabilities and BRICS.
- ASTRID substituted by default values per aircraft.

Like this the specific architectures for each aircraft are not relevant and only the impact of sharing the subsystems will be taken into account. Four different cases were analysed: no common subsystems among members, sharing between aircraft 1 and 2, sharing between aircraft 2 and 3 and sharing between all. The results are now shown in table 7.

It can be seen that the results are as expected. The bigger the commonality the more the fuel gets penalized. The masses are shown to control that the results actually make sense. As defined, the first aircraft is fixed and its mass does not change since it has the same on-board systems for now (pre-defined standard values). If the second aircraft shares subsystems with the first one then its MTOM increases

Share 1&2	Share 2&3	MTOM 1	MTOM 2	MTOM 3	Fuel
No	No	140786	94420	75342	6564
No	Yes	140786	94420	77240	6593
Yes	No	140786	96426	75342	6619
Yes	Yes	140786	96426	79143	6674

Table 7: Results of the first commonality design variables sensitivity analysis

two tons while the third one can share with the second or with the first and increases its MTOM also in around two tons.

It can be concluded that the model effectively takes into account the effect of sharing subsystems, and ASTRID will consider the effect of changing the subsystems' architectures so all the design variables are now tested and the tools work adequately. The fuel values of this analysis shall not be considered as exact ones since some approximations were done.

6.3. Workflow adjustments

Two more things were added before executing the workflow. One is the failure behaviour of the workflow components. This refers to what the workflow should do in case of a component failure. If ASTRID fails sending incorrect or inconsistent values the converger should fail and send the failure to the optimizer. The optimizer will write "not a value" as result and continue with another iteration. The same happens in case OpenAD fails or returns inconsistent results as outputs, the optimizer iteration should be discarded as well. This makes the workflow more automatic since it will not stop every time a component fails. It was ensured first that all the blocks were properly connected and working as expected to be able to make sure that if one component fails it is because there has been a failure in the values and not in the component itself.

The other enhancement was done to increase the speed of the algorithms. The main issue to solve is about the sharing variables. All the design variables can take whatever value, but when the sharing variables are active the workflow copies the design variables form the main aircraft and uses it for the other one. The problem is that if for example the sharing variable between the first and second member is active, the workflow ignores the design variables for the second aircraft. But if this sharing variable is not active the impact of the variables is enormous. The algorithms would take a really long time to understand this behaviour so a filter was implemented. This consisted on a script right after the optimizer. This script returns "not a value" to the optimizer in case there is one inconsistency or more in the design variables (for example different ECS architectures with active sharing variables). The filter returns the same values in case these inconsistencies are not existent in that iteration. Like this the whole workflow execution is avoided when the design variables are not exactly the ones that should be run, allowing us to obtain faster results. This filter allows to reduce the number of possible combinations from 2125764 to 544575. Which is a 25,6 % of the initial ones and will help a lot to reach

convergence sooner.

The input data for the next analysis is detailed in the Appendix. There are five tables. One with general TLARs information for each of the aircraft. Another one with the data for each of the mission segments. And three more with parameters for the wing, horizontal and vertical tailplanes. The coefficients for these three wing surfaces are shared among the three aircraft members, so there is just one table for the three of them. The information contained in the mission profile is just the Mach number and altitude on each segment. The rest of parameters, like for instance which subsystems are operating on that segment, are stored inside of ASTRID's files.

6.4. Design of Experiments (DOE)

A design of experiments consist in executing the workflow with some predefined points. So the optimizer is substituted by this DOE block that gives as an output some previously selected design points. This analysis is done in order to find some good points to give as initial points to the optimizer and to start having some initial results.

The same architectures were selected for the three aircraft in this DOE. This means that all the commonality values will be high since the aircraft will have common architectures during the iterations. The total number of different architectures is 24, if we also take into account the sharing variables this number increases. For this DOE only two cases regarding these variables were considered, both active or both inactive. This leads to a total number of DOE points of 48 (24 architectures that can be sharing subsystems or not). Less than 48 points were analyzed since the wanted conclusions were reached with around 30 iterations. The values for the fuel consumption and commonality for the conventional ECS architectures can be found in table 8. The white spaces are points that were not analyzed cause the commonality value is 1 and the fuel consumption is known to be between some values, so the are not yet needed. The values with an "x" are the ones in which convergence was not found. As said before, the commonality values are high since the architectures are the same in the three aircraft. The results regarding the bleedless ECS architectures are shown in table 9.

				Sharing	0	Sharing	1
ECS	**N_Wheels**	**Sub_Freezing**	**FCS**	**Fuel**	**Commonality**	**Fuel**	**Commonality**
0	2	0	0	6700	0,845	-	1
0	2	0	1	6610	0,845	-	1
0	2	1	0	x	x	x	x
0	2	1	1	6611	0,841	-	1
0	3	0	0	6698	0,845	-	1
0	3	0	1	6608	0,845	6786	1
0	3	1	0	6700	0,841	6886	1
0	3	1	1	6609	0,841	6787	1
0	4	0	0	6698	0,841	-	1
0	4	0	1	6608	0,841	-	1
0	4	1	0	6702	0,839	6888	1
0	4	1	1	6610	0,839	6788	1

Table 8: DOE results for the conventional ECS architectures

				Sharing	0	Sharing	1
ECS	**N_Wheels**	**Sub_Freezing**	**FCS**	**Fuel**	**Commonality**	**Fuel**	**Commonality**
1	2	0	0	6610	0,857	6766	1
1	2	0	1	6528	0,857	6677	1
1	2	1	0	6614	0,852	6770	1
1	2	1	1	6526	0,852	-	1
1	3	0	0	6616	0,857	-	1
1	3	0	1	6526	0,857	-	1
1	3	1	0	6617	0,852	6775	1
1	3	1	1	6524	0,852	-	1
1	4	0	0	6613	0,852	-	1
1	4	0	1	6524	0,852	-	1
1	4	1	0	6615	0,849	-	1
1	4	1	1	6523	0,849	6672	1

Table 9: DOE results for the bleedless ECS architectures

From this two tables we can see that the bleed-less configurations manage to have lower fuel burn values than the conventional architectures. While the commonality levels that both options can reach are similar. The number of wheels does not seem to have a big impact on the fuel burn, and the sub-freezing or non sub-freezing choice has the same behavior. The explanation for this would be provided afterwards. Regarding the flight control system, the electric actuators seem to make the fuel burn increase, this result was not expected and will also be explained later.

To be able to see why the fuel burn increases or decreases two main contributions shall be analyzed. The first one is the difference in mass and the second one is the difference in specific fuel consumption (SFC). An increase in mass will result in a bigger fuel burn while a decrease in SFC will do the opposite. In this case both effects are mixed and now they are going to be analyzed separately to check the impact of both of them on each architecture.

To do so, the masses and SFCs of the DOE results are now shown. First, the mass values can be seen in table 10, for the conventional ECS architectures and in table 11 for the bleed-less architectures. All of these cases are the ones with inactive sharing variables, like this effects of sharing subsystems will not be taken into account cause they can make unreliable conclusions for this specific analysis. The tables show the mass for each of the family members and the average among the three.

ECS	N_Wheels	Sub_Freezing	FCS	MTOM (1)	MTOM (2)	MTOM (3)	MTOM (average)
0	2	0	0	147.0	99.9	82.2	109.7
0	2	0	1	144.9	98.4	80.2	107.9
0	2	1	0	x	x	x	x
0	2	1	1	145.0	98.4	80.2	107.9
0	3	0	0	146.9	99.9	82.1	109.6
0	3	0	1	144.9	98.4	80.2	107.8
0	3	1	0	147.0	99.9	82.2	109.7
0	3	1	1	144.9	98.4	80.2	107.8
0	4	0	0	146.9	99.9	82.1	109.6
0	4	0	1	144.9	98.4	80.2	107.8
0	4	1	0	147.0	99.9	82.2	109.7
0	4	1	1	144.9	98.4	80.2	107.9

Table 10: Masses results for the conventional ECS architectures, in tons

ECS	N_Wheels	Sub_Freezing	FCS	MTOM (1)	MTOM (2)	MTOM (3)	MTOM (average)
1	2	0	0	147.2	100.3	82.6	110.0
1	2	0	1	145.3	98.9	80.8	108.4
1	2	1	0	147.3	100.3	82.7	110.1
1	2	1	1	145.3	98.9	80.8	108.3
1	3	0	0	147.3	100.4	82.7	110.1
1	3	0	1	145.3	98.9	80.8	108.3
1	3	1	0	147.4	100.4	82.8	110.2
1	3	1	1	145.3	98.8	80.7	108.3
1	4	0	0	147.3	100.3	82.7	110.1
1	4	0	1	145.3	98.8	80.7	108.3
1	4	1	0	147.3	100.4	82.7	110.1
1	4	1	1	145.2	98.8	80.7	108.3

Table 11: Masses results for the bleed-less ECS architectures, in tons

The results show that the mass is bigger for the bleed-less architectures, as expected. These configurations need extra compressors to substitute the bleeding systems and as a result the overall mass increases. It can also be seen how the electric flight control system has a noticeably higher mass than the hydraulic one, this result confirms why the fuel burn increased before and will be explained at the end of this chapter.

The difference between the sub-freezing and non subfreezing is small but appreciable. It resulted as expected, the sub-freezing architectures have more components since they need a re-heater and a condenser and hence the mass increases. The number of wheels did not affect the mass a lot since in all cases the number of components is similar. Comparing the 2 and 3 wheeled ACU, both need a fan, linked or not to the turbine. In the case of the 4 wheels ACU, the value is similar since the both turbines are smaller than the one-turbine of the 3 wheels case. So as a result the mass is similar in the three cases.

Having seen the effects on the mass we will now focus on the specific fuel consumption. The procedure is the same one as before, two tables will be shown. One with the values for the bleed-less architectures and another one with the ones for the conventional ones. Three columns show the SFC of each of the aircraft and the last one the average value for the whole family. The values are written in tables 12 and 13.

ECS	N_Wheels	Sub_Freezing	FCS	SFC (1)	SFC (2)	SFC (3)	SFC (average)
0	2	0	0	15.25	15.38	15.44	15.36
0	2	0	1	15.24	15.37	15.42	15.34
0	2	1	0	x	x	x	x
0	2	1	1	15.24	15.37	15.42	15.34
0	3	0	0	15.25	15.38	15.44	15.36
0	3	0	1	15.24	15.37	15.42	15.34
0	3	1	0	15.25	15.38	15.44	15.36
0	3	1	1	15.24	15.37	15.42	15.34
0	4	0	0	15.25	15.38	15.44	15.36
0	4	0	1	15.24	15.37	15.42	15.34
0	4	1	0	15.25	15.38	15.44	15.36
0	4	1	1	15.24	15.37	15.42	15.34

Table 12: SFC results for the conventional ECS architectures, in g/(kN · s)

The results show that the SFC is bigger for the conventional architectures, which was expected. These architectures penalize the fuel consumption since they need to bleed air from the compressors, lowering their performance and increasing the fuel consumption as a result.

The difference between electric and hydraulic FCS is really small in all cases. This explains why the fuel burn increases when choosing electric architectures since the mass increase is more noticeable than the sfc change. The difference between sub-freezing and non subfreezing architectures and between number of wheels is almost non-existent.

This analysis leads to some conclusions. Differences in ECS type are noticeable

ECS	N_Wheels	Sub_Freezing	FCS	SFC (1)	SFC (2)	SFC (3)	SFC (average)
1	2	0	0	14.98	15.08	15.15	15.07
1	2	0	1	14.97	15.07	15.12	15.05
1	2	1	0	14.98	15.08	15.15	15.07
1	2	1	1	14.97	15.07	15.12	15.05
1	3	0	0	14.98	15.08	15.15	15.07
1	3	0	1	14.97	15.07	15.12	15.05
1	3	1	0	14.98	15.08	15.15	15.07
1	3	1	1	14.97	15.07	15.12	15.05
1	4	0	0	14.98	15.08	15.15	15.07
1	4	0	1	14.97	15.07	15.12	15.05
1	4	1	0	14.98	15.08	15.15	15.07
1	4	1	1	14.97	15.07	15.12	15.05

Table 13: SFC results for the bleed-less ECS architectures, in g/(kN · s)

and according to the expectations while differences in FCS type are also noticeable but not as expected. The number of wheels does not have a great impact on the results since they do not affect off-takes. The necessary airflow comes from the analysis of the thermal loads between the cabin and the fuselage. For this reason the number of wheels only affects the mass and since the masses are similar the impact is really low. This decision has a greater impact on reliability and maintenance values. Regarding the subfreezing or non-subfreezing design variable. The cruise off-takes needed in both cases are the same since at high altitudes the external air is practically dry and the non-subfreezing architecture can work below zero temperatures. The differences in mass and SFC are appreciable but small so they do not have a big impact on the results.

6.5. Pareto Front

During the development of this thesis a paper for the 2020 AIAA was written on the same topic by the same author [72]. The results are the same in both cases. This thesis references a lot that paper since the Pareto front is the same one.

The result is the Pareto front between fuel consumption and commonality. The origin of this analysis was the trade-off between both variables. Each architecture showed different degrees of commonality and fuel burn which are represented in a two-axe graph. The frontier or border of the results is known as Pareto front and represents the limits reachable and the trade-off between the optimization objectives. Here one can see the impact that increasing one variable has on the other one. For instance, if increasing commonality penalizes fuel burn, how and in which degree.

After the DOE was done, the optimizer was launched from several promising points to help the genetic algorithms find the optimum solutions all along the Pareto front. Several runs were done. The obtained points and the Pareto front are shown in figure 44. This graph specifies which architectures allow the biggest degree of commonality and which one performs better from a fuel consumption perspective, as well as all the intermediate points. It shows the resulting points from the runs

as grey dots and the Pareto front as lines.

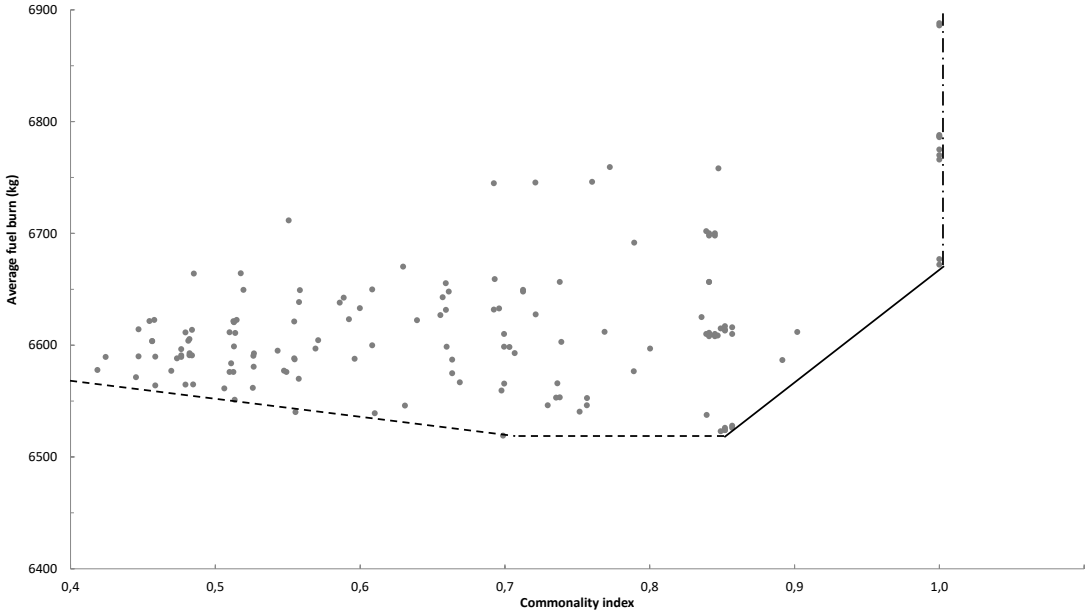


Figure 44: Pareto front

All the points are shown in the appendix in two tables: table 19 and table 20. All the design variables values as well as the results for both optimization objectives can be found in them.

The Pareto front is divided in three parts. The first one, with a dashed line, shows that the average fuel burn decreases as the commonality increases. The second one, represented with a continuous line, has the opposite behavior as the previous one and shows how increasing commonality penalizes fuel burn noticeably. And lastly, the dotted line represents the maximum commonality index reached, which corresponds to when the three aircraft share subsystems among all the three members.

The first area is affected by the fact that one of the architectures performs better than the other ones in terms of fuel burn. So the minimum fuel burn is reached when the three aircraft have a certain architecture. If the three family members have different architectures it's very likely that some of them will poorly perform in terms of fuel burn. As a result the average fuel tons increase. If the three architectures are different, the commonality value is low. Hence as we raise the commonality value, the minimum possible fuel burn decreases since we are getting closer to the minimum for fuel burn.

The second area has the opposite effect. The initial point starts in the inflection point (which is the global minimum for fuel burn) where the three aircraft have the same architecture with the minimum possible fuel burn and without sharing subsystems among them. This point represents the minimum fuel consumption among all the possible combinations and each of the aircraft has its optimum subsystems, no sharing variables are active in this point. The commonality obtained with this configuration is limited, and in order to increase it some subsystems shall be shared

among members. The sizing aircraft in case of sharing is always the bigger one, which is aircraft one. This causes that if components are shared the weight and specific consumption will increase, hence the family will perform worse on average. This trade-off between commonality and fuel consumption is common to family design problem [72].

The third area is a vertical line that represents the maximum commonality index reachable. It can be only reached when the three family members share the subsystems among them, hence they also have the same architectures. What this means is that the first aircraft is sized and then the other two are using its same subsystems. This causes the second member to be slightly over-sized. The third member is highly over-sized and far from its optimum design. This causes the fuel burn to substantially increase [72].

Figure 45 shows a detailed view of the Pareto front. Some interesting points have been highlighted.

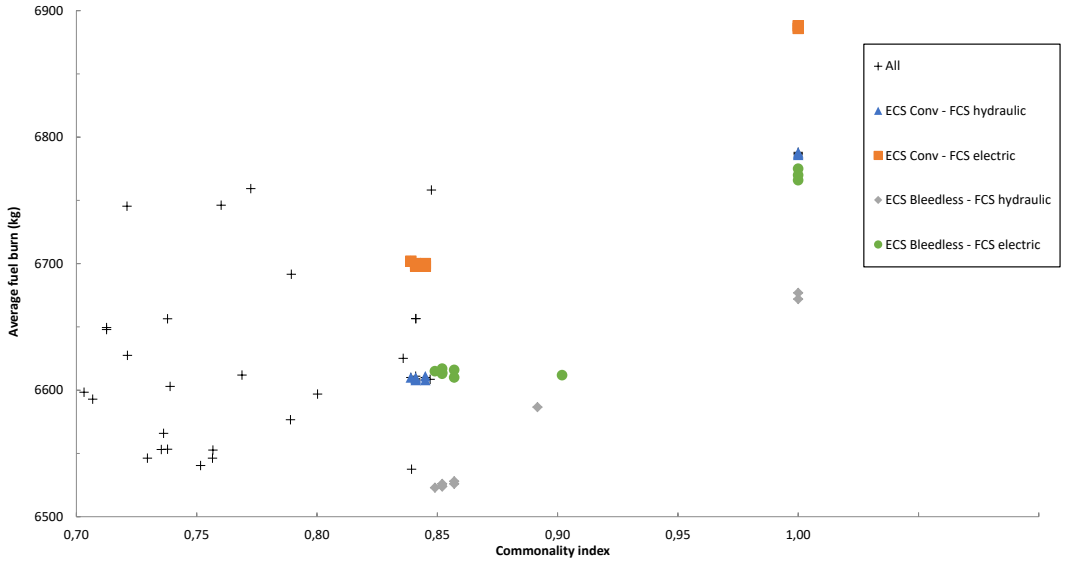


Figure 45: Pareto front, detailed view

The architecture that reaches the lowest fuel burn values corresponds to the bleedless ECS with hydraulic FCS. This result goes accordingly to the expectations regarding the ECS since the bleedless ECS has the lowest specific fuel consumption and compensates its mass increase. From the FCS perspective, the electric FCS has a bigger mass increase than SFC reduction in terms of fuel burn. This result was surprising and can be related to the fact that all the FCS actuators had the same architecture, a mix between electric and hydraulic FCS could reach better results. For instance, using only two hydraulic circuits an electric redundancies (like the A380) results in a global improvement. But there is not a commercial aircraft with all the FCS electric to compare the results with it.

The rest of the design variables regarding the ECS did not show a huge impact on

the results as the other two did. This is due to ASTRID sizing. The number of wheels does not have an impact on the off-takes since the necessary airflow comes from the analysis of the thermal loads between fuselage and cabin. The mass difference is minimum and the needed bleeding is the same one. Regarding the sub-freezing or non-subfreezing architecture, the cruise off-takes are the same in both cases since the external air is practically dry and the non-subfreezing configuration can work at minus-zero temperatures. The difference in mass and SFC were appreciable but small. As a result these two variables did not highly affect the results and only induced small variations around a certain point [72].

The difference between the minimum average fuel burn and the minimum reached with the maximum commonality index is approximately 150 kilograms per flight, which represents something close to a 2%. Multiplying this value with the total amount of flights the result is a difference of 454 000 fuel tons per year for the whole family. This number puts these quantities more into perspective.

7. Conclusions

First we presented the state of art of family concept design. In which we saw that several studies to be able to estimate commonality among members have been done. Some other studies tried to enhance these studies by adding a performance parameter and checking how commonality benefits can harm the overall family performance.

In aviation, some analysis have been done trying to design an aircraft family to fulfill some specific performance parameter (aerodynamic, structural...). In this thesis we focused on optimizing the family average fuel burn while also estimating somehow the commonality among the members. These two optimization objectives are also related to the operating and acquisition costs respetcively.

Then we reviewed the state of art for the environmental control system and the flight control system. We saw how the new more-electric aircraft tendencies are moving into new different architectures. The architectures chosen for the ECS were: conventional bleeding system or bleed-lees configuration, subfreezing or non-subfreezing air conditioning unit (also known as high or low pressure water extraction system) and finally the number of wheels that the bootstrap cycle has, which can be 2, 3 or 4. Regarding the FCS the only two architectures possible were using electric or hydraulic actuators. Also, we allowed the aircraft to share subsystems to increase the commonality.

A review on the tools that were used is presented. OpenAD was used to do the overall aircraft design. ASTRID to size the subsystems. BRICS to communicate tools run in differen servers. The fuel and commonality were estimated with two new Python tools designed specifically for this analysis. Finally everything was linked with RCE and CPACS was used as the common language to communicate among all of them. Hence with RCE we linked the collaborative multidisciplinary analysis toolchain. This setup allows the optimization of on-board system architecture for the aircraft family as a whole. Other aircraft subsystems than the ECS and FCS can be optimized this way by adding the corresponding architecture decisions and adding relevant tools to the multidisciplinary analysis toolchain.

The architecture design space was modeled and the multi-objective optimization problem is automatically formulated from this model using a novel method represented by the Architecture Design Space Graph (ADSG), which maps functions into components. An important difference compared to conventional system architecting methods is that in principle all possible system architectures are considered, and no pre-selection using expert experience is done. This way more architecture alternatives are considered by quantitative analysis, resulting in more complete knowledge about the behavior of the design space [72, 73].

The outputs of the multidisciplinary analysis toolchain are the family-level commonality and average fuel burn. Future improvements could include replacing these two objectives by their cost analysis counterparts: acquisition cost and operating cost, respectively. Another interesting trade-off might be present when looking at the design of more-electric architectures, where it is expected that maintenance costs

are lower, but development costs are higher. Both of the improvements can be easily integrated in the current toolchain due to the modular setup and flexibility of the architecture design space modeling method.

Analysis results show that the bleed-less environmental control system reaches the lowest fuel burn from all the architectures if combined with a hydraulic flight control system. Potential improvements in analysis and design fidelity can be achieved by integrating control of the electrification degree at actuator-level instead of at system-level. This would allow to analyze intermediate design points where the SFC decrease and the mass increase that the electric actuators cause results in an average fuel burn reduction. More variables to define the ECS architecture can also be included, like control over the spacial layout of the system.

A Pareto trade-off between commonality and fuel burn was found as expected, although it was seen that for lower commonality indexes commonality and fuel burn correlated positively. The inflexion point was found to be the one with minimum fuel burn, which corresponds to a family consisting of the three aircraft with the architecture that needs the least fuel and without sharing subsystems. This effect is due to the fact that there is an architecture that has a noticeably lower fuel burn than the others. If this architecture is chosen for the three family members, the fuel burn will be the lowest possible while the commonality index will be high since all the aircraft have the same components. If one member has a different architecture, the commonality will decrease and the fuel burn will be penalized, hence the resulting point will move to the left side of the Pareto front. If one member shares subsystems with another one, the commonality index will increase but the fuel burn will also be higher since the aircraft is being over-sized. This will result in a point on the right side if the Pareto front.

The further work for this thesis would consist on:

- Increase the number of architecting choices for the FCS
- Increase the number of subsystems analyzed, like for instance the landing gear
- Develop some cost models for operation cost and acquisition cost
- Change the optimization objectives to the respective costs
- Check if the new trade-offs and Pareto front suffer big changes from the previous ones

Appendix: Input data

Parameter	Aircraft 1	Aircraft 2	Aircraft 3
Desired Range (km)	6500	5084	3894
Cruise altitude (m)	10000	10000	10000
Cruise Mach	0,78	0,78	0,78
Fuel density (g/L)	785	785	785
Minimum Static Margin (%)	8	8	8
Wing load, MTOM (kg/m2)	750	635	588
Thrust/weight installed (%)	31,5	32,5	31,4
CL maximum TO	2	2	2
CL maximum L	2,5	2,5	2,5
sTOFL (m)	2100	2010	1880
Reserve fuel (%)	4	4	4
Max payload (kg)	27000	19800	17100
Wing AR	10,3	10,3	10,3
Leading edge sweep (°)	27,3	27,3	27,3
Passengers per row	6	6	6
Fuselage width (m)	3,9	3,9	3,9
Fuselage height (m)	4,14	4,14	4,14
Engine model	V2500 A5	V2500 A5	V2500 A5
Engine Bypass Ratio	5	5	5
Turbine exit temperature (°C)	1700	1700	1700
Enngine OPR	36	36	36

Table 14: TLARs for the three family members

Parameter	Horizontal Tail Plane	Vertical Tail Plane
CTH/CVT	1,11	0,092
AR	4,82	1,675
Dihedral	6	-
Taper Ratio	0,32	0,32
LE Sweep (°)	32,3	40,5
Average thinkness (%)	11	12

Table 15: Horizontal and vertical tailplanes TLARs for the three aircraft

The wing is divided in 4 segments (between 5 stations). The one in the symmetry plane is called center, the one in the fuselage union is the root, and then three more to define the trapezoidal shape, which are kink, middle and tip.

Wing Station	Parameter	Value
Center	Thickness (%)	15
	Reference AOA (°)	3
	Position (% of wingspan)	0
Root	Thickness (%)	15
	Reference AOA (°)	3
	Position (% of wingspan)	matching with fuselage
Kink	Thickness (%)	12
	Reference AOA (°)	1
	Position (% of wingspan)	0,34
Middle	Thickness (%)	11,5
	Reference AOA (°)	0,7
	Position (% of wingspan)	0,65
Tip	Thickness (%)	11
	Reference AOA (°)	0,4
	Position (% of wingspan)	1

Table 16: Wing stations TLARs

Wing segment	Parameter	Value
Center-Root	Dihedral (°)	0
	Taper ratio	1
Root-Kink	Dihedral (°)	10
	Taper ratio	0,535
Kink-Middle	Dihedral (°)	7
	Taper ratio	0,5616
Middle-Tip	Dihedral (°)	7,5
	Taper ratio	0,119

Table 17: Wing segments TLARs

Segment	Mach	Altitude (m)
Pre-flight checks	0	0
Engine start-up	0	0
Taxi out	0	0
Taxi out - flaps down	0	0
Take off run	0,22	0
Take off manoeuvre	0,28	0
Take off - landing gear up	0,3	15
Take off - flaps down	0,3	15
Climb	0,55	7000
Cruise	0,78	10000
Descent	0,55	7000
Descent - flaps down	0,4	2000
Approach - landing gear down	0,3	1500
Approach	0,3	1500
Landing manoeuvre	0,2	15
Landing run	0,15	0
Taxi in - flaps up	0	0
Taxi in	0	0
Engine shutdown	0	0
Emergency	0,7	10000

Table 18: Mission profile

Appendix: Pareto front values

ECS 1	ECS 2	ECS 3	FCS 1	FCS 2	FCS 3	NW 1	NW 2	NW 3	S_1and2	S_2and3	SF 1	SF 2	SF 3	Commonality	Fuel
0	0	0	0	0	1	3	4	4	0	0	0	1	1	0.639	6622
0	0	0	0	0	1	4	3	2	0	0	0	0	0	0.513	6621
0	0	0	1	0	1	4	4	3	0	0	0	0	0	0.660	6599
0	0	0	1	1	0	3	3	2	0	0	1	0	1	0.555	6588
0	0	0	1	1	0	4	3	3	0	0	1	0	0	0.555	6587
0	0	1	0	0	1	4	3	4	0	0	1	0	0	0.514	6611
0	0	1	0	1	1	2	3	4	0	0	0	1	1	0.459	6564
0	0	1	1	0	0	2	4	3	0	0	1	1	0	0.458	6623
0	0	1	1	0	1	2	2	4	0	0	0	1	1	0.459	6590
0	0	1	1	0	1	2	3	3	0	0	0	0	0	0.476	6589
0	0	1	1	1	0	2	3	3	0	0	1	1	1	0.510	6576
0	1	0	0	0	0	2	4	4	0	0	1	1	1	0.655	6627
0	1	0	0	0	1	3	2	2	0	0	1	1	0	0.482	6591
0	1	0	0	1	0	2	2	4	0	0	0	0	1	0.424	6589
0	1	0	0	1	0	3	3	3	0	0	0	1	0	0.511	6584
0	0	0	0	0	0	2	2	2	0	0	0	0	0	0.845	6700
0	0	0	1	1	1	2	2	2	0	0	0	0	0	0.845	6610
0	0	0	1	1	1	2	2	2	0	0	1	1	1	0.841	6611
0	0	0	0	0	0	3	3	3	0	0	0	0	0	0.845	6698
0	0	0	1	1	1	3	3	3	0	0	0	0	0	0.845	6608
0	0	0	0	0	0	3	3	3	0	0	1	1	1	0.841	6700
0	0	0	1	1	1	3	3	3	0	0	1	1	1	0.841	6609
0	0	0	0	0	0	4	4	4	0	0	0	0	0	0.841	6698
0	0	0	1	1	1	4	4	4	0	0	0	0	0	0.841	6608
0	0	0	0	0	0	4	4	4	0	0	1	1	1	0.839	6702
0	0	0	1	1	1	4	4	4	0	0	1	1	1	0.839	6610
1	1	1	0	0	0	2	2	2	0	0	0	0	0	0.857	6610
1	1	1	1	1	1	2	2	2	0	0	0	0	0	0.857	6528
1	1	1	0	0	0	2	2	2	0	0	1	1	1	0.852	6614
1	1	1	1	1	1	2	2	2	0	0	1	1	1	0.852	6526
1	1	1	0	0	0	3	3	3	0	0	0	0	0	0.857	6616
1	1	1	1	1	1	3	3	3	0	0	0	0	0	0.857	6526
1	1	1	0	0	0	3	3	3	0	0	1	1	1	0.852	6617
1	1	1	1	1	1	3	3	3	0	0	1	1	1	0.852	6524
1	1	1	0	0	0	4	4	4	0	0	0	0	0	0.852	6613
1	1	1	1	1	1	4	4	4	0	0	0	0	0	0.852	6524
1	1	1	0	0	0	4	4	4	0	0	1	1	1	0.849	6615
1	1	1	1	1	1	4	4	4	0	0	1	1	1	0.849	6523
0	0	0	1	1	1	3	3	3	1	1	0	0	0	1	6786
0	0	0	0	0	0	3	3	3	1	1	1	1	1	1	6886
0	0	0	1	1	1	3	3	3	1	1	1	1	1	1	6787
0	0	0	0	0	0	4	4	4	1	1	1	1	1	1	6888
0	0	0	1	1	1	4	4	4	1	1	1	1	1	1	6788
1	1	1	0	0	0	2	2	2	1	1	0	0	0	1	6766
1	1	1	1	1	1	2	2	2	1	1	0	0	0	1	6677
1	1	1	0	0	0	2	2	2	1	1	1	1	1	1	6770
1	1	1	0	0	0	3	3	3	1	1	1	1	1	1	6775
1	1	1	1	1	1	4	4	4	1	1	1	1	1	1	6672
0	0	0	0	0	0	3	4	3	0	0	1	1	0	0.738	6656
0	0	0	0	0	1	2	2	3	0	0	1	1	1	0.592	6623
0	0	0	1	0	1	2	4	3	0	0	0	0	0	0.513	6599
0	0	0	1	1	1	4	3	3	0	0	1	1	0	0.738	6553
0	0	1	0	0	1	2	3	2	0	0	1	1	1	0.510	6612
0	0	1	0	0	0	3	3	3	1	0	1	1	1	0.760	6746
0	0	1	0	0	1	3	3	3	1	0	0	0	1	0.551	6712
1	1	0	1	1	1	4	4	3	1	0	1	1	0	0.800	6597
1	1	1	0	0	1	3	3	3	1	0	0	0	1	0.693	6659
1	1	1	0	0	0	3	3	4	1	0	0	0	1	0.789	6692
1	1	1	1	1	1	3	3	3	1	0	1	1	1	0.892	6587
0	0	0	0	0	0	3	3	3	1	0	1	1	0	0.848	6758
0	0	1	1	1	1	3	3	3	1	0	0	0	1	0.692	6632
0	0	0	0	0	0	3	3	2	1	0	1	1	1	0.772	6759
0	0	1	0	0	0	3	3	4	1	0	1	1	1	0.721	6745
1	1	0	1	1	1	2	2	3	1	0	0	0	0	0.739	6603
0	0	1	1	1	1	3	3	4	1	0	0	0	0	0.696	6633
0	0	1	1	1	0	3	3	2	1	0	0	0	1	0.485	6664
1	1	0	1	1	0	2	2	3	1	0	1	1	0	0.600	6633
0	0	1	1	1	1	3	3	4	1	0	0	0	1	0.659	6632
0	0	1	0	0	0	3	3	3	1	0	0	0	1	0.692	6745
1	1	0	0	0	1	3	3	3	1	0	0	0	1	0.630	6670
0	0	1	1	1	0	3	3	4	1	0	0	0	1	0.518	6664
0	0	0	1	0	0	3	2	2	0	1	1	0	0	0.558	6649
0	1	1	0	0	0	3	3	3	0	1	0	0	0	0.836	6625

Table 19: Pareto front values, first table

ECS 1	ECS 2	ECS 3	FCS 1	FCS 2	FCS 3	NW 1	NW 2	NW 3	S_1and2	S_2and3	SF 1	SF 2	SF 3	Commonality	Fuel
1	1	1	1	0	0	4	3	3	0	1	1	1	1	0.707	6593
0	0	0	1	0	0	4	2	2	0	1	1	0	0	0.519	6649
0	1	1	1	1	1	2	3	3	0	1	0	0	0	0.752	6540
1	0	0	1	1	1	2	3	3	0	1	0	1	1	0.669	6567
0	1	1	1	1	1	3	3	3	0	1	1	1	1	0.839	6538
1	1	1	0	1	1	2	2	2	0	1	1	1	1	0.757	6546
1	1	1	0	0	0	3	3	3	0	1	0	0	0	0.902	6612
0	1	1	0	1	1	3	3	3	0	1	1	1	1	0.698	6559
0	0	0	1	0	0	3	3	3	0	1	1	0	0	0.661	6648
0	0	0	0	1	1	3	2	2	0	1	0	1	1	0.608	6600
1	0	0	1	0	0	4	3	3	0	1	1	1	1	0.586	6638
0	0	0	1	1	1	4	3	3	0	1	0	1	1	0.789	6577
1	1	1	0	1	1	2	2	2	0	1	0	1	1	0.730	6546
0	0	0	1	0	0	2	3	3	0	1	0	1	1	0.608	6650
0	0	0	1	0	0	3	4	4	0	1	0	0	0	0.713	6648
1	0	0	1	1	1	4	3	3	0	1	0	1	1	0.700	6566
1	1	1	0	0	0	2	2	2	0	1	1	0	0	0.847	6609
0	0	0	1	0	0	3	3	3	0	1	0	1	1	0.713	6649
1	0	0	1	1	1	3	3	3	0	1	0	1	1	0.736	6566
1	0	0	1	0	0	4	3	3	0	1	0	1	1	0.558	6639
0	0	0	0	0	0	2	4	4	0	0	1	0	0	0.659	6655
0	0	0	0	0	0	3	3	3	0	0	1	1	1	0.841	6657
0	0	0	0	0	1	2	3	3	0	0	1	0	1	0.555	6621
0	0	0	0	0	1	3	3	2	0	0	1	0	0	0.513	6621
0	0	0	0	0	1	3	4	2	0	0	1	0	1	0.512	6621
0	0	0	0	0	1	3	4	2	0	0	1	1	0	0.515	6623
0	0	0	0	1	1	3	2	4	0	0	1	0	1	0.512	6576
0	0	0	1	0	1	3	3	3	0	0	0	0	0	0.703	6598
0	0	0	1	1	0	4	3	3	0	0	1	1	0	0.596	6588
0	0	0	1	1	1	3	4	3	0	0	0	0	0	0.757	6553
0	0	1	0	0	0	3	3	4	0	0	0	1	1	0.657	6643
0	0	1	0	0	1	2	2	3	0	0	1	1	0	0.484	6614
0	0	1	0	1	0	2	3	3	0	0	0	0	0	0.476	6596
0	0	1	0	1	0	4	3	4	0	0	1	1	1	0.569	6597
0	0	1	0	1	1	3	2	3	0	0	0	1	1	0.485	6565
0	0	1	1	0	1	2	3	2	0	0	0	0	0	0.476	6591
0	0	1	1	0	1	4	2	3	0	0	0	0	0	0.447	6590
0	0	1	1	1	1	4	4	2	0	0	1	1	0	0.631	6546
0	1	0	0	0	0	3	3	3	0	0	1	1	1	0.721	6628
0	1	0	0	1	1	3	3	4	0	0	0	0	0	0.513	6551
0	1	0	1	0	0	2	2	4	0	0	1	1	0	0.456	6604
0	1	0	1	0	0	4	2	3	0	0	1	0	1	0.456	6604
0	1	0	1	0	0	4	4	3	0	0	1	1	1	0.571	6604
0	1	0	1	0	1	2	3	2	0	0	0	1	0	0.445	6571
0	1	0	1	0	1	3	3	3	0	0	0	0	0	0.558	6570
0	1	1	0	0	0	2	2	2	0	0	0	1	1	0.769	6612
0	1	1	0	0	1	2	4	2	0	0	0	1	1	0.527	6581
0	0	0	0	0	0	3	3	3	0	0	1	1	1	0.841	6657
0	0	0	0	1	0	3	3	3	0	0	1	1	1	0.699	6610
0	0	0	0	1	1	3	3	3	0	0	0	1	1	0.664	6575
0	0	0	1	0	1	4	4	4	0	0	0	0	0	0.699	6599
0	0	0	1	1	0	3	4	4	0	0	0	0	0	0.664	6587
0	0	0	1	1	1	4	3	4	0	0	0	1	0	0.735	6553
0	0	1	0	0	0	2	4	3	0	0	0	0	0	0.589	6642
0	0	1	0	0	1	3	2	2	0	0	0	1	0	0.447	6614
0	0	1	0	0	1	4	3	3	0	0	1	0	0	0.479	6611
0	0	1	0	1	1	3	3	4	0	0	1	0	0	0.479	6565
0	0	1	1	0	0	2	4	3	0	0	1	0	1	0.455	6621
0	0	1	1	0	1	2	3	3	0	0	1	1	0	0.484	6591
0	1	0	0	0	1	3	3	2	0	0	1	0	1	0.482	6593
0	1	0	1	0	0	3	3	2	0	0	1	1	0	0.482	6605
0	1	0	1	0	0	4	2	3	0	0	1	1	1	0.481	6604
0	1	1	0	1	1	4	3	3	0	0	0	1	0	0.555	6540
0	1	1	0	1	1	4	4	4	0	0	0	1	0	0.610	6539
0	1	1	1	0	0	2	3	3	0	0	0	0	1	0.527	6593
0	1	1	1	0	0	3	2	3	0	0	0	1	0	0.526	6590
0	1	1	1	0	1	2	4	2	0	0	1	1	0	0.526	6562
0	1	1	1	1	1	2	3	3	0	0	1	1	0	0.699	6519
1	0	0	0	1	0	4	4	3	0	0	0	0	0	0.543	6595
1	0	0	0	1	1	3	3	3	0	0	0	1	0	0.506	6561
1	0	0	1	0	1	3	3	4	0	0	0	0	1	0.473	6588
1	0	0	1	1	0	2	3	3	0	0	0	0	0	0.470	6577
1	0	0	1	1	0	2	4	3	0	0	0	0	1	0.419	6578
1	0	0	1	1	0	3	3	3	0	0	0	1	1	0.547	6577
1	0	0	1	1	0	3	3	3	0	0	1	1	0	0.549	6576

Table 20: Pareto front values, second table

References

[1] Alberto Jose and Michel Tollenaere. "Modular and platform methods for product family design: literature analysis". In: *Journal of Intelligent Manufacturing* 16.3 (2005), pp. 371–390. ISSN: 1572-8145.

[2] Timothy W. Simpson, Jonathan R. Maier, and Farrokh Mistree. "Product platform design: method and application". In: *Research in Engineering Design* 13.1 (2001), pp. 2–22. ISSN: 1435-6066.

[3] Timothy W Simpson. "Product platform design and customization: Status and promise". In: *Ai Edam* 18.1 (2004), pp. 3–20.

[4] Karl Ulrich. "The role of product architecture in the manufacturing firm". In: *Research policy* 24.3 (1995), pp. 419–440.

[5] Rhea Liem, Gaetan Kenway, and Joaquim Martins. "Multi-point, multi- mission, high-fidelity aerostructural optimization of a long-range aircraft configuration". In: *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2012, p. 5706.

[6] Atif Riaz, Marin D Guenov, and Arturo Molina-Cristobal. "Set-based approach to passenger aircraft family design". In: *Journal of Aircraft* (2017), pp. 310–326.

[7] Gregory T Mark. "Incorporating flexibility into system design: a novel framework and illustrated developments". PhD thesis. Massachusetts Institute of Technology, 2005.

[8] Achille Messac, Michael P Martinez, and Timothy W Simpson. "Introduction of a product family penalty function using physical programming". In: *J. Mech. Des.* 124.2 (2002), pp. 164–172.

[9] Sridhar Kota, Kannan Sethuraman, and Raymond Miller. "A metric for evaluating design commonality in product families". In: *J. Mech. Des.* 122.4 (2000), pp. 403–410.

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

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

[12] Tim Lammering and Eike Stumpf. *Integration of aircraft systems into conceptual design synthesis*. Tech. rep. Lehrstuhl und Institut für Luft-und Raumfahrtsysteme (ILR), 2014.

[13] Imco van Gent, Gianfranco La Rocca, and Leo L Veldhuis. "Composing MDAO symphonies: graph-based generation and manipulation of large multidisciplinary systems". In: *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2017, p. 3663.

[14] Pier Davide Ciampa et al. "Streamlining Cross-Organizational Aircraft Development: Results from the AGILE Project". In: *AIAA Aviation 2019 Forum*. 2019, p. 3454.

[15] Arthur Rizzi et al. “Towards a unified framework using CPACS for geometry management in aircraft design”. In: *50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*. 2012, p. 549.

[16] Marko Alder et al. “Recent Advances in Establishing a Common Language for Aircraft Design with CPACS”. In: *Aerospace Europe Conference*. 2020.

[17] Andrew B Lambe and Joaquim RRA Martins. “Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes”. In: *Structural and Multidisciplinary Optimization* 46.2 (2012), pp. 273–284.

[18] Joaquim RRA Martins and Andrew B Lambe. “Multidisciplinary design optimization: a survey of architectures”. In: *AIAA journal* 51.9 (2013), pp. 2049–2075.

[19] Olivier L de Weck, Eun Suk Suh, and David Chang. “Product family and platform portfolio optimization”. In: *ASME 2003 international design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers Digital Collection. 2003, pp. 175–185.

[20] Timothy W Simpson et al. “From user requirements to commonality specifications: an integrated approach to product family design”. In: *Research in Engineering Design* 23.2 (2012), pp. 141–153.

[21] Gaetan KW Kenway and Joaquim RRA Martins. “Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration”. In: *Journal of Aircraft* 51.1 (2014), pp. 144–160.

[22] Karen Willcox and Sean Wakayama. “Simultaneous optimization of a multiple-aircraft family”. In: *Journal of Aircraft* 40.4 (2003), pp. 616–622.

[23] *Federal Aviation Regulations, ”FAR 25”.*

[24] *European Aviation Safety Agency, ”Certification Specifications for Large Aeroplanes CS-25”.*

[25] Daniel Bender. “Exergy-based analysis of aircraft environmental control systems and its integration into model-based design”. In: (2019).

[26] Sabin Poudel. *Modelling of a Generic Aircraft Environmental Control System in Modelica*. 2019.

[27] Leo Mäkelä. “Model-based Fault Diagnosis of an Aircraft Environmental Control System”. In: (2016).

[28] Alexander Pollok and Francesco Casella. “Comparison of control strategies for aircraft bleed-air systems”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 14194–14199.

[29] Xiong Peng. “Aircraft environmental control systems modeling for configuration selection”. In: (2013).

[30] *Paolo, A. Numerical Models for Aircraft Systems – lecture notes, Chapter 6 – Environmental Control System, Politecnico Di Milano. 2019.*

[31] Michael Sielemann et al. “A flexible toolkit for the design of environmental control system architectures”. In: *Proceedings of the First CEAS European Air and Space Conference*. 2007.

[32] *Mathieu, K. Simulation of Components from the Environmental Control System. 2006.*

[33] Mike Sinnett. "787 no-bleed systems: saving fuel and enhancing operational efficiencies". In: *Aero Quarterly* 18 (2007), pp. 6–11.
[34] Justin Hale. "Boeing 787 from the ground up". In: *Aero* 4.24 (2006), p. 7.
[35] Victor Crespo Cavalcanti and Cláudia Regina de Andrade. *A trade-off study of a bleedless and conventional air conditioning systems*. Tech. rep. SAE Technical Paper, 2008.
[36] Boeing database: *boeing.com/commercial/aeromagazine/articles/2012_q1/4/*. Tech. rep.
[37] Andrea Lazzaretto and George Tsatsaronis. "SPECO: a systematic and general methodology for calculating efficiencies and costs in thermal systems". In: *Energy* 31.8-9 (2006), pp. 1257–1289.
[38] Christian Müller, Dieter Scholz, and Tim Giese. "Dynamic simulation of innovative aircraft air conditioning". In: *DGLR: Deutscher Luft-und Raumfahrtskongress. Bonn: Deutsche Gesellschaft für Luft-und Raumfahrt* (2007).
[39] Paul Feliot et al. "Design of a commercial aircraft environment control system using Bayesian optimization techniques". In: *arXiv preprint arXiv:1610.02271* (2016).
[40] Dieter Scholz et al. "FLECS: Functional Library of the Environmental Control System-A Simulation Tool for the Support of Industrial Processes". In: *Proceedings of the 1st International Workshop on Aircraft System Technologies*. 2007, pp. 143–157.
[41] Isidoro Martinez. "Aircraft environmental control". In: *academic website webserver. dmt. upm. es/isidoro* (2014).
[42] Isabel Pérez-Grande and Teresa J Leo. "Optimization of a commercial aircraft environmental control system". In: *Applied thermal engineering* 22.17 (2002), pp. 1885–1904.
[43] APP Santos, CR Andrade, and Edson Luiz Zaparoli. "A thermodynamic study of air cycle machine for aeronautical applications". In: *International Journal of Thermodynamics* 17.3 (2014), pp. 117–125.
[44] *Paolo Maggiore, Matteo D. L. Dalla Vedova, Lorenzo Pace, Marco Tosetti, Andrea Piovano. "Development of an environmental control system pack simulation model for a more electric aircraft"*. 2014.
[45] Maria Aranda Rosales. "A preliminary study for selecting and optimizing the environmental control system". In: (2018).
[46] Daniel Perez Linares. "Modeling and simulation of an aircraft environmental control system". PhD thesis. École Polytechnique de Montréal, 2016.
[47] *Nicole Viola, Roberta Fusaro, Marco Fioriti, Davide Ferreto, Valeria Vercella, Laura Babetto, Gabriele Ferrari. Progetto dei sistemi aerospaciali integrati: Environmental Control System. Support slides for lectures 2018-2019.*
[48] *A319/A320/A321 Environmental Control System. Familiarization Training. Liebherr Aerospace, Lindenberg. 2004.*
[49] Ian Moir and Allan Seabridge. *Aircraft systems: mechanical, electrical, and avionics subsystems integration*. Vol. 52. John Wiley & Sons, 2011.
[50] Haider Al-Lami et al. "The Evolution of Flight Control Systems". In: (2015).
[51] Pascal Traverse, Isabelle Lacaze, and Jean Souyris. "A Process Toward Total Dependability Airbus Fly-by-Wire Paradigm". In: *EDCC*. Springer. 2005, p. 1.

[52] Oliver Unruh, Thomas Haase, and Martin Pohl. “Application of a load-bearing passive and active vibration isolation system in hydraulic drives”. In: *Journal of Physics: Conference Series*. Vol. 744. 1. IOP Publishing. 2016, p. 012018.

[53] U Persson and C Schallert. “The 728 JET flight control system”. In: *Deutscher Luft-und Raumfahrtkongress 2001* (2001).

[54] Martin Recksiek. “Advanced high lift system architecture with distributed electrical flap actuation”. In: *Workshop on Aviation System Technology (AST)*. 2009.

[55] *https://forums.x-plane.org/index.php?/forums/topic/109151-magknight-787 - 9 -development-thread /&page=42*.

[56] Navatha Alle et al. “Review on electro hydrostatic actuator for flight control”. In: *International Journal of Fluid Power* 17.2 (2016), pp. 125–145.

[57] Kang Rongjie et al. “Design and simulation of electro-hydrostatic actuator with a built-in power regulator”. In: *Chinese Journal of Aeronautics* 22.6 (2009), pp. 700–706.

[58] Guan Qiao et al. “A review of electromechanical actuators for More/All Electric aircraft systems”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 232.22 (2018), pp. 4128–4151.

[59] B Aigner et al. *Assessment of electromechanical flight control actuators with regard to direct operating costs*. Deutsche Gesellschaft für Luft und Raumfahrt Lilienthal Oberth eV, 2016.

[60] *Airbus S.A.S, ”Airbus Training - A320 Flight Crew Operating Manual: Flight Controls”*.

[61] XL Tron. “A380 flight controls overview”. In: *Presentation at Hamburg University of Applied Sciences* (2007).

[62] *Airbus S.A.S, ”Airbus Training - A350-900 Flight Deck and Systems Briefing for Pilots”*.

[63] Böhnke Daniel et al. “An Integrated Method for Determination of the Oswald Factor in a Multi- Fidelity Design Environment”. In: (2011).

[64] Daniel Böhnke, Björn Nagel, and Volker Gollnick. “An approach to multi-fidelity in conceptual aircraft design in distributed design environments”. In: *2011 Aerospace Conference*. IEEE. 2011, pp. 1–10.

[65] S. Wöhler et al. “Preliminary Aircraft Design within a Multi-disciplinary and Multi-fidelity Design Environment”. In: *Aerospace Europe Conference, AEC2020 - 032*. 2020.

[66] S. Chiesa et al. “ASTRID - Aircraft On Board Systems Sizing and Trade-Off Analysis in Inital Design”. In: 2012.

[67] Erwin Moerland et al. “Collaborative aircraft design using an integrated and distributed multidisciplinary product development process”. In: *30th Congress of the international council for aeronautical sciences*. 2016.

[68] EH Baalbergen, J Kos, and WF Lammen. “Collaborative multi-partner modelling & simulation processes to improve aeronautical product design”. In: (2013).

[69] *Bureau of Transportation Statistics, United States Department of Transportation, Washington, D.C., http://www.transtats.bts.gov/*.

[70] Luis Bahamonde Jacome and Ali Elham. "Wing aerostructural optimization under uncertain aircraft range and payload weight". In: *Journal of Aircraft* 54.3 (2017), pp. 1109–1120.

[71] Snorri Gudmundsson. *General aviation aircraft design: Applied Methods and Procedures*. Butterworth-Heinemann, 2013.

[72] Carlos Cabaleiro de la Hoz et al. "Environmental & Flight Control System Architecture Optimization from a Family Concept Design Perspective, (to be published in AIAA 2020)". In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (AIAA Aviation Forum), American Institute of Aeronautics and Astronautics* (2020).

[73] J.H. Bussemaker, P.D. Ciampa, and B. Nagel. "Modeling the System Architecture Design Space and Automatically Formulating an Optimization Problem". In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (AIAA Aviation Forum), American Institute of Aeronautics and Astronautics* (2020).

[74] Andreas Page Risueño et al. "MDAx: Agile Generation of Collaborative MDAO Workflows for Complex Systems". In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (AIAA Aviation Forum), American Institute of Aeronautics and Astronautics* (2020).

[75] J Brombach et al. "Converter topology analysis for aircraft application". In: *International Symposium on Power Electronics Power Electronics, Electrical Drives, Automation and Motion*. IEEE. 2012, pp. 446–451.