CRANFIELD UNIVERSITY

ALEX GEORGIADES

SET-BASED DESIGN AND OPTIMISATION OF AIRCRAFT SYSTEMS

School of Aerospace, Transport and Manufacturing

PhD Thesis
Academic Years: 2014–2018

Academic Supervisors: Dr. Timoleon Kipouros
Prof. Mark Savill
Industrial Supervisor: Dr. Sanjiv Sharma

April 2018

CRANFIELD UNIVERSITY

ALEX GEORGIADES

SET-BASED DESIGN AND OPTIMISATION OF AIRCRAFT SYSTEMS

School of Aerospace, Transport and Manufacturing

PhD Thesis
Academic Years: 2014–2018

Academic Supervisors: Dr. Timoleon Kipouros
Prof. Mark Savill
Industrial Supervisor: Dr. Sanjiv Sharma

April 2018

To my grandfather…

"Natural science has discovered chaos. Social science has encountered complexity. But chaos and complexity are not characteristics of our new reality; they are features of our perceptions and understanding. We see the world as increasingly more complex and chaotic because we use inadequate concepts to explain it."

–J. Gharajedaghi

Abstract

During the early stages of any system design, a thorough exploration of the design space can prove to be challenging and computationally expensive. The challenges are further exacerbated when dealing with complex systems, such as an aircraft, due to the high dimensionality of their design space. Arising from the Toyota Product Development System, Set-Based Design allows parallel evaluation of multiple alternative configurations in the early design stages. At the same time, optimisation methods can be employed at later stages to fine-tune the engineering characteristics of design, or architecture, variants. As part of this project, the ADOPT framework has been developed that integrates the aforementioned areas. This allows for a thorough exploration of the design space while ensuring the optimality of the selected designs. Furthermore, assessment methods are introduced to evaluate, not only the performance of each architecture variant, but also its sensitivity to design changes and the costs associated with them. Different visualisation tools are employed, including matrix methods and parallel coordinates, to act as design decision making mechanisms. Due to the wealth of information that such an approach generates, traceability of each architecture variant is also taken into consideration, and the knowledge acquired can be used for future design projects. The framework has been developed using a process-independent and tool-agnostic approach so that it can be applied to the design process of varying kinds of systems. To demonstrate the implementation and potential benefits, the framework has been applied to the design of a generic aircraft fuel system. The results from the case study and the framework itself are discussed, with a number of areas for further development and future work being identified and presented.

Keywords

Set-Based Design; Optimisation; Systems Engineering; Engineering Design; Visualisation; Aircraft Fuel Systems;

Acknowledgements

I am grateful to Cranfield University, Airbus, and the Engineering and Physical Sciences Research Council (EPSRC) for giving me the opportunity, and supporting me financially, to undertake this project. The work presented in this thesis would not have been possible without the support from a number of people.

First and foremost I would like to thank my industrial supervisor Dr. Sanjiv Sharma. Having spent more than two years at Airbus under his supervision, this project would not have been up to the standard it is today without his unparalleled support and guidance. I am truly grateful for the knowledge I acquired, for challenging me to see things from a different point of view, and for how much he has helped me become better in my work and as a person. Thank you Sanjiv.

I want to express my sincere gratitude to my academic supervisors, Dr. Timoleon Kipouros and Prof. Mark Savill, for their guidance and counsel since the first day of the project. Their support, feedback, and constructive comments were vital to the success of this project.

I would also like to thank Emmanuel Domingues and Olivier Broca from Siemens PLM Software not only for the provision of licenses for software, but also for their outstanding technical support throughout the project.

None of this however would be possible without my family supporting me and encouraging me throughout my life. Without them I would not be where I am today and I will forever be grateful for that.

To my friends back home in Cyprus, you know who you are, thank you for being there for me when I visited, and for keeping in touch with me while I was away. My trips back home were something I always looked forward to.

This journey would not have been nearly as fun or as interesting without my fellow PhD colleagues and friends Jean Demange, Spyros Kontogiannis, Kamal Haider, Jesus Nieto Martin, Joe Loxham, Fabio Crescenti, Simone Simeone, Gabriele Mura and Andrea Castrichini.

To the EZMM team in Airbus: Adrian Murton, Judith Crockford, Helene Sebrier, Jonathan Taylor, Steve Allwright, and John Calder, thank you for making me feel welcome from day one, and I will miss the coffee breaks and the cakes.

Last but not least, I would also like to thank Gillian Hargreaves from Cranfield University for her support and assistance, and for being patient with all my enquiries over the years.

Contents

List of Figures

List of Tables

List of Abbreviations

ACT	Additional Centre Tank
ADOPT	Augmented set-based Design and Optimisation
AHP	Analytic Hierarchy Process
AMESim	Advanced Modelling Environment for Simulations
ASO	Asymmetric Subspace Optimisation
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Cambridge Advanced Modeller
CBA	Choosing By Advantages
CFD	Computational Fluid Dynamics
CE	Concurrent Engineering
C-FAR	Change Favorable Representation
CoG	Centre of Gravity
CP^2	Cost Prediction for Change Propagation
CPI	Change Propagation Index
CPM	Change Propagation Methods
CPU	Central Processing Unit
CSSO	Concurrent SubSpace Optimisation
CSV	Comma-Separated Values

DBT	Design-Build-Test cycle
DMM	Domain Mapping Matrix
DSM	Design Structure Matrix
EASA	European Aviation Safety Agency
EC	Engineering Change
FaHS	Fuel as a Heat Sink
GA	Genetic Algorithms
GPU	Graphics Processing Unit
HDF	Hierarchical Data Format
HoQ	House of Quality
HTP	Horizontal Tail Plane
IDF	Individual Discipline Feasible
INCOSE	International Council On Systems Engineering
LeanPPD	Lean Product and Process Development
MCC	Method of Controlled Convergence
MCDM	MultiCriteria Decision Making
MDF	MultiDisciplinary Feasible
MDM	Multi-Domain Matrix
MDO	Multidisciplinary Design Optimisation
MLW	Maximum Landing Weight
MTOW	Maximum Take-Off Weight
NEA	Nitrogen Enriched Air
NSGA	Nondominated Sorting Genetic Algorithm
PDE	Partial Differential Equations
PSO	Particle Swarm Optimisation
QFD	Quality Function Deployment

RQ	Research Question
SAND	Simultaneous ANalysis and Design
SATM	School of Aerospace, Transport and Manufacturing
SBCE	Set-Based Concurrent Engineering
SBD	Set-Based Design
SE	Systems Engineering
TPDS	Toyota Product Development System
UERF	Uncontained Engine Rotor Failure
VDD	Value-Driven Design
xDSM	eXtended DSM
XML	eXtensible Markup Language

Nomenclature

x	Design variable
$\mathbf{x}$	Vector of design variables
$f(\mathbf{x})$	Objective function
pp_n	Probability of path n
dp_{ij}	Direct probability between components i and j
Cp_{ij}	Combined probability of all paths between components i and j
pc_n	Cost of path n
cw_i	Cost weight of component i
Ac_{ij}	Aggregate cost of all paths between components i and j
V_W	Wingbox volume
b_S	Structural semispan
C_{S_1}	Unusable part of the leading edge as a percentage of the chord
C_{S_2}	Unusable part of the trailing edge as a percentage of the chord
Λ_S	Sweep angle of the structural semispan at the quarter-chord line
R_{t_R}	Thickness to chord ratio at root
R_{t_T}	Thickness to chord ratio at tip
C_R	Root chord of wing at fuselage intersection
C_T	Tip chord
w_C	Width of carrythrough structure/Fuselage diameter

A	Wing Area
AR	Aspect ratio
b	Wing span
C'_R	Theoretical root chord
TR	Taper ratio
r	Pipe radius

Part I

Introduction and Literature Review

Chapter 1

Introduction

Moore's Law states that the number of transistors within an integrated circuit is doubling every two years [1]. This directly translates to more computational power; power that can be used for computational simulations. During the past few years, due to this increase in available computational power, aircraft design has been progressively relying on computational design and simulations for providing data for design decision-making and design evolution.

From Computer-Aided Design (CAD), to Computer-Aided Engineering (CAE) such as Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEA), computational methods and processes have allowed for a rapid progress in aerospace design practices [2]. The ability to solve systems of Partial Differential Equations (PDE) was the key enabler of this advancement; due to the ability to predict performance before physical systems have to be built [3].

New regulations are constantly set by the appropriate governing bodies to meet emission and noise reductions. This entails a continuous improvement of the aircraft design in order to meet said regulations. In order to do so, improving and enhancing computational methods and tools is vital, as the design process relies more and more on computational design and simulations [4]. Not only the tools, but the processes as well need to be constantly improved in terms of speed, agility, reliability, and robustness, in order to provide the necessary data for design decision-making, in a rapid and reliable way.

1.1 Problem Statement

The design of an aircraft involves a large number of different interacting systems, disciplines, and functions. The process of designing such a system comprises tightly coupled, loosely coupled and uncoupled sub-processes, making the aircraft a very complex system with a high-dimensional design space.

At present, the challenge is that thorough design space exploration of such systems, especially during the early, less-constrained, design stages, can be a very complex and computationally expensive process [5, 6, 7]. The challenge is further exacerbated due to the need to ensure the feasibility and desirability of the selected design configurations. Furthermore, due to the uncertainty in the early stages of design, the sensitivity to design changes needs to be assessed; as any design changes during later stages of the development process tend to be costly. These issues give rise to six Research Questions (RQ) that this work aims to answer:

RQ1. How can a thorough and detailed exploration of the design space of a complex system take place, in a fast and computationally efficient way?

RQ2. How are the designs that are selected, assured to be the optimum ones?

RQ3. How can the robustness and feasibility of the selected designs be ensured when requirements or design parameters change during the design process?

RQ4. How can the sensitivity of the system to the propagation of design changes, and the cost associated with it, be predicted and evaluated?

RQ5. How can the data required for design decision making be visualised in a meaningful way?

RQ6. How can the knowledge acquired be utilised in future projects?

1.2 Aims & Objectives

This work investigates two major areas that can address the problem stated in the previous section, and provide solutions to the research questions that have arisen. The first, a Set-Based Design approach to identify and evaluate multiple system configurations in parallel while discarding the infeasible, the non-robust, and the undesirable ones. The second concerns Multidisciplinary Design Optimisation (MDO) tools that can fine tune the engineering characteristics of the system, to optimise the performance of it against pre-defined objectives.

Integration of the above areas will provide the capability for a rapid and efficient exploration of the design space while identifying the optimum configurations. This will assist in speeding up the process to reach the trade studies stage where the configurations can be further evaluated and assessed.

In order to achieve that, there is a need to bring the design and development of domain-specific systems, from different levels of a system's architecture, as early in the design process as possible. This will facilitate the identification of multiple configurations in a less constrained design space where novel thinking is promoted. The selected configurations may then be matured,optimised, and made available for selection by the time the aircraft reaches a more detailed and constrained design stage.

The configurations are constantly assessed and evaluated against a number of value drivers and performance metrics. One of the most important ones is the sensitivity to design changes. To address that, change prediction methods are employed along with methods to predict the cost associated with the propagation of the design changes. The Set-Based Design approach ensures that alternative and backup configurations are always available, while MDO methods can identify multiple non-dominated solutions for each configuration.

Other metrics, such as complexity penalties and fast numerical or analytical simulations provide further information for the assessment of configurations. As undesirable configurations are being progressively discarded, the computational power that was allocated to them can now be transferred to the remaining active configurations for more detailed simulations.

Such an approach entails generation of huge amounts of data. It is essential that appropriate visualisation methods are employed in order to communicate the compelling information to the designer or engineer in a meaningful way. Design structure matrices and critical path networks can be used for modelling system architectures and design change propagations, while Parallel Coordinates can be utilised for multidimensional data interrogation. All of the above methods act as mechanisms for communicating design decision-making information.

The proposed approach is expected to benefit the design process in both a direct and an indirect way. The direct will be in the case when designing a new product where, as mentioned before, it will make the decision making process faster and more reliable. The indirect one, is due to the wealth of knowledge that an approach like this will generate, which can then be used in future projects. In other words, any data that have been generated during one programme, even discarded options, can be reused in future projects. This can assist in further minimising the time required to get to the trade studies stage, by avoiding exploration of previously visited areas. The time saved, can also be used to explore new areas.

In order to demonstrate the benefits and process of the proposed approach, and integration of methods, the Augmented set-based Design and OPTimisation (ADOPT) framework has been developed. Split into two stages, the framework takes into consideration everything, from the generation of configurations to their evaluation and assessment, using a Set-Based workflow with integrated optimisation tools. The framework has been applied to the design of a generic aircraft fuel system, taking into account the fuel system level as well as higher level design parameters, such as the geometric characteristics of the wing. This enables the demonstration of how domain-specific parameters and configurations can be brought forward, in early, highly uncertain, stages of the aircraft development process.

The architectural framework has been developed using a tool-agnostic and process-independent approach. This ensures that ADOPT can be adapted and applied to a range of different system design scenarios. The area of aircraft fuel systems will be used as a case study, to demonstrate the process and potential benefits of ADOPT.

1.3 Thesis Outline

The thesis consists of four parts that address the following areas respectively: Introduction, Theoretical Framework, Application, Summary. A more detailed description of each part is given below.

Part I - Introduction and Literature Review

The first part of the thesis has already presented and introduced the topic of the research project while outlining the problem and the research questions arising from it. Following, is a literature review of the areas that this project is investigating and identification of any gaps in literature.

The literature review covers the origins of Set-Based Design,what Engineering Design is and problems associated with it, the Toyota Product Development System, and the Lean concept. An outline of matrix-based tools for engineering design is also presented, with a more detailed look on Design Structure Matrices. Predicting Engineering Change propagation using the aforementioned tools is also an area explored in this project.

The second major area that the work, and consequently the literature review, covers is optimisation methods. An overview of what optimisation is, the process, and methods is presented with emphasis on multiobjective and multidisciplinary applications.

The two major areas of the project are then compared and contrasted, identifying previous research utilising both. The final chapter of this part deals with visualisation methods in engineering design and optimisation.

Part II - The ADOPT Framework

Based on the gaps in literature, the problem statement, and research questions that were presented in the first part, the second part of the thesis presents the theoretical framework that was developed to address the aforementioned issues. It first gives an overview of the framework, named ADOPT, with its features and potential benefits and drawbacks.

The rest of Part II examines in detail the two stages of the framework. Stage 1 deals with the identification of the design parameters in a system, the imposition of undesirability and infeasibility constraints, and finally, the generation of design configurations. Stage 2 optimises the selected configurations and introduces additional methods and metrics for assessment and evaluation, including visualisation. A method that has been developed for predicting how cost propagates due to engineering changes is also presented.

Part III - Case Study

In order to demonstrate the process of the developed framework, the design of a generic aircraft fuel system is used as a case study, and is described in Part III. A descriptive overview of an aircraft fuel system is presented, along with its major subsystems and design drivers for each one.

The framework is then applied step by step on the design of a fuel system, following the process outlined in Part II. Stage 1 identifies the design parameters of the fuel system in consideration, imposes a set of infeasibility and undesirability constraints, and generates the configurations that are passed on to the second stage. The configurations with their description and results are traced and documented, the process of which is also demonstrated. The second stage identifies the objectives for optimisation, selects the cases to be optimised, and presents the additional methods that are used for evaluation and assessment of the selected configurations. A separate chapter shows the computational implementation of the framework, and how the workflow is managed using the pSeven suite, a design space exploration software by DATADVANCE.

The final chapter of part III presents the results for both stages along with further assessment and evaluation metrics. A discussion takes place for the results of each stage.

Part IV - Discussion, Future Work, and Conclusion

The fourth and final part of the thesis begins with a discussion on the previous parts regarding the framework, the case study, and the results presented. The aim of the discussion is to demonstrate how the framework and its application have addressed the problem statement, and how the research questions have been answered. The discussion also addresses the benefits and drawbacks of the presented approach.

During this research, a number of areas for improvement have been identified and presented in a separate chapter for future work. The improvements and proposals are split into two sections, one dealing with framework improvements and the other one with the application area and case study.

Part IV, and consequently the thesis itself, concludes with an overview of the project and what has been achieved.

Chapter 2

Background Research and Literature Review

2.1 Systems Engineering

2.1.1 Introduction

The International Council on Systems Engineering (INCOSE) defines systems engineering as “an interdisciplinary approach and means to enable the realization of successful systems” [8]. A system is a collection of interacting elements that when combined together produce a result that would not otherwise be obtainable by the elements alone. Elements and parts can include, but not limited to, people, hardware, software or facilities that are required to achieve the system-level objective(s) [9]. Even though a system itself does not have any levels, they are usually imposed as levels of abstraction to separate the contents [10]. Starting from top, a system of systems is usually present and each system can contain a number of subsystems. Breaking it further down, subsystems consist of parts and/or components.

An aircraft is an example of a very complex system. It includes a vast number of subsystems including avionics, hydraulics, fuel systems etc. and each of the subsystems can contain thousands of parts and components. Designing and implementing such a system is a long and complex process [11].

In the early 1900s Henry Ford changed the automotive industry by introducing the Model T. It was a revolution in mass production, which was introduced by General Motors(GM), as it managed to integrate truly the entire manufacturing process. By lining up the fabrication steps, it managed to reduce the production time but it meant that this kind of manufacturing process could not provide any variability [12].

This concept of “eliminating waste” is the main idea behind Lean manufacturing (Lean Production) that Toyota revolutionized 20 years later after what Ford did with the Model T. Womack et al. [12] in *The machine that changed the world* took a thorough look at how lean manufacturing compared with mass production. It comes to show how much the concept and practices of Lean manufacturing have helped Toyota, because at the time of publication of the book (1990), Toyota was only half the size of GM and two thirds the size of Ford. Seventeen years later, when the book had been republished, Toyota had already passed Ford and it was well on track to overtake GM as well.

Since then, Lean manufacturing has evolved into Lean Thinking therefore it is now applicable beyond production and manufacturing. A number of research projects and case studies have been undertaken with regards to Lean applications; not just in the automotive industry but also in Clinical research [13], food supply chains [14] and in governments [15]. It has also been shown that Lean can help in complex product development [16].

Lean is also present in the Aerospace industry. In 1993, Massachusetts Institute of Technology set up the Lean Aerospace Initiative to investigate the applications of lean methodologies in the aerospace industry, which led to major cost and schedule savings [17, 18]. A UK-based initiative, formed by four Universities and a consotrium of Industrial partners, has also undertaken similar research [19].

Further, a large European project called LeanPPD (Lean Product and Process Development) was set up to address the need of European manufacturing companies for a new lean model beyond just manufacturing [20, 21].

While Ford and GM showed a glimpse of what Lean production was capable of, Toyota applied the Lean thinking to the whole product development process that came to be known as the Toyota Product Development System (TPDS) [22]. The TPDS is a socio-technical framework that involves people, processes and technology in order to eliminate anything that doesn't add value to the final product. Such waste can come from, for example, overproduction, waiting times in materials and information, processing and corrections. Generally, any activity that lengthens the lead time without adding value and adds extra cost is considered waste.

In order to meet the defined value and produce the minimum viable product, Toyota linked all disciplines and departments to work together; the design studio, engineering, product planning, finance and others. Front-loading, which is the allocation of the majority of the resources at the beginning of a process, is also required to ensure the availability of either people or materials, as early as possible in the design process. Cross-functional participation is also essential in order to maximize effectiveness in the development process. This cross-functional simultaneous development became known as Set-Based Concurrent Engineering (SBCE) and it is now a core element in the TPDS [23].

2.1.2 Engineering Design

Engineering design can be described as a process to devise a system in order to satisfy a set of predefined requirements [24, 25]. The process varies considerably from one approach to the other. Howard [26] compared more than twenty different engineering design processes and outlined how they differ in each step. Choosing the appropriate process is a critical factor for a successful design as Khandani states [27]. Regardless of the approach used, the process aims to provide a framework that makes the process efficient, clarifies the problem, drives innovation, manages complexity, and promotes collaboration [28]. Engineering design has been of research interest for decades with Pahl et al. stating that it could even be dated back to Leonardo Da Vinci [29].

Engineering design faces a number of challenges which usually arise from problems that are not well defined or structured. Requirements and objectives tend to change multiple times in the design phase alone, especially in large and complex projects such as an aircraft development where the design phase lasts for years. With such design problems, not having just one correct solution makes a thorough and robust decision making process necessary in order to find the best option [30].

Most literature that deals with engineering design uses a requirements-based, or requirements-first, approach [8]. However, Ring [31] proposed that a value-seeking approach could produce better results by first focusing on understanding and describing the problem; rather than starting with setting requirements. Furthermore, arising from the large European program CRESCENDO, Value-

Driven Design(VDD) was introduced which focuses on understanding what drives the generation of stakeholder value. VDD aims to "strengthen the value and requirements maturation process" [32].

Regardless of the specific approach, the general process for engineering design usually involves the following steps:

1. Problem Definition: The first step is to identify and define what the problem is and why it needs to be solved

2. Background Research: After the problem has been identified, a background research is necessary to review what has already been done or any alternative solutions to the problem. This step also assists in avoiding any previously made mistakes.

3. Requirements Specification: In order for the solution that will be proposed to be able to solve the problem, a number of requirements must be met.

4. Identification of potential solutions: Having the requirements specified, a process of generating potential solutions to meet those requirements begins.

5. Development: After the desired solution or solutions have been selected, the design maturity begins. This involves continuous evaluation, refinement, and improvement of the design, usually in an iterative process.

6. Prototype: A prototype, whether physical or digital, is the first operating version of the solution that will undergo further tests.

7. Test and refinement: The prototype undergoes a series of tests and is being refined. This will give rise to any necessary changes that need to be made

or any improvements identified.

8. Results: The completion of the design phase needs to produce the necessary results that will be communicated to the appropriate shareholders. The results contain information with regards to the reasoning behind the selected design and performance metrics.

The process can be split into two main approaches. The first, the iterative process, which is also known as point-based approach or point-based design. In this case, only one initial design is selected from a pool of candidate solutions. The design is then refined and reworked in an iterative process until the final desirable version emerges. Variations of the point-based approach have been proposed by French [33], Pahl [29], Pugh [34], and Ullman [35] amongst others.

In contrast to the iterative approach, the second one is the convergent approach. In this case, a number of potential solutions are selected, which are then matured and evaluated in parallel. The convergent approach has an iterative process embedded but in a different way. Whereas the iteration in point-based approaches is for redesigning and refining the solution, the iterations in a convergent approach primarily aim to discard undesirable, infeasible, or non-robust solutions. The remaining desirable ones, will then undergo further development. Convergent approaches include the Method of Controlled Convergence(MCC) by Pugh [34], the Design-Build-Test (DBT) cycle by Wheelwright and Clarke [36], and the Set-Based Design (SBD) [37].

Raudberget, in a comparison between Pugh's MCC and the SBD, found the MCC process and description "questionable and contradictive" and hard to implement [38]. The same research also found that the firm in which the comparison took place, had a preference to the results that have arisen from the SBD approach. Also, the SBD approach allows different disciplines to independently analyse their respective areas, hence reducing and sometimes eliminating iterative paths; this is a benefit that the other two approaches do not necessarily provide [39, 40]. The SBD approach is described in more detail in the next section.

2.1.3 Set-Based Design

Conventionally the design of a system follows an iterative or point-based approach. An initial design is selected and reworked over and over again until a desirable outcome emerges. Arising from the TPDS, Set-Based Design (also referred to as Set-Based Concurrent Engineering) allows parallel evaluation of multiple alternative configurations in the early stages of design by using sets of possible solutions [37, 40] .

The procedure, as described by Sobek [23], starts by mapping the design space and defining the feasible regions while exploring the trade-offs by designing different alternatives. A number of feasible concepts are developed based on sets that are fully functional across all disciplines and looks for possible intersections between those feasible sets. Finally, the sets are narrowed down and increase the detail of the designs; given that the narrowing down is done consensually amongst multidisciplinary teams, therefore creating commitment from each of the disciplines.

When dealing with physical systems that have discrete design parameters (i.e. subsystem options or components), generating the configurations is straight forward. The first step is to identify the design parameters of the system and tabulate their respective possible options. Different combinations of options generate a set of competing configurations. The configurations are then evaluated against predefined performance metrics, and the non-robust or undesirable ones are discarded; the rationale for discarding is recorded as information for future use. The remaining configurations are then matured, and the evaluation process is repeated until the desirable configurations emerge. Figure 2.1 shows the outline of the process.

The Set-Based Design approach offers a number of advantages over traditional development processes; it has been well documented with some authors even claiming that the approach (and related Lean practices) can be up to four times more productive [22, 41, 42]. Delaying critical design decisions, in order to fully understand the customer specifications and requirements, can reduce the possibility of design changes later on, which otherwise could have been costly [43, 44].

The approach has also been shown to have positive effects on the resulting product and the development process itself. Not only does it avoid costly reworks, the risk of failure is also reduced (due to other feasible solutions being available at all stages). It also promotes innovative thinking and creativity along with organisational knowledge and learning [21, 45].

The topic of SBD is an active area of research and, though limited, industrial applications of Set-Based Design have been documented [46]. Ward and Sobek argued that SBD was a key to Toyota's and TPDS success [23, 43]. Raudberget et al. [47, 48] investigated how Set-Based Design can be implemented in existing product development companies through industrial case studies, while Mebane et al. [49] applied the SBD approach to the design of a Ship to Shore Connector. Set-based concurrent engineering has also been the main enabler of the large European project, LeanPPD, which aimed to address the need of Eu-

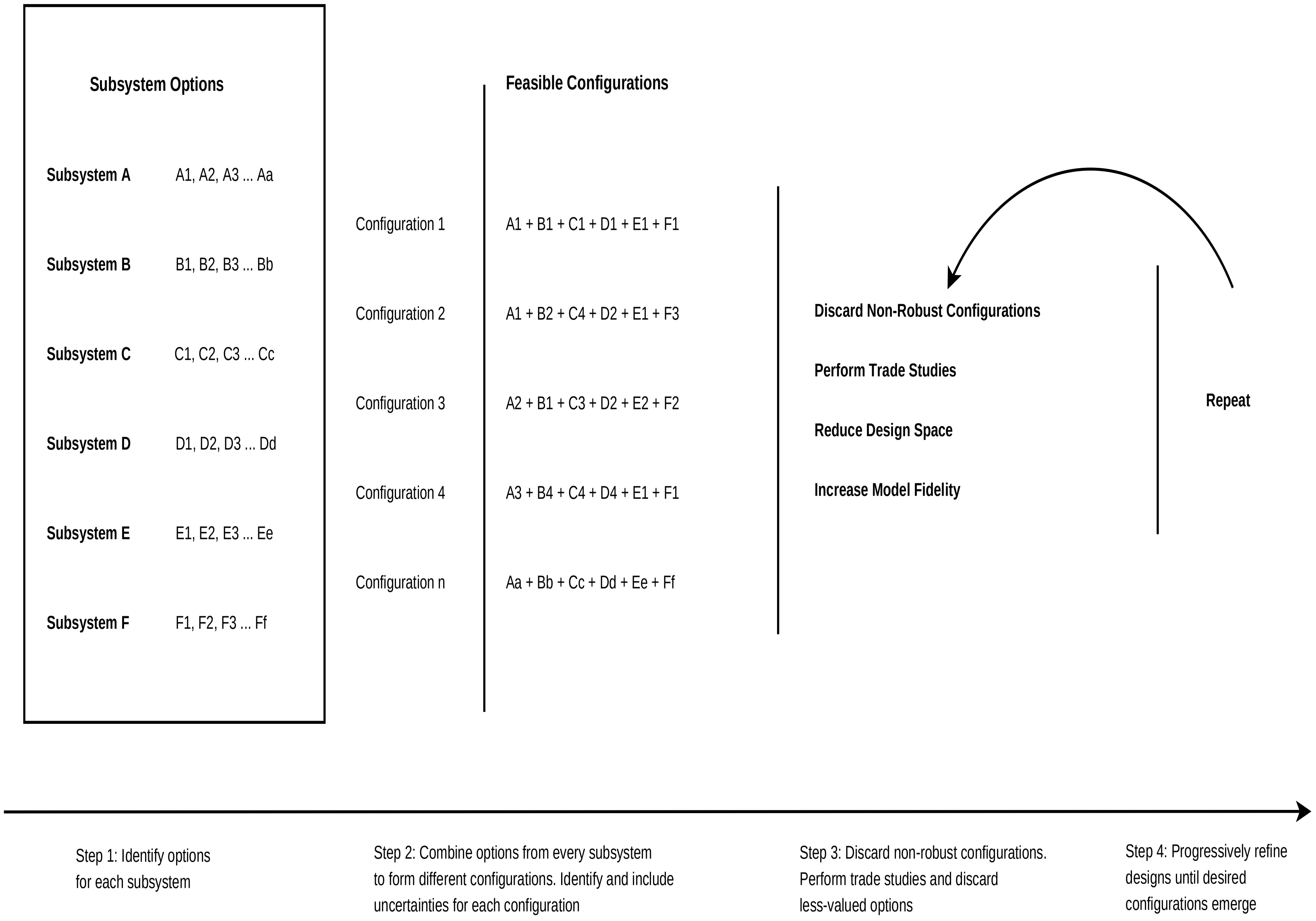


Figure 2.1: Set-Based Design Approach

ropean manufacturing companies for a new lean model beyond just manufacturing [20, 21, 50, 51]. Furthermore, Guenov et al. [52] developed an interactive computational tool that facilitates a set-based design workflow and applied it to the design of a conceptual aircraft .

2.1.4 Comparison to Iterative approaches

Where the set-based approach considers a lot of design options both at system level and subsystem level, and eventually narrows down to one, point based deals with just one design configuration from the beginning. This main difference means that both approaches have their benefits and drawbacks when compared against each other [20].

With the ability to consider many alternative options simultaneously, the Set-Based approach offers a lot more flexibility. This is particularly true in multidisciplinary cases where, due to the modularity of the approach, it offers a significant advantage over the point-based approach. However, the flexibility of having competing configurations, synthesised by different options for each design parameter, brings a potential drawback in the process. If the type and range of uncertainties varies between those different options, it can complicate the uncertainty management process; compared to an iterative approach where only one option is considered.

Furthermore, the flexibility and consideration of multiple options means that there is a delay in the decision making process in order to converge to the final chosen design configuration; unless the narrowing is done aggressively. As a result, it can make it a more resource demanding process, especially in early stages due to front-loading.

Even though the initial costs might be higher for SBD, the availability of alternative solutions at any point in the design process, reduces the impact of a possible design change. Such a feature is not provided by a point based approach, increasing the cost of a potential design change or design failure. It is down to the management to decide whether or not the extra upfront costs and investment is valued over the flexibility and modularity that is provided by the Set-Based approach.

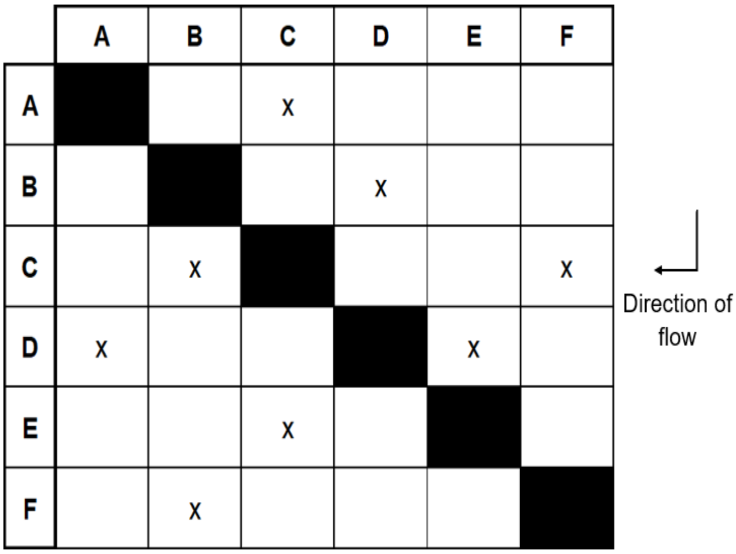
Due to the knowledge generation that SBD provides, future implementations of it in similar projects can reduce the development time because the information is readily available. In an iterative process, the knowledge generated is very limited because of the confined and constrained design space such an approach takes place in.

It is the author's opinion, that in the case of complex systems such as an aircraft, SBD is the better suited option. This is due to the fact that an aircraft design is a long, costly, and complex process that involves many disciplines. This creates a multidimensional design space that a point-based approach would struggle to explore thoroughly. Furthermore, due to the high possibility of changes in initial design and requirements, flexibility in the initial stages is essential, as design changes occurring in later stages using a point based approach could yield prohibitively high costs. Adding to that, aircraft manufacturers tend to provide different configurations for each aircraft (aircraft families), something that SBD and platform-based design can accomplish [53, 54, 55, 56].

2.2 Matrix Methods in Engineering Design

2.2.1 Design Structure Matrices

Design Structure Matrices(DSM) are square(NxN) matrices that are widely used as compact network modelling tools in engineering design for visualising and decomposing system architectures [57, 58, 59]. When designing any complex multidisciplinary system, it is not only helpful but also vital to be able to visualise the dependencies between the elements of the system [10]. A DSM does exactly that, by showing the couplings/interactions between activities, people, components or parameters of a system, depending on the type of DSM (Organisation, Process or Product). A simple example of a DSM is shown in Figure 2.2.



	A	B	C	D	E	F
A	■		X			
B		■		X		
C		X	■			X
D	X			■	X	
E			X		■	
F		X				■

Figure 2.2: Example of a Design Structure Matrix (DSM)

The elements of the system are represented in each row and column. A mark in the upper right or lower left part of the DSM indicates a connection or coupling between two elements. The elements in a DSM can be clustered together depending on their coupling or rearranged for sequencing, depending on the DSM that is being considered. A number of algorithms for automatic clustering and sequencing have been developed [60, 61].

There are three possible dependencies between two tasks: Independent (parallel), Dependent (sequential), and interdependent (coupled). Figure 2.3 shows a visual representation of the three states along with their respective representation in a DSM. In the case where an element is dependent on another (sequential), the column element is considered the initiator, and the row element is considered the receiver. In the example given in Figure 2.3, B is dependent on A. This relational notation is not standardised as it varies from author to author; this notation is the one this project is using.

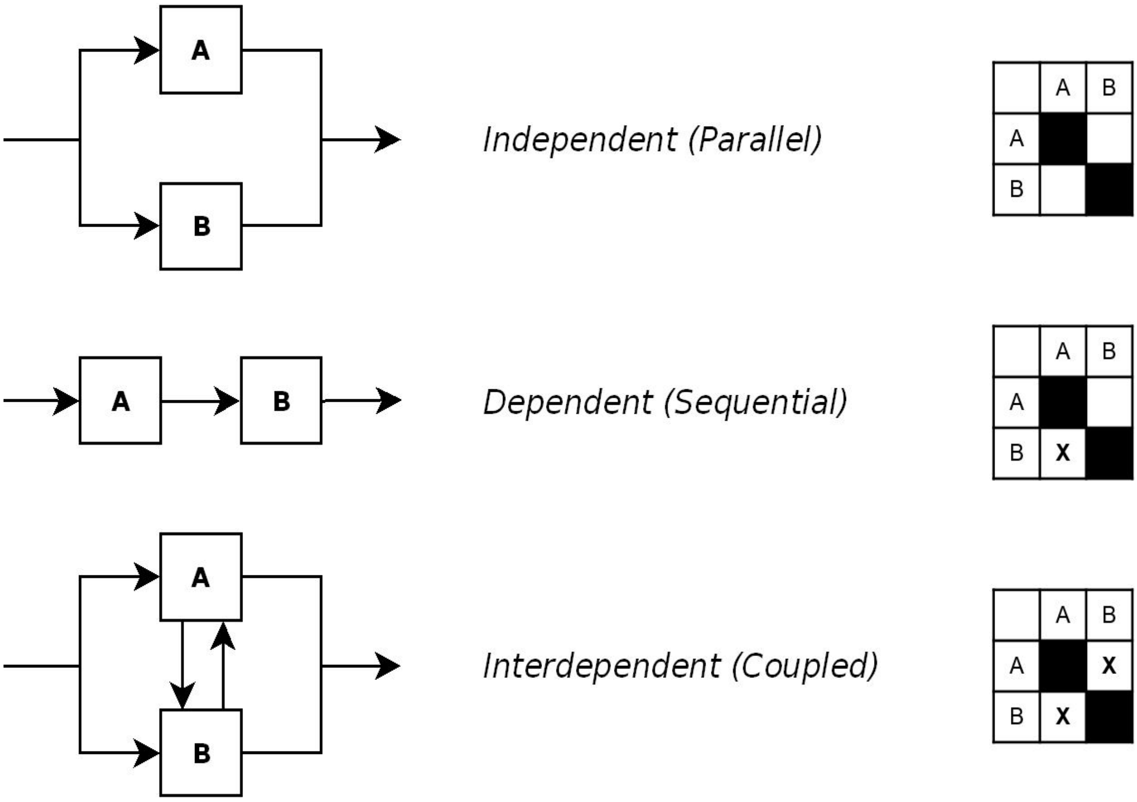


Figure 2.3: Design Structure Matrix representation of the three dependency types

DSMs have been widely used as an effective tool in systems design and project management whether the system is a product, process, or even an organization and in a wide range of industries. An extensive and detailed study in the use of DSMs, their applications and extensions was undertaken in a survey by Browning [62]. One of the useful extensions of DSM are the MultiDomain Matrices (MDM) or Domain Mapping Matrices (DMM) that map elements from one DSM to another [63]. This is particularly useful in mapping different DSMs, for example people to processes and processes to components [57]. An eXtended DSM (xDSM) was developed by Lambe and Martins [64] that captures not only dependencies, but also data flows, and it has been primarily used in Multidisciplinary Design Optimisation.

2.2.2 Other Methods

Besides DSMs, a plethora of tools have been developed to assist the engineering design process, varying from computational to paper based methods. Depending on the design stage, system in consideration, resources and other factors, different tools are applicable in each scenario. A large number of those tools were implemented using matrices, which have come to be known as matrix methods, or, matrix-based tools. A brief overview of the most common ones are presented in Table 2.1.

House of Quality, one of the tools in Quality Function Deployment (QFD), was originally developed to map customer requirements, and expectations, to the system's characteristics and properties [65]. The HoQ is made up of two parts: The HoQ roof and a relationship matrix. Anderson [66] also demonstrated how the HoQ can be used to demonstrate the dependencies between

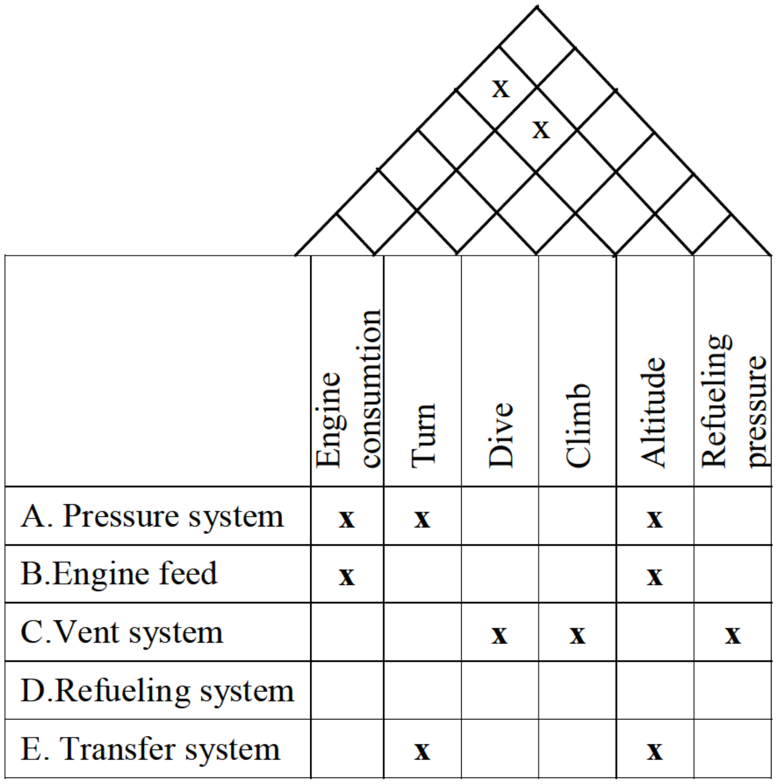
Matrix Method	**Use**
Quality Function Deployment(QFD) and House of Quality(HoQ)	Mapping customer requirements to engineering characteristics
Design Structure Matrices (DSM)	Network Modelling, System Architecture decomposition
Morphological Matrices	Synthesize Configurations / System architectures
Axiomatic Design	Mapping functional requirements to design parameters
Analytic Hierarchy Process(AHP)	Decision making support for Trade Studies

Table 2.1: Overview of common matrix methods in engineering design

subsystem and top level requirements. Gavel [67] showed an example of a HoQ for a simplified aircraft fuel system as shown in Figure 2.4. In the example, the roof of the HoQ shows the dependencies between top level requirements; for example, the altitude affects the engine consumption and the turn. Furthermore, the relationship matrix shows the connections between the subsystems and the top-level requirements; the engine feed subsystem for example is impacted by the engine consumption and altitude requirements.

Whilst the HoQ is useful for mapping requirements to characteristics and properties, for the design stage and for system decomposition there are better suited tools, such as the Design Structure Matrix which was previously presented.

Morphological matrices are very useful tools for decomposing systems and synthesising configurations, hence its applicability in SBD. Each row in the matrix represents the subsystems or elements of the system being design, while the



	Engine consumtion	Turn	Dive	Climb	Altitude	Refueling pressure
A. Pressure system	x	x			x	
B.Engine feed	x				x	
C.Vent system			x	x		x
D.Refueling system						
E. Transfer system		x			x	

Figure 2.4: Example of a House of Quality for a simplified aircraft fuel system [67]

columns represent the design options for each one. Figure 2.5 shows a simple example of a morphological matrix. By taking one of three options for each of the four subsystems, a configuration (A) is generated. Different combinations of different options will generate competing configurations.

Different algorithms have been developed for quantifying morphological matrices and forming optimisation problems [68, 69]. Morphological matrices can be useful for generating concepts for simple systems; however, they can become very complicated and a time consuming process for large, complex systems with multiple elements and options. Furthermore, as Frank et al. [70] have mentioned, even though morphological matrices do offer a good first approach to generating configurations, they can't ensure their feasibility.

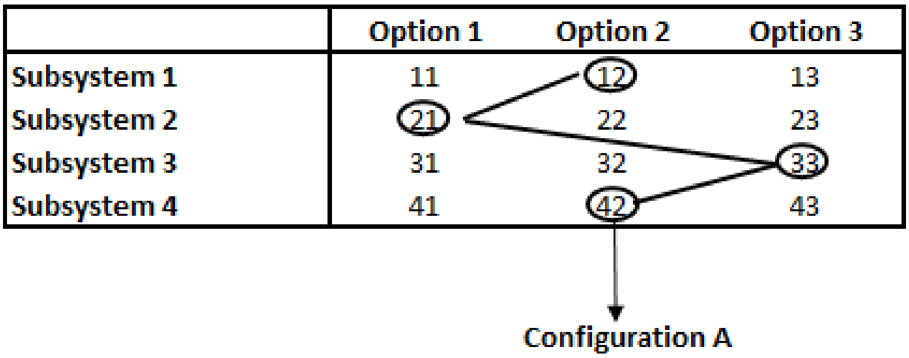


Figure 2.5: Example of a morphological matrix

Axiomatic Design was developed by Suh in 1990 [71] to map functional requirements to design parameters. It has since then evolved considerably, showing how a systematic approach to design can improve efficiency and productivity [71, 72]. Axiomatic design matrices consist of a Functional Requirements (FR) array, a Design Parameters(DP) array and a relationship/mapping matrix (A) as shown below:

$$\begin{bmatrix} FR_1 \\ FR_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \end{bmatrix}$$

The method is based on two fundamentals called axioms (hence the name):

- Independence Axiom: FR and DP, where possible, should remain completely independent. To achieve that, the relationship matrix (A) must be a diagonal matrix.
- Information Axiom: Information and hence complexity should be kept to a minimum.

Finally the Analytic Hierarchy Process (AHP) is one of the most widely used tools for assisting the decision-making process in engineering design [73]. AHP is part of a number of methods called MultiCriteria Decision Making (MCDM).

It aims to rank different solutions against performance criteria, with each criterion having an importance or preference weight. As with many other MCDM methods, such as the Choosing By Advantages (CBA), AHP has its benefits and drawbacks [74].

Even though not a matrix-based method, a decision tree is another well used decision support tool. However they are not considered for this work due to their limitations and impracticalities when dealing with larger systems [75].

2.3 Engineering Change Propagation

2.3.1 Introduction

In the current age, where product development is driven by rapidly changing market requirements, design changes can occur at any point during the development of any system or product. As the development process progresses, the cost associated with any design change increases, sometimes by a factor of 10 [76, 77]. This calls for an effective engineering design change management; this is defined as the process where design changes are assessed, evaluated, implemented, and controlled [78].

Such design changes do not only affect the element of the system that needs to be redesigned but might also lead to changes to the whole system. This phenomenon is known as change propagation and has been a topic of interest, especially in engineering design. Substantial research has been undertaken to develop methods that can predict and manage change propagations. Clarkson and Eckert presented an overview of the methods [79]. Further, Keller, Hamraz, Caldwell and Clarkson compare and contrast many of these methods [80, 81].

Change propagation becomes even more important when dealing with more complex systems especially in cases where elements, subsystems or components, are heterogeneously (tightly and/or loosely) coupled. The ability to predict the propagation of design changes, and quantify their impacts, can assist in a better evaluation and analysis of the system's sensitivity to design changes.

A number of different approaches have been developed in order to model change propagation, such as the Change FAvorable Representation (C-FAR) [82]. C-FAR uses vectors to define the components and their attributes, with their characteristics as the vector elements, and then uses a matrix to capture dependencies amongst these characteristics. It starts by defining the propagation paths, then multiplies the change vector with the respective dependency matrix, and then aggregates all possible paths. This method can be slow and computationally expensive for large, complex systems because it is based on a large number of matrix and vector multiplications to calculate all possible propagation paths.

2.3.2 Change Propagation Method

The other major approach for predicting design changes, is CPM (Change Propagation Method); a method that makes use of Design Structure Matrices [83]. A DSM succinctly represents the dependencies amongst the components of a system, which is why it makes it such a powerful tool for predicting change propagation.

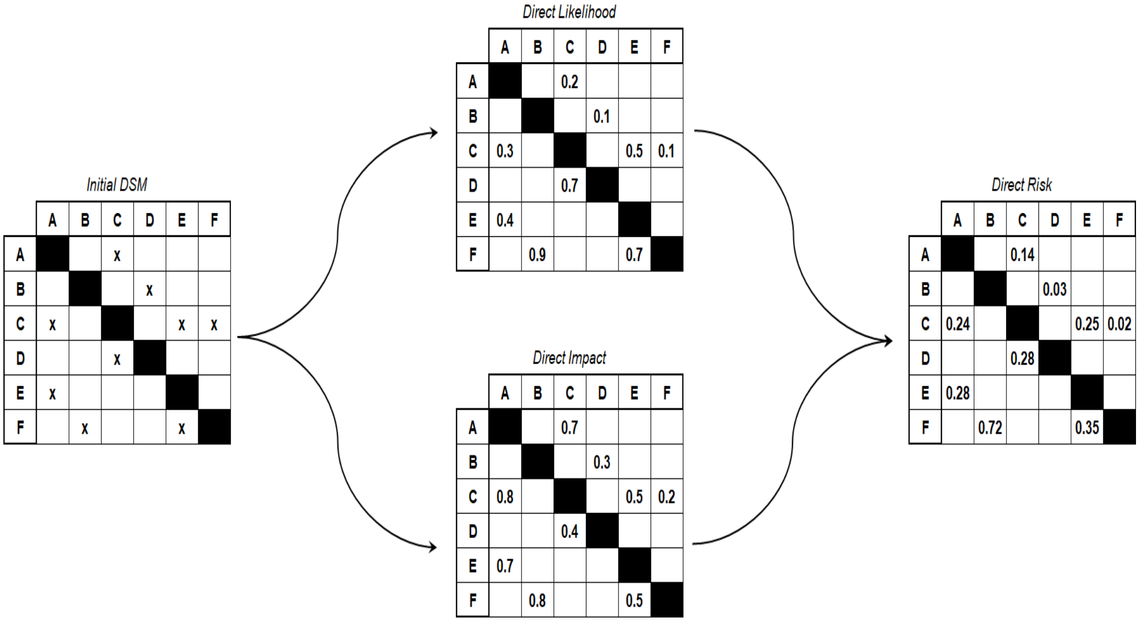
In the case of CPM, a way to interpret the DSM is that for any connection in the matrix, the corresponding column is the component that triggers the change (transmitter), and the corresponding row is the component that receives the change (receiver).

CPM makes use of 3 main values: Likelihood, Impact and Risk. Likelihood measures the probability of one element having an effect on another element (ie, how likely it is that the change will propagate), while impact measures the magnitude of the effect from the initiating element to the receiving one. Risk is the product of likelihood and impact.

In order to calculate the change propagation, initially two quantified instances of the DSM are generated as shown in Figure 2.6. One includes the value of how likely the change will propagate directly from one component to the other; the other DSM measures the impact of that design change on the receiving component. All values of Likelihood, Impact and Risk are measured between 0 and 1 (sometimes expressed as a percentage). The values of direct Likelihood and Impact are obtained through experts' opinions and the risk is the product of the two values.

The first two quantified DSMs indicate direct Likelihoods and direct Impacts, and consequently the product is the direct Risks. Direct (or 1st order) indicates that there are no intermediate components and one component directly connects to the other. A 2nd (and higher) order of change propagation takes into consideration indirect connections as well, due to change propagating through intermediate components, therefore adding other paths of propagation.

To illustrate this using critical path networks, Figure 2.7 demonstrates the direct, 2nd order and 3rd order propagation of a change between components A and C; as defined in the DSM presented in Figure 2.6. It is important to notice however that the algorithm avoids cyclic routes, meaning that a component cannot be affected twice in the same path.



Initial DSM

	A	B	C	D	E	F
A	■		x			
B		■		x		
C	x		■		x	x
D			x	■		
E	x				■	
F		x			x	■

Direct Likelihood

	A	B	C	D	E	F
A	■		0.2			
B		■		0.1		
C	0.3		■		0.5	0.1
D			0.7	■		
E	0.4				■	
F		0.9			0.7	■

Direct Impact

	A	B	C	D	E	F
A	■		0.7			
B		■		0.3		
C	0.8		■		0.5	0.2
D			0.4	■		
E	0.7				■	
F		0.8			0.5	■

Direct Risk

	A	B	C	D	E	F
A	■		0.14			
B		■		0.03		
C	0.24		■		0.25	0.02
D			0.28	■		
E	0.28				■	
F		0.72			0.35	■

Figure 2.6: Quantifying the initial Design Structure Matrix with direct Likelihoods and direct Impacts, and calculating direct Risks

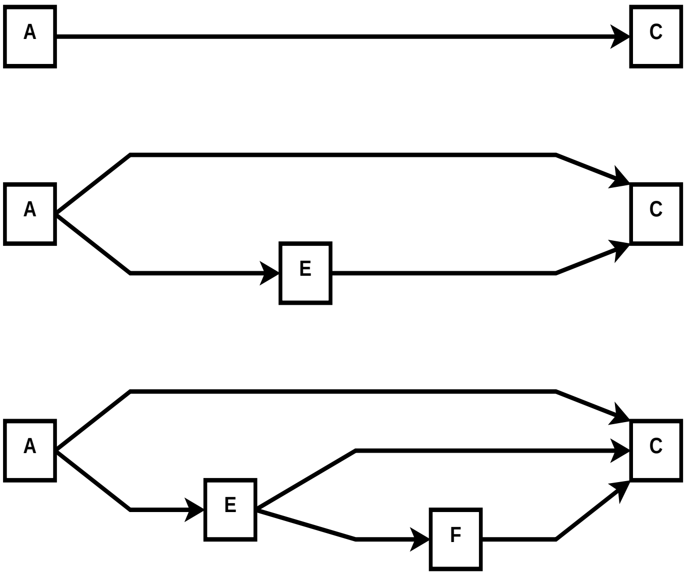


Figure 2.7: Different paths of change propagating from A to C, depending on order of propagation (1st, 2nd and 3rd)

A number of algorithms have been developed to calculate the combined Likelihoods, Impacts and Risks, depending on the order of propagation that is of interest. The most prominent one is the forward-CPM algorithm [83]. It uses a brute-force search method and it analyses all possible paths one by one. Other algorithms for calculating the combined values have been developed and compared, for example, the trail counting algorithm, and matrix multiplication-based [80, 81].

Suh, Weck, and Chang [84] also introduced a Change Propagation Index (CPI), that identifies whether a component is a multiplier, carrier, absorber, or constant; these terms were introduced by Keller [80].

The limitation with the methods developed so far is that none of them considers the cost that is associated with change propagation. In their paper, Koh, Caldwell, and Clarkson, do include cost in their methodology, but they only consider the cost of the final affected component [85]. It does not consider how the cost is accumulated by the intermediate changes between the initiating and receiving components.

2.4 Optimisation

2.4.1 Introduction

Optimisation methods aim to reach an optimum or a non-dominated solution by minimising or maximising an objective function $f(\mathbf{x})$; the generic process is shown in Figure 2.8. The function is dependent on a number of design variables that form the design vector $[\mathbf{x}]$, which can be bound by constraints [86].

$$\min f(\mathbf{x}) \tag{2.1}$$

subject to:

$$x_{iL} < x_i < x_{iU} \qquad\qquad x_i = 1, 2, 3 \ldots n$$

where x_{iL} and x_{iU} are the Lower and Upper bounds of each design variable respectively.

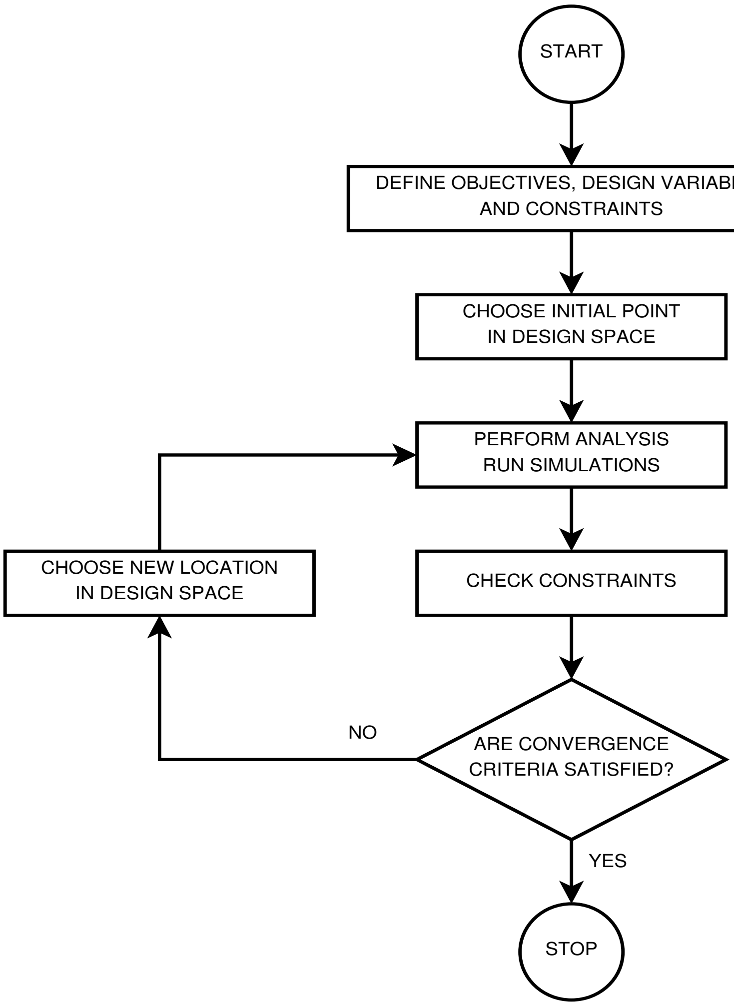


Figure 2.8: Generic optimisation process

Optimisation has been an active field of research in the aerospace industry for many years with applications ranging from aerodynamic and aeroelasticity optimisation [87, 88], to structures [89], weight and manufacturability [90].

Most real-world problems, especially in multidisciplinary design optimization (MDO), involve more than just one objective. Multiobejctive optimisation (sometimes referred to as multicriterion) deals with two or more objective functions:

$$min[f_1(\mathbf{x}), f_2(\mathbf{x})...f_n(\mathbf{x})] \tag{2.2}$$

In such cases, there is usually a trade-off between objectives therefore there is no unique optimum solution but a range of non-dominated solutions.That is, solutions that perform better at one objective but worse towards others.

When dealing with biobjective optimisation (two objective functions), a common method to visualise the non-dominated solutions is the pareto front shown in Figure 2.9.

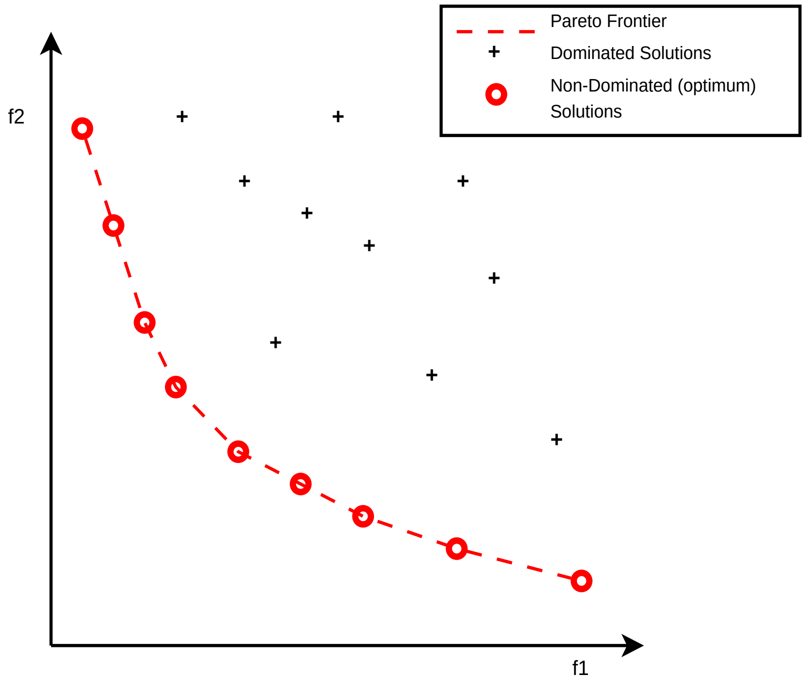


Figure 2.9: Pareto graph for a 2-objective optimisation (minimisation) problem

The pareto front method is useful for visualising the trade-offs between two objectives. From there on however it is limited in the sense that the respective variables for each solution are not visualised and the graph cannot be applied in optimisation problems with more than two objectives.

2.4.2 Multidisciplinary Design Optimisation

When designing complex systems such as an aircraft, the optimisation process needs to take into consideration the different disciplines involved, such as structures, aerodynamics, aeroelasticity and costs. Optimisation methods that aim to address and take into consideration more than one discipline are known as Multidisciplinary Design Optimisation(MDO) methods. Sobiesky and Haftka [91] defined MDO as “a methodology for the design of systems in which strong interaction between disciplines motivates designers to simultaneously manipulate variables in several disciplines”. However, the term has evolved to encompass any kind of optimisation process that involves more than one discipline [91] or strongly coupled elements [92].

A large number of optimisation algorithms (optimisers) that search for solutions have been developed. The selection of the right algorithm is very important depending on the type of problem that is considered, as the process usually requires many iterations until it converges to a solution. Algorithms can be categorised in a lot of different ways such as stochastic, deterministic, heuristic, gradient and gradient free. Different algorithms perform better depending on number of objectives, number of design variables, and whether the design space is discrete or continuous.

One of the issues that can arise from MDO problems, is a design space with multiple local minima or maxima where optimisers can get trapped in; in such cases, finding the global optimum solution becomes even more complicated and hard. A number of surveys and benchmarks throughout the years have focused specifically on comparing and evaluating optimisers [93, 94, 95, 96, 97, 98].

Another important factor that needs to be taken into consideration when implementing MDO methods, is the architecture that will be used; that is,the process that will be followed for the optimisation problem to be solved. MDO architectures can be split into two categories: Monolithic and Distributed. There has been considerable interest in comparing and benchmarking MDO architectures [99, 100, 101].

Monolithic architectures form one objective function which then try to solve. Starting from the All-At-Once architecture where all variables and constraints are considered and controlled by the optimizer, the three other major monolithic architectures can be derived [102]. The Simultaneous Analysis and Design (SAND) and Individual Discipline Feasible (IDF) architectures can be derived by removing the consistency and discipline analysis constraints respectively [103]. Removing both types of constraints results in the Multidisciplinary Feasible(MDF) architecture [102].

On the other hand, distributed architectures decompose the single function into smaller problems and then solve them. This allows for individual groups or disciplines to run independently while communicating between them. This is particularly important in multi-fidelity optimization where one discipline requires more computational power than others. This is not possible in monolithic architectures where all discipline-analysis are run the same number of times. All Distributed architectures are derived from the monolithic IDF and

MDF architectures. They usually consist of system level and discipline level optimisers. There are a lot of different distributed architectures. Starting from the Concurrent Subspace Optimisation (CSSO) which is one of the oldest, to the newly developed Asymmetric Subspace Optimisation (ASO) which focuses on multi-fidelity optimisation [104, 105].

2.5 Comparison between Set-Based Design and Multidisciplinary Design Optimisation

The two main areas that have been presented, SBD and MDO, are evidently very useful for different applications and at different stages of design. Table 2.2 outlines the main differences between the two.

Set-Based Design	**Multidisciplinary Design Optimisation**
Multiple Initial Designs	One Initial (Reference) Design
Aim to identify infeasible and undesirable solutions and reduce design space	Aim to identify optimum and non-dominated solutions
Usually used in early stages for concept selection and physical configurations	Used in later stages for fine-tuning of engineering characteristics
Engineering Design approach	Methodology for fine-tuning engineering characteristics

Table 2.2: Differences between Set-Based Design and MDO

As it is evident from Table 2.2, there are a number of important differences, the most important one being that SBD is an approach to designing systems, whereas MDO is used as methodology to fine-tune the engineering characteristics of a system. Set-Based Design is primarily used in early stages of design where the design space is large and not as constrained as it will be at later stages. A number of different options are available for a each domain and subsystems, and a combination of those forms a number of configurations that are used as initial designs. Those designs will then be assessed and evaluated in order to remove the infeasible, non-robust, or less desirable ones. The assessment, evaluation, and discardment process is repeated until a final desirable design emerges. At each iteration, the configurations that haven't been discarded are matured and assessed against new performance metrics and criteria.

On the other hand, MDO tools are employed at a different stage of the design process and for different reasons. It is primarily used to fine-tune engineering characteristics in order to achieve pre-defined performance objectives while meeting the required constraints and parameters. As a result, a tool like this is used at later stages of the design process where an overall configuration has been selected. The aim in this case is to find the optimum and non-dominated variations of that design as opposed to the set-based approach where the aim is to eliminate infeasible and undesirable ones.

There are a number of previous research publications that attempted to bring together the two areas of Set-based design and optimisation. Hannapel and Vlahopoulos, introduced principles of Set-Based Design in MDO where a new algorithm was developed that studies the design variables in terms of sets [106]. Veenhuis modified the Particle Swarm Optimisation (PSO) algorithm by replacing the position and velocity vectors of the particles with sets [107].

However the aforementioned researches only use principles of Set-Based Design for developing novel optimisation methods and algorithms. In other words, they integrated a Set-Based approach in an MDO algorithm. The framework developed as part of this project aims to address what no previous research attempted to do; to integrate MDO tools in a Set-Based Design approach. This will bring the two areas together in a unified framework that exploits the benefits of both.

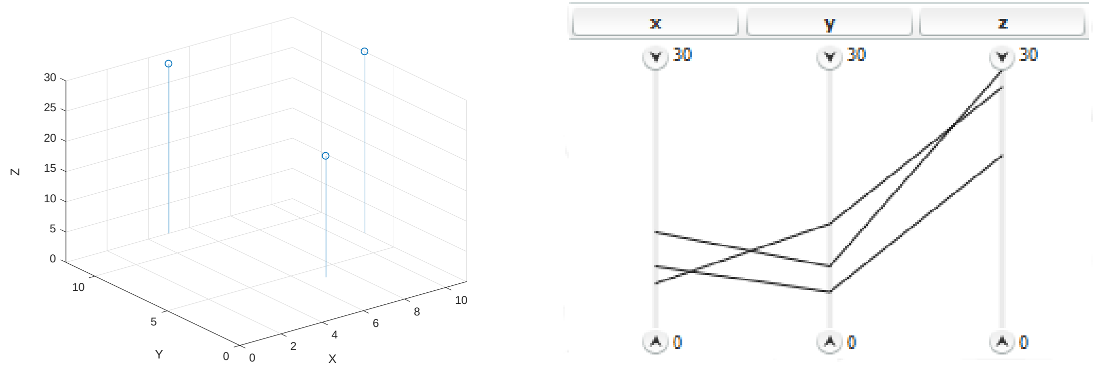
2.6 Visualisation

Throughout this literature review a number of visualisation methods have been presented, signifying the importance of appropriate visual representation depending on the kind of information provided. Starting with DSMs that can be used not only as network modelling and system decomposition, they also assist in visualising connections between system elements and dependencies. DSMs are also important in change propagation methods to show design dependencies and quantify the probability and impact of those dependencies. Change propagation methods also make use of critical path networks to demonstrate how a design change in one element can reach another one with all the possible paths in between them.

When it comes to optimisation, the pareto front in a scatter plot is useful for visualising the trade-offs between two objectives. From there on however it is limited in the sense that the respective variables for each solution are not visualised and the graph cannot be applied in optimisation problems with more than two objectives.

For this reason, parallel coordinates are employed to help visualise the different solutions for more than three objectives along with their respective variables. This not only allows for multidimensional data visualisation, but it also assists in identifying positive and negative relationships between objectives and variables [108]. This makes parallel coordinates a very powerful tool in visualising multidimensional data especially in engineering design, but also driving optimisation processes [109, 110].

Parallel coordinates consist of parallel axes that each represent the objectives functions and any other parameters that are of interest. Polylines are then being used, passing through all of the axes, with each line representing a solution. Figure 2.10 illustrates a simple example of how three random 3-Dimensional points in a scatter (stem) plot can be converted to parallel coordinates. Each point has 3 values, one for each dimension, and those values are connected with a polyline between the axes. There are only 3 axes since there are 3 dimensions. Each polyline represents one point in a multidimensional space. Each additional dimension requires an additional parallel coordinate axis to be introduced.



(a) 3D Stem plot

(b) Parallel Coordinates

Figure 2.10: Stem Plot for three points and the respective parallel coordinates visualisation

Part II

The ADOPT Framework

Chapter 3

Introduction

3.1 Overview

As mentioned in the previous Chapter, integration of optimisation tools within a Set-Based Design approach, has never been attempted before. A number of other tools were also identified that can assist in the proposed approach, such as DSMs and change propagation methods; along with visualisation tools to communicate the generated data. However, since current change propagation approaches do not consider the associated costs, a new method had to be developed to address that gap.

A novel framework has been developed to address the problem and research questions stated in the Introduction. The Augmented set-based Design and OPTimisation (ADOPT) architectural framework, integrates features from all the areas covered in the literature review, and addresses the gaps previously stated.

ADOPT is built on the theory of Set-Based Design, integrating optimisation tools, and has been developed in order to improve the converging process in numerous ways. The framework is split in two main stages. The first stage deals with generating the configurations and checking for initial feasibility constraints and desirability. The second stage deals with the optimisation of each configuration and generation of alternative designs in the case of Multiobjective optimisation. The second stage is iterative where at each iteration, configurations are eliminated, reducing the design space and allowing the remaining configurations to be evaluated in more detail. Overall, the framework aims to provide a process and methods for a more thorough design space exploration in an efficient and robust approach.

The framework also includes methods for assessing the sensitivity to design changes of the different configurations. Furthermore, CP^2[111] was developed as a method for predicting the cost associated with the propagation of design changes. This acts as an additional metric for evaluating and assessing a configuration beyond the conventional performance ones.

In order to communicate the results and manage the information generated, a number of tools and methods for traceability and visualisation are used.

This chapter introduces ADOPT and outlines the main features of it along with their potential benefits. The second chapter of this Part examines the two stages in detail with their respective steps and processes.

3.2 Framework Features

Exploring a large and complex design space, seeking non-dominated and robust solutions, can be a challenging and computationally expensive process as previously mentioned. In a continuous space, there are infinite number of solutions, in the same way that there are infinite numbers between any two real numbers. By segregating that continuous space in ranges, a finite number of areas are created, that can be explored independently. This ”discretisation” of the design space allows optimisers to search for solutions in more confined spaces making the process faster, while ensuring that only the areas that are of interest are explored.

Furthermore, the framework aims to improve the conventional Set-Based approach in a number of ways:

1. Taking into consideration the different levels of a system, from high level geometry to low level components. This will allow for early identification and elimination of incompatible, infeasible, or undesirable combinations and configurations. It also brings forward the development of domain-specific systems, earlier in the design process.

2. It can handle any kind of design parameter whether a continuous or discrete one. Continuous parameters are discretised into ranges, which are then combined with the remaining discrete ones to form architectures / configurations. In these cases they will then be reverted back to continuous ranges, the bounds of which will act as optimisation constraints.

3. By integrating optimisation tools, each architecture can be fine-tuned with regards to its engineering characteristics. This will lead to generation of different non-dominated and robust design alternatives for each architecture, while respecting the constrains of each search area.

4. Optimising each area in the discretised design space assists in a more thorough exploration; as opposed to having an optimiser for the entire continuous design space.

5. By introducing infeasibility and undesirability constraints allows areas in

the design space to be discarded. This ensures that the areas in the design space that are being explored are the desirable ones at each stage.

6. Not all design parameters will directly affect the optimisation objective. Therefore, the configurations that only differ in dormant parameters, that is, parameters that do not have an effect on the objective, will yield the same results. Identifying those configurations with shared active parameters will lead to a substantial reduction in the number of configurations that need to be optimised and avoid duplicate results.
7. With each configuration being able to be optimised independently allows for a concurrent (parallel) search for non dominated solutions in each area of the design space.
8. Employing a range of visualisation tools to offer appropriate methods for data interrogation, depending on the information required. Such tools act as mechanisms for design decision making; something that can be particularly useful when performing trade studies and assessing configurations.
9. The configurations that make it to the trade studies stage are ensured to fulfil certain predefined requirements, whether in terms of performance, robustness, or sensitivity to design changes and the costs associated with such changes.

From the aforementioned features, it becomes apparent that ADOPT enables the facilitation of optimisers in a fragmented design space. Only the desirable fragments will be explored in order to generate non-dominated configurations and only the areas that will yield unique solutions. In a conventional optimisation process the function needs to be evaluated first before being checked if it violates any constraints; If it does, this costs time and computational power unnecessarily. By completely discarding areas of the design space allows omitting undesirable areas entirely without first having to evaluate them. A visual representation of the transformed design space is presented in Figure 3.1 where **X** marks areas that are either undesirable or infeasible.

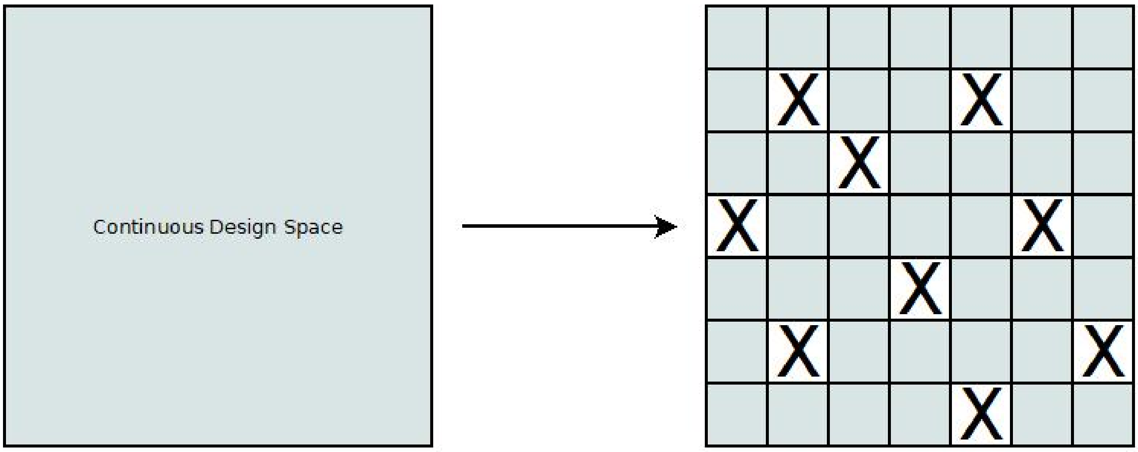


Figure 3.1: Design space transformation with ADOPT

Chapter 4

The Framework

4.1 Stage 1

Stage 1 deals with the decomposition of the system and the generation of configurations. It is assumed that the requirements have already been defined at a previous stage. The first stage is also concerned with converting the continuous parameters to discrete and handling the initial infeasibility and undesirability constraints, which leads to the transformation of the design space as previously described. Figure 4.1 shows the flow diagram of the first stage.

4.1.1 Stage 1 Overview

Since ADOPT deals with the abstraction of multiple system levels, the first step is to identify the levels, the disciplines or the areas that form the system. In this case the system is partitioned into two levels: the high level, which considers external parameters that affect the system but are not within the system itself, and the low level which includes the subsystems and components of the sys-

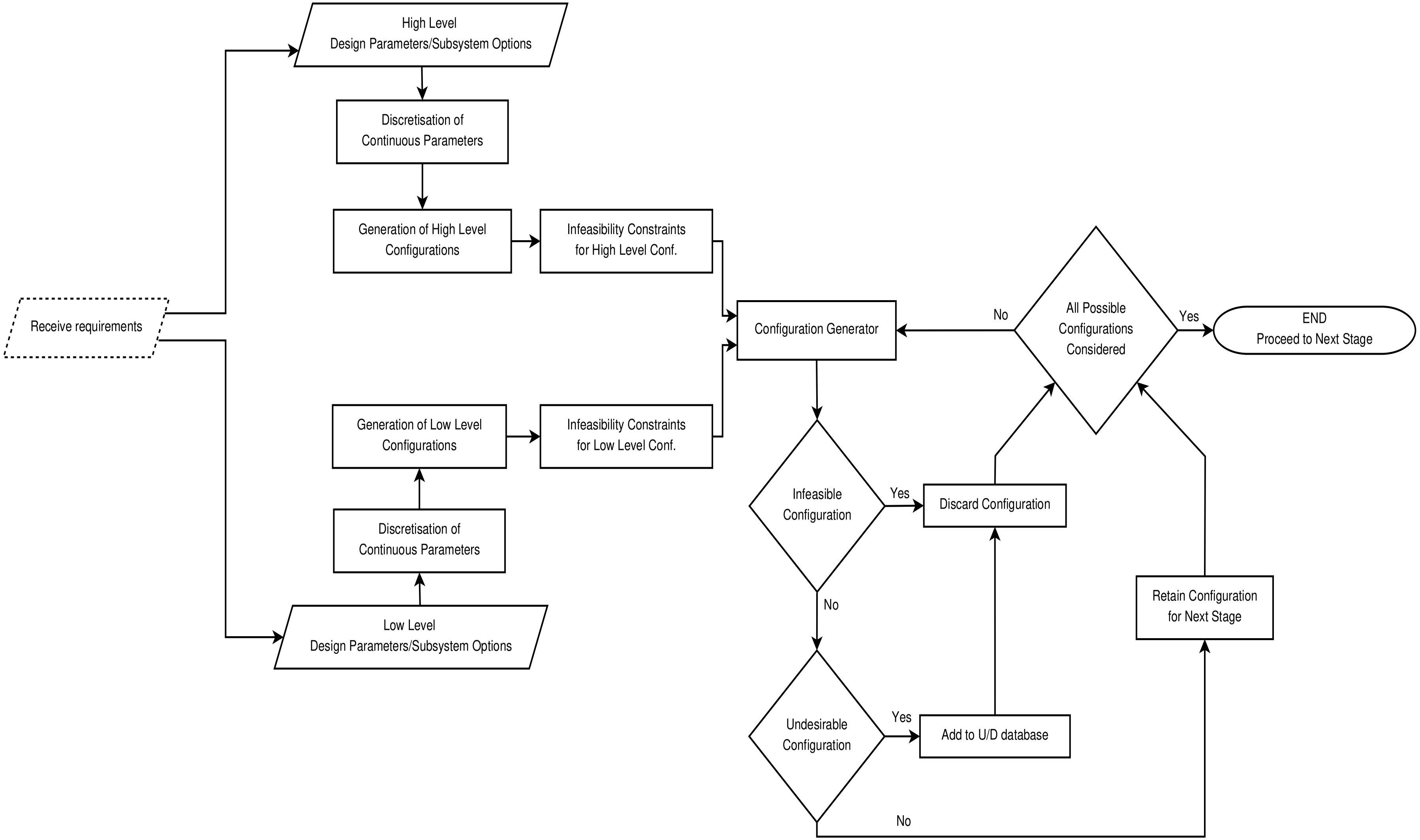


Figure 4.1: ADOPT Framework Stage 1

tem being designed. Additional intermediate levels of the system can be added where appropriate.

Following the process of the Set-Based Design, as presented in Figure 2.1, each level is further elaborated into its design parameters. Each design parameter has a number of possible options, be it discrete ones in the case of the low level subsystems and components, or continuous ones in the form of ranges in the case of high level dimensions. Before proceeding to generate the configurations, the continuous parameters need to be converted to discrete. This is done in order to generate a finite number of possible combinations amongst all options of each design parameter. Continuous parameters would not allow such combinations because an infinite number of values exists between any range of continuous values.

Completing the identification of the levels and their respective design parameters, and after the continuous parameters have been discretised, the process of generating the configurations begins. By taking one option for each design parameter the configurations are formed. Initially, the combination of options happens at each level independently; meaning that the combinations of high level parameters are independent from the low level ones. This is to allow better handling of the configurations and for performing infeasibility checks. After the infeasibility checks take place at each level and the infeasible configurations are discarded, the high level combinations are combined with the low level ones to form the full design configurations.

The combined configurations are then filtered through new infeasibility constraints before assessing them against undesirability constraints. The difference between the two is that the undesirable case studies are the ones that are currently not of interest, not feasible or not desirable due to limitations in technology or otherwise, but could become feasible in the future. For this reason the undesirable configurations are stored in a separate database to be re-evaluated at a future stage. The infeasible ones are the ones that are practically or otherwise impossible to implement and are discarded completely.

All the configurations that have passed the infeasibility and undesirability constraints are stored and proceed to Stage 2 of the framework for further evaluation. It is vital to be able to trace the characteristics and properties of each configuration, therefore a documentation system for record keeping of decisions and their rationale is mandatory. Such a tracing mechanism will not only allow to keep records of active configurations that are being considered, but also retain details with regards to undesirable, discarded ones that can be useful knowledge for future projects.

4.1.2 Discretisation of Continuous Parameters

In order to form a fragmented design space as described in the previous chapter, and shown diagrammatically in Figure 3.1, the continuous parameters of the system need to be discretised. This allows the formation of a finite number of regions in the design space; each of which refers to a different architecture / configuration.

The formation of smaller areas ensures that the optimisers can perform a more thorough exploration of each area. Whereas in a conventional optimisation in the continuous space, the risk of the optimiser being trapped in a local minimum would be high, that risk is substantially reduced due to the optimisers being forced to search all of the fragmented areas.

The most straightforward approach to discretising a continuous range is to split the range into smaller ranges and assign a linguistic term to each one such as “High”,“Medium”, or “Low”. Combining options afterwards becomes easier and straightforward; for example a “High” option for one design parameter with a “Medium” option for another design parameter. The linguistic term used for the variables, depends on the type of design parameter being described. A simple example of a range being discretised is shown in Figure 4.2.

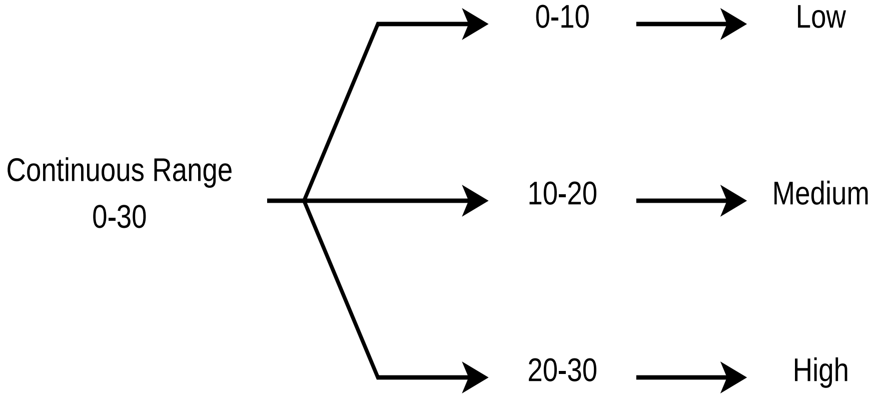


Figure 4.2: Example of discretising a continuous range

The discrete values will be converted back to their respective ranges in the second stage of the framework for optimisation purposes.

4.1.3 Infeasibility and Undesirability Constraints

Introducing infeasibility and undesirability constraints, allows the elimination and discardment of configurations before moving to the second stage. Discarding configurations, directly translates to rejecting areas in the fragmented design space. Allowing the optimisers in the second stage to skip areas in the design space, eliminates the unnecessary use of computational power and time that would have been used to evaluate configurations that violate constraints.

The first sets of constraints are applied at each level independently. This allows the domain or discipline-specific teams to remove any configurations that cannot be designed or produced. This could be due to for example, physical restrictions, manufacturing limitations, performance requirements, or technological readiness. At this stage only infeasible constraints are considered, since any undesirable configurations (at the domain/discipline level) may still be the best option overall for the whole system.

After the initial configurations at each level have been discarded, the full system architecture combinations take place by following the process described previously. At this stage, a new set of infeasibility constraints are introduced that deal with the architecture as a whole. The reasons for discarding configurations at this stage would typically be due to incompatibility amongst configurations from each level, domain, discipline, or subsystem.

Having discarded the infeasible configurations, a new set of constraints is introduced; the undesirability constraints. Undesirability constraints aim to eliminate those configurations that provide no value in studying them further, either due to technological limitations, previously evaluated configurations, or any other reason that deems them as not worthy of further assessment. However, configurations that have been eliminated due to undesirability are not entirely discarded. Instead, they are made “inactive”, which means they do not get passed on to the second stage of the framework, and get stored on a separate database along with the reasoning behind their discardment. This is useful for future projects either to avoid already visited configurations, or to re-evaluate them when sufficient technological advancement has been achieved.

The inclusion of constraints this early is very important, due to the large number of configurations the first stage of the framework can generate. The number of possible configurations can be in the thousands even for simple systems. Discarding a portion of those configurations can make the second stage of the framework faster, and enable the allocation of people and resources to feasible and desirable configurations.

4.2 Stage 2

Stage 2 deals with the evaluation and assessment of the configurations that have been generated in Stage 1 and have passed the initial infeasibility and undesirability constraints. The second stage follows an iterative and converging process, where at each iteration the undesirable configurations are eliminated, hence progressively narrowing the design space. The process of the second stage of ADOPT is presented in Figure 4.3.

4.2.1 Stage 2 Overview

The first step of the second stage is to transform the discretised parameters of each configuration, back to their respective continuous ranges. The configurations are then passed through to the iterative stage of the framework where each one is optimised against a set of predefined objectives. The ranges of the continuous parameters become optimisation constraints for each configuration in order to obtain the optimum, or non-dominated, values for each parameter.

However, in many cases, not every design parameter will affect the optimisation objective(s). Therefore, two competing configurations that only differ in a design parameter that does not affect the objective (a “Dormant” parameter), will yield the same optimisation results. Identifying the “Active” and “Dormant” parameters for optimisation is crucial as it can vastly reduce the optimisation

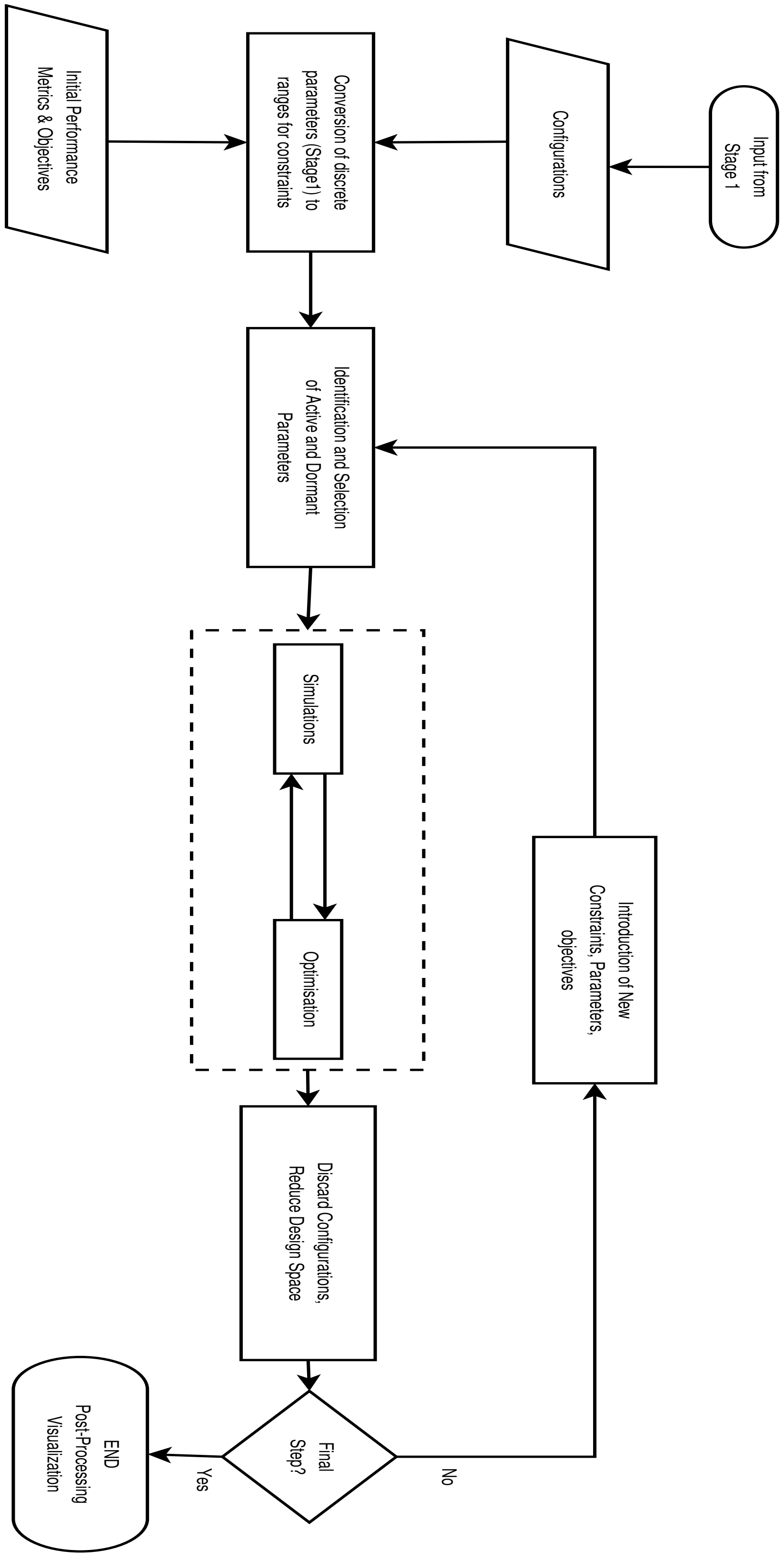


Figure 4.3: ADOPT Framework Stage 2

time due to eliminating duplicate optimisation runs.

In the case of multiobjective optimisation a number of non-dominated solutions will emerge for each configuration, thus expanding the design space around each one, but also providing alternatives for each configuration. In the case where a selected configuration has to undergo design changes at a later stage due to a change in requirements, there are alternative solutions to select from. This can reduce or even eliminate the need for major redesigns and subsequently the cost of the rework.

If necessary, further simulations can take place for each configuration, to obtain performance metrics for the value drivers at each iteration. In addition, the configurations can be assessed for sensitivity to design changes,how costly a design change propagation could be, and evaluated using different penalty weights (cost, weight, complexity etc.).

At the end of each iteration the optimised configurations are evaluated through the results obtained from the trade studies and assessment stage. Parallel coordinates are employed to help visualise the trade-offs between the configurations and the performance of each one, because there will be a large number of configurations, objectives, and performance metrics, .

Whereas optimisation methods aim at finding the optimum and non dominated solutions, discarding the rest, in Set-Based Design the approach is the other way around. The least desirable or least robust ones are identified and discarded. This leads to a lot of non-optimum solutions being retained for later stages, but robust enough to be used as back-up options.

After the set of feasible and desirable configurations has been narrowed down, the computational power can be transferred to the remaining active configurations where it can now be utilised for a more detailed evaluation of them. This allows for new performance metrics to be introduced and more detailed simulations to take place.

The iterative step is performed over and over again until the desirable number of configurations have remained, which by this stage will have been matured enough and evaluated thoroughly against the other options. The process of the second stage ensures that the final configurations are robust, optimised, and have available alternatives in case of a change in requirements. Feasibility of alternative designs, in the case of requirements change, can be assessed using interactive parallel coordinates by alternating the bounds of the affected design parameters.

4.2.2 Conversion of Discrete Parameters

Before optimising and evaluating the configurations, their discretised design parameters need to be transformed back to continuous ranges. Using Figure 4.2 as an example, the discrete values can be reverted back to their ranges in a straightforward way; a ”High” option will revert to a range of 20-30 and so on. The ranges of continuous parameters now become the constraints for the optimisation process.

The ranges of each parameter become the upper and lower boundaries of each design parameter for each configuration. This creates a more confined space, in which the optimiser seeks non-dominated solutions.

Having such confined design spaces for the optimisers to run in, along with the complete elimination of areas in the design space due to undesirability or infeasibility will vastly benefit the process. It will enable the optimisation process to be more efficient and enable a thorough exploration of the desirable areas in the design space.

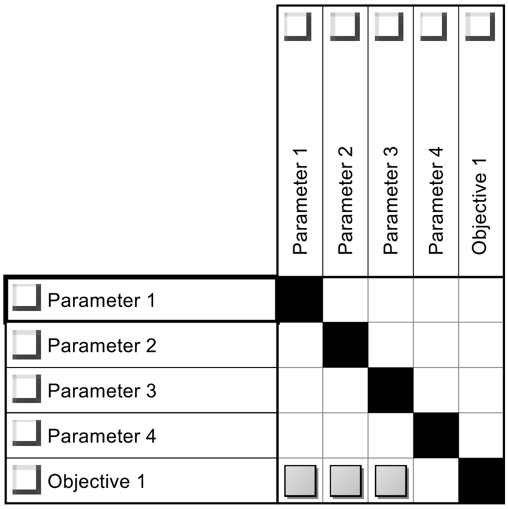
4.2.3 Identification of active and dormant parameters

The last step before moving to the optimisation and simulation process, is the identification of which design parameters are "Active" and which are "Dormant". An active design parameter is one that affects the optimisation objective while dormant ones are the ones which do not have any effect. Objectives can change at each iteration of the second stage and consequently so can the active and dormant parameters.

Identifying which parameters are active and which are dormant, allows for duplicate configurations to skip the optimisation process. Duplicate is with regards to the optimisation results and not with the synthesis of the configurations. The optimisation process is usually the most computationally expensive process; therefore, reducing the number of configurations that are optimised will considerably reduce the time required for the stage to be completed. Configurations that will yield the same results are placed in optimisation clusters.

The identification of the parameters, and consequently the selection of configurations, can be accomplished in numerous ways. A simple and straightforward approach would be to use a DSM to visualise the connection between design parameters and optimisation objectives. Configurations that differ only in dormant parameters options will yield the same optimisation results since they share the same active parameter options. Therefore only one of those configurations need to go through the optimisation process and have the results shared with the duplicate configurations.

For example, Figure 4.4 presents a simple example of a system with four de-



	☐ Parameter 1	☐ Parameter 2	☐ Parameter 3	☐ Parameter 4	☐ Objective 1
☐ Parameter 1	■				
☐ Parameter 2		■			
☐ Parameter 3			■		
☐ Parameter 4				■	
☐ Objective 1	▨	▨	▨		■

Figure 4.4: Example of a DSM for identifying active and dormant parameters

sign parameters and one optimisation objective. At this stage, the interdependencies between the design parameters are not being considered, instead only the parameters that directly affect the objective are considered. It becomes immediately apparent that only the first three design parameters are affecting the optimisation objective and not the 4th one. This makes the first three parameters Active and the last one Dormant.

This means that configurations that share the same options for the first three design parameters will belong in the same optimisation cluster; since they will yield the same optimisation results. Any variation in the 4th parameter will not alter the outcome of the optimisation process. It is worth noting however that even though the 4th parameter is dormant for this specific objective, it might affect the system in other ways, such as performance or optimisation objectives in future iterations.

Assuming no infeasibility or undesirability constraints have been set, if the 4th design parameter has even just two options, using this approach would mean that only half the total possible combinations would need to be identified. In realistic cases where more parameters are dormant, each with more options, using this approach can reduce the number of optimisation cases from the order of thousands to the order of tens.

4.2.4 Optimisation

Following the process described in the previous section, the number of cases that need to be optimised will have been reduced considerably. The combination of imposing the constraints and identifying the active parameters, allows the optimisers to focus only in regions that are, not only desirable and feasible, but that will also yield unique results.

The optimiser operates within the confined regions that are bound by the ranges that have been converted from the discrete values. In the case of Multiobjective MDO, this approach can generate multiple non-dominated design alternatives for each configuration, adding further flexibility and reducing the cost of potential design changes that might occur later in the design lifecycle. This is due to, not only having multiple configurations available at each stage, but also different design variations of each configuration.

At each iteration of the second stage of the framework, new objectives can be introduced along with new performance metrics and value drivers. The main factors that need to be taken into consideration for any optimisation process, is the selection of the right search algorithm and the right architecture. However, since the second stage is iterative, with the optimisation process having different objectives, parameters and constraints each time, the optimiser and the architecture needs to be adaptable. With increasing cost and complexity, especially of the evaluation functions, the optimiser and the MDO architecture needs to be able to change at each iteration.

After the results have been obtained for the unique optimisation configuration, they can be shared with the remaining active configurations within the cluster. The results can be plotted using parallel coordinates, which will help visualise trade-offs between the configurations. Even though the optimisation results will be the same for configurations that belong in the same cluster, the configurations are expected to yield different results in other areas;for example, by introducing additional value drivers and performance metrics.

4.3 Evaluation and Assessment of Configurations

The optimisation process will generate information that can be used for trade studies. It can identify where a configuration is performing better than others as well as the trade-offs between them. However, the optimisation results might not be sufficient to properly assess a configuration on its own. Therefore, a number of additional performance metrics and value drivers are introduced to help differentiate among the different configurations.

4.3.1 Additional Value Drivers

The most important factor that needs to be addressed when assessing and evaluating a configuration, is whether it meets the expected performance requirements. Simulation models can assist in determining just that, as well as how it performs with regards to other performance metrics under different states and scenarios. Simulation models can also identify the flexibility and sensitivity in design changes that the configuration can withstand without violating any constraints.

Using the CPM process as described by Clarkson [83], the sensitivity of a configuration to engineering changes can be assessed; as well as how those changes can propagate through the system. This can be a useful tool to identify high risk components or subsystems that are prone to design changes, or elements that need to have their designs frozen in order to avoid any potential changes later on. One of the drawbacks of the CPM approach however is that it does not consider the cost associated with the propagation of a design change. To address that, a new method was developed and is presented in the next section.

A useful measurement of performance is the use of penalty weights for each option. Depending on the system in consideration, each option can have penalties with regards to complexity, cost, or weight. Penalties are usually assigned on a range from 1-10, with 10 being the highest, and are primarily knowledge-based. Adding up the penalties for the selected options of each subsystem that make up the configuration, will give a metric of how costly (for example) that configuration would be compared to others. Such an approach will benefit the early stages of design decision-making for use in trade studies where a fast approach is required due to the large number of configurations being considered. During later stages, these metrics can be replaced by using higher fidelity methods, such as more computationally expensive simulations.

4.3.2 Cost Propagation due to Engineering Changes

Besides using the CPM approach to assess the sensitivity and impact of design changes, it is also critical to be able to measure the costs associated with the changes. The proposed change prediction method, CP^2[111], shares some characteristics with the CPM method described in Section 2.3.2. CP^2 uses a DSM to quantify the probability of propagation (i.e. the Likelihood) but the impact measurement is removed in this case. A new measurement, cost weight, is introduced, which is knowledge-based, as is the case with probability of propagation. The obvious difference between impact and cost is that the former measures the design impact on the other component; the latter measures how costly a design change of the affected component will be. This means that while impact is defined at the connection between two components, using a DSM (same way as the probability), the cost is a property of each component independently. It is important to notice however that cost does not necessarily mean directly a financial one and it only represents a relative cost weight, hence, it is expressed as a value in the range between 0 and 1. Figure 4.5 shows an example of a critical path network between two components in a system. Values above the paths indicate the probability of propagation between two components, while values above the components itself, indicate the respective cost weight of each component.

Probability of propagation

Calculation of probability of propagation between two components (considering direct and indirect change) is straightforward; simply by using probability rules for independent events. As with most propagation methods , indepen-

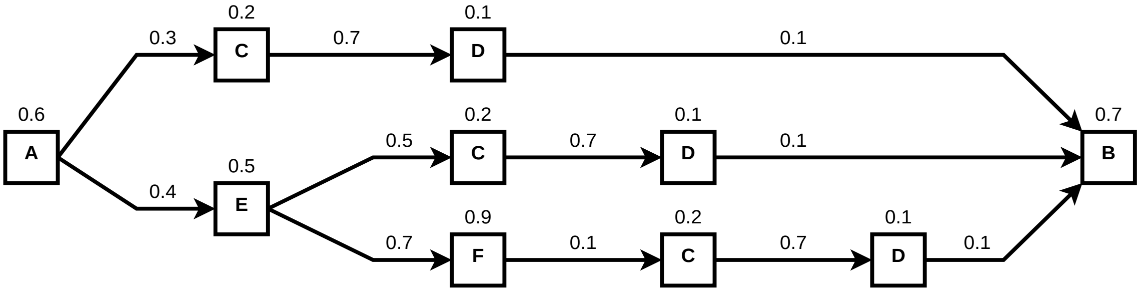


Figure 4.5: Example of a critical path network with path probabilities and component cost weights

dence between direct probabilities is assumed. For each critical path between components, the path probability (pp_n) is the product of all the direct probabilities (dp_{ij}) between components. In the formulation, i and j represent the components with n being the path number:

$$pp_n = \prod dp_{ij} \tag{4.1}$$

To get the combined probability Cp_{ij} of all paths from the initiating component to the receiving:

$$Cp_{ij} = 1 - \prod (1 - pp_n) \tag{4.2}$$

Cost Weights and Aggregated Cost

As mentioned before, each component is assigned a relative cost weight. The cost weight is a value between 0 and 1 and represents how costly a design change of that component will be, with 1 being the most costly. For each critical path, assuming that the design change reaches the receiving component, the path cost pc_n is the sum of all cost weights cw_i, including the weights of the initiating and receiving component:

$$pc_n = \sum cw_i \tag{4.3}$$

To aggregate the costs of all paths between two components, each path probability is multiplied by the respective path cost and then all paths are summed. The sum of all paths is then divided by the combined probability to get the aggregate cost Ac_{ij}:

$$Ac_{ij} = \frac{\sum (pp_n \times pc_n)}{Cp_{ij}} \tag{4.4}$$

As expected the aggregate cost of a connection with only one path is the same as the total cost of the path since the path probability is equal to the combined probability.

Aggregate costs allow the user to identify which elements in a system are more likely to occur higher costs than others; when a change originates from them. This is due to considering the probabilities and costs of all paths in a single metric.

DSMs can be generated from there on for combined probabilities between all components, and aggregate costs using the respective equations. As opposed to the traditional CPM approach though, where a combined impact is calculated along with risk, an aggregated cost is not as useful on its own for decision making in the method presented here. This is due to each path having a different cost and different probability of it occurring, and combining those values leads to loss of important information. It is therefore a lot more useful to be able to see all possible path costs with their respective probability values. This will lead to a more detailed and thorough assessment of the system.

In order to be able to visualize all possible propagation paths between components when all the data have been calculated and obtained, a multidimensional data visualization tool is mandatory. For this reason, using a DSM in this case is not possible. Therefore, parallel coordinates are employed to help visualize and analyse the paths. The parallel axis will provide the initiating and receiving components, the combined probability between components, the aggregated costs, as well as the individual path probabilities, and the path costs. Additional axes can be added to show the intermediate components in each path.

The two quantified DSMs with combined likelihoods and aggregated costs can still be used if necessary. Further DSMs can be produced depending on the needs and data that are required to be visualized such as the worst-case scenario (highest path cost), and the most probable scenario (highest path probability).

4.3.3 Visualisation

Through the previous chapters where the process of ADOPT has been outlined, it becomes apparent that a range of visualisation methods are employed, depending on the stage and the type of information that is considered. DSMs are used in multiple instances, from network modelling purposes to outline the connections between the elements of the system in consideration, to design dependencies for predicting change propagation. They can also be used to identify the active and dormant parameters to improve the optimisation process, and present the risks, impacts and cost weight that are associated with change propagation between pairs of elements.

With regards to change propagation, one of the most useful tools to visualise propagation paths, are critical path networks. They offer the ability to visualise the different possible paths that originate from a design change at one component, propagating through intermediate components, until they reach the final affected element.

One of the most useful visualisation tools however is parallel coordinates. A scatter plot is useful when visualising 2-dimensional or even 3-dimensional design spaces, but cannot visualise further dimensions. Parallel coordinates can overcome this limitation and be employed to not only demonstrate the tradeoffs between objectives in a multiobjective optimisation problem, but also to visualise the associated design parameters for each configuration. In the case of costs associated with change propagation, parallel coordinates can be used to visualise the probabilities and associated cost of the different paths of propagation between any two components of the system.

All of the above methods aim to act as tools that will assist the design decision-making process. Each one has different use depending on the type of data that is being visualised, the design stage, but also on the person who requires the information to make decisions.

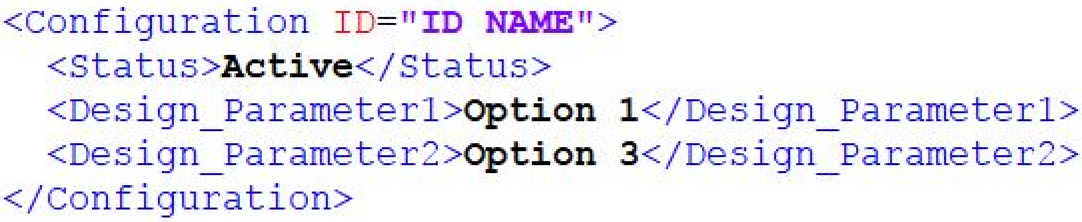
4.4 Traceability

The proposed approach with the ADOPT framework is ensured to generate large amounts of data due to multiple configurations. It is essential to be able to track the configurations along with their properties throughout the design process. A method for tracing and documenting the configurations along with their status (active/inactive), and their performance characteristics, is vital in order to be able to manage the large amount of information generated.

A widely used method that can manage something of this scale, besides the use of conventional databases, is using eXtensible Markup Language (XML) files.

The main benefit of using XML files is that they are both machine and human readable and can be platform and interface agnostic. Further, regulated industries like aerospace, are required to archive and retain such information over decades; therefore the information storage mechanism has to be in a neutral format, like XML.

In the case of the ADOPT framework, the configurations are stored as elements with an identification for each one (Configuration ID). The design parameters are listed as subelements in each configuration and the selected option of each design parameters is assigned as an attribute. Using this approach, the status of each configuration can also be traced. Figure 4.6 shows how a configuration with two design parameters would appear in an XML file.



```
<Configuration ID="ID NAME">
  <Status>Active</Status>
  <Design_Parameter1>Option 1</Design_Parameter1>
  <Design_Parameter2>Option 3</Design_Parameter2>
</Configuration>
```

Figure 4.6: Example of XML syntax for ADOPT

Other performance metrics can be recorded on the XML entry of each configuration. However large datasets such as optimisation results would yield a very large number of data points which could prove difficult to manage and read in an XML file. Therefore for large datasets, comma-separated values (.CSV) files are utilised, and can be linked to the respective entry in the XML file. Future implementations may use other emerging data standards, like Hierarchical Data Format (HDF), for storing and organising large amounts of simulation data.

Part III

Case Study

Chapter 5

Aircraft Fuel Systems

In order to demonstrate the process and benefits of ADOPT, the design of an aircraft fuel system is used as a case study. The main function of the fuel system in an aircraft, as defined by EASA, is to ensure a fuel flow at the correct rate and pressure to the engines under all probable operating conditions [112]. It consists of a complex network of pipes, pumps, connectors, sensors and valves (see Figure 5.1). As with any aircraft system, safety is a priority, especially in this case, where flammable fuel is being considered. In order to ensure the reliability and safe operation of the system, there are a number of things that need to be taken into consideration, such as the flammability and pressurization of the tanks. Provisions must also be made in order to minimise the effects of abnormal events, like an uncontained engine rotor failure (UERF).

The system can be broken down in several subsystems, each with it's own function. The geometric characteristics of the wing will define the available volume in the wingbox, which will then define the shape and size of the tanks. The tank location and size is then used to design the subsystems, while taking into consideration the higher level requirements such as the aircraft's intended mission. This chapter aims to present the different subsystems present in a typical aircraft fuel system and the design drivers behind each of the subsystems.

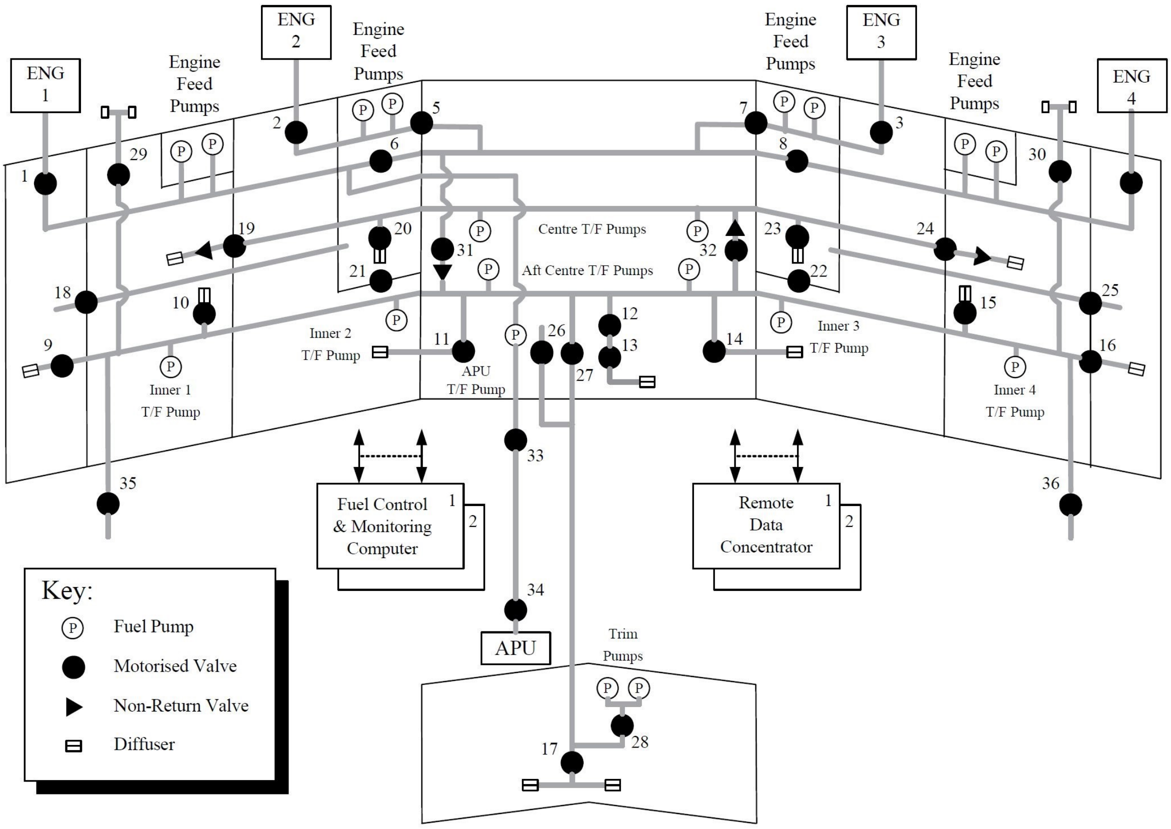


Figure 5.1: Airbus A340 Fuel System Schematic [113]

5.1 Storage

A typical Airbus A350-900 can carry up to 140,000L (110,523 Kg) of fuel, which given the aircraft's MTOW (280,000Kg) means that the fuel quantity can account for almost 40% of the total weight. Depending on the intended mission of the aircraft, the appropriate provisions need to be made in order to provide sufficient space for fuel storage, for the given range and passenger / cargo capacity.

The most common location for storing fuel, is within tanks located in the wings. Additional fuel tanks can be present in the fuselage (Center Tank and Additional Center Tanks - ACT) or even in the horizontal tail plane (Trim Tanks). Each engine has its own feed tank, which is usually a cell within the wing or center tanks. The purpose of the cell is to ensure that fuel is always immediately available to the engines. A vent box (sometimes referred to as a surge tank) is located on the edge of the wings. Its purpose is to collect any fuel that might get into the vent lines and transfer it back to the main fuel tanks. Figure 5.2 presents the location and arrangement of the aforementioned tanks on an Airbus A380.

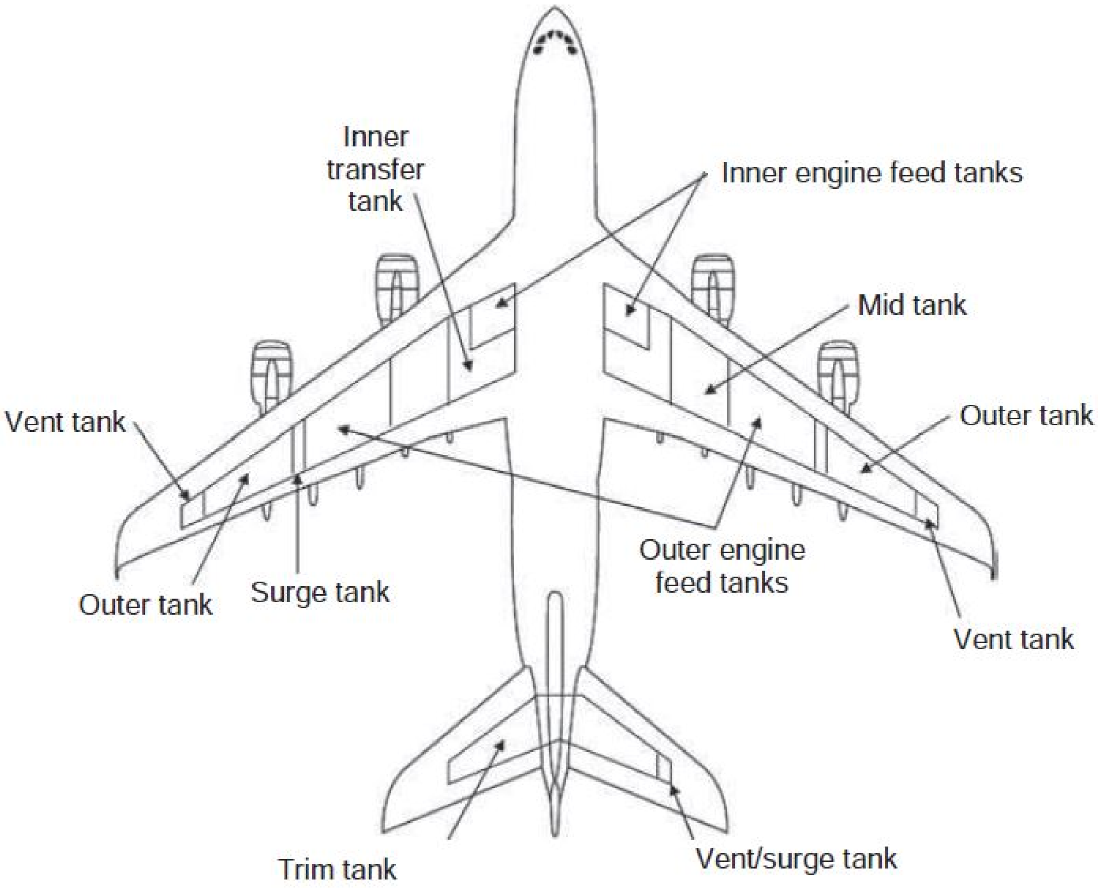


Figure 5.2: Airbus A380 Fuel Tank Arrangement [113]

When considering the locations of the tanks one major safety factor that needs to be taken into consideration is the case of an Uncontained Engine Rotor Failure (UERF). Such an event can produce debris that can have the energy to penetrate the aircraft structure, which can lead to fuel loss. It is therefore essential to include dry areas where fuel can't be stored as shown in Figure 5.3.

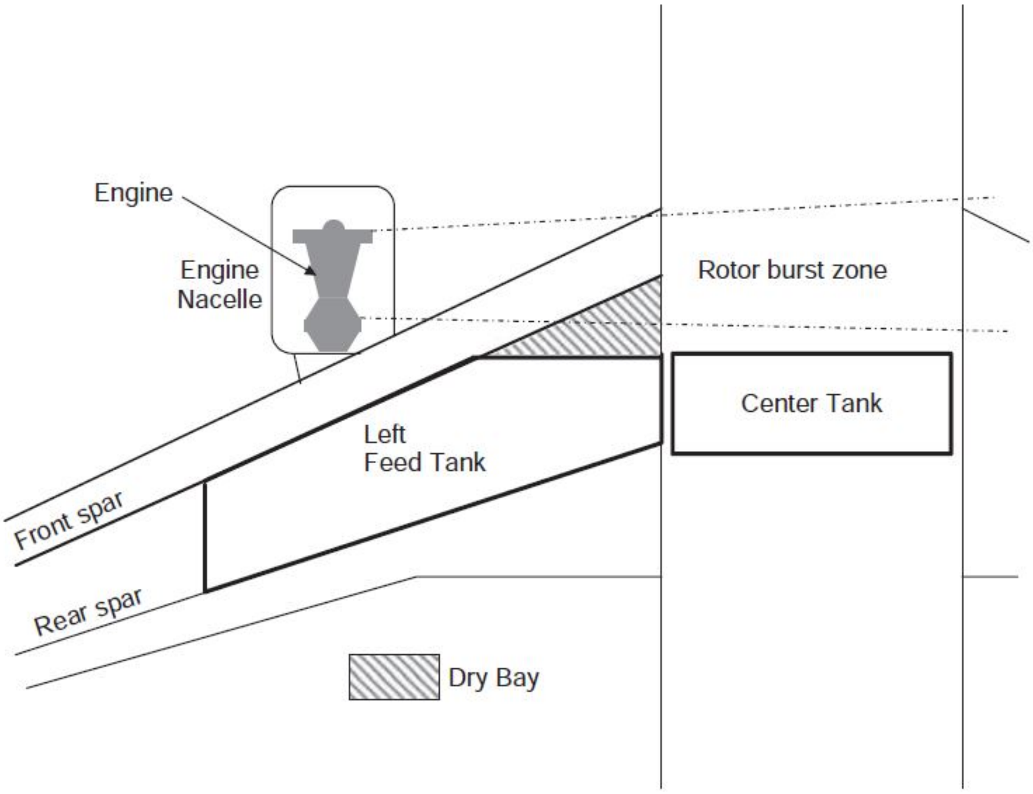


Figure 5.3: Example of an Uncontained Engine Rotor Failure [113]

Due to the thin structure of wing, storing and gauging fuel is a challenge. Few research papers have presented work on multidisciplinary design optimization of the wing with consideration of fuel storage, fuel systems, and provisions for a UERF [114].

5.1.1 Fuel management and Burning Sequence

In order for the collector cells of the engine feed tanks to be constantly full (or close to), fuel needs to be transferred from other tanks. Due to the large volume of fuel present in an aircraft, it can significantly affect the Centre of Gravity (CoG) of the aircraft. As fuel is being consumed the CoG shifts along the longitudinal axis. Lateral changes are minimal unless in the case where fuel can't be transferred symmetrically from one of the wing tanks. In the case where a trim tank is also present, the longitudinal shift of CoG increases, therefore it becomes even more important to manage the refuelling and burning sequences.

An onboard fuel management computer manages the transferring of fuel to the feed tanks for the engines, but also between the trim tank and the centre-wing tanks for controlling the CoG. In the case of an engine failure, it can manage the cross-feed fuel from one tank to another.

5.1.2 Wing Load Alleviation

For aircraft with aluminium wings one important aspect to consider is the structure fatigue due to wing loads. Fuel can be used as a loads alleviation mechanism by keeping fuel in the outer wing tanks during flight to act as a balancing force to the lift. This is usually the case for long-range aircraft where the wing tanks are split into a number of cells. It is important, where possible, to keep the fuel in the inner tanks while on ground to avoid structure stress due to the weight and then transfer them to the outer tanks after take-off.

This is where the fuel burning sequence gets even more complicated. As mentioned in the previous section, the feed tanks need to be always close to full ($> 95\%$) and this can be accomplished by transferring fuel from the other tanks onboard. In such cases, the most common sequence is moving outwards, meaning that consumption starts from the centre tank, then from the inner wing tanks and finally from the outer tanks. Presence of a trim tank only adds to the complexity since there are considerations that need to be made due to CoG shifts as mentioned previously.

In the case of carbon fibre composite wings, the fatigue issue is removed but only to be replaced by other issues such as protection of electronics from electro-magnetic and electro-static phenomena.

5.2 Fuel Subsystems

5.2.1 Transfer System

As the name suggests, the purpose of the Transfer System is to transfer fuel between the tanks either by using gravity or pumps. In the case where fuel is transferred from the outer wing tanks to the inner ones and the engine feed tank (considering a dihedral wing case), this is done using gravity due to tank cascading. In the case of transferring fuel from the centre tank to the wing tanks, or from the trim tank, this is done using pumps.

The Transfer System also enables the fuel burning schedule to ensure fuel availability in the feed tanks, manage the shifts in CoG and transfer fuel to and from the outer wing tanks for load alleviation. In order to minimise the complexity and weight of the pipe network, in some cases the transfer system shares most of its pipe network with other systems such as the refuelling and/or the jettison; this also assists in maintenance.

5.2.2 Refuel/Defuel System

For airlines to minimize turnaround times, quick refuelling is an important factor to be considered by the manufacturer. The most common method of refuelling a large transport aircraft is with pressure refuelling. The onboard Refuel System transfers the pumped fuel from the ground station to the onboard tanks.

The system must be able to handle the pressure of the refuelling station, while ensuring the right amount of fuel is transferred and stored at the right location onboard. Defuelling only occurs for maintenance purposes or in the rare case where the amount of fuel needs to be reduced before a flight.

5.2.3 Feed System

The Feed System is responsible for transferring fuel from the collector cells within the feed tanks to the engines and the Auxiliary Power Unit(APU). Collector cells tend to be as close to the engine as possible to avoid long fuel lines; therefore ensuring the immediate availability of fuel to the engines.

It is an airworthiness requirement that each engine has it's own dedicated feed tank [112]. In the event of an engine failure a crossfeed system is in place which enables the operating engine(s) to consume fuel from the collector cell of the inoperable engine.

5.2.4 Venting System

Pressurization of the tanks due to climb, descend and refuelling, needs to be avoided, as it can create large forces on the wing structure and tank walls. The main function of the Venting System is to connect the ullage (empty space within a tank above fuel level) with the outside air to prevent large pressure differences.

The vent lines (pipes) start at the top of each tank, and end at the surge tanks located at the wing tips, which are connected to the outside air. The purpose of the surge tank is to collect any fuel that might have entered the vent lines, and return it to the main fuel tanks. If the surge tank overflows, fuel is dumped overboard.

When the system is being designed, the main driver is the consideration and ability for the system to handle maximum descend cases. In cases such as loss of cabin pressure where the aircraft needs to descend rapidly to a ”breathable” altitude, the venting system must be able to handle the large mass flow rate of air in order to avoid large pressure differences between the tank and the atmosphere.

5.2.5 Tank Inerting

One of the major safety concerns that needs to be addressed is the flammability of the ullage due to the combination of fuel vapours and oxygen concentration. The centre tank has a much higher risk of fuel vapour explosion since in most cases it has little to no fuel (the first tank to consume fuel from) creating a large, highly combustible, ullage. Due to it being located in the fuselage, it does not benefit from airstream cooling the same way that wing tanks do, thereby increasing the risk factor.

To reduce the oxygen concentration in the ullage (and hence the possibility of combustion), an inerting system must be present in the centre tank. The purpose of the inerting system is to inject Nitrogen Enriched Air (NEA) into the ullage that reduces the oxygen concentration. This is done by forcing oxygen out of the tank through the vent lines. Nitrogen is an inert gas, therefore not combustible. EASA defines a tank as inert when the oxygen concentration in the ullage is 12% or less [112].

The inerting system is usually operable during descent where atmospheric air is flowing in the tank. Injecting NEA inside the tank displaces the oxygen back out thereby keeping the ullage inert. Difficulties arise when considering a half-empty tank in the take-off stage where air is flowing out of the tank therefore larger NEA flowrates are required in these cases.

5.2.6 Jettison

For aircraft where the Maximum Landing Weight (MLW) is substantially less than the Maximum Take-Off Weight (MTOW), provisions need to be made for a fuel jettison system. In cases of emergency landing immediately after take-off (such as engine failure) the aircraft needs to dump fuel overboard as fast as possible, in order to meet the MLW and land safely. In worst case scenarios, where not enough fuel can be dumped in time, the aircraft can still land above the designed MLW but doing so, there is risk of landing gear failure and subsequent system failures. In such cases, this will lead the aircraft to be grounded until formal investigations have been carried out, and the aircraft is cleared to operate again.

Since the jettison system is not a primary subsystem, and is one that will rarely be used, design considerations need to be made to reduce the complexity and weight of it. For this reason, a number of aircraft that employ a jettison system, use the pipe network of other subsystems such as the Transfer system.

Chapter 6

Case Study Definition

6.1 Introduction

To demonstrate the process of the ADOPT framework, a computational model of it has been developed. The design of an aircraft fuel system is used as a case study to show how each step is performed.

This chapter describes in detail the problem formulation in terms of assumptions, parameters, constraints and objectives. The performance metrics and value drivers are also defined and a template for traceability is proposed.

The remaining sections of this chapter demonstrate how the framework is implemented computationally. An overview of the tools used for this case study is presented. However, the framework has been defined generically and even for this case study other tools can be used. The choice of the tools can be based on their suitability to the task and the experience of the user.

6.1.1 Tools and Software

Since ADOPT integrates a number of different approaches, methods and tools, various computational tools and software are employed to develop the framework computationally.

First and foremost, a workflow manager is vital for such a process in order to integrate, monitor, and manage the different steps of the process. pSeven suite, a software by DATADVANCE has been selected for this task. The suite is advertised as a design space exploration tool [115]. It offers a number of pre-built blocks that can be connected; the connections define the data flows between the blocks. pSeven offers a number of readily available blocks but only a number of them are used for this case study:

- Script: Custom python script as provided by the user. This can also be used for function evaluations in an optimisation loop.
- Program: A configurable block to call external programs or scripts.
- Optimizer: The optimisation block defines and solve the optimisation problem. It uses SmartSelection to select the most appropriate optimisation approach depending on the complexity of the problem and the computational cost of the evaluation function. This is covered in more detail later.
- Map: Traverses an input list and outputs the elements one by one. Mapping blocks are also used to control a processing loop.
- CSVGenerator: Creates a CSV file from an input matrix.
- CSVParser: Receives data from a CSV and outputs a matrix.

The implementation of the framework using pSeven, along with the respective process flows, is covered in detail in a following chapter.

Cambridge Advanced Modeller (CAM) is a software tool for modelling and analysing the dependencies and flows in complex systems [116]. In this case, CAM is used to generate the DSMs of the systems as well as to calculate the propagation of a change in the system using the embedded CPM algorithm. A custom script developed, will also calculate the cost associated with the change using the previously described method CP ^{2}. CAM is also employed to generate the critical path networks.

In order to validate the technical performance of the configurations a software for simulating the fuel system is required. LMS AMESim by Siemens, provides an integrated 1-D simulation platform, including a library with predefined models specifically for aircraft fuel systems [117].

Finally for visualisation purposes, High-D by Macrofocus and XDAT were used [118, 119]. The software offer interactive parallel coordinates visualisation with clustering and filtering, and can handle large datasets.

6.1.2 Assumptions

A number of assumptions are made for the case study. The assumptions that are concerned with the geometry of the wing are obtained from the NASA technical memorandum that was used for the wingbox volume calculation [120]. The assumptions concerned with the fuel systems and the tank sizing are obtained from expert judgement. All assumptions are necessary in order to calculate the optimisation objectives and the fuel system performance.

- The wings are untwisted, tapered and swept with straight leading and trailing edges.

- The resulting shape of the wing will be trapezoidal with the root and tip chord being parallel.
- A part of the chord, both at leading and trailing edges, at any spanwise location is used for high lift devices, hence not considered for the wingbox volume calculations.
- The upper and lower surfaces of the wing (and the wingbox) are assumed to be flat.
- The structural semispan of the wing is assumed to lie on the quarter-chord line.
- Wing tank cells (where more than one present) are equal in volume.
- The volume taken up by the internal structure is not considered.
- The width of the carrythrough structure equals the diameter of the fuselage.
- The length of the vent pipeline on each wing is equal to the structural semispan of the wing.
- The radius of the venting pipe is 0.05m.
- A trim tank is only considered in terms of weight and complexity (no performance calculations).

6.2 Design Parameters

6.2.1 Aircraft Level Parameters

The high-level system considered here is concerned with the geometry, material, and engines of the aircraft. For continuous parameters, the ranges are knowledge-based and obtained from historical data [120, 121]. Table 6.1 presents the design parameters of the aircraft level.

Parameter	Type	Options
Aspect Ratio	Continuous	7 – 10
Wing Span	Continuous	28 – 80
Sweep	Continuous	24.5 – 37.5
Fuselage Width/Diameter	Continuous	3.60 – 7.05
Taper Ratio	Continuous	0.16 – 0.33
Thickness/Cord Ratio (root)	Continuous	0.12 – 0.14
Thickness/Cord Ratio (tip)	Continuous	0.07 – 0.11
Wing Material	Discrete	Aluminium , Composite
Engines	Discrete	2 , 4

Table 6.1: Aircraft level design parameters

Out of the 7 continuous parameters, three of them (taper ratio and thickness/cord ratios) will not be discretised in order to generate the configurations. Instead, all the configurations will share the same ranges for those three parameters and consequently will have the same bounds for optimisation. This is due to their small range and small effect on the result, compared to the rest.

6.2.2 System Level Parameters

As opposed to the aircraft level that comprised of both continuous and discrete parameters, the fuel system level is entirely comprised of discrete parameters.

The parameters are primarily with regards to the fuel tanks configuration and subsystem options. Table 6.2 shows the design parameters of the level with their respective options.

Parameter	Type	Options
Wing Tank Cells	Discrete	1(Single),2,3,4
Centre Tank	Discrete	Yes, No
ACT	Discrete	0,1,2
Trim Tank	Discrete	Yes, No
Inerting System	Discrete	Yes, No
Transfer System	Discrete	Pressure, Gravity, Both
Jettison System	Discrete	Yes,No

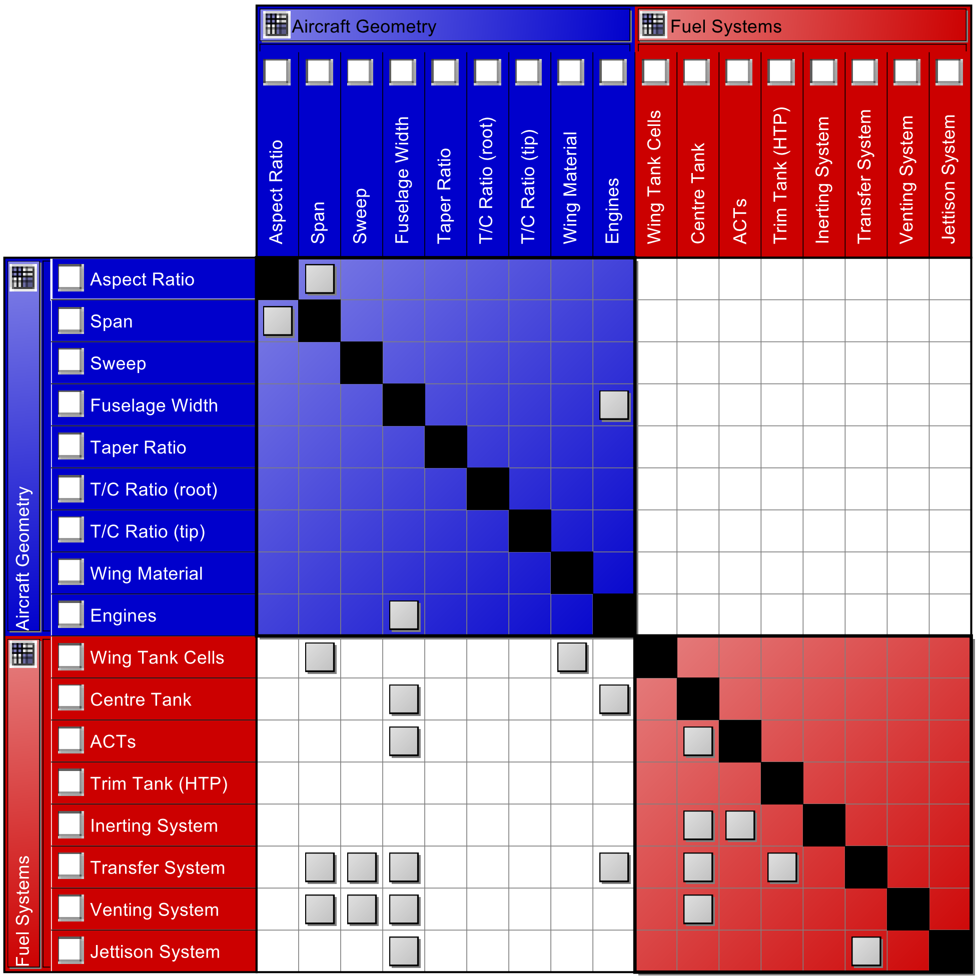
Table 6.2: Fuel system level design parameters

The venting system is a mandatory subsystem but has no options, which is the reason why it is not included in the table. It is also a subsystem that is defined and sized according to the rest of the design parameters.

6.2.3 Matrix view of the System

To create a model of the system network, the connections and design dependencies need to be visualised within and between the different levels of the system. Matrix methods can be employed to achieve that. Figure 6.1 presents a DSM representation of the overall system and the interconnections between the design parameters.

It is important to notice that some of the connections represent design dependencies while others represent constraints that involve the 2 connected elements. For example the connections with the wing tank cells element, are both design dependencies, since the number of tank cells are dependent on the length of the span and the material of the wing. On the other hand, the mirrored



		Aircraft Geometry									Fuel Systems							
		Aspect Ratio	Span	Sweep	Fuselage Width	Taper Ratio	T/C Ratio (root)	T/C Ratio (tip)	Wing Material	Engines	Wing Tank Cells	Centre Tank	ACTs	Trim Tank (HTP)	Inerting System	Transfer System	Venting System	Jettison System
Aircraft Geometry	Aspect Ratio	■	☐															
	Span	☐	■															
	Sweep			■														
	Fuselage Width				■					☐								
	Taper Ratio					■												
	T/C Ratio (root)						■											
	T/C Ratio (tip)							■										
	Wing Material								■									
	Engines				☐					■								
Fuel Systems	Wing Tank Cells		☐						☐		■							
	Centre Tank				☐					☐		■						
	ACTs				☐							☐	■					
	Trim Tank (HTP)													■				
	Inerting System											☐	☐		■			
	Transfer System		☐	☐	☐					☐		☐		☐		■		
	Venting System		☐	☐	☐							☐					■	
	Jettison System				☐											☐		■

Figure 6.1: DSM view of the system

connections between span and aspect ratio, represent a constraint between the two elements. The constraints considered for this case study are further explained at a later section.

The DSM can assist in identifying elements that have a large design impact on the rest of the system, but can also identify elements that are highly dependent on others. This can assist in the case of sequential design, where for example, the design of the transfer system is affected by 6 other elements, but only has an impact on the design of the jettison system since they usually share the same pipelines.

6.3 Generation of Configurations

After the system has been elaborated down to it's design parameters and elements, and the options for each one have been identified, the generation of configurations begins. This is the stage where one option from each parameter is selected and combined to form a unique system architecture. By selecting different options for each design parameter leads to the generation of multiple unique system architectures / configurations.

In large complex systems, where a large number of design parameters and options are present, the combination of all available options can lead to an unmanageable amount of combinations. This makes it necessary to set some restrictions, in the form of constraints, to avoid exploring areas that will lead to undesirable or infeasible configurations.

6.3.1 Discretisation of Continuous Parameters

In order to generate a finite number of all possible configurations, all the continuous parameters need to be discretised into ranges and assign a linguistic term to each one. The number of ranges that each continuous parameter is split into, depends on a number of factors including the type of parameter and the size of the range. Table 6.3 presents how the continuous parameters are discretised for the current case study.

Parameter	Discrete Options	LB	UB
Aspect Ratio	Low	7	8.5
	High	8.5	10
Wing Span	Short	28	45
	Medium	45	63
	Long	63	80
Sweep	Low	24.5	31
	High	31	37.5
Fuselage Width/Diameter	Narrow Body	3.6	4.9
	Wide Body	4.9	7.05

Table 6.3: Discretisation of Continuous Parameters (LB: Lower Bound; UB: Upper Bound)

6.3.2 Stage 1 Constraints

The narrowing down of options can be done systematically in two ways:

- Generate one configuration at a time and check against all constraints each time.
- Generate all possible configurations, and eliminate the ones that violate the constraints.

For this case study, the second option was chosen to avoid the excessive number of loops of generating and checking each configuration. The process is demonstrated in Figure 6.2.

Infeasibility Constraints

Infeasibility constraints are meant to remove configurations that are either completely impossible to be implemented and/or manufactured, or even if they could be, they would yield highly undesirable results that wouldn't change in the future even with further research. These kind of constraints are imposed at two

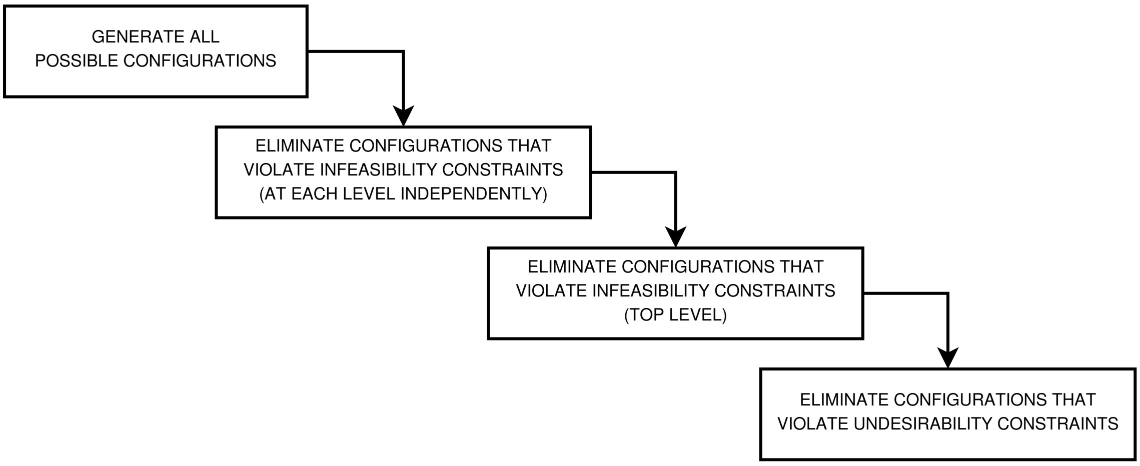


Figure 6.2: Process of discarding configurations using constraints

stages, as previously mentioned. The first stage is at each level independently, in this case, the geometry level has some specific infeasibility constraints, and the fuel system has it's own independent constraints. The second stage is with regards to the combined configurations from the different levels of the system, ie. a constraint between the element of one level and another element from a different level.

With the design parameters under consideration, the following constraints are imposed:

- A centre tank is present without an inerting system. This would violate safety regulations.
- No centre tank but 1 or more ACTs. ACTs can only be present with a main centre tank (hence the "Additional" in ACT)
- No centre tank with an inerting system present. That would render the inerting system useless since wing tanks do not require inerting.
- Narrow body with 4 engines. Even though this could go under the unde-

sirability constraints, no narrow body aircraft has 4 engines since it adds unnecessary complexity, weight, and it is not as efficient.

- A gravity-only transfer system with a trim tank present. A pressure system needs to be present for transferring fuel between the trim and wing tanks regardless of the aircraft's attitude.
- A gravity-only transfer system with more than 1 wing tank cell. A pressure system needs to be present in order to transfer fuel outboard for load alleviation purposes and to the engine feed tanks.
- A gravity-only transfer system with a centre tank present. The transfer system needs to be able to transfer fuel from the centre tank to the wing tanks, which requires a pressure-based system.
- A single wing tank cell with 4 engines. Each engine needs to have it's own feed tank which entails the need for more than one cell on each wing.

Any configuration that has any of the above features, is considered infeasible, and is discarded.

Undesirability Constraints

A number of configurations might evidently, and due to previous knowledge, yield undesirable results. However this might be due to technological limitations or other factors that might not be present at a future stage. Undesirability constraints aim to remove configurations that fall under that category but without discarding them completely (as opposed to infeasibility constraints where configurations are completely discarded). Undesirable configurations get stored in a separate database for future evaluations but do not proceed to the second stage of the framework.

The undesirability constraints that are imposed after the combination of the system-specific configurations are:

- A composite wing with more than one wing tank cell, increases the complexity unnecessarily. Composite wings do not require wing load alleviation since they do not suffer from structural fatigue the same way aluminium ones do.

- An aluminium wing, with long span and just one wing tank cell. Due to the large span, wing load alleviation mechanisms, using the wing tank cells, are highly desirable in order to reduce the fatigue of the wing.

- A narrow-body aircraft with a trim-tank. Smaller aircraft, in the narrow body range, do not usually require a trim tank. Therefore, having a trim tank would unnecessarily increase both weight and complexity.

- A narrow-body aircraft with a jettison system. Jettison systems are only found in large, wide-body aircraft where the MLW is substantially less than the MTOW.

Any configuration that falls under the above criteria, will no longer be considered for the second stage of the framework. They will however remain in the database but their status becomes ’inactive’ so that no further assessment on them takes place.

6.4 Traceability

As previously mentioned, the approach presented as part of this work, entails the generation of large amounts of data and information. The knowledge that results from this approach need to be traced and documented throughout the entire process. In order to do so, the configurations need to have a unique identification number that tracks the performance of each one. Furthermore, each configuration needs to have a record with the synthesis of it, the results for the performance metrics, and any further associated data.

6.4.1 Identification of Configurations

Before the records are created for each configuration, a unique identifier needs to be assigned to each one. In this example, the approach taken is by using the initials for each of the options that make up each configuration. Table 6.4 shows the initials corresponding to each option.

Using Table 6.4, the unique identification number (Configuration ID), can be synthesised for each configuration. In the case of a (L)ow aspect ratio, (S)hort span, (L)ow sweep angle, (N)arrow body, (A)luminium wings, (2)Engines, (1) wing tank cell, (Y)es to a centre tank, (0) ACTs, (N)o Trim tank, (Y)es to an inerting system, (B)oth a gravity and pressure transfer system, and (N)o Jettison system would yield the configuration ID LSLNA21Y0NYBN. A schematic representation of the above configuration and a configuration of the other extreme (High aspect ratio, Long Span, etc.) is shown diagrammatically in Figure 6.3.

The approach used here with the initials of each option, is useful for this application area where a relatively small number of parameters is considered. The approach not only gives meaning to the IDs, but also makes them human-readable. However, larger and more complex systems would require an alternative approach, such as a machine-readable ID, rather than a human-readable one which is used in this example.

Parameter	Options
Aspect Ratio	(L)ow, (H)igh
Wing Span	(S)hort, (M)edium, (L)ong
Sweep	(L)ow, (H)igh
Fuselage Width/Diameter	(N)arrow, (W)ide
Wing Material	(A)luminium, (C)omposite
Engines	(2), (4)
Wing tank cells	(1),(2),(3),(4)
Centre Tank	(Y)es, (N)o
ACT	(0),(1),(2)
Trim Tank	(Y)es, (N)o
Inerting System	(Y)es, (N)o
Transfer System	(P)ressure, (G)ravity, (B)oth
Jettison System	(Y)es, (N)o

Table 6.4: Initials for Configuration IDs

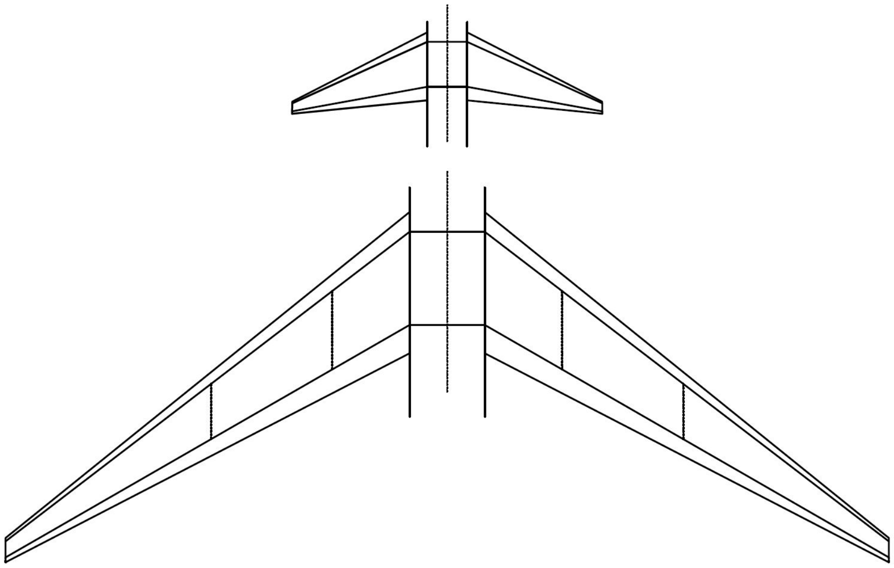
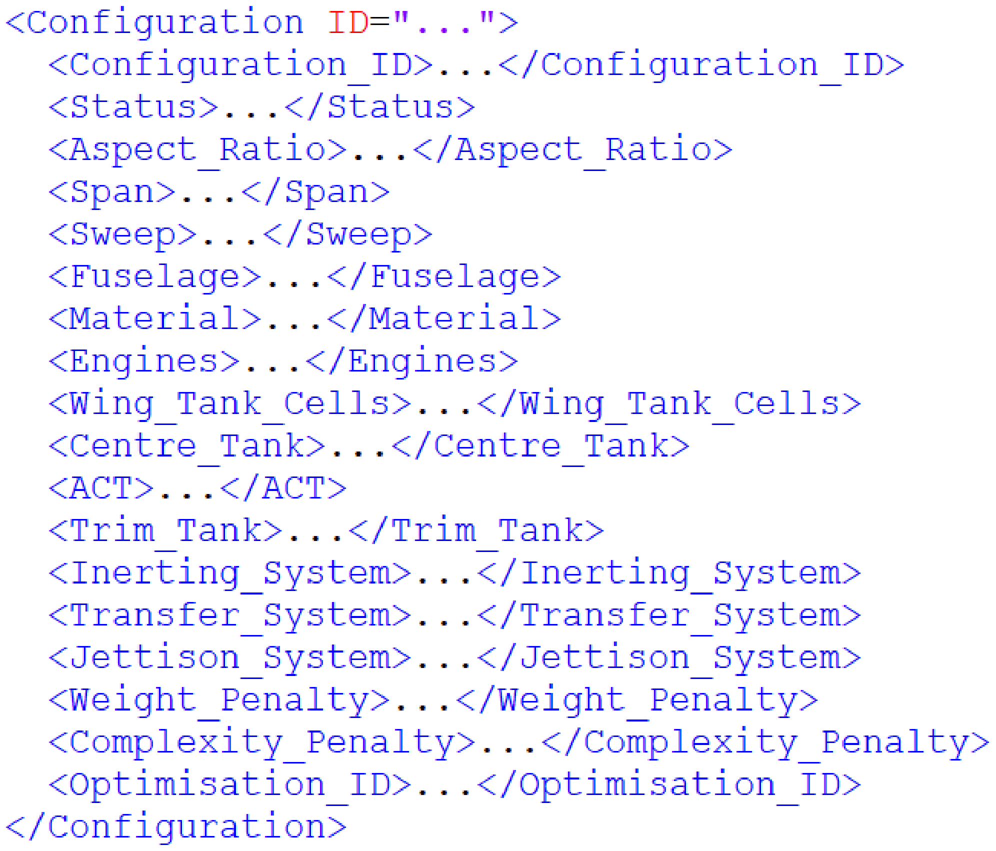


Figure 6.3: Scaled representation of the LSLNA21Y0NYBN and HLHWA23Y0NYBN configurations

6.4.2 XML Definitions

After the configuration IDs have been generated for all configurations, the records can start being built for each one. Using an XML file as described earlier, the entry for each configuration contains the synthesis of it with the options selected for each design parameter. Furthermore, each configuration has an entry for the main performance metrics; in this case the weight and complexity penalties. Figure 6.4 shows an example of an empty configuration entry in the XML file.



```xml
<Configuration ID="...">
  <Configuration_ID>...</Configuration_ID>
  <Status>...</Status>
  <Aspect_Ratio>...</Aspect_Ratio>
  <Span>...</Span>
  <Sweep>...</Sweep>
  <Fuselage>...</Fuselage>
  <Material>...</Material>
  <Engines>...</Engines>
  <Wing_Tank_Cells>...</Wing_Tank_Cells>
  <Centre_Tank>...</Centre_Tank>
  <ACT>...</ACT>
  <Trim_Tank>...</Trim_Tank>
  <Inerting_System>...</Inerting_System>
  <Transfer_System>...</Transfer_System>
  <Jettison_System>...</Jettison_System>
  <Weight_Penalty>...</Weight_Penalty>
  <Complexity_Penalty>...</Complexity_Penalty>
  <Optimisation_ID>...</Optimisation_ID>
</Configuration>
```

Figure 6.4: XML Document Structure for a Configuration

Data that cannot be included in the XML either due to their size or dimensionality, are referred to using file identifications. In this case, since optimisation results tend to be large in size and dimensions, they are saved in comma-separated variables (.CSV) files.

The optimisation IDs, that refer to the optimisation clusters, are similar to the configuration IDs but are reduced to the active parameters only. The naming convention for the optimisation clusters is outlined in Section 6.5.5.

6.5 Optimisation

The most important step of the second stage of the framework is the optimisation process. The step aims to explore the confined design space areas for each configuration and find the optimum or the non-dominated solutions for each one.

In this example, two objectives are considered to demonstrate the trade-off in multi-objective optimisation when considering competing objectives.

6.5.1 Conversion of Discrete Parameters

Before the optimisation process takes place, the design parameters that were previously discretised, need to be reverted back to their continuous ranges. Those ranges become the optimisation constraints for each continuous parameter.

Following the discretisation in Table 6.3 each configuration defines its constraints using the ranges previously selected. In this case, a Low Aspect ratio configuration, will have its Lower Bound (LB) at 7 and its Upper Bound (UB) at 8.5 for the Aspect Ratio variable. Any value above 8.5 would violate the constraints since it falls into the High aspect ratio area where other configurations lie.

6.5.2 Objectives

The optimisation process for this example aims to demonstrate the performance of two competing objectives that can be calculated analytically. The first objective aims to maximise the volume of the wingbox while the second one aims at minimising the volume of the surge tank. Since both objectives are dependent on common parameters like the span and the fuselage width, and their functions are opposite (maximisation and minimisation), a number of non-dominated solutions are expected to arise.

6.5.3 Wingbox Volume Calculation

One of the objectives for optimisation is to maximise the volume of the wingbox. The larger the volume, the more fuel can be stored in the wing. To calculate the volume, an analytical approach is used as presented in the NASA technical memorandum by Ardema et al. [120].

$$V_W = \frac{b_S(1 - C_{S_1} - C_{S_2} cos(\Lambda_S)}{3} \times [R_{t_R} C_R (2C_R + C_T) + R_{t_T} C_T (C_R + 2C_T)]$$

$$\underbrace{+(1 - C_{S_1} - C_{S_2}) R_{t_R} C_R^2 w_C}_{\text{Carrythrough Structure}} \tag{6.1}$$

$$A = \frac{b^2}{AR} \qquad b_S = \frac{b - w_C}{2cos(\Lambda)}$$

$$C'_R = \frac{2A}{b(1+TR)} \qquad C_T = TR \times C'_R$$

$$C_R = C'_R - \frac{w_C}{b}(C'_R - C_T)$$

where V_W is the wingbox volume, b_S is the structural semispan of the wing, C_{S_1} is the unusable part of the leading edge as a percentage of the chord, C_{S_2} is the unusable part of the trailing edge as a percentage of the chord, Λ_S is the sweep angle of the structural span at quarter-chord line, R_{t_R} is the thickness-to-chord ratio at root, R_{t_T} is the thickness-to-chord ratio at tip, C_R is the root chord of wing at fuselage intersection, C_T is the tip chord, w_C is the width of the carrythrough structure of the wing, A is the area of the wing, AR is the Aspect Ratio, b is the span if the wing, C'_R is the theoretical root chord and TR is the taper ratio. Some of the aforementioned parameters are shown in Figure 6.5.

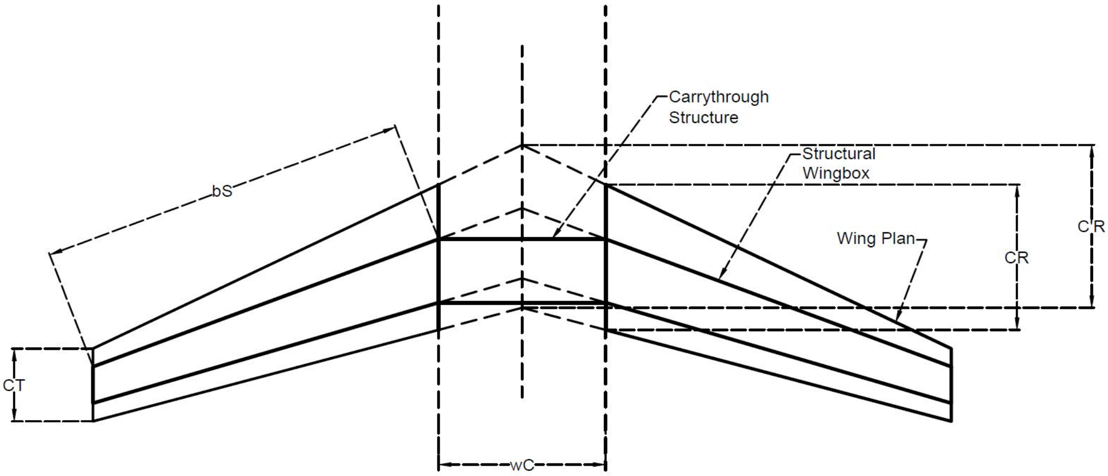


Figure 6.5: Structural wingbox and carrythrough structure

We consider the centre tank volume equal to the volume of the carrythrough structure. Therefore, the second part of the equation for the carrythrough, becomes zero for the configurations where no centre tank is present.

Each configuration has different bounds for the aspect ratio, wing span, sweep and the fuselage width, which are defined in the first stage during the generation of the configurations. In order to calculate the wingbox volume using the formula presented above, a number of other variables are required, the bounds of which are shared between all configurations. The three parameters that have the same range are the Taper Ratio and the Thickness/Chord ratio for both the tip and the root chord. Furthermore, the unusable parts of the leading and trailing edges, C_{S_1} and C_{S_2}, are constant at 0.14 and 0.2 respectively; as a percentage of the chord.

6.5.4 Surge Tank Volume Calculation

The volume of each surge tank is assumed to be three times the volume of the vent pipe for each wing. For simplification purposes, we assume that the length of the vent pipe is equal to the structural semispan of the wing plus 10% of the width of the carrythrough structure and the vent pipe has a constant radius of 0.05m.

$$V_S = 3(b_S \times \pi r^2) + \underbrace{3(0.1 w_C \times \pi r^2)}_{\text{Centre Tank section}} \tag{6.2}$$

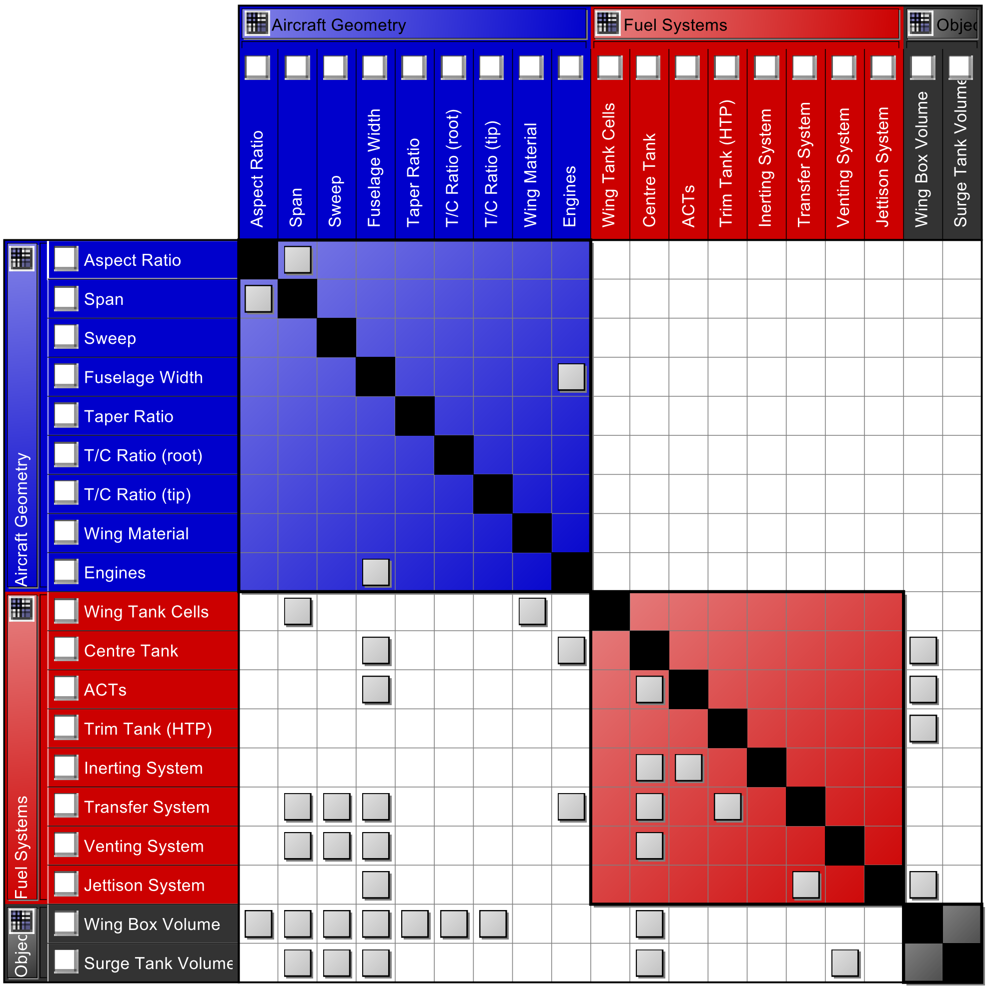
For configurations where a centre tank is not present the second part of the equation becomes zero.

The main difference between the optimisation of the wingbox volume and the surge tank is that the objective is to maximise the former and minimise the latter, hence, there is a trade-off.

6.5.5 Identification of Active and Dormant Parameters

Following the approach outlined in Part II, a DSM is employed to map the dependencies between the design parameters and the optimisation objectives.

Using the DSM in Figure 6.1, two additional elements are added to represent the two objectives. From there on the dependencies can be mapped; using the objective functions and identifying the variables that affect them. Figure 6.6 shows the expanded DSM.

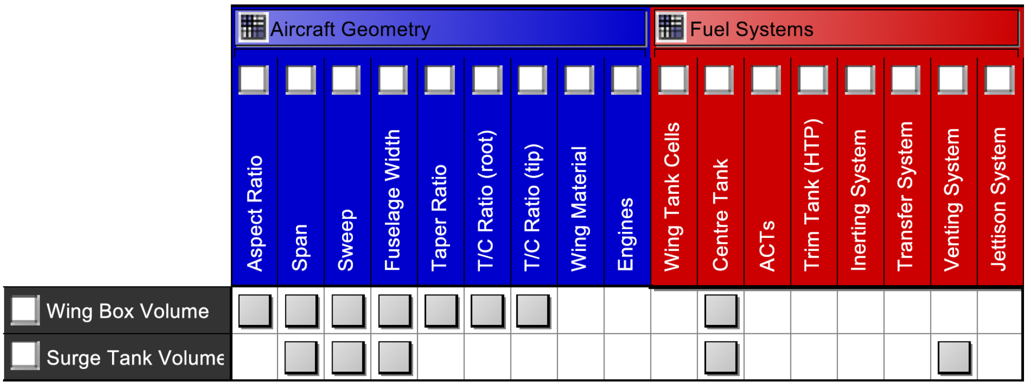


		Aspect Ratio	Span	Sweep	Fuselage Width	Taper Ratio	T/C Ratio (root)	T/C Ratio (tip)	Wing Material	Engines	Wing Tank Cells	Centre Tank	ACTs	Trim Tank (HTP)	Inerting System	Transfer System	Venting System	Jettison System	Wing Box Volume	Surge Tank Volume
Aircraft Geometry	Aspect Ratio	■	☐																	
	Span	☐	■																	
	Sweep			■																
	Fuselage Width				■					☐										
	Taper Ratio					■														
	T/C Ratio (root)						■													
	T/C Ratio (tip)							■												
	Wing Material								■											
	Engines				☐					■										
Fuel Systems	Wing Tank Cells		☐						☐		■									
	Centre Tank				☐					☐		■							☐	
	ACTs				☐							☐	■						☐	
	Trim Tank (HTP)													■					☐	
	Inerting System											☐	☐		■					
	Transfer System		☐	☐	☐					☐		☐		☐		■				
	Venting System		☐	☐	☐							☐					■			
	Jettison System				☐											☐		■	☐	
Objec	Wing Box Volume	☐	☐	☐	☐	☐	☐	☐				☐							■	▨
	Surge Tank Volume		☐	☐	☐							☐					☐		▨	■

Figure 6.6: Identification of active parameters for optimisation

At this point, the interest lies on the connections between objectives and variables; therefore, the interconnections between the variables can be removed.

This allows the DSM to be truncated in order to identify the active parameters more easily as shown in Figure 6.7.



	Aircraft Geometry									Fuel Systems							
	Aspect Ratio	Span	Sweep	Fuselage Width	Taper Ratio	T/C Ratio (root)	T/C Ratio (tip)	Wing Material	Engines	Wing Tank Cells	Centre Tank	ACTs	Trim Tank (HTP)	Inerting System	Transfer System	Venting System	Jettison System
Wing Box Volume	■	■	■	■	■	■	■				■						
Surge Tank Volume		■	■	■							■					■	

Figure 6.7: Reduced view of the DSM for the identification of active parameters.

Using Figure 6.7 the active parameters become clear. For example, any competing configurations that only differ in the number of engines, or the number of ACTs, will yield the same optimisation results and will therefore belong in the same optimisation cluster. The active parameters that **do affect** the optimisation are:

- Span
- Aspect Ratio
- Sweep Angle
- Fuselage width
- Taper Ratio, Thickness to chord Ratios (these are shared ranges between all configurations)
- Centre Tank
- Venting System

Using the parameters above the optimisation clusters IDs can be formulated. Using the example of configuration LSLNA21Y0NYBN, the optimisation ID of its cluster would be LSLNY. This is due to the active parameters being: (L)ow aspect ratio, (S)hort span, (L)ow sweep angle, (N)arrow body, and (Y)es to a centre tank; the venting system does not have options. The other initials can be removed as they do not affect the objective. The same optimisation ID is shared with other configurations in the same cluster that only differ at inactive parameters.

6.6 Further Assessment and Evaluation

The optimisation process alone will often not yield enough information for design decision making. It is therefore important to evaluate the configurations further, in order to gather more information.

Further assessment and evaluation for this example considers complexity and weight penalties, if the system meets certain performance requirements such as tank venting in rapid descent cases, and sensitivity to design changes (change and cost propagation).

For the system performance and change propagation, the two extreme configurations presented before (LSLNA21Y0NYBN and HLHWA23Y0NYBN) will be described.

6.6.1 Complexity and Weight Penalties

In the early stages of a design lifecycle, knowledge-based and empirical methods can provide additional information for decision making. Following the optimisation process outlined in the previous Section, a configuration with a long span, will result in a larger wingbox volume when compared to a configuration with a short span; given that all other design parameters remain the same. However, the Long span configuration will result in a heavier structure, amongst other possible drawbacks.

For this reason, knowledge-based penalties can be introduced for each option of every design parameter. At this stage, two different penalties are considered: weight and complexity. Complexity includes difficulties in design, manufacturing, and maintenance. The penalties are assigned on a range from 0 to 10, with 10 being the most heavy/complex option. The lightest or simplest option of each design parameter always has a zero value for that penalty; for example, a short span wing has a zero weight penalty since it's the lightest of the three options (Short, Medium, Long). Table 6.5 shows the associated weight and complexity penalties for all options.

When all the configurations have been generated, their total weight and complexity penalties can be calculated; this is done by summing up the penalties of the options that form the configuration. Using this approach provides a fast approach to introduce an additional metric with which the configurations can be further assessed. However, such an approach entails high levels of uncertainty and it is only applicable during the early stages of design; where a rapid assessment is required due to the large number of configurations being considered. At later stages, higher fidelity methods will need to be employed to assess each configuration in more detail through simulations.

Parameter	Option	Weight Penalty	Complexity Penalty
Aspect Ratio	Low	0	0
	High	0	0
Wing Span	Short	0	0
	Medium	+1	0
	Long	+2	0
Sweep	Low	0	0
	High	0	0
Fuselage Width/Diameter	Narrow Body	0	0
	Wide Body	+4	+1
Wing Material	Aluminium	+4	0
	Composite	0	+3
Engines	2	0	0
	4	+2	+2
Wing Tank Cells	1	0	0
	2	0	+1
	3	0	+2
	4	0	+3
Centre Tank	Yes	0	+1
	No	0	0
ACT	0	0	0
	1	0	+1
	2	0	+2
Trim Tank	Yes	+2	+5
	No	0	0
Inerting System	Yes	0	+1
	No	0	0
Transfer System	Pressure	+1	+1
	Gravity	0	0
	Both	+1	+2
Jettison System	Yes	0	+1
	No	0	0

Table 6.5: Penalties for each design parameter option

6.6.2 System Performance

When the configurations have been generated the initial simulation models can be constructed; the details of the models however, such as the tank geometries, will be dependent on the optimisation results. The models can be used to ensure that the selected configurations meet certain performance criteria, and can also be used for further optimisation studies.

Siemens AMESim (Advanced Modelling Environment for Simulations), part of the LMS Imagine.Lab suite, is a 1D modelling and simulation environment [117]. In this work it is employed for modelling, simulating and assessing selected configurations by determining their technical performance.

The main drivers behind choosing AMESim are the ease of use of the software, its extensive capabilities, and the fact that it has an in-built library with models of aircraft fuel system components. It also provides a range of input and output analysis tools, design space exploration and custom scripting.

The aircraft fuel systems library comes with ready-to-use models for components such as the fuel tanks, pipes, orifices, etc.; all of which are fully customisable. Other libraries such as pneumatics, thermal, thermal hydraulic, and gas mixing were also used. The fuel tanks were defined by tables that specify the volume of fuel, and wet/dry areas at different aircraft attitude during a flight, depending on the geometry of the tank.

One of the performance metrics that were evaluated in this case study, was the performance of the venting system. As stated previously, the vent line lies on the structural semispan of each wing. For configuration LSLNA21Y0NYBN the venting system is presented in Figure 6.8 where the blue dashed lines represent the venting pipelines.

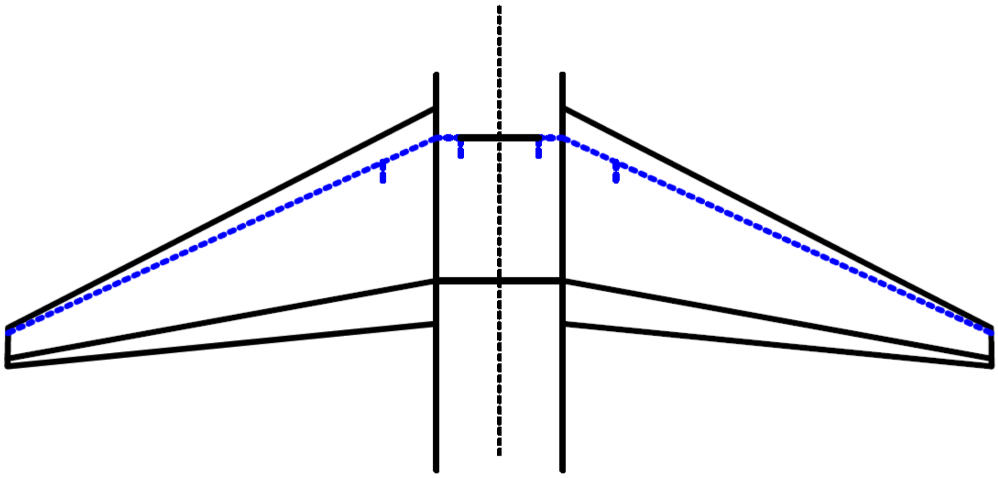


Figure 6.8: Venting System Schematic for configuration LSLNA21Y0NYGN.

To model the above configuration in AMESim, a number of model libraries were used, including the aircraft fuel systems library and the gas mixing library. Figure 6.9 presents the AMESim model of the LSLNA21Y0NYBN tank configuration with only the venting and inerting systems; this is the initial configuration of the system as the tank geometries and pipe lengths were defined using the optimisation results.

Each block in the model represents a mathematical model; details of each one can be found in the appendix. The components are coloured depending on the library they belong in; for example, the aircraft fuel systems library has a blue colour while the signal processing library is red.

It becomes apparent that models for each configuration, even for simple systems, can become complex and difficult to manage quite fast. This becomes evident when more complex configurations start being modelled as shown in Figure 6.10; where the venting and inerting system of the HLHWA23Y0NYBN configuration is presented.

Schematically, the only thing that changes between the two models at this

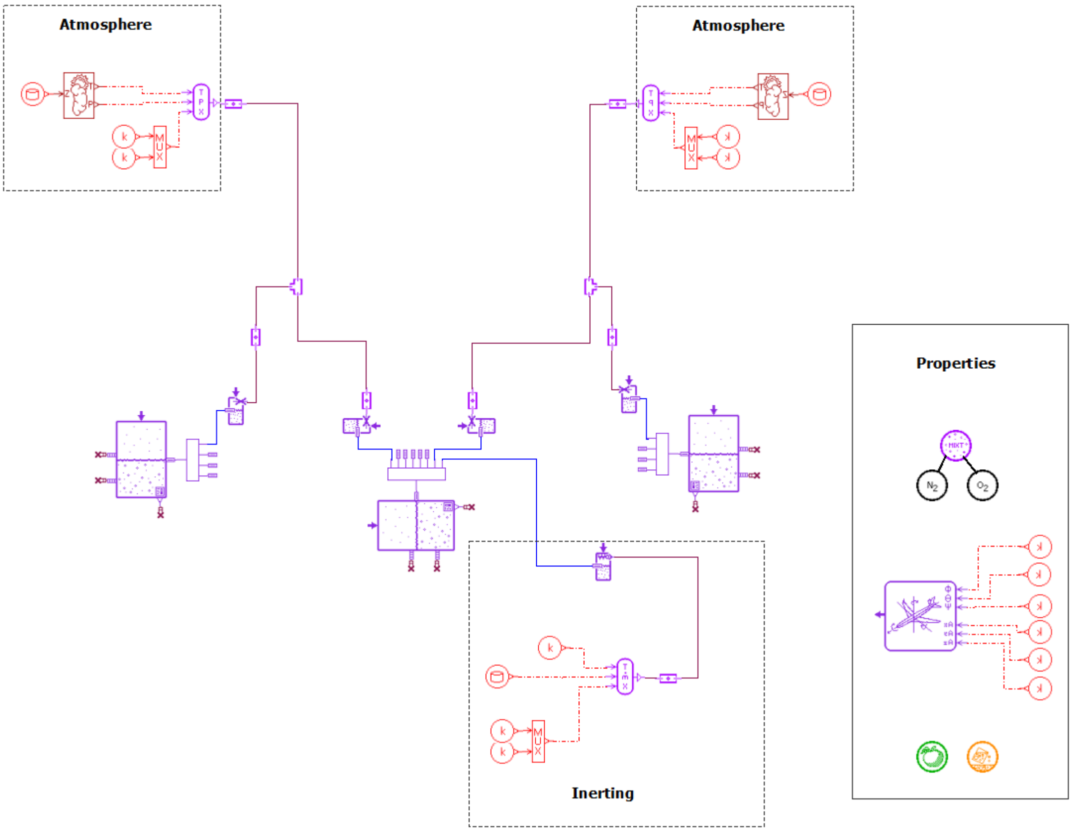


Figure 6.9: AMESim Model for the venting and inerting systems of configuration LSLNA21Y0NYBN.

point is the number of wing tank cells and consequently the venting line. The main difference between the 2 models is the size and location of the tanks which will be defined after the optimisation takes place; the results of which will also define the length and performance of the venting system.

The models can be further expanded to include the refuelling system; the pipelines of which are also used for the transfer system in most cases. Using the initial HLHWA23Y0NYBN model, the refuelling system was added to it as shown schematically in Figure 6.11; blue lines represent the venting system, while orange lines represent the refuelling system.

The venting system performance can be evaluated in two scenarios: the

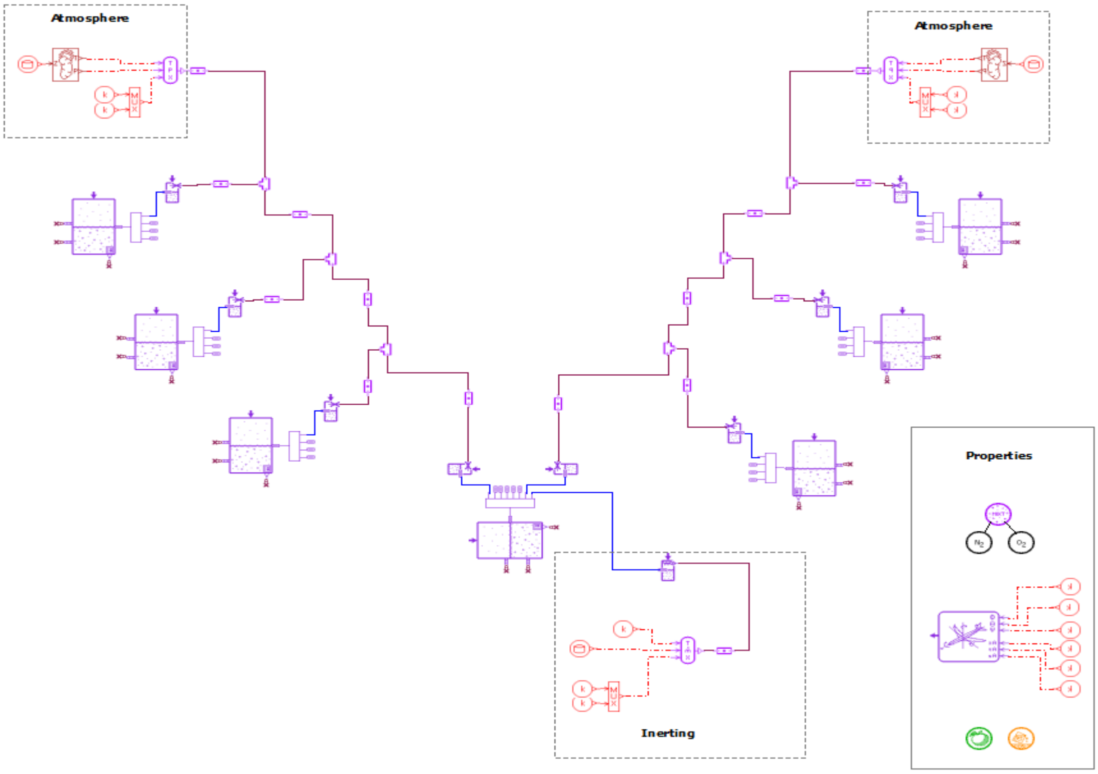


Figure 6.10: AMESim Model for the venting and inerting systems of configuration HLHWA23Y0NYBN.

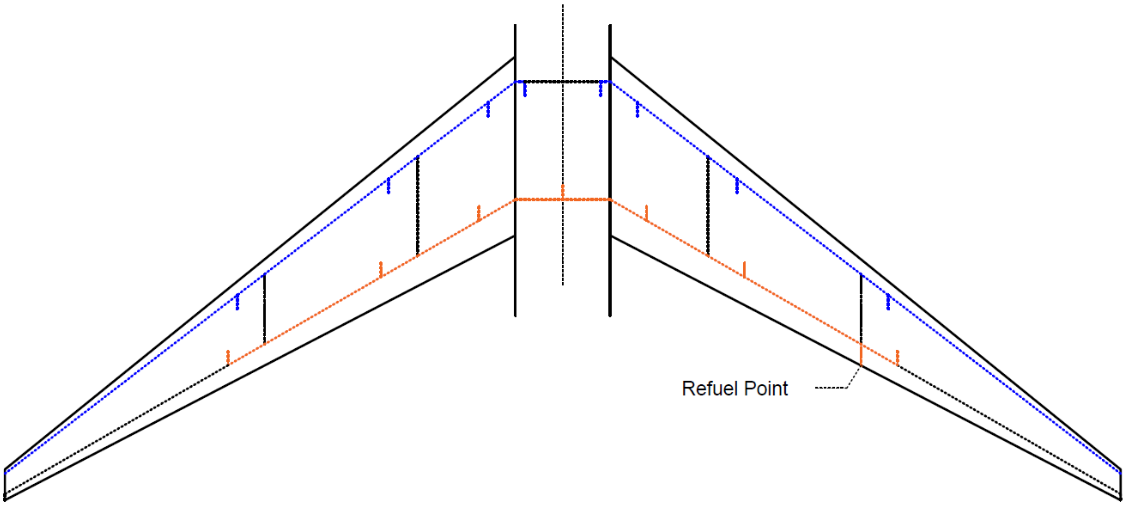


Figure 6.11: Schematic representation for the venting and refuelling systems of configuration HLHWA23Y0NYBN.

maximum descend case, and on ground refuelling; in both cases, there must be sufficient air flow in and out of the tank ullage so that no pressurisation occurs. Using the schematic representation, the AMESim models can be expanded to include the refuelling system as shown in Figure 6.12. However, the inerting system is removed at this stage since the refuelling case will be used for the evaluation of the venting system alone.

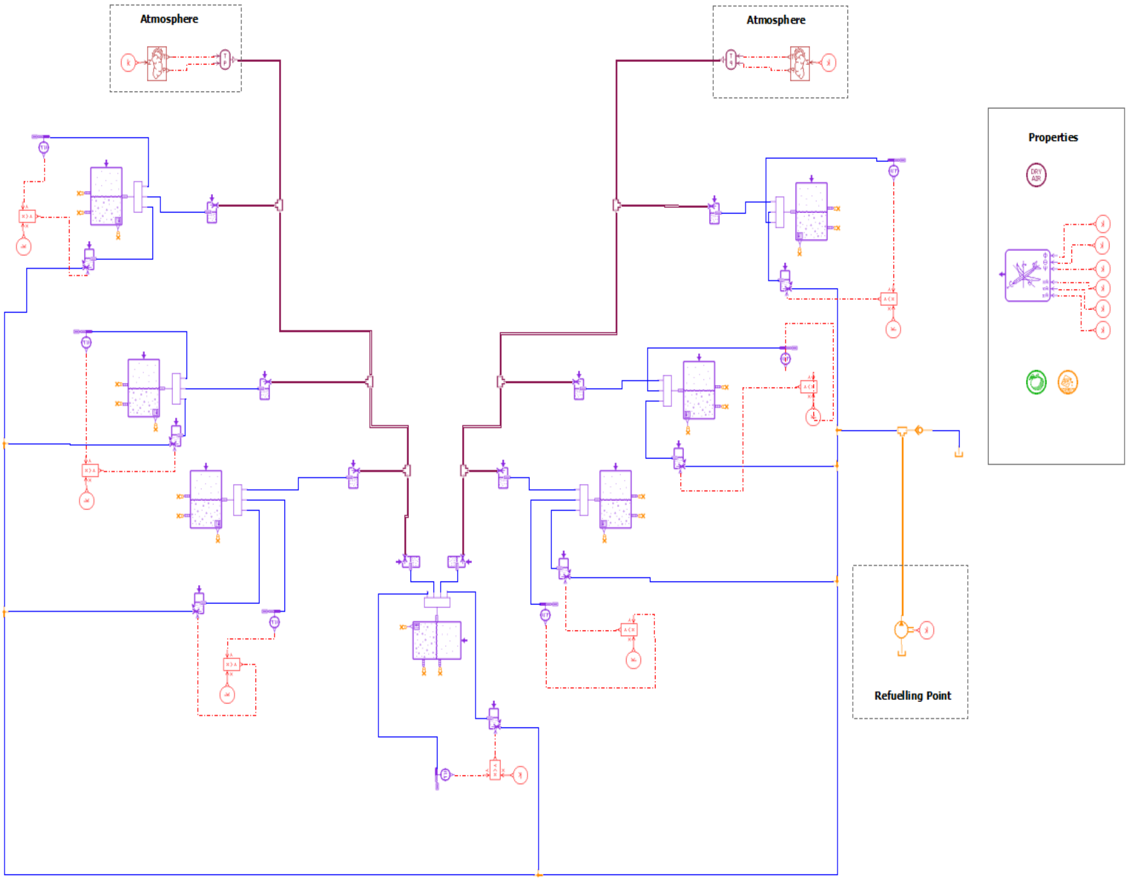


Figure 6.12: AMESim Model for the venting and refuelling systems of configuration HLHWA23Y0NYBN.

The simulations ran for the two scenarios described above in order to evaluate the performance of the venting system. For the maximum descent case, a table with the change of altitude over time was provided as shown in Figure 6.13.

The rate of descent was approximately 6000 feet per minute until the aircraft reached an altitude of 25000 feet and then it continued with a normal descend rate.

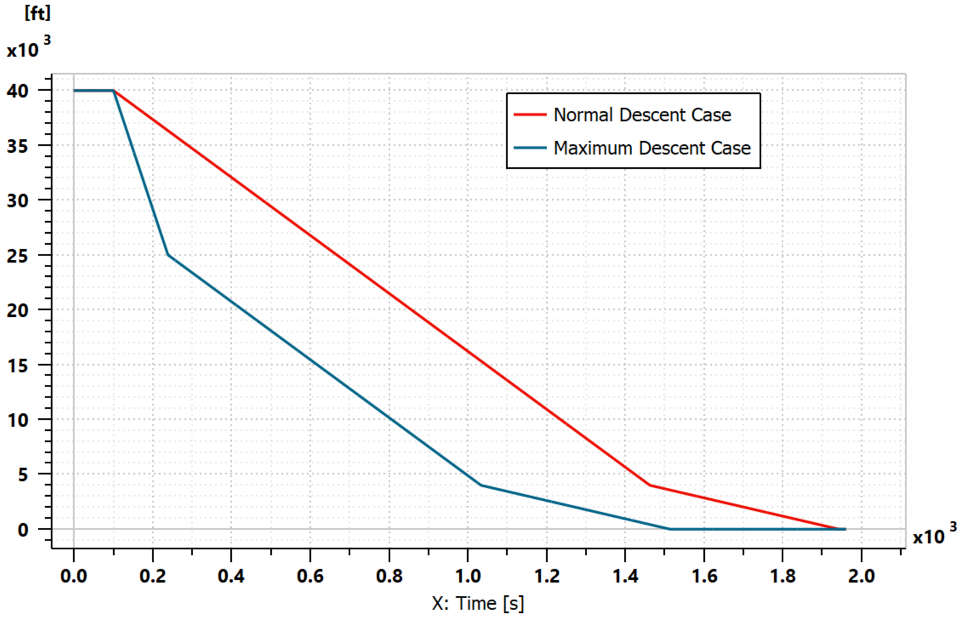


Figure 6.13: Altitude change for both descent cases

The inerting system was also evaluated during the maximum descent scenario. It is expected that if the system can keep the centre tank inert in such extreme case it will be able to meet the performance requirements under normal operating conditions as well. The mass flow rate of the NEA was set at a maximum of 70g/s. The inerting system starts operating right before the aircraft starts its descend, which is where there would be a large flow of atmospheric air in the tank. It was also assumed that the centre tank is empty when the descend begins. Figure 6.14 shows the NEA flow rate over time for both descent cases.

Finally, for the refuelling case, the refuelling rate was set at a constant 6000 Litres per Minute. Sensors were used for each tank that indicated when a tank had reached 95% capacity, at which point the inflow valve of the tank was closed

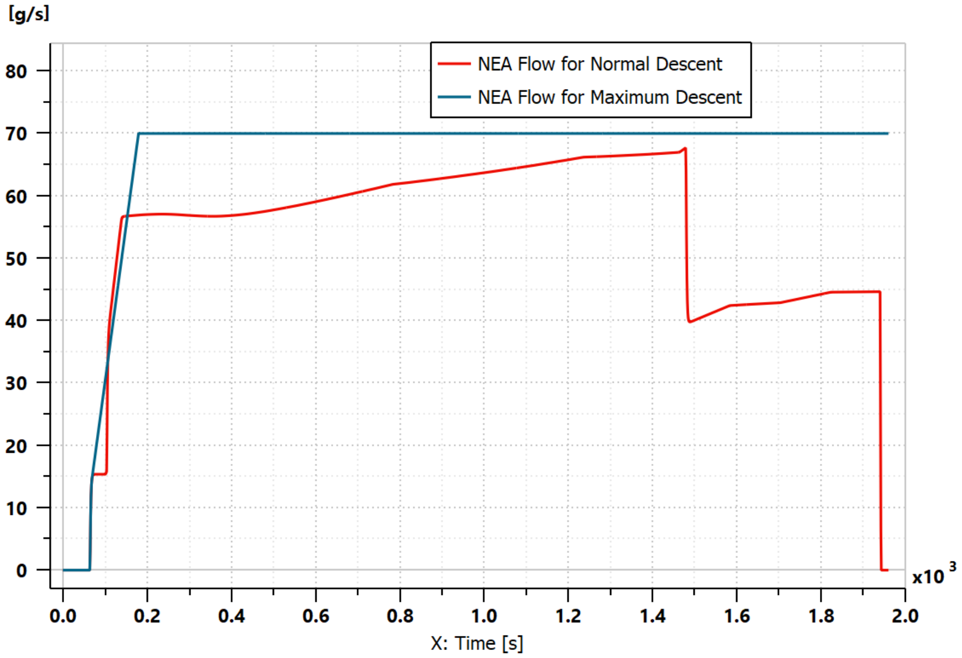


Figure 6.14: NEA Flow rates for both descent cases

in order to avoid overfilling. Since, at this stage, the conditions considered were ideal (frictionless flow), it was expected that the tanks would be refuelled at the same rate in parallel. However since the Centre Tank had a smaller capacity than the wing tank cells, it was expected to be the first one to fill.

Besides the performance of the system in terms of flowrate etc., AMESim provides the capability to observe the thermal characteristics of the system, as well as mass properties (Centre of gravity), for all stages of flight. In cases where a trim tank is present, a fuel burning sequence using in-built statecharts can be introduced for controlling the CoG. Even though those metrics are not considered in the current work, they are areas for further exploration in future work.

6.6.3 Change and Cost Propagation

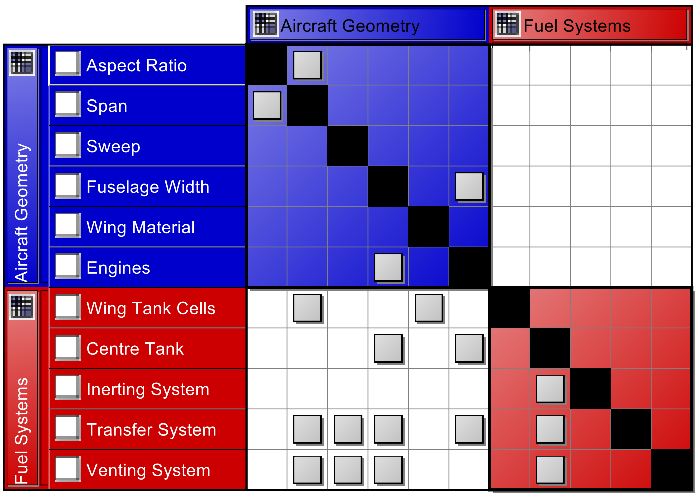
Further to the penalties and the system performance, the configurations were also assessed in terms of sensitivity to design changes and the associated costs. The DSMs that were used for modelling the change propagation of each configuration were modifications of the fuel system base DSM presented in Figure 6.1.

The modified DSMs were quantified with values for likelihood and impact for the purposes of calculating the change propagations, and the elements were assigned a cost weight for the cost propagation. The same selected configurations that were used for the system performance metrics, were also used as example configurations for the prediction of cost and change propagation.

The DSM can be reduced by removing the elements that are not present in the configurations. For example, neither configuration has an ACT, a trim tank or a jettison system, therefore those elements can be removed from the DSM. Furthermore, the elements for Taper Ratio, and both T/C ratios can be removed since they have no connections. This means that both configurations will share the same reduced DSM shown in Figure 6.15.

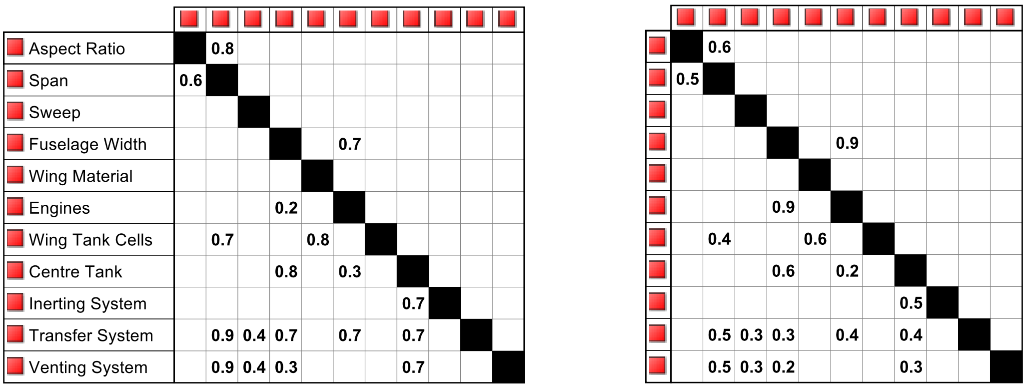
In order to calculate the change propagation and the combined risks, the DSM connections needed to be quantified with values for direct likelihood and impact. The quantified DSMs are shown in Figure 6.16. All values are empirical and based on expert judgement.

Furthermore, to calculate the costs associate with the change propagation, besides the probability DSM, a cost weight needed to be assigned to each system element. Table 6.6 shows the respective cost weight for each element. As with the change propagation prediction, the cost weights are also empirical.



		Aircraft Geometry						Fuel Systems				
Aircraft Geometry	Aspect Ratio	■	☐									
	Span	☐	■									
	Sweep			■								
	Fuselage Width				■		☐					
	Wing Material					■						
	Engines				☐		■					
Fuel Systems	Wing Tank Cells		☐			☐		■				
	Centre Tank				☐		☐		■			
	Inerting System								☐	■		
	Transfer System		☐	☐	☐		☐		☐		■	
	Venting System		☐	☐	☐				☐			■

Figure 6.15: Reduced DSM for both configurations



	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥 Aspect Ratio	■	0.8									
🟥 Span	0.6	■									
🟥 Sweep			■								
🟥 Fuselage Width				■		0.7					
🟥 Wing Material					■						
🟥 Engines				0.2		■					
🟥 Wing Tank Cells		0.7			0.8		■				
🟥 Centre Tank				0.8		0.3		■			
🟥 Inerting System								0.7	■		
🟥 Transfer System		0.9	0.4	0.7		0.7		0.7		■	
🟥 Venting System		0.9	0.4	0.3				0.7			■

(a) Likelihood/Probability

	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥	■	0.6									
🟥	0.5	■									
🟥			■								
🟥				■		0.9					
🟥					■						
🟥				0.9		■					
🟥		0.4			0.6		■				
🟥				0.6		0.2		■			
🟥								0.5	■		
🟥		0.5	0.3	0.3		0.4		0.4		■	
🟥		0.5	0.3	0.2				0.3			■

(b) Impact

Figure 6.16: Quantification of the reduced DSM with values for Likelihood and Impact

Element	Cost Weight
Aspect Ratio	0.6
Wing Span	0.6
Sweep	0.4
Fuselage Width	0.9
Wing Material	0.7
Engines	0.8
Wing tank cells	0.5
Centre Tank	0.4
Inerting System	0.1
Transfer System	0.4
Venting System	0.3

Table 6.6: Cost weight for system elements

Chapter 7

Computational Workflow Implementation

7.1 Introduction

As mentioned in Section 6.1.1, in order to integrate the steps and processes of ADOPT, a workflow management tool is required. For this case study, pSeven Suite by DATADVANCE was employed to manage the workflow and dataflow of the two stages of ADOPT. The scripts were coded in Matlab for the generation of configurations, while Python was used for all other tasks such as optimisation and data management.

The workflow was built upon multiple levels with the top level presented in Figure 7.1. The top level consists of 4 Blocks: 3 Composite ones and 1 Python script; composites are blocks that contain another workflow within them when expanded. From the top level view of the workflow, the first step was to process Stage 1 of the framework which deals with the generation of configurations and the associated XML and CSV files for record-keeping.

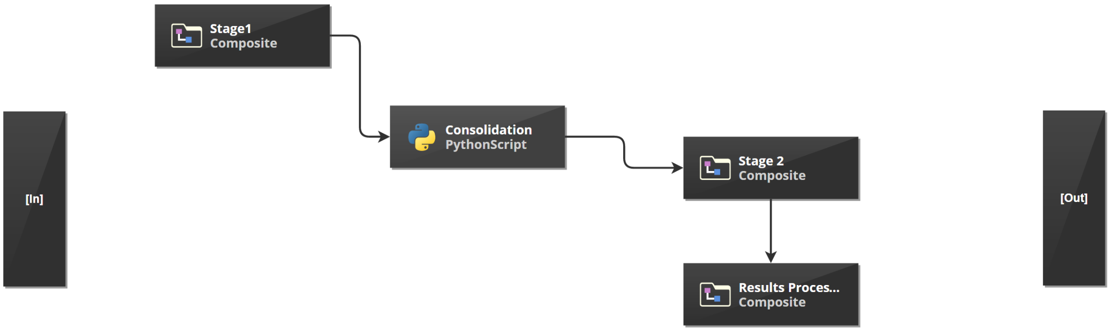


Figure 7.1: Top level view of the Workflow

Before proceeding to the second stage, the active cases are passed from the 'Consolidation' block. A python script consolidates the active cases using the selection of Active and Dormant parameters. It then passes on the cases that will yield unique optimisation results.

Stage 2 receives the consolidated configurations with their optimisation IDs, and optimises each configuration against the predefined objectives, with its respective constraints. The results from Stage 2 are transferred to the final composite block, which deals with the calculation of penalties and processing of results.

A detailed description of the ADOPT implementation in pSeven is presented in the following sections along with the expanded composite blocks.

7.2 Stage 1

The first stage of the workflow(and the framework) is made up of 4 blocks as shown in Figure 7.2. The stage is concerned with the following processes:

- Generation of Configurations

- Infeasibility and Undesirability checks
- Creation of the XML database for record keeping and the CSV file for the optimiser
- Conversion of discretised parameters back to ranges for optimisation purposes

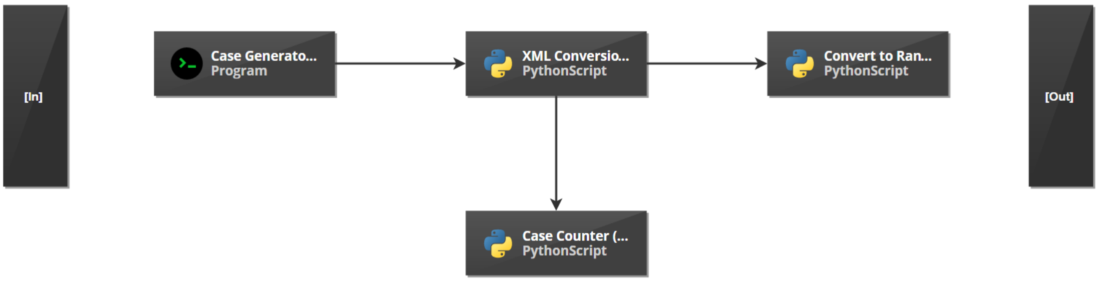


Figure 7.2: Stage 1 composite block

The first and main block is a Matlab program that generates the configurations and imposes the infeasibility and undesirability constraints. The program consists of the 7 scripts described in detail in Table 7.1.

The options for the aircraft level and the fuel system level are predefined in their respective scripts. The main script, CaseGenerator.m calls the two scripts to generate all possible combinations at each level, and discards the infeasible ones. The combine.m script then combines all the feasible options from the two level, before those configurations are checked against another set of infeasibility constraints and undesirability.

The results from the Matlab program are then saved in a CSV-format file. The file contains the parameters of all feasible configurations along with their respective Configuration IDs and status; cases are active if they have passed the

Matlab script name	Details
CaseGenerator.m	Top level script. This is the script that manages all sub-processes and the script that gets called from the pSeven workflow.
systemoptions.m	Combines options at the fuel system level and performs infeasibility checks.
aircraftoptions.m	Combines options at the aircraft geometry level and performs infeasibility checks.
combine.m	Combines the configurations from the two levels to get the full system configurations
convert.m	Generates the Configuration IDs, and adds the 'Status' of each case to the database
feasibility.m	Checks the combined configurations from the two levels against infeasibility constraints. Configurations that violate the constraints get discarded.
desirability.m	Checks the combined configurations against undesirability constraints, and changes the "Status" parameter to inactive if any of those constraints are violated.

Table 7.1: Matlab scripts for the generation of configurations

undesirability constraints and inactive if they failed.

The CSV file is then passed through a Python script that extracts all the important information (IDs, parameters, status,) and creates the XML file for record keeping. The original CSV file is then passed through a second python script that converts the discrete parameters back to their respective ranges. This is done in order to prepare the data for the optimiser that follows in the next stage.

However, before the workflow proceeds to the second stage, there is an intermediate python script between the two composite blocks(Stage 1 and Stage

2). The script processes the CSV file from the first stage, and removes the duplicate configurations based on the dormant parameters. The active and dormant parameters again are predefined before the workflow starts. The intermediate script also generates the optimisation IDs for the unique configurations; unique in this context refers to the configurations that yield different optimisation results. The output of this script is a consolidated version of the original CSV file that contains the configurations to be optimised, along with their optimisation IDs and upper and lower bounds for optimisation.

7.3 Stage 2

Stage 2 of the workflow deals exclusively with the optimisation process. The breakdown of the composite block for the stage is presented in Figure 7.3 below.

The second stage makes use of the consolidated CSV file since, at this stage, only the optimisation process is considered. The first step in the block is to check if the results are already available (from previous runs) and create a new entry for the current run.

The consolidated CSV file passes through a python script that splits the file in two; both contain the configurations for optimisation but the first includes the lower bounds for each parameter, while the second one contains the upper bounds for each one.

The "map" blocks of pSeven are responsible for processing the CSV files row by row. At each iteration, the "map" blocks send a row, and when they receive a signal back they proceed to send the second row. This way, at each iteration, each mapping block sends the respective bounds(upper and lower) for the configuration that is to be optimised.

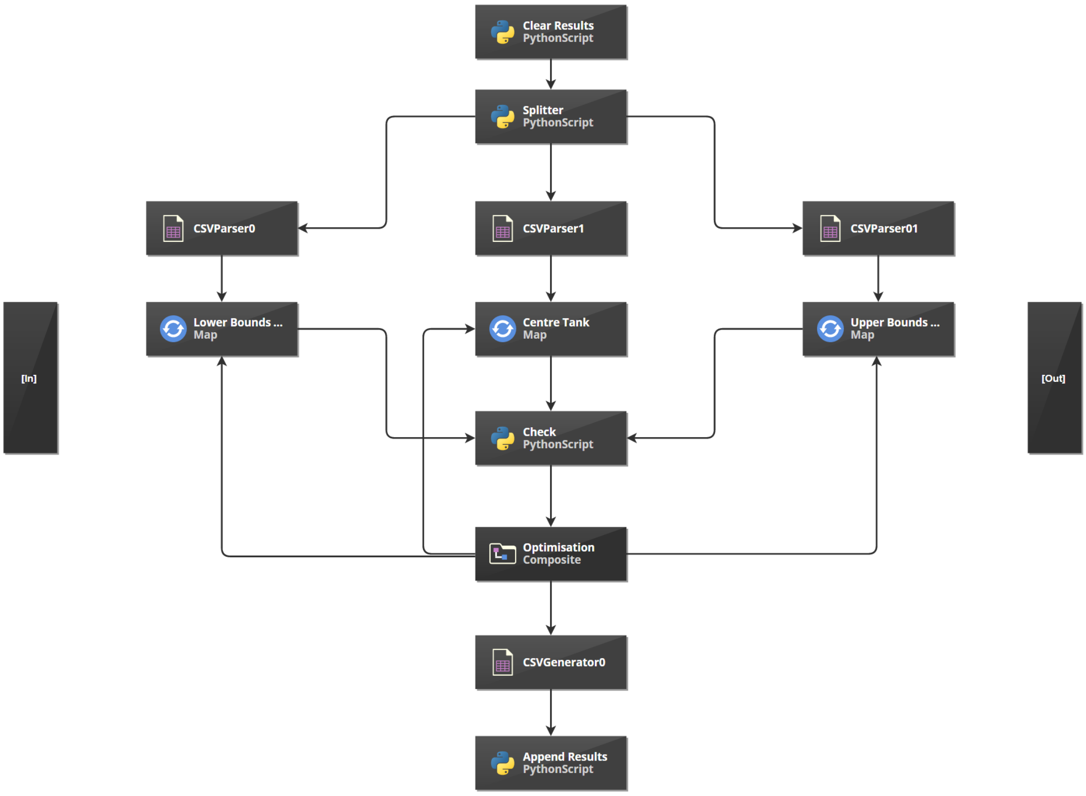


Figure 7.3: Stage 2 composite block

A third CSV Generator and Map pair is also used. The purpose of the third one is to inform the optimiser whether the configuration that will be optimised contains a centre tank or not. It is essential for the optimiser to know whether a centre tank is present since the inclusion of one or not changes the objective functions as shown in equations 6.1 and 6.2.

A python script is present between the mapping blocks and the optimiser to ensure that the bounds and the centre tank information that each mapping block sends to the optimiser are the correct ones.

Following the python script that checks the bounds, the composite block that consists of the optimiser and the evaluation functions performs the optimisation process for each configuration. The expanded composite block for optimisation is shown in Figure 7.4.

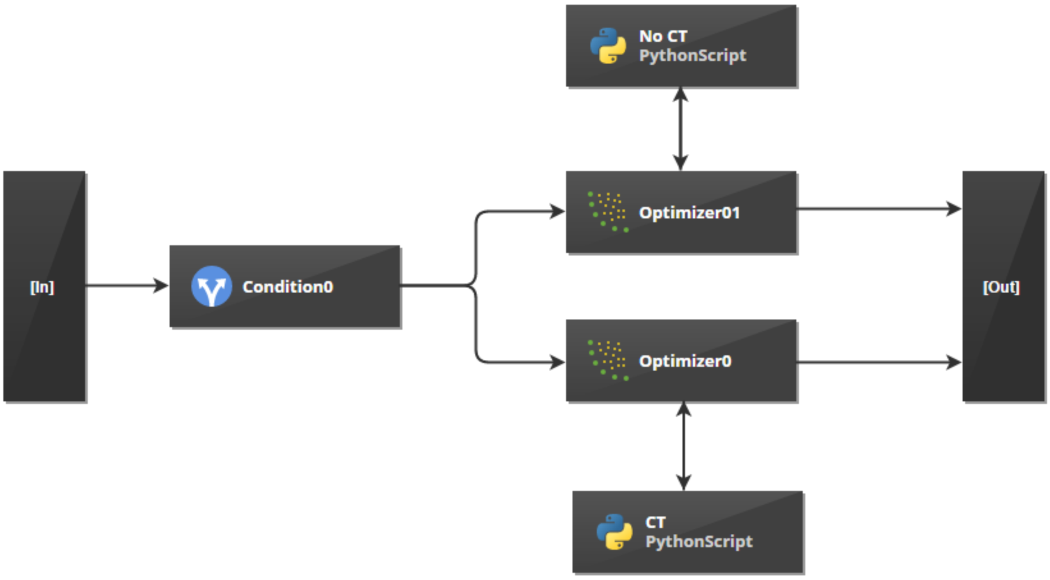


Figure 7.4: Optimisation composite block

The composite optimisation block contains a condition check as the first step. The condition block checks whether the configuration being optimised contains a centre tank or not. Depending on that, the data continues to flow to the respective ”Optimizer” block. The top optimizer uses the modified evaluation functions shown in equations 6.1 and 6.2 without the second part that concerns the centre box. The bottom optimizer deals with configurations that do include a centre tank. Furthermore, the respective optimizer block receives and sets the upper and lower bounds for each variable.

The optimizer blocks are pre-configured with regards to the number and type of objective functions to be evaluated. The user provides ”Hints” as to the evaluation cost type (Cheap / Expensive) and the linearity type (Generic / Linear / Quadratic). The user also provides the blocks with the type of problem that needs to be solved whether it's analytical, noisy, or smooth. Using the inputs, the block uses the SmartSelection tool to automatically chose the appropriate optimisation algorithm for the problem.

In design processes, such as the one presented in this work, a tool like this can be very useful as it can adapt to the difficulty and type of the optimisation problem at hand. For this case study the inputs provided to the optimiser are cheap and generic evaluations, and the problem is set as analytical.

The python scripts accompanying each optimizer block deals with the evaluation of the analytical functions for the wingbox and surge tank volumes. They receive a set of variables from the optimiser within the predefined bounds, calculate the volumes and return the results to the respective optimiser. This process goes on for a predefined number of steps until a number of non-dominated solutions are obtained.

After each configuration has been optimised, and the non-dominated results have been acquired, they are stored in a CSV file with the appropriate structure. The CSV file gets appended with the non dominated solutions each time a configuration completes the optimisation process. The end result is a CSV file containing all the optimised configurations with the respective solutions for each one.

7.4 Results Processing

When both stages of the framework have been completed, a final composite block manages the results and the processing of them; the results processing block is shown in Figure 7.5. Two main files are used for this stage: the first is the XML file of all the configurations along with their status and design parameters, and the second is the CSV file with the non-dominated solutions for each Optimisation case.

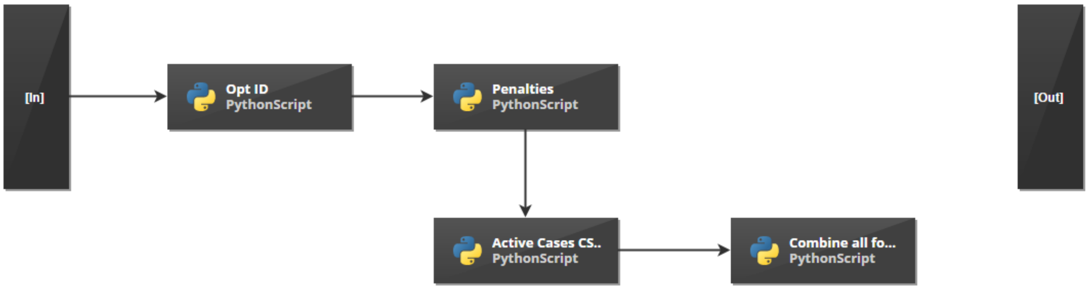


Figure 7.5: Results Processing composite block

The first step of the post-optimisation process is to format the CSV file of the optimisation results to add the Optimisation IDs for each set of solutions(and respective variables for each one) using the order they were optimised.

The second step (which can be processed in parallel with the previous one) is the calculation of the penalties for each active case. Following the process described in the previous chapter, each configuration receives a value for complexity and weight penalty. After the calculation is completed, each 'Active' entry in the XML file is updated with the respective penalties.

All the active configurations from the XML file are gathered in a new CSV file that contains the Case ID, Complexity and weight penalties, and their respective Optimisation ID. Using the newly generated CSV file, the final step of the workflow, merges it with the optimisation results, to generate the full dataset required for plotting the parallel coordinates. Using this approach, the resulting dataset will contain each configuration separately with its penalties, and the associated non-dominated options for each one.

Chapter 8

Results

8.1 Stage 1 Results

The first stage of the workflow, as presented in the previous section, deals with the generation of configurations. The system in consideration consists of 13 design parameters (not considering the common ones such as the taper ratio), and each design parameter has a number of possible options. To get the total number of possible combinations the number of options of each parameter needs to be multiplied together as shown in Table 8.1.

From Table 8.1 it becomes apparent that even for a small number of design parameters, each with just a few options, the resulting number of possible configurations is very high. Evaluating and optimising 55296 configurations would not only require a prohibitive amount of time, it would also generate a large amount of data that would be difficult to manage.

For this reason, infeasibility and undesirability constraints were imposed to reduce the number of configurations to be evaluated. The first set of infeasibility constraints was set at each level independently, while the second was set after

Design Parameter	Number of options
Aspect Ratio	2
Span	3
Sweep	2
Fuselage Width	2
Wing Material	2
Engines	2
Wing Tank Cells	4
Centre Tank	2
ACTs	3
Trim Tank	2
Inerting System	2
Transfer System	3
Jettison System	2
Total Configurations:	55296

Table 8.1: Calculation of all possible configurations

the levels merged to generate the complete configurations of the system. The last step is the undesirability check where the configurations are not remove, they are however made inactive. The sequential reduction in configurations is shown in Table 8.2 in terms of number of configurations that were filtered out at each step, and the reduction percentage of desirable configurations remaining; compared to the previous step.

Imposing the Infeasibility constraints at each level separately (before the configurations of each level are combined), eliminates 41472 configurations bringing the total number down to 13824. The second set of infeasibility constraints, after the configurations of each level have merged, eliminates a further 5280, bringing the number down to 8544. Finally, imposing the undesirability constraints, 5092 configurations are made inactive, leaving 3452 feasible, desirable configurations to be further evaluated. With all constraints taken into consider-

Configurations	Total Number	% Reduction
All Possible Configurations	55296	-
Infeasible (constraints at each system level)	41472	75%
Infeasible (constraints after merging levels)	5280	38%
Undesirable	5092	60%
Feasible & Desirable Configurations:	3452	93.7%

Table 8.2: Reduction of configurations due to constraints

ation, the number of configurations was decreased by 93.7%.

From there on, the first stage creates the associated XML record with the details of each desirable configuration, and generates a CSV file with the respective bounds of each configuration for optimisation.

8.2 Stage 2 Results

After the first stage of the workflow is completed, the feasible and desirable configurations are retained for further evaluation. The configurations are first checked for active and dormant parameters based on the optimisation objectives, and get reduced to a consolidated list for optimisation.

After the optimisation process is complete, the configurations are then evaluated using weight and complexity penalties. Furthermore, they are assessed for cost and change propagation as well as the performance of the models using simulations.

8.2.1 Optimisation Cases

After the feasible and desirable configurations have been obtained, the process for consolidating the list for optimisation purposes begins.

The process was described in Section 4.2.3 while the relevant active and dormant parameters for this case study were shown in Figure 6.6 and Figure 6.7.

Using the active parameters allows the desirable configurations to be reduced from 3452 down to just 48 clusters of configurations to be optimised. The full list of clusters for optimisation is shown in Table 8.3.

LSLNY	LSLNN	HSLNY	HSLNN
LMLNY	LMLNN	HMLNY	HMLNN
LLLNY	LLLNN	HLLNY	HLLNN
LSHNY	LSHNN	HSHNY	HSHNN
LMHNY	LMHNN	HMHNY	HMHNN
LLHNY	LLHNN	HLHNY	HLHNN
LSLWY	LSLWN	HSLWY	HSLWN
LMLWY	LMLWN	HMLWY	HMLWN
LLLWY	LLLWN	HLLWY	HLLWN
LSHWY	LSHWN	HSHWY	HSHWN
LMHWY	LMHWN	HMHWY	HMHWN
LLHWY	LLHWN	HLHWY	HLHWN

Table 8.3: List of all configurations for optimisation

The optimisation IDs presented in Table 8.3 are the initials for each option for each active parameter. For example, LSLNY stands for (L)ow aspect ratio, (S)hort span, (L)ow Sweep Angle, (N)arrow Body and (Y)es to the presence of a centre tank. Any other configuration that shares those options will also share the same optimisation results.

8.2.2 Optimisation Results

All of the clusters presented in Table 8.3 are processed through the optimisation step. When the process concludes, the results can be visualised using parallel coordinates. Initially, the two extreme optimisation clusters, LSLNY and HLHWY are plotted as shown in Figure 8.1 with green and blue polylines respectively.

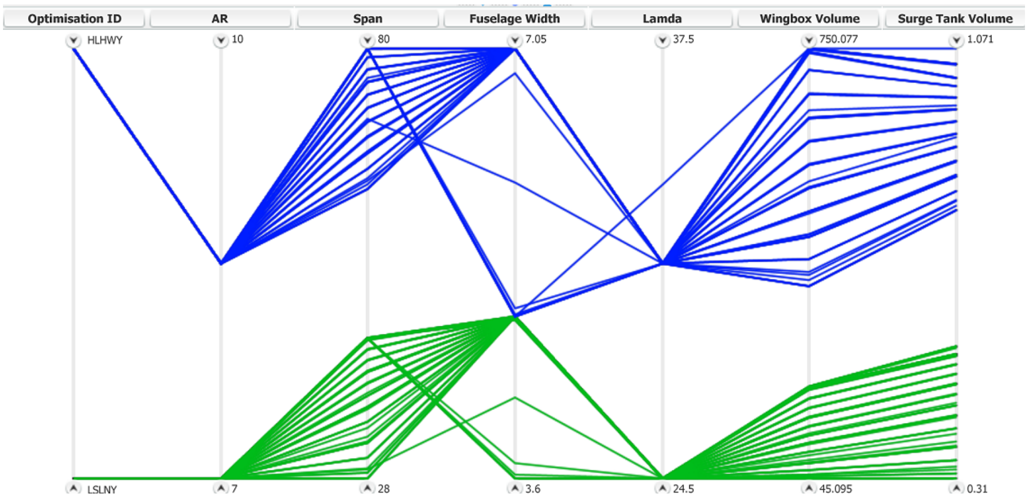


Figure 8.1: Optimisation results for LSLNY (green) and HLHWY (blue)

Some observations can be made immediately. First, the Aspect Ratio for all solutions is always at the lowest bound for each configuration. This is logical because for a constant span, the lower the Aspect Ratio value, the larger the wing area. Therefore, this would result in a bigger wingbox volume and, since span is constant in this case, it does not affect the length of the vent line and consequently the surge tank volume.

In order to examine the relationship between span and the two objectives, the axes of the parallel coordinates graph can be rearranged as required. Figure 8.2 shows the relationship between Span, Wingbox Volume, and Surge Tank Volume.

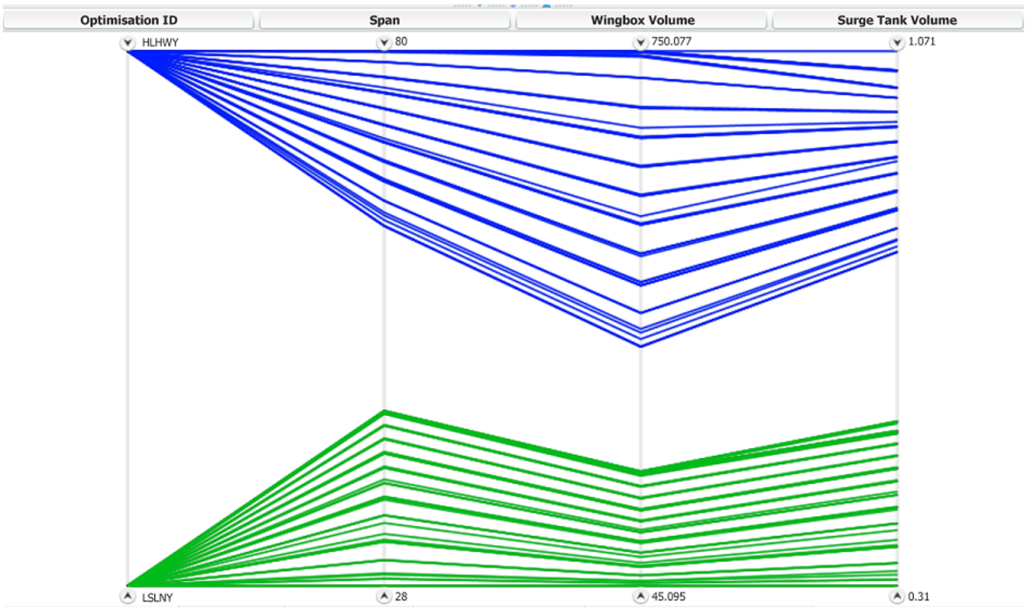


Figure 8.2: Relationship between span and the two optimisation objectives

The strong correlation between the Span and the two objectives becomes quite obvious from the Figure 8.2. Furthermore, this can be demonstrated using scatter plots, such as the one shown in Figure 8.3. The size of the points in the scatterplot are dependent on the size of the surge tank volume. Besides the wingbox volume, the surge tank volume also increases with span; this demonstrates the trade-off when dealing with competing objectives in multiobjective optimisation.

Plotting the optimisation objectives between them on a scatter plot allows the pareto front to be visualised as shown on Figure 8.4. The trade-off between the two objectives becomes even more evident.

The way that the results have been presented for the two extreme optimisation clusters can also be used for all configurations as shown in Figure 8.5. Long Span configurations are presented in blue, Medium Span in black, and Short Span in green.

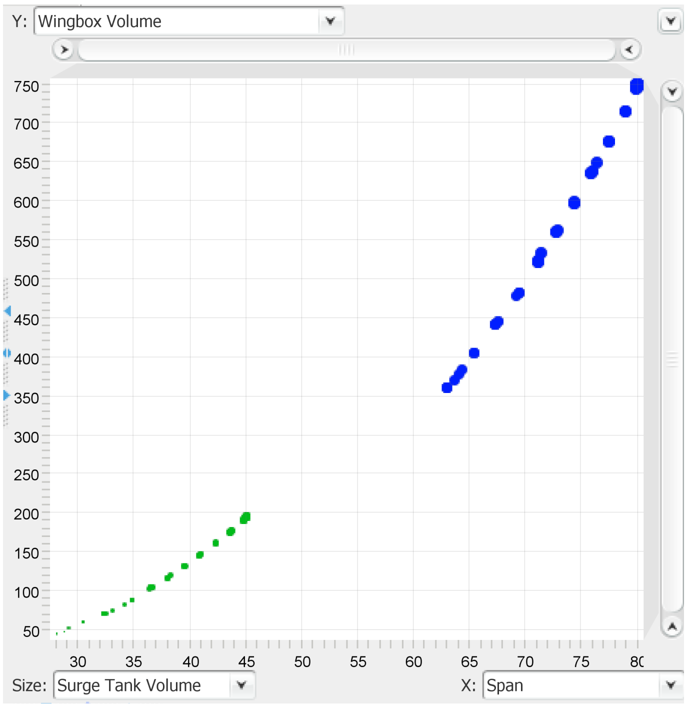


Figure 8.3: Scatter plot for Span versus the Wingbox Volume (relative sizes of the points indicate relative volume of surge tank)

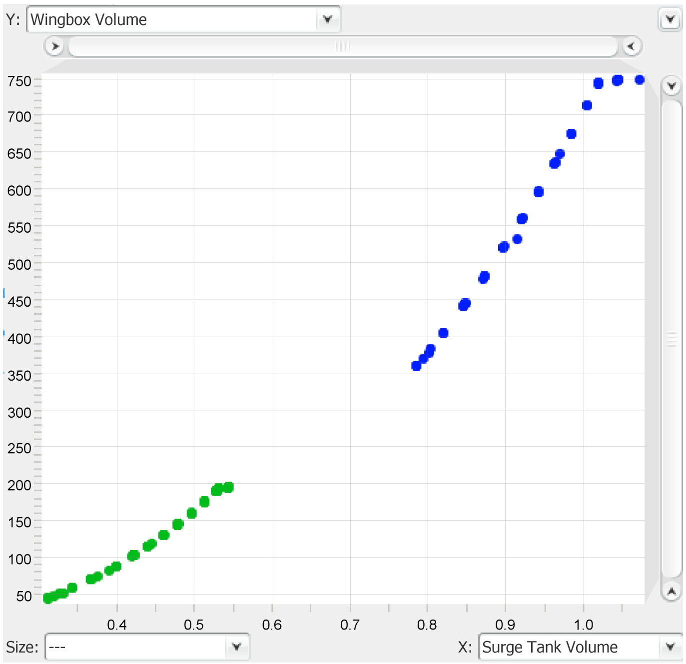


Figure 8.4: Pareto front for the two extreme optimisation clusters

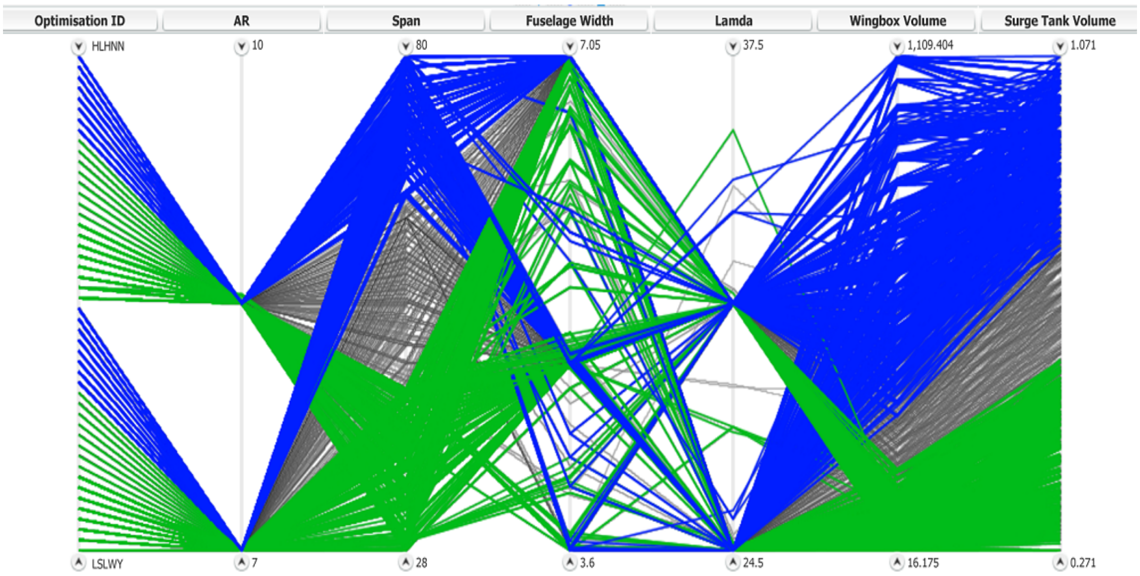


Figure 8.5: Parallel Coordinates graph for all optimisation results

8.3 Assessment and Evaluation Results

With just the optimisation results, all the configurations that share the same Optimisation ID (ie. belong in the same optimisation cluster) would yield the same results. The results of the optimisation process alone are insufficient for decision making. Further information is needed, therefore more metrics are introduced.

8.3.1 Complexity and Weight

Complexity and Weight penalties were introduced to assess and differentiate the competing configurations; especially in cases where configurations belonged in the same optimisation cluster. Summing up the penalties for each configuration, using Table 6.5, the results can be plotted again using the parallel coordinates graph.

Initially, the two extreme optimisation clusters are used to demonstrate the range of different penalties for different configurations. Figure 8.6 presents the complexity and weight penalties of all configurations that belong in the optimisation clusters LSLNY (Green) and HLHWY (Blue).

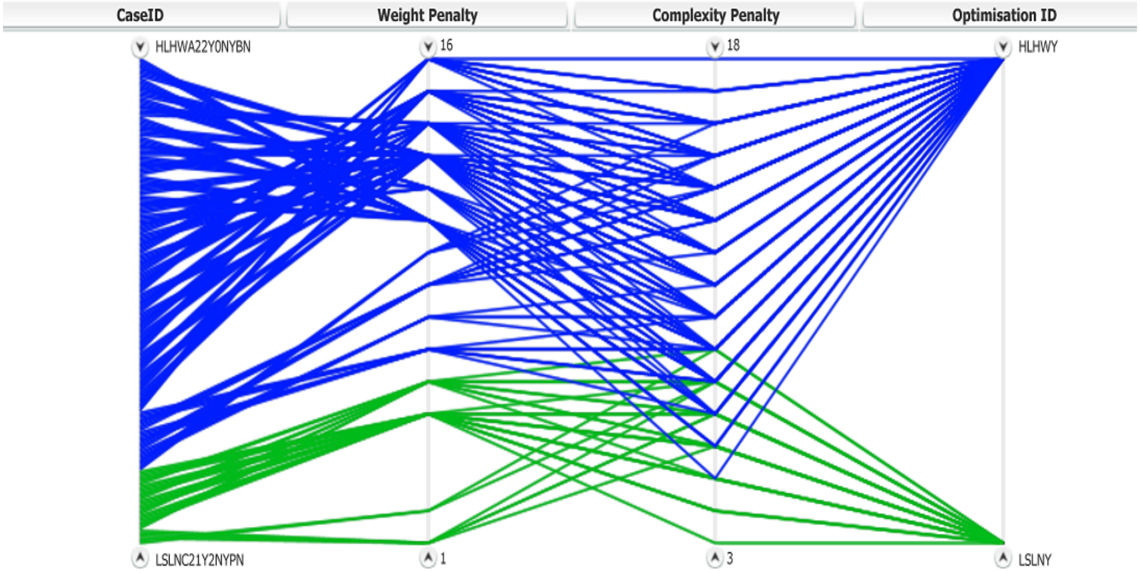


Figure 8.6: Parallel Coordinates graph for Complexity and Weight penalties of all configurations in the LSLNY and HLHWY optimisation clusters

Using the same approach, the penalties for all configurations can be plotted as shown in Figure 8.7. Furthermore, a scatter plot as shown in Figure 8.8 demonstrates an almost linear correlation correlation between Weight and Complexity penalties.

The penalties and the respective configurations IDs can now be appended to the results obtained from optimisation. Merging the two sets of data, generates the full results and non-dominated solutions for wingbox volume, surge tank volume, complexity penalties, and weight penalties for all desirable configurations. The complete results are presented in Figure 8.9.

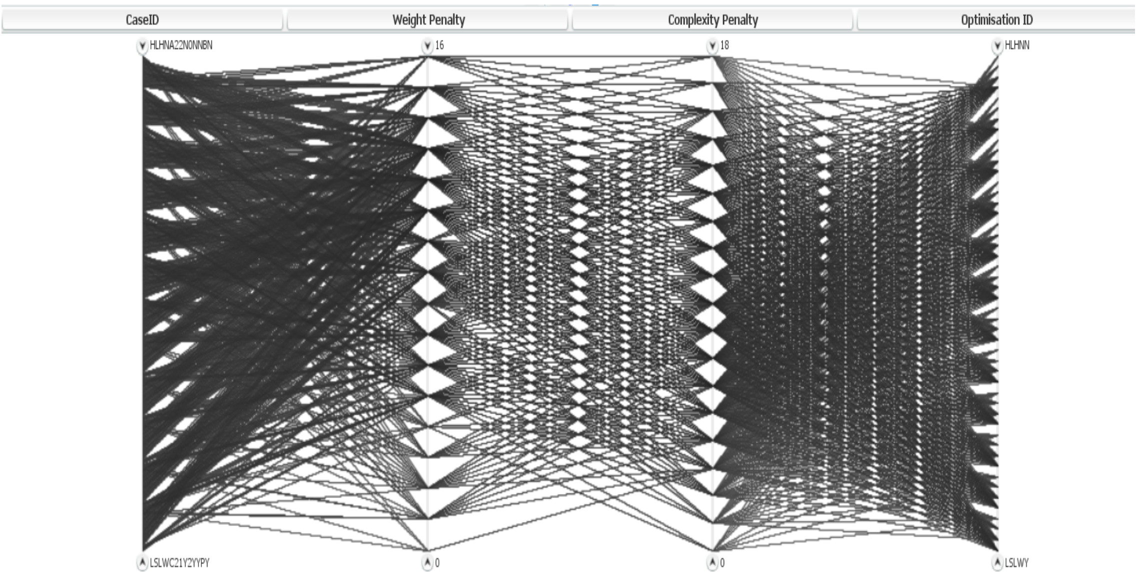


Figure 8.7: Parallel Coordinates graph for Complexity and Weight penalties for all configurations

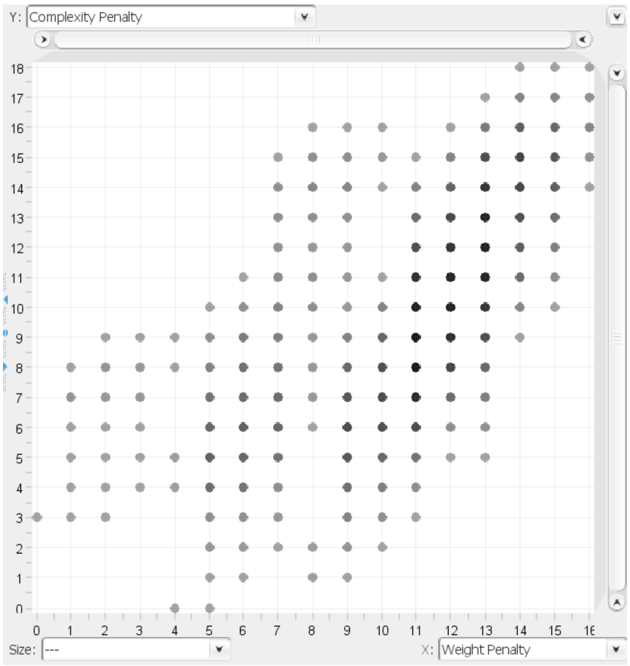


Figure 8.8: Scatter plot for Complexity penalties against Weight Penalties

Using the Parallel Coordinates plot to visualise the full set of results, a range of adjustments can be made to identify correlations, eliminate configurations, or highlight different sets. This is why the interactivity of parallel coordinates makes it a very powerful visualisation tool to assist in design decision-making.

One of the most likely scenarios would be to eliminate configurations that have high penalties, whether it is weight or complexity. Another elimination would be configurations that have both a low aspect ratio and a long span; as this would create an excessively large wing area. Finally, configurations that lie on the edges of the optimisation objectives can be eliminated. This is done to avoid extreme cases or risking the violation of constraints due to change in requirements and/or uncertainty in the data. The different filterings are presented in Figure 8.10.

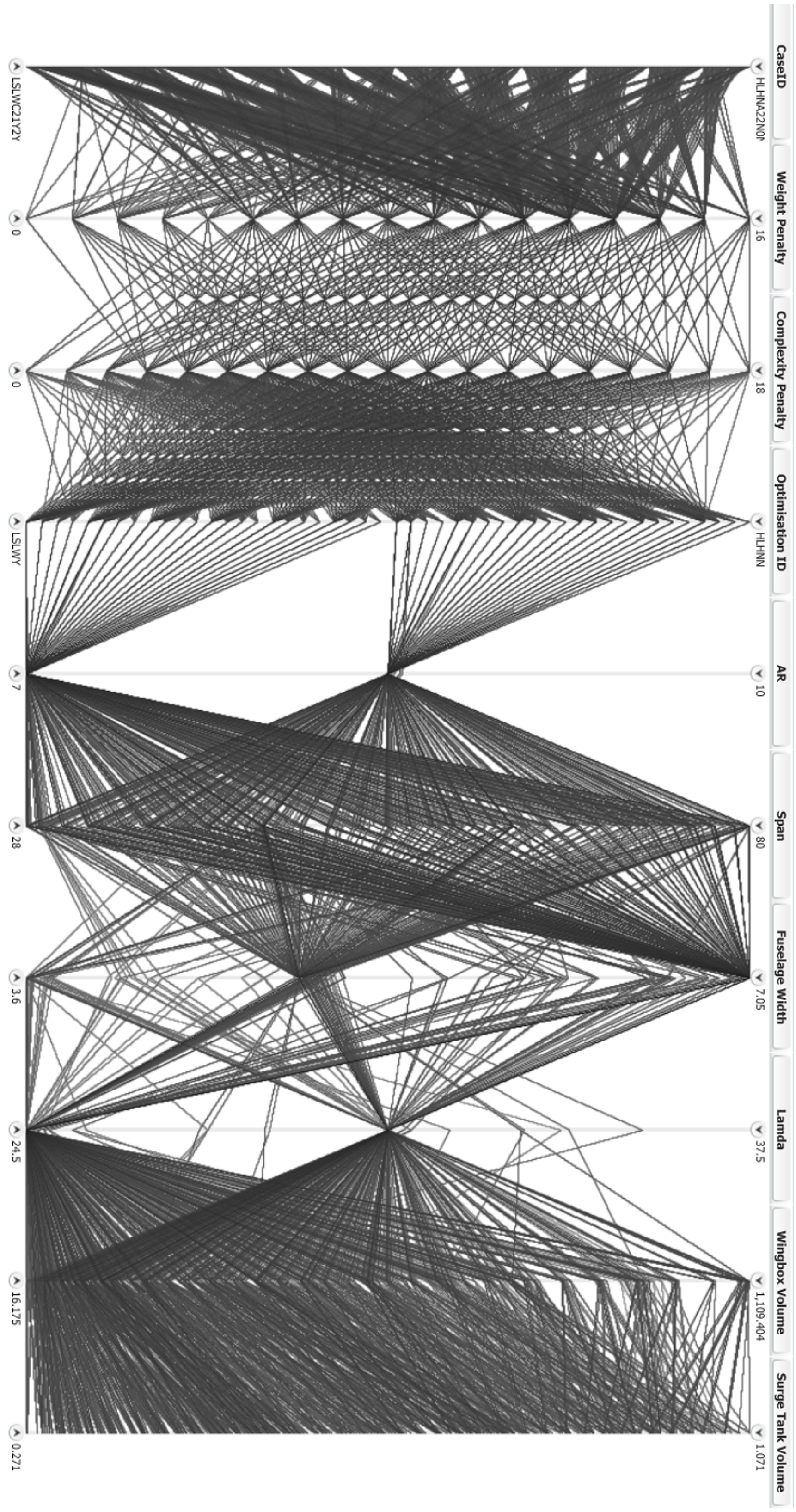
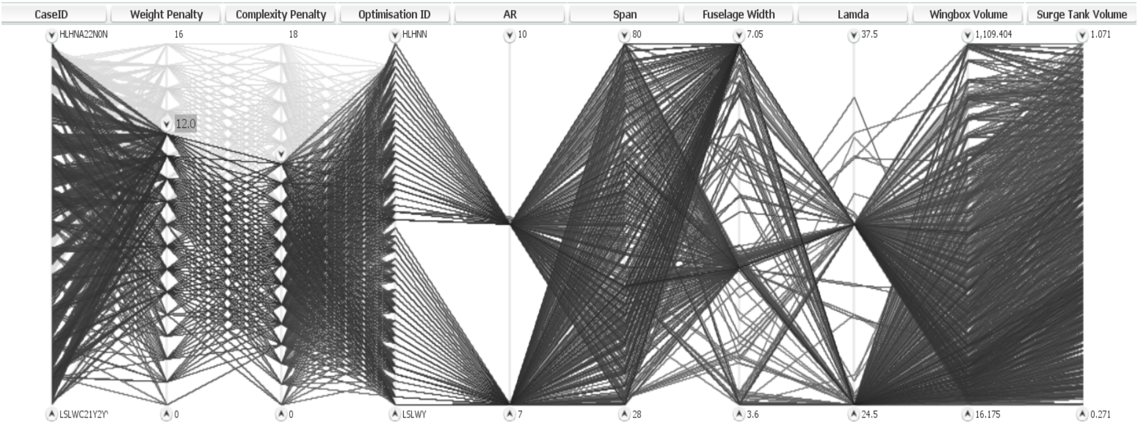
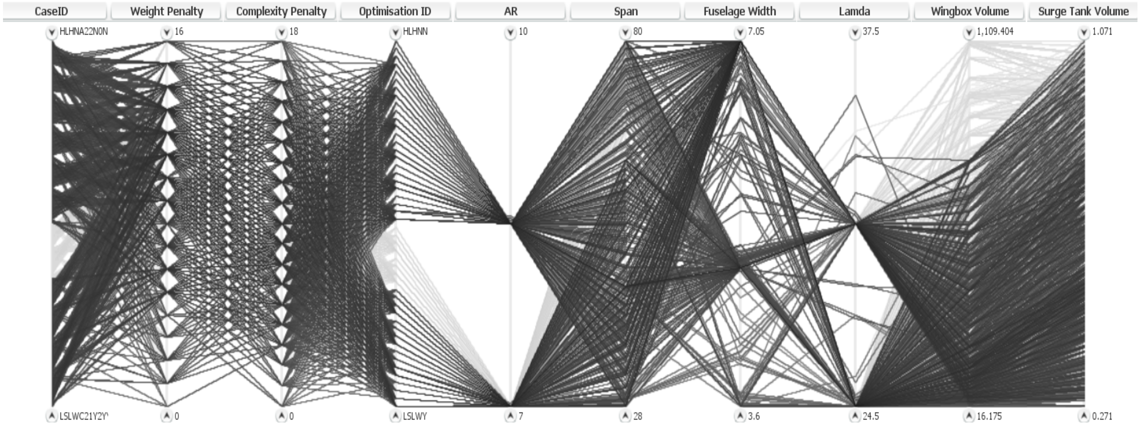


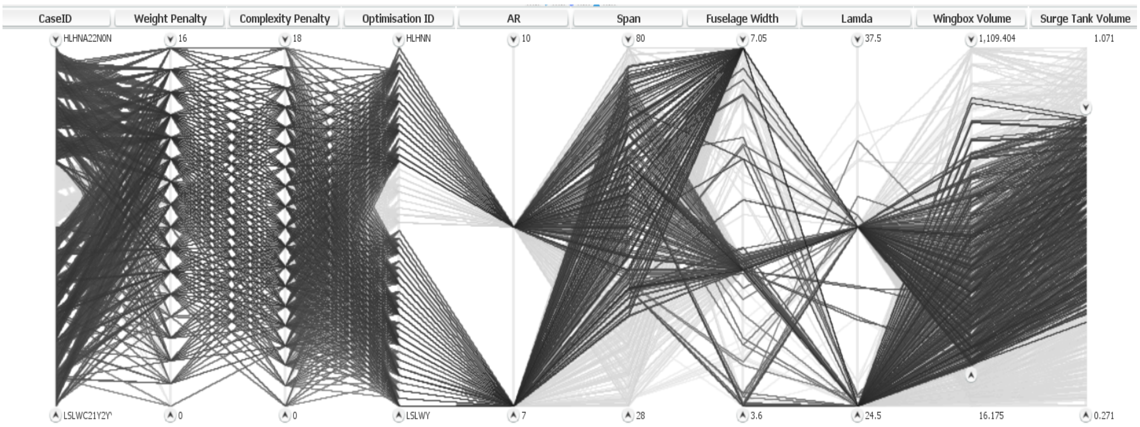
Figure 8.9: Parallel Coordinates graph for all Configurations including optimisation and penalties results



(a) Elimination of configurations with high complexity and high weight



(b) Elimination of configurations with Low aspect ratio and Long span



(c) Elimination of configurations with Small wingbox volume and Large surge tank volume

Figure 8.10: Different filtering applied to the full results

8.3.2 Performance

In order to assess the performance of the venting and inerting system, different scenarios were considered. Using configuration LSLNA21Y0NYBN, a maximum decent case was considered in order to demonstrate the performance of the two systems and compare it to a normal descend scenario. Configuration HLHWA23Y0NYBN on the other hand was used to evaluate the venting system in a maximum refuelling rate case.

Starting with the first configuration the venting and inerting systems were evaluated for the two descent cases using the data shown in Figure 6.13 and Figure 6.14. The venting system needs to be able to handle the large flow of air out of the tank ullages in order to avoid pressurisation of the tanks; the tanks are assumed to be empty. Figure 8.11 shows a comparison between the atmospheric pressure and the tank pressures over time for normal descend.

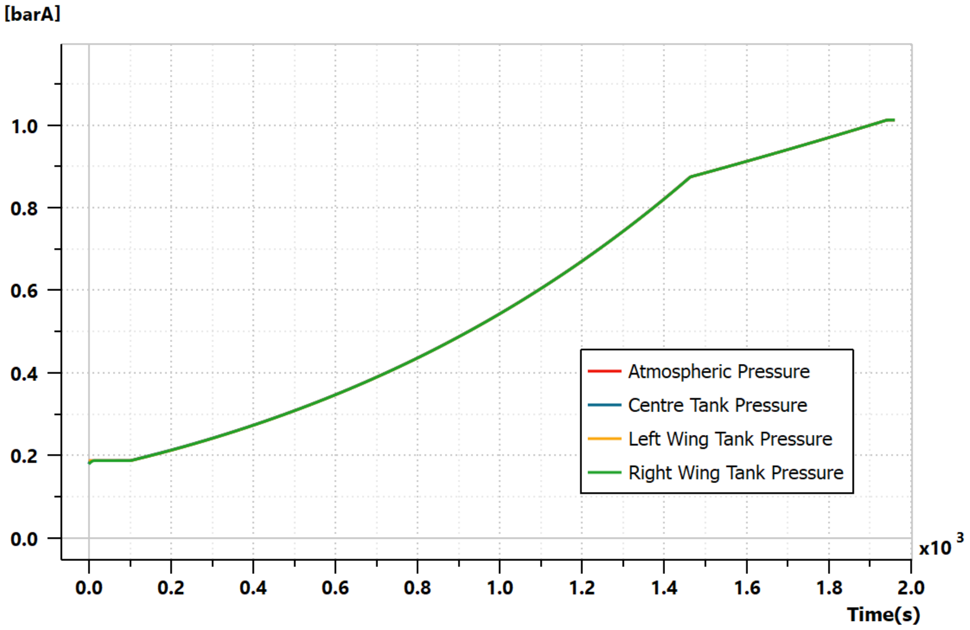


Figure 8.11: Pressure over time comparison for normal descend

Figure 8.11 shows that all lines overlap therefore at any given time the tank pressures equal the atmospheric pressure. This means that the venting lines can handle the air flow fast enough in order to avoid the tanks being pressurised. If there was a delay in the tank pressures reaching the atmospheric pressure that would mean that the tanks are becoming pressurised since the venting lines can't manage the airflow fast enough; in which case, a larger-diameter pipe would be required.

When the maximum descend case is considered, as shown in Figure 8.12, the pressures seem to be equal between the atmosphere and the tanks.

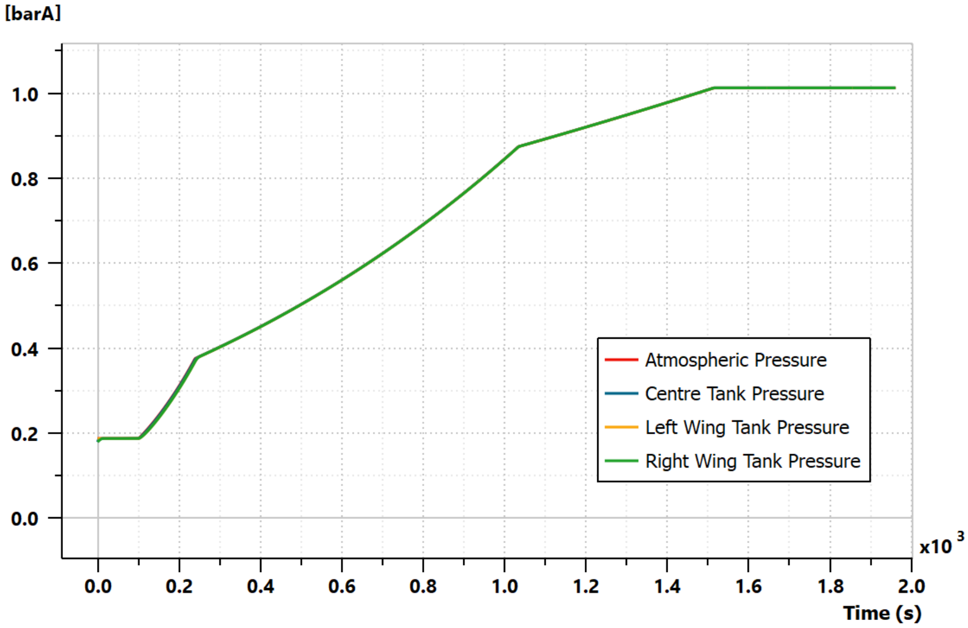


Figure 8.12: Pressure over time comparison for maximum descend case

However upon closer inspection during the maximum descend rate interval there seems to be an offset between the pressures as shown in Figure 8.13.

Even though the difference is not substantial, it does show that there is a delay in equalising the pressure between the tanks and the atmosphere; the de-

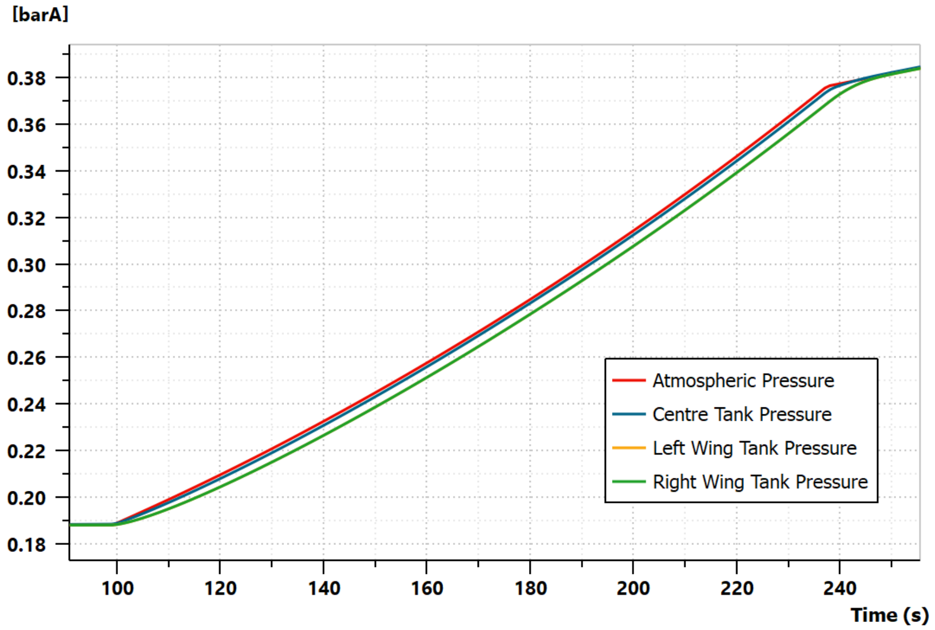


Figure 8.13: Pressure difference for maximum descend rate interval

lay is approximately 3.5 seconds. This causes minor pressurisation in the tanks but as soon as the rate of descend decreases, the pressures are equalised almost immediately.

There is also a difference between the wing tank pressures and the centre tank pressure. This is due to the centre tank having an additional flow of air due to the inerting system operating at maximum rate.

The inerting system needs to be able to maintain low levels of oxygen in the centre tank. During the rapid descent case this becomes challenging as a large portion of the injected air will be flowing out of the tank. Figure 8.14 shows the nitrogen concentration % in the centre tank for a normal descend case and for the maximum descend case.

As expected, in a maximum descend case, the inerting system takes a longer time to reach a 95% nitrogen concentration in the centre tank. In a normal de-

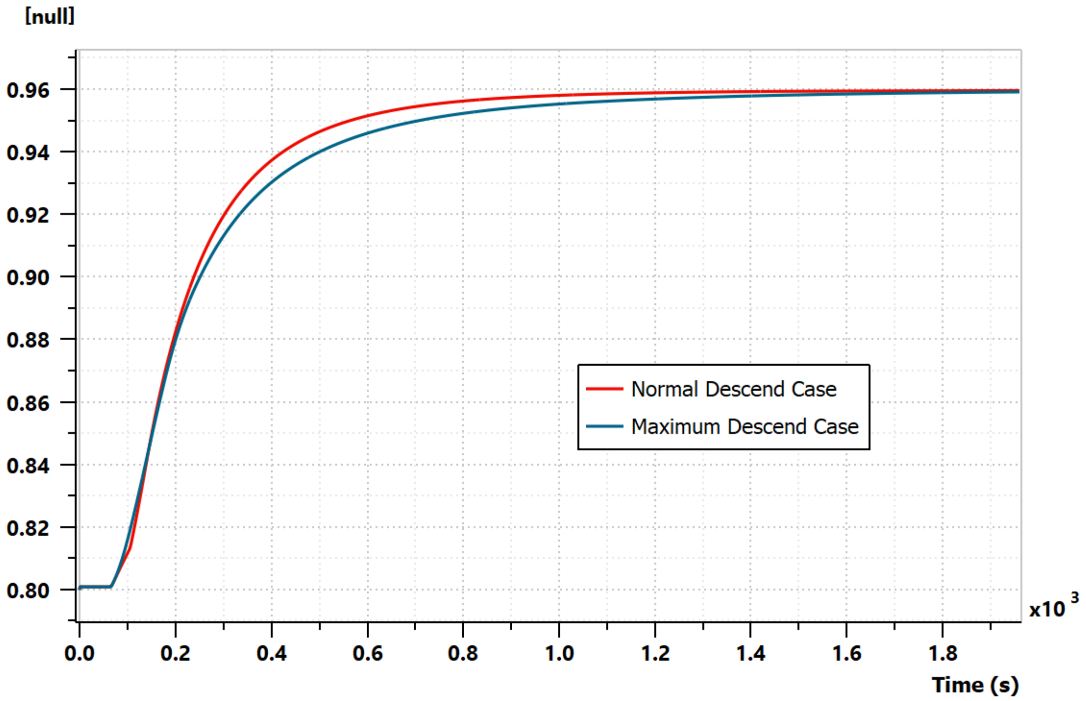


Figure 8.14: Nitrogen concentration in centre tank for the two scenarios

scend case the tank reaches a 95% nitrogen concentration at 561 seconds while in the maximum descend the 95% is reached at 705 seconds; a difference of more than 2 minutes.

However, since a tank is considered as inert if the oxygen concentration is less than 12%, the difference between the two cases for reaching 88% in Nitrogen concentration is only 4 seconds; which is negligible.

The last scenario is the refuelling case as presented in Figure 6.12. Using a constant refuelling rate of 6000 L/min, each of the tank is filled up to 95%. Figure 8.15 shows the fuel volume over time for the centre tank and one of the wing tanks; all the wing tanks share the same characteristics, therefore the refuelling rate is identical.

The aim of the refuelling case is to evaluate if the venting system can handle

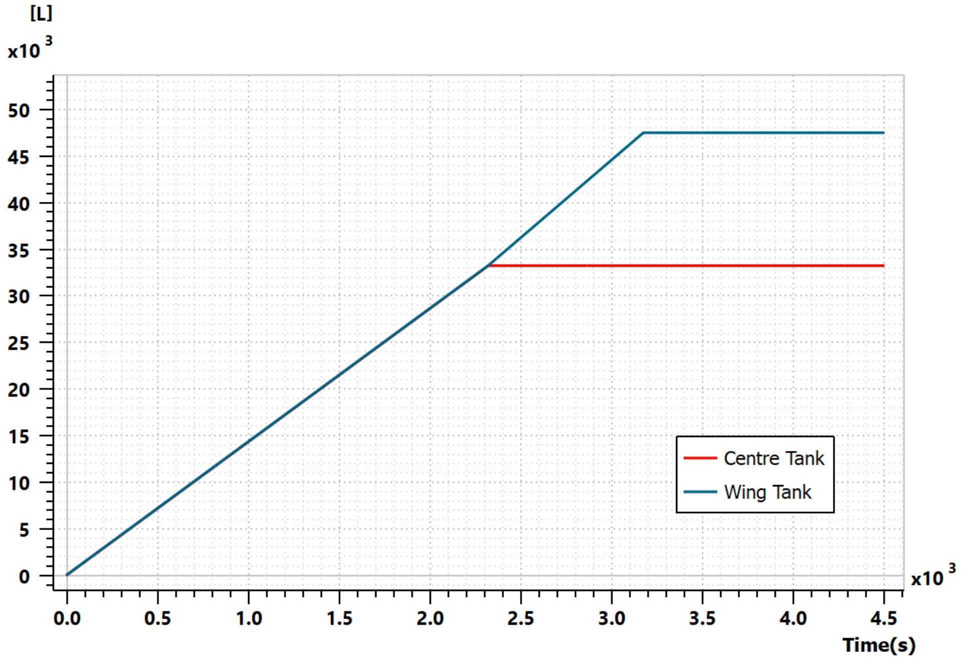


Figure 8.15: Fuel volume over time for the centre tank and a wing tank

the displaced air from the ullage as the tank fills up. Figure 8.16 shows a comparison between the ullage pressure of the centre tank and the pressure in the wing tank; the atmospheric pressure is also visualised for comparison.

The first observation that becomes apparent from Figure 8.16 is the immediate increase in the ullage pressure as soon as the refuelling begins. That difference is maintained throughout without increasing which means that the venting system displaces air fast enough to counterbalance the inflow of fuel. The pressure difference caused by the initial inflow of fuel is minor; almost negligible.

As soon as the centre tank is at 95% the inflow valve shuts completely, redirecting all fuel flow to the wing tanks. This explains both:

1. The rate increase of fuel volume in Figure 8.15 for the wing tanks.

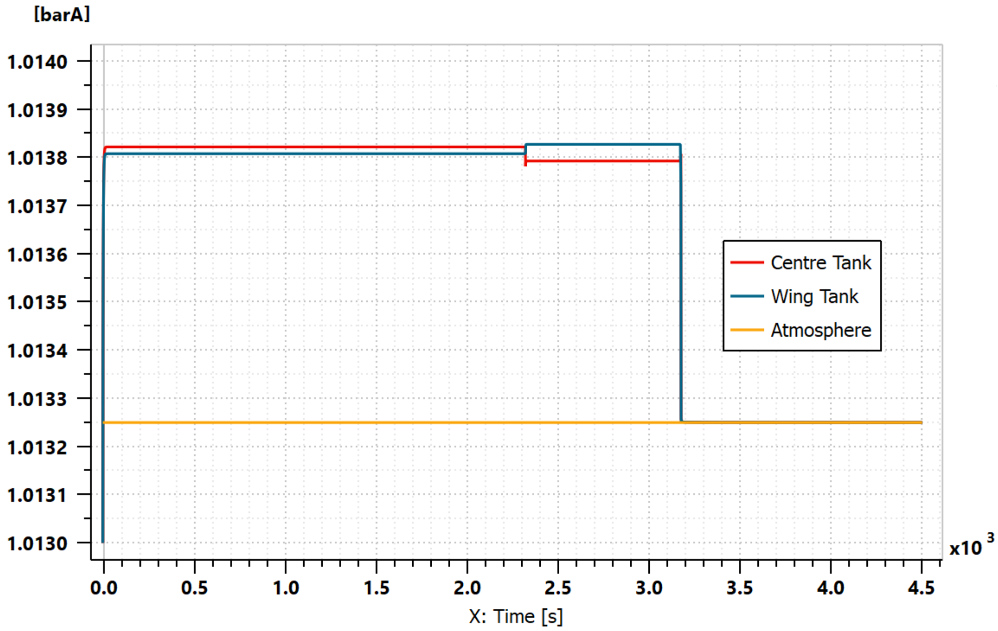


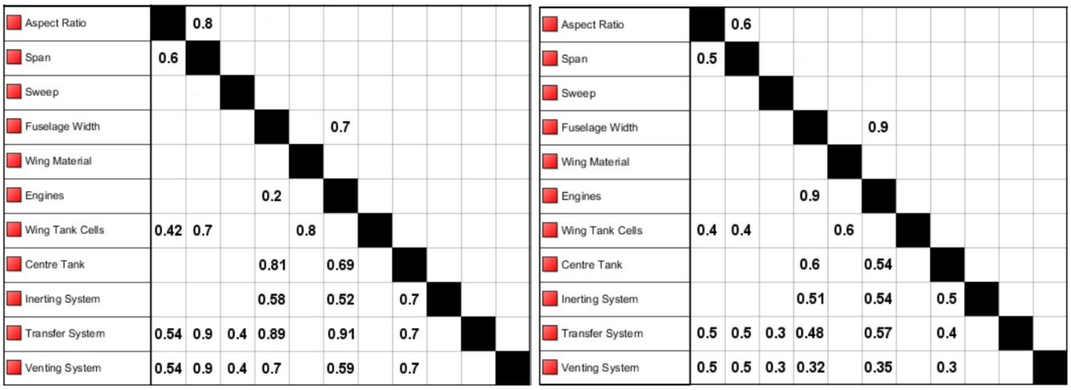
Figure 8.16: Ullage pressure over time for the centre tank and a wing tank

2. The initial pressure increase in the wing tanks due to the additional fuel flow as shown in Figure 8.16.

As soon as the centre tank valve is shut, there is a pressure drop in the ullage due to air being displaced, without any fuel flowing inwards. However, the pressure can't yet be equalized with the atmosphere due to the pressure increase in the ullage of the other tanks. Since they share the same venting line, the higher pressure of the wing tanks restricts the equalisation of the centre tank.

8.3.3 Change and Cost Propagation

Using the reduced DSM shown in Figure 6.15 and its quantified instances shown in Figure 6.16 the change propagation can be calculated. For this example, only four steps of change propagation are considered. Cambridge Advanced Modeller is utilised at this stage to calculate the combined values for Impact, Likelihood, and Risk, as well as to visualise the critical path networks. Figure 8.17 presents the quantified instances of the DSM with values for Combined Likelihood and Combined Impact after four steps of change propagation.



🟥 Aspect Ratio	■	0.8									
🟥 Span	0.6	■									
🟥 Sweep			■								
🟥 Fuselage Width				■		0.7					
🟥 Wing Material					■						
🟥 Engines				0.2		■					
🟥 Wing Tank Cells	0.42	0.7			0.8		■				
🟥 Centre Tank				0.81		0.69		■			
🟥 Inerting System				0.58		0.52		0.7	■		
🟥 Transfer System	0.54	0.9	0.4	0.89		0.91		0.7		■	
🟥 Venting System	0.54	0.9	0.4	0.7		0.59		0.7			■

(a) Likelihood/Probability

🟥 Aspect Ratio	■	0.6									
🟥 Span	0.5	■									
🟥 Sweep			■								
🟥 Fuselage Width				■		0.9					
🟥 Wing Material					■						
🟥 Engines				0.9		■					
🟥 Wing Tank Cells	0.4	0.4			0.6		■				
🟥 Centre Tank				0.6		0.54		■			
🟥 Inerting System				0.51		0.54		0.5	■		
🟥 Transfer System	0.5	0.5	0.3	0.48		0.57		0.4		■	
🟥 Venting System	0.5	0.5	0.3	0.32		0.35		0.3			■

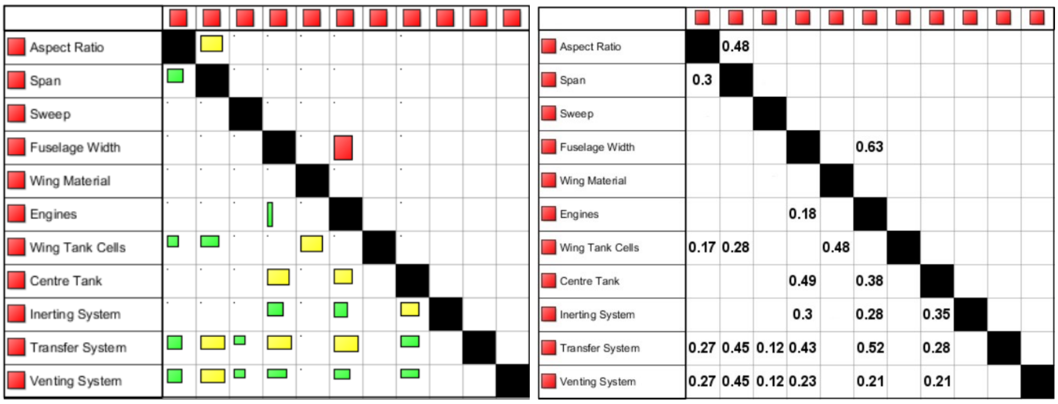
(b) Impact

Figure 8.17: Combined Likelihood and Combined Impact after four steps of change propagation

To calculate the Combined Risk value, the process is the same with the direct values; a multiplication between the Likelihood and Impact of each connection. This results in another quantified DSM with the values of Combined Risk at each connection.

However, for larger systems, having number in a DSM can get confusing. For this reason the connections can be colour-coded using rectangles. The width of the rectangle represents the Likelihood while the height represents the Impact. Consequently this means that the bigger the area of the rectangle, the higher the Risk which is also represented in colour codes: Green for low Risk, Yellow for medium Risk, and Red for high Risk. Both the colour-coded and the numerical representations are shown in Figure 8.18.

The first thing that becomes apparent by the resulting DSMs is that now



(a) Colour Coded Risk

	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Aspect Ratio	■	🟨									
Span	🟩	■									
Sweep			■								
Fuselage Width				■		🟥					
Wing Material					■						
Engines				🟩		■					
Wing Tank Cells	🟩	🟩			🟨		■				
Centre Tank				🟨		🟨		■			
Inerting System				🟩		🟩		🟨	■		
Transfer System	🟩	🟨	🟩	🟨		🟨		🟩		■	
Venting System	🟩	🟨	🟩	🟩		🟩		🟩			■

(b) Numerical Risk

	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Aspect Ratio	■	0.48									
Span	0.3	■									
Sweep			■								
Fuselage Width				■		0.63					
Wing Material					■						
Engines				0.18		■					
Wing Tank Cells	0.17	0.28			0.48		■				
Centre Tank				0.49		0.38		■			
Inerting System				0.3		0.28		0.35	■		
Transfer System	0.27	0.45	0.12	0.43		0.52		0.28		■	
Venting System	0.27	0.45	0.12	0.23		0.21		0.21			■

Figure 8.18: Combined Risk

there are connections between elements that were not there previously. This arises due to indirect propagations, ie changes that pass through intermediate components.

The Transfer system and the Venting system are affected by most of the other elements of the system either directly or indirectly. However, they do not affect any element themselves which makes them Absorber elements; that is, elements that are affected by changes, but do not propagate the change. Absorber elements can therefore be designed at later stages since any design changes on them won't have any effect on other elements. On the same note, since they are affected by so many other elements, the absorber elements will most probably have to undergo design changes therefore leaving them for later stages avoids multiple redesigns.

On the other hand, elements such as the Engines, that affect a lot of other components but are not affected themselves, are considered change emitters. The design of such elements should be brought forward so that their design can be frozen earlier in order to avoid changes propagating at later stages.

Information such as the above, can drastically help the design process of a system where a sequential process needs to be followed. Using critical path networks, the propagation paths between two elements of the system can also be visualised. Figure 8.19 shows an example of the possible propagation paths initiating from the Engines and reaching the Transfer System.

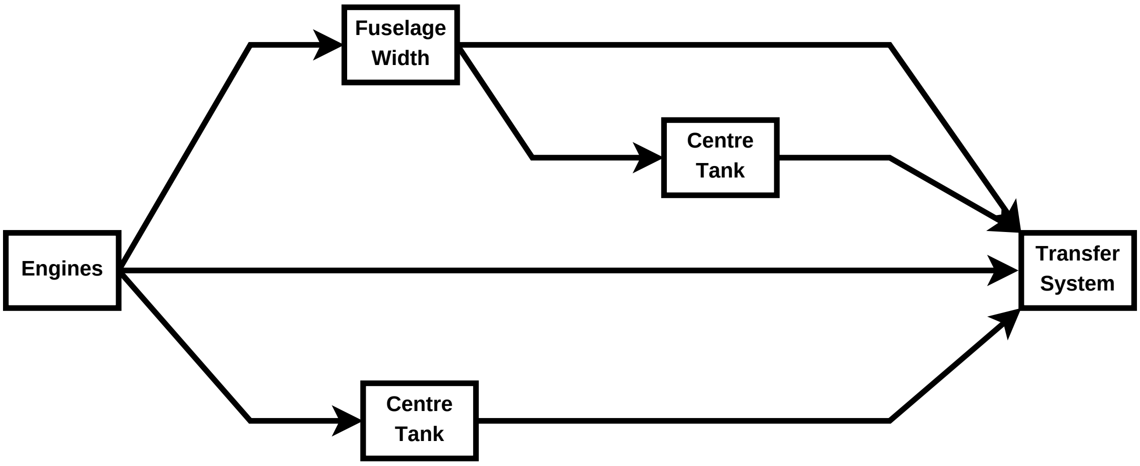


Figure 8.19: Change propagation paths from Engines to Transfer System

However, as previously mentioned, change propagation only tells part of the story. A complete redesign of a cheap element might be a better option than a partial redesign of a very expensive element. Applying the CP^2 method, the cost associated with each path of propagation can be calculated and visualised.

Using the example from Figure 8.19, there are four propagation paths from the Engines to the Transfer System. From the top path to the bottom one the paths are:

1. Indirect (Second order) Through the Fuselage Width
2. Indirect (Third order) Through the Fuselage Width and the Centre Tank
3. Direct (No intermediate elements)
4. Indirect (Second order) Through the Centre Tank

Using the equations for CP^2 the probabilities and costs associated with each path can be calculated. Using the top path (Number 1) as an example, the path probability is calculated using Equation 4.1 which is the product of all direct probabilities between elements. In this case the direct probabilities are between the Engines and Fuselage width (0.7) and between the Fuselage Width and the Transfer System. Therefore:

$$pp_1 = 0.7 \times 0.7 = 0.49 \tag{8.1}$$

Following the same procedure above, the remaining path probabilities are:

$$\begin{aligned} pp_2 &= 0.7 \times 0.8 \times 0.7 = 0.392 \\ pp_3 &= 0.7 \\ pp_4 &= 0.3 \times 0.7 = 0.21 \end{aligned}$$

The combined probability between the initiating and receiving element has already been calculated as ”Combined Likelihood” using the CPM method above. In this example, following the DSM in Figure8.17, the combined Probability is 0.91.

To calculate the associated cost for each path, Equation 4.3 is used where the total cost of each path is the sum of the cost weights of the elements. Using Table 6.6, for path 1, the costs are: Engines(0.8), Fuselage Width(0.9), and Transfer System (0.4), Therefore:

$$pc_1 = 0.8 + 0.9 + 0.4 = 2.1 \tag{8.2}$$

For the remaining paths:

$$pc_2 = 0.8 + 0.9 + 0.4 + 0.4 = 2.5$$

$$pc_3 = 0.8 + 0.4 = 1.2$$

$$pc_5 = 0.8 + 0.4 + 0.4 = 1.6$$

Finally, the aggregate cost between the initiating and receiving element following Equation 4.4:

$$Ac = \frac{(0.49 \times 2.1) + (0.392 \times 2.5) + (0.7 \times 1.2) + (0.21 \times 1.6)}{0.91} = 3.5 \tag{8.3}$$

Now that all associated values have been calculated, the results can be plotted using parallel coordinates as shown in Figure 8.20.

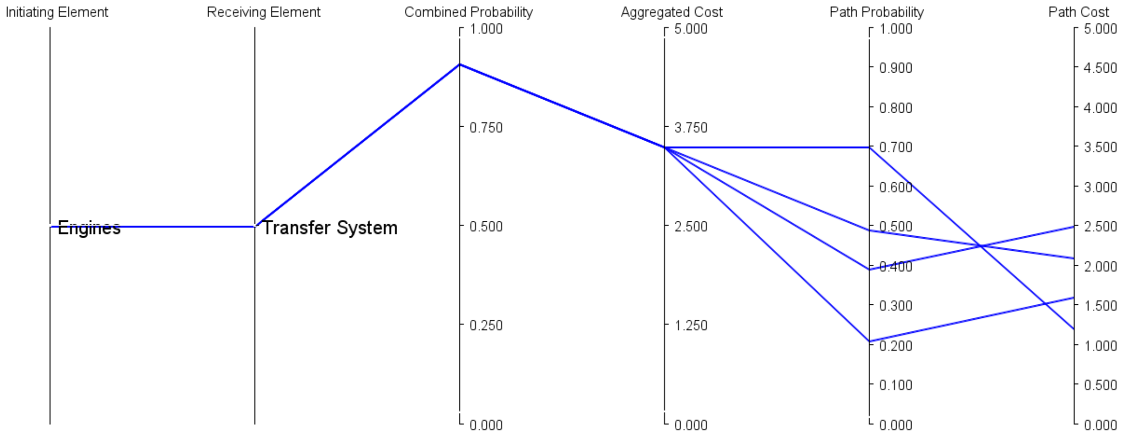


Figure 8.20: Change Propagation paths between Engines and Transfer system using Parallel coordinates

Figure 8.20 demonstrates the propagation paths using polylines to indicate the respective values of each path. The first two axes represent the initiating and receiving elements, followed by the combined probability and aggregated cost between the two. Only one line is visible up to here since there is only one value for Combined Probability and Aggregated Cost regardless of the number of paths between two elements. The one line expands however to a number of lines depending the number of propagation paths, in this case 4. Each line shows the respective probability and cost value for each path.

Following the process above, all of the values for Path Probability, Path Cost, and Aggregated Cost can be calculated for any combination of initiating and receiving elements; the Combined Probability has already been calculated using CPM. The results are added to the initial parallel coordinates presented in Figure 8.20 to present all paths of propagations between all elements of the system. The resulting visualisation is presented in Figure 8.21.

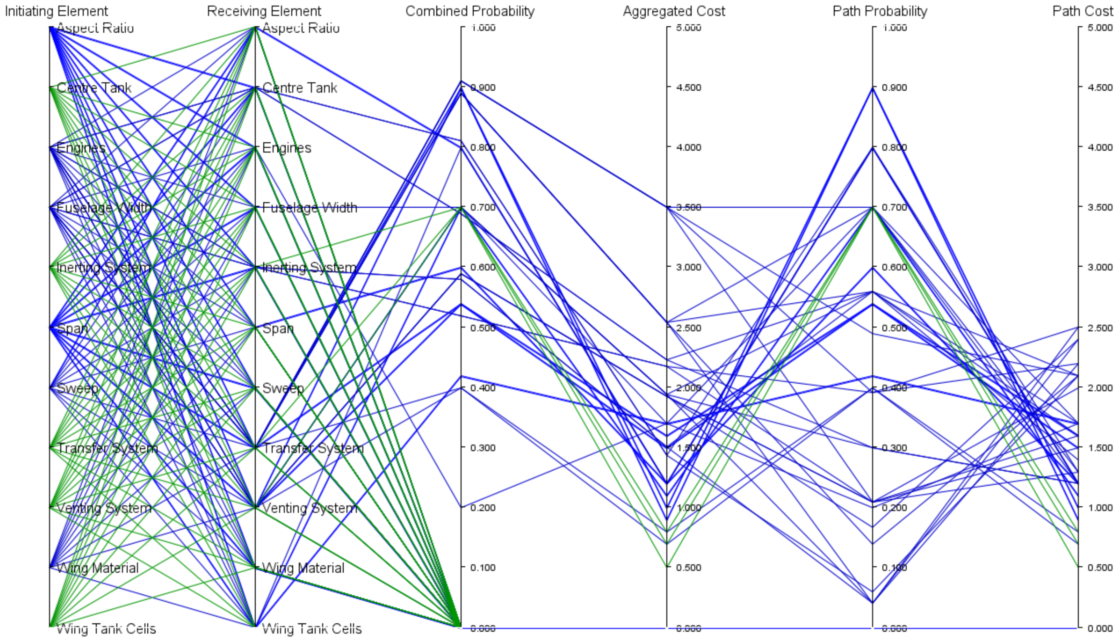


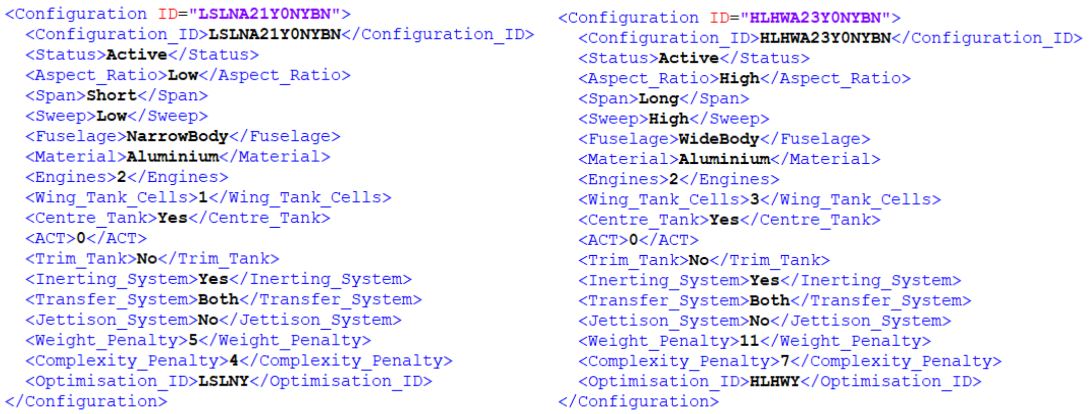
Figure 8.21: Parallel Coordinates graph for Cost and Change propagation of the whole system

In Figure 8.21 Blue lines represent changes that initiate from an Aircraft Level element, while green lines represent changes that initiate from a System Level element. The visualisation comes to reaffirm the fact that most design change propagations are initiated by Aircraft Level elements; since there are more blue lines than green.

Furthermore, the ones that do initiate from an Aircraft Level element have usually higher Combined Probability and higher Aggregated Costs. Aggregated values, as mentioned before, might cause important information to be lost. However, they help in identifying which elements will probably result in higher costs, so that they can be studied in more detail; by examining each propagation path originating from them.

8.4 Traceability & Processing of Results

After the workflow has been completed and the final results have been obtained, the information generated can be further examined. In order to manage the configurations effectively, the XML records of the desirable configurations needs to be updated to include the optimisation IDs and the associated penalties. Figure 8.22 shows the records for the two configurations that were examined in more detailed.



```xml
<Configuration ID="LSLNA21Y0NYBN">
  <Configuration_ID>LSLNA21Y0NYBN</Configuration_ID>
  <Status>Active</Status>
  <Aspect_Ratio>Low</Aspect_Ratio>
  <Span>Short</Span>
  <Sweep>Low</Sweep>
  <Fuselage>NarrowBody</Fuselage>
  <Material>Aluminium</Material>
  <Engines>2</Engines>
  <Wing_Tank_Cells>1</Wing_Tank_Cells>
  <Centre_Tank>Yes</Centre_Tank>
  <ACT>0</ACT>
  <Trim_Tank>No</Trim_Tank>
  <Inerting_System>Yes</Inerting_System>
  <Transfer_System>Both</Transfer_System>
  <Jettison_System>No</Jettison_System>
  <Weight_Penalty>5</Weight_Penalty>
  <Complexity_Penalty>4</Complexity_Penalty>
  <Optimisation_ID>LSLNY</Optimisation_ID>
</Configuration>
```

(a) LSLNA21Y0NYBN

```xml
<Configuration ID="HLHWA23Y0NYBN">
  <Configuration_ID>HLHWA23Y0NYBN</Configuration_ID>
  <Status>Active</Status>
  <Aspect_Ratio>High</Aspect_Ratio>
  <Span>Long</Span>
  <Sweep>High</Sweep>
  <Fuselage>WideBody</Fuselage>
  <Material>Aluminium</Material>
  <Engines>2</Engines>
  <Wing_Tank_Cells>3</Wing_Tank_Cells>
  <Centre_Tank>Yes</Centre_Tank>
  <ACT>0</ACT>
  <Trim_Tank>No</Trim_Tank>
  <Inerting_System>Yes</Inerting_System>
  <Transfer_System>Both</Transfer_System>
  <Jettison_System>No</Jettison_System>
  <Weight_Penalty>11</Weight_Penalty>
  <Complexity_Penalty>7</Complexity_Penalty>
  <Optimisation_ID>HLHWY</Optimisation_ID>
</Configuration>
```

(b) HLHWA23Y0NYBN

Figure 8.22: XML Records for the two extreme cases

The updated XML records the current status of each configuration. Additionally, they now have the respective penalties included, as well as the optimisation cluster ID for each one. Using the full configuration ID (instead of the consolidated optimisation one) the respective simulation models and change propagation data can be traced.

The approach has generated a wealth of information which was presented in a number of ways. Parallel coordinates were extensively used in order to visualise multidimensional data in the stages of optimisation, penalties and cost propagation. The interactivity of Parallel Coordinates allowed for the data to be manipulated in order to generate useful knowledge.

The optimisation results assisted in identifying the non-dominated variants for each configuration. Using selection and filtering also allowed for the elimination of excessively high wing area configurations, as well as ones that were close to the bounds and might be considered as non-robust options. Interactivity would also allow to explore alternative options in the case where parameter bounds change, or become more constrained.

However, the optimisation results alone would have been restrictive for a thorough assessment due to a large number of configurations sharing the same active design parameters; hence the same optimisation results. The addition of penalties to the optimisation results, enabled a more detailed assessment of configurations; especially ones that share common active parameters. Even if the penalties are entirely empirical, they can be useful for the early stages where a rapid assessment and elimination is required before moving to more detailed stages.

In addition to the trade-off studies and comparisons between configurations, each one can be further evaluated individually. Starting by the simulation models, the configurations were put through different scenarios in order to study the performance of major subsystems. Using analytical calculations for the optimisation and 1D simulations allowed the process to run in a short amount of time. The simulations can help assess whether a configuration meets the expected performance requirements, but can also assist in further improving them. For example, pipe diameters can be reduced in order to reduce their weight; as long as they can still meet the performance and safety requirements.

Each configuration can be further assessed by examining its sensitivity to design changes, as well as the associated costs. Using the already established CPM method, the combined risks can be calculated and visualised using DSMs and critical path networks. With the newly-developed CP^2 the associated costs can also be calculated, with each of the aggregated and path costs visualised using parallel coordinates. Configurations that exhibit DSMs with densely populated connections will result in a higher probability of changes propagating easier through the system; the costs associated with them also increases depending on the affected elements.

The pSeven-implemented workflow took 14 minutes to be executed from start to finish, for all configurations. Considering the number of different stages, the different programming languages used, and the large number of configurations that were initially considered, the running time was kept short. However this was primarily due to being a completely analytical process for a high level view of the overall case study. On the other hand, a number of different areas have been identified that can improve the efficiency of the process; those areas are outlined in the final Part of this thesis.

Part IV

Discussion, Future Work and Conclusion

Chapter 9

Discussion

The aim of this project was to investigate how a thorough design space exploration of complex systems, with high dimensionality, can be carried out in a fast and computationally efficient process. At the same time, the desirability and feasibility of the configurations that are selected and matured during the process, needs to be ensured. This task becomes even more challenging for the early stages of design where a higher degree of uncertainty is present and the design space is less constrained.

These challenges gave rise to a number of research questions that this work aimed at addressing, as presented in Part I of this thesis. In order to answer those questions sufficiently, and address the problem stated, certain aims and objectives were set.

The first stage of this project was to carry out a thorough literature review on the, initially two, main areas of interest: Set-Based Design and optimisation methods. The background research presented the origins and latest developments in the respective fields, but also identified gaps in literature. Furthermore, investigating the two areas in depth allowed for a cross comparison between them, outlining their main differences, but also identifying how they can be integrated in a unified process where they can complement each other.

During the literature review and background research on Set-Based Design and optimisation, a number of additional areas, that were deemed worthy of further exploration were identified. Numerous tools used for engineering design were assessed with DSMs being the more applicable and more useful to this project. Not only were they used for modelling the network and the connectivity between the elements of the system at hand, but also for modelling, predicting, and calculating engineering change propagations.

On the topic of predicting how changes propagate through a system, the different methods previously developed were examined. Even though CPM was deemed as the most useful one for this work, there was the limitation that cost was not considered. Therefore a new method was developed, CP^2, that takes into consideration the cost weight of each element.

Integrating all of the above would result in generating large sets of data and information which posed two issues. The first is how can the information generated be presented and visualised in a meaningful way to the relevant user. A range of different visualisation tools were considered, depending on the type of data that required to be visualised. DSMs proved to be suitable for modelling the network of the system as well as the CPM results. Furthermore, the results that arise from CPM also employed critical path networks to show the change propagation paths. Scatter plots were initially considered for visualising trade-offs between two sets of data such as in the case of a pareto front. However since most of the applications studied dealt with multidimensional data, parallel coordinates was deemed as the most suitable method. Parallel coordinates were therefore employed not only to visualise the results arising from optimisation but also as part of CP^2 to visualise the associated costs with each path of a change propagating through the system.

The second issue arising from the large data generation is with regards to storing, managing, and tracing the resulting information. Since the approach aimed at being tool independent, XML and CSV files were selected as the most appropriate filetypes for storing data. The reasoning behind it is that they are platform and interface agnostic and, with regards to the XML files, they are both machine and human readable.

All of the above areas were integrated into the ADOPT architectural framework. As the acronym suggests, Augmented set-based Design and OPTimisation, the framework is built on a, newly-developed, modified and improved SBD approach while also incorporating optimisation tools. Furthermore, ADOPT makes use of a number of engineering design tools, change propagation methods with associated costs, and varying visualisation tools.

The SBD approach was enhanced by a number of ways. The conversion of continuous parameters to discrete, enables the generation of a finite number of possible configurations. Instead of selecting which configurations to be evaluated, ADOPT uses an elimination approach initially. The imposition of infeasibility and undesirability constraints, allows for discretised areas of the design space (ie. the configurations) to be entirely discarded.

The optimisation process within ADOPT, assists in finding the optimum design of each configuration within their respective design-space area. In the case of competing objectives in multiobjective optimisation, it provides a number of non-dominated alternatives. One of the important steps that ADOPT uses, is the identification of active and dormant parameters. The process allows to identify which parameters have an effect on the optimisation objectives (active parameters) and which ones do not (dormant parameters). As a result, configurations that share the same active parameters, and only differ in dormant parameters, will yield the same optimisation results and will be placed in the same optimisation cluster. Therefore, not all the configurations have to be optimised; instead, only the ones that will produce unique results need to be considered for this stage.

The steps described above, provide the answers to the first two research questions posed in Part I of this thesis. Fragmenting the design space, created a finite number of areas, which translates to configurations. This ensures that all possible configurations can be considered, allowing for a thorough exploration of the design space. The inclusion of undesirability and infeasibility constraints, eliminated unwanted areas, making the process faster, and more efficient; since computational power was not wasted on exploring undesirable areas.

Optimising the configurations within the bounds of their respective design space areas, ensures that the configurations that result from each area are the optimum, or non-dominated ones. Furthermore, optimising only the areas that generate unique results, and not all the areas, made the process even more efficient both in terms of time and computational power utilisation.

The introduction of penalties for metrics such as weight and complexity add a further measurement of desirability of each configuration. They also assisted in differentiating the configurations that shared the same optimisation results. Furthermore, additional performance metrics with regards to the systems performance, provided information as to whether or not the configuration that is being considered can meet the expected performance requirements.

Each configuration was also assessed as to its sensitivity to design changes. Using CPM, the resulting DSMs can provide information about which elements of the system are most likely to cause further changes (propagation), but also identify elements that can absorb them without any further impact. The CPM approach was preferred to the C-FAR in order to avoid the high computational costs due to the large number of matrix and vector multiplications required. The inclusion of CP^2 also provides a metric as to how costly each propagation path will be. Integrating CPM with CP^2 addresses research question 4.

The optimisation results, along with the change and cost propagation, complexity and weight penalties, and the performance metrics, provide the user / designer with a large amount of information that acts as a design decision-making mechanism. Different visulisation methods are employed for each set of results depending on their dimensionality, size, and nature of the set. This allows designers and the people involved in the design process, to extract useful information, as required, using visualisation methods that are useful to them or the process itself. The relevant data can then be used from the respective interested parties depending on which aspect of the design process they are concerned with. To address research question number 5, the kind of data required depends on the task and process at hand. However, the wealth of information generated along with the use of the appropriate visualisation tools, ensures the availability of said data to the respective party while presented in a meaningful way.

The storage, management and traceability of information is a critical aspect of the framework not only for the system that is being designed at the time but also for future projects. To address research question number 6, the knowledge that has been generated during the design of a system, can be used in future projects in order to reduce the time of the design process. By avoiding previously explored areas, and making use of results that have already been generated, allows future projects to focus entirely on new areas. Furthermore, previous configurations that were deemed as undesirable due to manufacturing or other limitations, might be re-examined and explored in the future, when the required technological capability has been achieved.

Finally, to address the remaining research question, number 3, a combination of all of the above is required. The use of interactive parallel coordinates to eliminate or filter selections, allows to identify which configurations would remain feasible in the case where there is a change in design parameters or requirements. Furthermore, the SBD approach and the non-dominated solutions arising from the optimisation process, ensure that there are always backup configurations available at every stage of the design process. The inclusion of knowledge-based undesirability and infeasibility constraints allows for the elimination of any configurations that are known beforehand to produce undesirable or non-robust results. Using CPM and CP^2, can provide further insight on which configurations are more susceptible and costly when considering engineering design changes.

The research questions that have been arisen from the problem statement have all been addressed. Given the assumptions that were previously presented, a brief overview of the proposed solutions to each one is presented in Table 9.1.

Addressing all of the research questions translates to successfully satisfying the aims and objectives that were set out for this project. The integration of the two major areas, which was the primary aim of this work, into a unified methodology formed the backbone of the ADOPT architectural framework. From there on, ADOPT was further developed to make it more efficient, faster, and with added capabilities. However, in order for the benefits of the framework to be validated, it had to be applied to a case study.

Research Question	Proposed Solution
RQ1: How can a thorough and detailed exploration of the design space of a complex system take place in a fast and computationally efficient way?	Fragmenting the design space by discretising continuous parameters. Generation of configurations using a Set-Based approach. Elimination of infeasible or undesirable areas. Optimisation of each area that will produce unique results.
RQ2: How can the optimality of the selected configurations be ensured?	Optimisation of each configuration within the bounds of its respective design space area. Generation of non-dominated solutions
RQ3: How can the robustness and feasibility of the selected configurations be ensured when requirements or design parameters change, during the design process?	Interactive parallel coordinates for filtering and selection. Alternative/Backup configurations from SBD and alternative non-dominated designs due to optimisation.
RQ4: How can the sensitivity of the system to the propagation of design changes, and the cost associated with it, be predicted and evaluated?	Integration of CPM and CP^2
RQ5: How can the data required for design decision making be visualised in a meaningful way?	Parallel coordinates for optimisation results with complexity penalties and for cost propagation, DSMs and critical path networks for change and cost propagation, simulation models for performance metrics.
RQ6: How can the knowledge acquired be utilised in future projects?	Storage, management, and traceability of information generated. Future projects can avoid re-evaluation of previously visited areas, and can focus on exploring areas that were previously undesirable due to lack of technological capabilities or otherwise.

Table 9.1: Proposed solutions to the research questions

The design of an aircraft fuel system was used as a case study to demonstrate how the ADOPT architectural framework can be configured and adapted to the design of a system. The process and methodologies outlined in Part II of this Thesis, along with domain-specific tools and design inputs as presented in part III were used to set up the case study. The process was able to generate a wealth of information for design decision-making in a short amount of time as presented in the final chapter of Part III. By bringing the design of a domain-specific system, such as the fuel system, early in the design process, allows for the identification and consideration of novel configurations with other domains. Furthermore, it can eliminate configurations that are infeasible or incompatible with other disciplines, which assists in avoiding design changes later in the design process.

Using the discretisation approach and generating all possible combinations, resulted in a total number of 55296 unique configurations. Imposing the sets of constraints allowed the configurations to be reduced by 93.7% which comes to show the benefit of the constraints in early stages. The generation and elimination of configurations at this stage, and the generation of the initial XML record with respective configuration IDs and properties, took less than half a minute to be completed.

Optimising the remaining 3452 desirable configurations would still require a prohibitively long amount of time even for analytical evaluations. The identification of active and dormant parameters helped to identify the configurations that would yield unique optimisation results; a further reduction of 98.6%, down to just 48 configurations to be optimised.

Visualising the optimisation results using parallel coordinates, presents how the configurations compare against each other. It also demonstrates the trade-off between the optimisation objectives as well as the high correlation between the wing span and the two objectives. Scatter plots help in better visualisation of the pareto front arising from the non-dominated solutions of each configuration.

The introduction of complexity and weight penalties, allows for an additional measurement metric, especially when considering configurations that share the same optimisation results. Adding the penalties to the optimisation results enables a full visualisation of all desirable configurations. Using selection and filtering can check the configurations for robustness as well as eliminating results that do not satisfy certain requirements.

Besides the trade-off studies that are performed using the above metrics, each configuration can be further evaluated on its own. The integration of CPM and CP^2 calculates the combined risk of change propagation, as well as the potential cost of each propagation path. Not only this can assist in identifying sensitive configurations with high risks and costs, it can also help in the design process by sequencing which components are to be designed first.

The fuel system of an aircraft needs to fulfil some essential performance criteria. One of them is for the venting line to be able to handle the large mass flow rate of air in and out the tanks in extreme conditions. The use of 1D models allows for a rapid evaluation of the performance and check if the configuration meets the performance requirements. Furthermore, simulation models can help identify areas for potential improvement, such as weight minimisation due to the reduction of pipe diameters.

The status and results of configurations are stored in an XML file that makes it easy to access and is readable both by machines and humans. Tracing the status of configurations, can provide useful information in future projects as to which ones did not perform as expected, or which ones were not previously evaluated due to technological constraints.

The computational implementation of ADOPT was adapted to the case study in consideration. In order to apply the implementation for the design of other systems a number of things need to be modified. The design parameters, discretisation ranges, infeasibility and undesirability constraints, active and dormant parameters, and optimisation objectives, were all explicitly defined in their respective scripts and blocks. Therefore to generalise the implementation there are two approaches:

- Explicitly define each of the aforementioned inputs to fit the requirements of the desired case study (fully automated approach)
- Keep the human in the loop by removing full automation. Ask the user to provide the necessary inputs at each stage.

Keeping the human in the loop is another important aspect that needs to be considered regardless of the implementation approach. Important decisions should involve the relevant parties due to their ability to bring expert judgement in the process. In the case of ADOPT, the discretisation of continuous parameters, the identification of active and dormant parameters, and most importantly the reduction of the design space, are stages that should involve human interaction.

The implementation of ADOPT for the case of designing an aircraft fuel system demonstrated the benefits that were expected and previously outlined. However a number of potential limitations and drawbacks of the framework were identified:

1. Even for a case study such as the one presented, with less than 20 design parameters, the number of possible combinations is quite large. Each option added, even for one design parameter, increases the number of possible configurations substantially. When dealing with multiple hierarchical levels of a system, and multiple disciplines, this can lead to an unmanageable number of configurations. The imposition of constraints would also be challenging due to the multidisciplinary and multi-level nature of such systems.

2. The large generation of data might pose difficulties in storing and/or visualising them. Computational tools for parallel coordinates, struggled to plot the results generated from the case study that was presented.

3. At this stage, ADOPT does not have an integrated requirements management process. When dealing with more complex multidisciplinary systems, such a process is critical in order to keep track of requirements. This becomes even more vital in long design processes where requirements might change with time. Furthermore there is no information exchange method defined between disciplines.

4. The discretisation of continuous parameters poses an important trade-off. The more discrete values a range is divided into, the more thorough the exploration will be and the constraints can be more accurate. However this increases the complexity, due to the increase in the number of options, and would lead to the issues described in point number 1.

5. Crisp sets in discretisation might lead to loss of potentially desirable configurations due to constraints. For example, generally a high aspect ratio

and a short wing span would yield a relatively small wing area and that would be undesirable in most cases. However, the values closer to the lower bound of a high aspect ratio, and the values close to the upper bound of a short wing span, might yield desirable configurations.

6. With larger systems that include substantially more design parameters, using the case identification method that was outlined in this work would be counterproductive as it would generate excessively long IDs. Even if they would theoretically still be human-readable, they would not be practical. In such cases a machine-generated and machine-readable approach would be the most logical.

7. When dealing with simple systems both in terms of number of elements, and number of options for each design parameter, a point-based approach might be more suitable. This is also the case when dealing with systems that do not have a multidimensional design space, or if the design space is highly constrained. Using a Set-Based approach in such cases, would unnecessarily require more upfront resources, and would not be able to fully exploit the potential benefits of the approach. In general, systems that have design options with limited flexibility, might be more suited for a point-based approach.

Resulting from the work presented in this thesis is a novel methodology for the early stages of designing complex systems. A methodology that provides a thorough exploration of the system's design space while utilising the computational power available in an efficient way. All of the above limitations and drawbacks do not subtract from the novelty, usability, and adaptability of the ADOPT architectural framework and the work presented. Instead, they demonstrate that there are still a lot of areas that can be further developed and improved. The framework should be seen as the foundations, on top of which further research can be undertaken; not only just to improve the framework but also for adapting it for ad hoc purposes.

Chapter 10

Future Work

10.1 Introduction

As previously stated, the work presented in this thesis, serves as the foundations for future research. During this work a number of areas for further improvement or for new developments have been identified. This chapter presents a number of proposals for future work; the first section addresses further improvements and additions to the framework itself while the second section deals with the case study and application areas. Furthermore, the proposals presented here aim at addressing and resolving the limitations and drawbacks that were previously outlined. It is expected that implementing the following proposals will improve the scalability and management of the framework while the case study recommendations will help validate the potential benefits of the improved framework

10.2 Framework Improvement

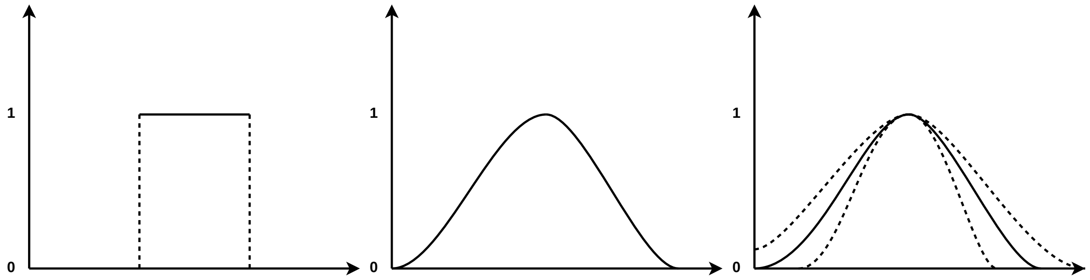
One of the limitations of the current state of the framework is the lack of a requirements management process and a formal method of exchanging information between disciplines. The inclusion and implementation of a Model-Based Systems Engineering (MBSE) approach would resolve both those issues and drastically improve the scalability of the framework [122, 123]. Implementing the MBSE methodology, will enable the identification and traceability of the system's and subsystems' requirements, as well as facilitate the information exchange and collaboration between disciplines. Such a development would drastically improve the framework as it will offer a method to manage the processes within it and provide support for analysis and design activities. Furthermore, the information exchange will allow designers to request the information that is useful to them in the format and method required; this will ensure that only the requested data are transmitted and visualised.

At the current state of the framework, the configurations are being generated, optimised and assessed sequentially; which for analytical evaluations of this scale does not pose an issue. However as the framework is scaled for larger and more complex problems, it will be critical for it be parallelised. By having a number of configurations being evaluated concurrently, it will allow the workflow to be completed in a much shorter time and make better use of modern computational processing capabilities. Such a parallelisation could be implemented using Graphics Processing Units (GPUs) instead of the conventional way of using exclusively CPU clusters. Even though underpowered when compared to CPU cores, GPUs offer thousands of logical cores that can run in parallel. Furthermore, a cluster of CPUs would not only offer a fraction of parallel threads for computation, it will probably cost substantially more than a consumer GPU. It has been proven that, depending on the application case, GPUs can offer substantial time savings for parallel applications when compared to a CPU-based approach of equal costs [124, 125, 126]. However, CPUs are still the prevailing solution for computationally expensive problems which is why a hybrid approach would be more beneficial: A GPU-based parallel evaluation for the early stages where thousands of analytical calculations take place, and gradually moving to a CPU-based evaluation for later stages where only a handful of configurations are considered with more expensive simulations taking place.

The early stages of designing a system, encompasses a large number of uncertainties that can have a critical impact on the robustness of the configurations. There is need to ensure that the feasibility and performance of the system is not compromised due to variations and uncertainties and be able to obtain the target values for performance [127]. Uncertainty quantification and management has been a topic of interest in the aerospace industry and a number of industrial research studies have been undertaken to study the effects of uncertainty [128, 129]. The ability to manage uncertainties in early design stages can ensure the feasibility and manufacturability of the final product even if that has to go through design changes; especially design changes that can propagate through the whole system. There are two main categories of uncertainty: the aleatory (random), and the epistemic (subjective). How uncertainty propagates is also an area of interest, with a number of modelling methods such as the Polynomial Chaos Expansion, Full Factorial Numerical Integration etc. [130]. Due to the nature of the Set-Based Design approach that ADOPT incorporates the initial designs usually contain a high degree of uncertainty. The ability therefore to quantify and manage uncertainty, as well as its propagation, is something that needs to be implemented in ADOPT. This will further ensure the robustness of configurations even if requirements change later in the product development process.

One of the framework processes that caused a number of limitations and drawbacks was the approach used for the discretisation of continuous parameters and the potential loss of configurations due to constraints. The reason behind this is the use of crisp sets for the ranges arising from the discretisation. In crisp sets a value can either be a member of the set or not; in other words, the membership degree is binary, either yes or no (1 or 0). For example, in the case study presented in this work, a 44-meter wing span is considered ”Short” (1 membership in short, 0 in Medium and 0 in Long) even though it is very close to the medium range. Introducing fuzzy sets instead of crisp sets would remove this limitation and provide a number of further benefits [131]. Whereas in crisp sets, a value’s membership degree can either be 0 or 1, in fuzzy sets the value can have a membership degree of anything between 0 and 1 using distributions (Gaussian, triangular, etc.). Furthermore, fuzzy sets allow memberships in more than one range. By using fuzzy sets the uncertainty accompanying how each person perceives what is ”short” and what ”medium” is (in the case of wing span for example) is also captured. From there on, the constraints can also be adapted by using fuzzy logic and various membership functions. Fuzzy logic has two types, 1 and 2; Type-2 also captures the uncertainty (or fuzziness) of the membership itself [132]. Figure 10.1 demonstrates the difference between a crisp set and the two types of fuzzy sets. Fuzzy sets can also be introduced in CPM and CP^2 in order to capture the uncertainty due to experts judgement for probability, impact, and cost; instead of using crisp values.

The final area for improving the framework is with regards to the storage and



(a) Crisp Set (b) Fuzzy Type-1 (c) Fuzzy Type-2

Figure 10.1: Comparison between different types of membership functions

traceability of the generated configurations. As previously mentioned, the approach used for assigning identification strings to configurations would not be applicable to larger systems; this would entail more design parameters hence longer strings. To resolve this a machine-generated and machine-readable identification needs to be defined and used. Furthermore, new filetypes for storing large amounts of data such as the HDF are constantly being developed. These can offer a better alternative to XML files for storing the configuration data and respective results.

10.3 Proposed Case Study Enhancements

The appropriate case study, or application area, is necessary in order to demonstrate the potential benefits and process of any kind of newly developed methodology. The case study presented as part of this work was sufficient to demonstrate those aspects for this stage of the architectural framework. However, when the framework improvements mentioned above are implemented, a more detailed and extensive case study will be required.

The most important area that needs to be validated is the scalability of the framework. In order to assess that, a larger, more complex, and more detailed case study is required. Assuming that the one presented in this work is used as the baseline, a number of improvements can be made. The first is to increase the size of the system by introducing more levels; more than the system level and the aircraft geometry. The new levels can go, for example, down to the component level of the fuel subsystems. Additional levels of the same hierarchy can be introduced, such as the hydraulic or flight control systems.

Optimisation of the fuel system itself is another aspect that can be taken into consideration. From the network and sizing of the pipes, to the geometry of the tanks, a number of aspects can be tuned to improve the performance of the system; they can also provide savings in terms of weight, cost and complexity.

On the topic of the fuel system assessment, a major area when evaluating the performance of such a system, is the thermal analysis of it. The temperature of the fuel needs to be maintained within a certain range; higher than the flash point creates a risk of combustion and lower than the water freezing point creates a risk of ice formation. This is a very complex and crucial area due to the different types of heat exchanges taking place between the fuel subsystems and the surrounding environment, including other systems.

Due to the large thermal capacity of jet fuel (approx $2000 J/(kgK)$ at room temperature) [133], on board fuel has been used as a heat sink for cooling of other systems. Research on using Fuel as a Heat Sink (FaHS) has been an area of research for many years going back to 1973 [134]. Using FaHS can enable better thermal management of the aircraft. Not only it allows for waste heat to be removed from other sources, it also allows the fuel to warm up which can be beneficial in order to avoid freezing conditions, given that it doesn't exceed the flammability limits. Examples of waste heat sources are: the engine oil system, hydraulics and electrical generators.

Two major approaches to using FaHS have been researched. The first is by using a passive cooling approach, where the hot fluid is guided in pipes through the fuel tanks where it is cooled. Even though this approach is very efficient, it poses major safety risks. Having a hot component in a fuel tank is prohibitive, and there is also the possibility of oil leaks in the tank. Using such a concept also wouldn't allow for a good heat distribution in the tank. The second one, which is widely used, is the case of an open fuel loop where instead of bringing the heat to the fuel tank, fuel is directed to the heat source. This is accomplished by pumping fuel out the tank, passing it through a heat exchanger with the hot liquid (i.e. engine oil), and then returning the fuel to the tank. By pumping fuel from the lower part of the tank, where the liquid temperature tends to be lower, and returning it to a higher level warmer, ensures a good heat distribution within the tank. It also makes the cooling process in the heat exchanger more efficient while ram air can also be used to cool the heat exchanger. Improving the overall performance of heat exchangers is also a research area of interest [135].

A number of additional optimisation objectives and performance metrics need to be introduced in order to provide a more detailed evaluation of the configurations. Additional disciplines need to be considered in the process such as the aerodynamic performance and structural analysis. In the case study presented the aspect ratio was constantly close to the lower bound, which provided a larger wingbox volume without affecting the surge tank volume. However, the aspect ratio has a critical impact on the lift and aeroelasticity of the wing. Therefore additional trade-offs will arise as new objectives are being introduced. Furthermore, structural and aerodynamic simulations using Computational Fluid Dynamics, tend to be very expensive; depending on the approach used. Therefore an appropriate management of computational power becomes crucial. Cheap and analytical simulations can be used in the early stages due to the large number of configurations considered, whereas more detailed numerical simulations can take place at later stages for a more detailed evaluation and assessment.

Finally, it is the author's opinion that the framework should also be applied to the design of an entirely different system. Not just moving away from aerospace and aircraft systems, but from physical systems altogether. Applying ADOPT to the design of an organisational system, for example, will further solidify its adaptability to the design of any kind of system in consideration.

Chapter 11

Conclusion

As the design of systems has been relying more and more on computational methods, new approaches need to be developed in order to take advantage of the increase in processing power. The new approaches need to be able to perform a thorough design space exploration in a computationally efficient way.

At the beginning of the project, the aim was to compare and contrast two areas that aimed to address that: The Set-Based Design approach and Multidisciplinary Design Optimisation tools. Since then, the project has evolved considerably with the outcome being a novel architectural framework which introduces optimisation tools within an improved Set-Based Design process. The new aims and objectives were addressed fully which in turn addressed the research questions and the problem statement.

The ADOPT framework was developed in such a way that allows it to be applied to the design of any physical system. It is also expected that it can be expanded to the design of any system; physical or not. Furthermore, it introduces a number of new modifications to the Set-Based Design approach in order to improve the process.

The framework takes into consideration the multiple hierarchical levels of the system by bringing the design of domain-specific subsystems, earlier in the overall design process. The discretisation of continuous parameters transforms the design space by fractioning it in a finite number of small areas. A large number of those areas can then be eliminated by imposing constraints, allowing the computational power to be directed at areas that are expected to yield desirable configurations. With the ability to identify the active and dormant design parameters of the system, the framework can then be used to optimise only the areas that will yield unique non-dominated results. As configurations are eliminated, the computational power is redistributed to the remaining configurations, for more detailed and potentially more computationally expensive simulations.

All of the above enable the framework to be used to perform a thorough design space exploration with better utilisation of computational power. Furthermore, the inclusion of engineering change prediction methods, and the newly developed CP^2, allow for a more detailed assessment of the sensitivity to design changes of each configuration.

Different methods and approaches can be used for evaluating and assessing the performance of the configurations. From analytical calculations and penalties for the early stages where a large number of configurations are still considered, to numerical simulations for the late stages.

The use of appropriate visualisation methods is critical in cases such as this where a large amount of data is generated. Presenting the data to the relevant parties in a meaningful way is essential for the design decision making process. In addition to that, knowledge generated can be used for future projects to reduce the time required to reach the trade-off stage; this makes the storage, management, and traceability of the data another important factor that needs to be considered.

Applying ADOPT to the design of aircraft fuel systems, presented how the framework can be adapted to a specific case study. The implementation of it, and the results generated, demonstrate the benefits of the framework as were expected, but have also assisted in identifying areas for further improvement.

At this stage, ADOPT offers a very strong proven foundation for future research to be carried out. The areas that have been identified will enable the framework to overcome any kind of limitations or drawbacks currently present. Furthermore, the proposed improvements will make ADOPT faster, more efficient, more powerful, and more adaptable.

References

[1] Moore, G. The future of integrated electronics. Technical report, Fairchild Semiconductor, 1964.

[2] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D. CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. Technical report, NASA, 2014.

[3] Keane, A. and Nair, P. *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. Wiley, 2005.

[4] Isaksson, O., Keski-Seppl, S., and Eppinger, S. D. Evaluation of design process alternatives using signal flow graphs. *Journal of Engineering Design*, 11(3):211–224, 2000.

[5] Langen, P. H. V. and Brazier, F. M. Design space exploration revisited. *AI EDAM*, 20(02), 2006.

[6] McKenney, T. A., Kemink, L. F., and Singer, D. J. Adapting to changes in design requirements using set-based design. *Naval Engineers Journal*, 123(3):66–77, 2011.

[7] Nunez, M., Datta, V. C., Molina-Cristobal, A., Guenov, M., and Riaz, A. En-

abling exploration in the conceptual design and optimisation of complex systems. *Journal of Engineering Design*, 23(10-11):852–875, 2012.

[8] International Council on Systems Engineering. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities.* Wiley-Blackwell, 4th edition, 2015.

[9] National Aeronautics and Space Administration. *NASA Systems Engineering Handbook.* CreateSpace Independent Publishing Platform, 2007.

[10] Sharman, D. and Yassine, A. Characterizing complex product architectures. *Systems Engineering*, 7(1):35–60, 2004.

[11] Ulrich, K. and Eppinger, S. *Product Design and Development.* McGraw-Hill Education Ltd, 2015.

[12] Womack, J., Jones, T., and Roos, D. *The Machine that changed the world. How Lean production Revolutionized the Global Car Wars.* Simon and Schuster Ltd, new edition, 2007.

[13] Mavilio, V. The application of lean management system in early clinical research. Master's thesis, School of Health, Cranfield University, 2009.

[14] Sjogren, P. Usefulness of lean as a sustainable strategyin food supply chains. Master's thesis, School of Applied Sciences, Cranfield University, 2014.

[15] Krings, D., Levine, D., and Wall, T. The use of lean in local government. *ICMA Public Managament Magazine*, 88(8), 2006.

[16] Siyam, G. I., Wynn, D. C., and Clarkson, P. J. Review of value and lean in complex product development. *Systems Engineering*, 18(2):192–207, 2015.

[17] Murman, E., Allen, T., and Cutcher-Gershenfeld, J. *Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative.* Palgrave Macmillan, 2002.

[18] Walton, M. Strategies for lean product development. Technical report, Lean Aerospace Initiative, Massachusetts Institute of Technology, 1999.

[19] Harrison, A., Pavitt, J., and Alexander, J. The status of lean thinking in uk lean aerospace initiative (uk-lai). supply chains: a survey. Technical report, Cranfield School of Management, 2002.

[20] Al-Ashaab, A., Golob, M., Attia, U., Khan, M., Parsons, J., Andino, A., Perez, A., Guzman, P., Onecha, A., Kesavamoorthy, S., Martinez, G., Shehab, E., Berkes, A., Haque, B., Soril, M., and Sopelana, A. The transformation of product development process into lean environment using set-based concurrent engineering: A case study from an aerospace industry. *Concurrent Engineering: Research and Applications*, 21:268–285, 2013.

[21] Khan, M. S., Al-Ashaab, A., Shehab, E., Haque, B., Ewers, P., Sorli, M., and Sopelana, A. Towards lean product and process development. *International Journal of Computer Integrated Manufacturing*, 26(12):1105–1116, 2013.

[22] Morgan, J. and Liker, J. *The Toyota Product Development System. Integrating People Processes and Technology.* Productivity Press, 2006.

[23] Sobek, D., Ward, A., and Liker, J. Toyota's principles of set-based concurrent engineering. *Sloan Management Review*, 40(2):67–83, 1999.

[24] Ertas, A. and Jones, J. C. *The Engineering Design Process.* JOHN WILEY & SONS INC, 1996.

[25] Cross, N. *Engineering Design Methods: Strategies for Product Design.* Wiley, 4 edition, 2008.

[26] Howard, T., Culley, S., and Dekoninck, E. Describing the creative design process by the integration of engineering design and cognitive psychology literature. *Design Studies*, 29(2):160 – 180, 2008.

[27] Khandani, S. Engineering design process- education transfer plan. Technical report, Initiatives for Science and Math Education, 2005.

[28] Holston, D. *The Strategic Designer: Tools & Techniques for Managing the Design Process.* F+W Media, 2011.

[29] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K. H. *Engineering Design: A Systematic Approach.* Springer-Verlag GmbH, 2006.

[30] Ullman, D. G. Robust decision-making for engineering design. *Journal of Engineering Design*, 12(1):3–13, 2001.

[31] Ring, J. A value-seeking approach to the engineering of systems. In *SMC98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics.* IEEE, 1998.

[32] Isaksson, O., Kossmann, M., Bertoni, M., Eres, H., Monceaux, A., Bertoni, A., Wiseall, S., and Zhang, X. Value-driven design - a methodology to link expectations to technical requirements in the extended enterprise. *INCOSE International Symposium*, 23(1):803–819, 2013.

[33] French, M. J. *Conceptual Design for Engineers.* Springer-Verlag GmbH, 1998.

[34] Pugh, S. *Total Design: Integrated Methods for Successful Product Engineering.* Addison-Wesley, 1991.

[35] Ullman, D. *The Mechanical Design Process.* McGraw-Hill Science/Engineering/Math, 2002.

[36] Wheelwright, S. C. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency and Quality.* Free Press, 2011.

[37] Ward, A. C. and Liker, J. K. Set-based concurrent engineering and toyota. *Proceedings of the ASME DTM,* 1994.

[38] Raudberget, D. The decision process in set-based concurrent engineering - an industrial case study. *International Design Conference - DESIGN,* 2010.

[39] Bernstein, J. Design methods in the aerospace industry: Looking for evidence of set-based practices. Master's thesis, Massachusetts Institute of Technology, 1998.

[40] Singer, D. J., Doerry, N., and Buckley, M. E. What is set-based design? *Naval Engineers Journal,* 121(4):31–43, 2009.

[41] Raudberget, D. *Industrial Application of Set-based Concurrent Engineering Managing the design space by using Platform System Families.* PhD thesis, Chalmers University of Technology, 2015.

[42] Ward, A. and Sobek, D. *Lean Product and Process Development.* Lean Enterprises Inst Inc, 2014.

[43] Ward, A., Liker, J. K., Cristiano, J. J., and Sobek, D. K. The second toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review*, 36(3):43–61, 1995.

[44] Kennedy, B. M., Sobek, D. K., and Kennedy, M. N. Reducing rework by applying set-based practices early in the systems engineering process. *Systems Engineering*, 17(3):278–296, 2014.

[45] Khan, M., Al-Ashaab, A., Doultsinou, A., Shehab, E., Ewers, P., and Sulowski, R. Set-based concurrent engineering process within the leanppd environment. In *Improving complex systems today : proceedings of the 18th ISPE International Conference on Concurrent Engineering*. Springer, 2011.

[46] Malak, R. J., Aughenbaugh, J. M., and Paredis, C. J. Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design*, 41(3):214 – 227, 2009. Computer Support for Conceptual Design.

[47] Raudberget, D. Practical Applications of Set-Based Concurrent Engineering in Industry. *Journal of Mechanical Engineering*, 56(11):685–695, 2010.

[48] Raudberget, D., Levandowski, C., Isaksson, O., Kipouros, T., Johannesson, H., and Clarkson, J. Modelling and assessing platform architectures in pre-embodiment phases through set-based evaluation and change propagation. *Journal of Aerospace Operations*, 3(3, 4):203221, 2015.

[49] Mebane, L. W., Carlson, M. C., Dowd, C., Singer, J. D., and E., B. M. Set-based design and the ship to shore connector. *Naval Engineers Journal*, 123(3):79–92, 2011.

[50] Castelli, F. Using set-based concurrent engineering to enhance rolls-royce product development. Master's thesis, School of Applied Sciences, Cranfield University, 2012.

[51] Carlotta, R. Supporting the development of turbine sub-system for a helicopter engine using set-based concurrent engineering. Master's thesis, School of Applied Sciences, Cranfield University, 2012.

[52] Guenov, M., Nunez, M., Molina-Cristobal, A., Datta, V., and Riaz, A. Aircadia -an interactive tool for the composition and exploration of aircraft computational studies at early design stage. *Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences*, 2014.

[53] Levandowski, C., Raudberget, D., and Johannesson, H. Set-based concurrent engineering for early phases in platform development. *Advances in Transdisciplinary Engineering*, 2014.

[54] Levandowski, C., Muller, J., and Isaksson, O. Modularization in concept development using functional modeling. *Advances in Transdisciplinary Engineering*, 4(Transdisciplinary Engineering: Crossing Boundaries):117–126, 2016.

[55] Landahl, J., Levandowski, C., Johannesson, H., and Isaksson, O. Assessing producibility of product platforms using set-based concurrent engineering. *Advances in Transdisciplinary Engineering*, 2016.

[56] Riaz, A., Guenov, M. D., and Molina-Cristobal, A. Set-based approach to passenger aircraft family design. *Journal of Aircraft*, 54(1):310–326, 2017.

[57] Eppinger, S. D. and Browning, T. R. *Design Structure Matrix Methods and Applications (Engineering Systems).* The MIT Press, 2012.

[58] Danilovic, M. and Browning, T. R. Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management*, 25(3):300 – 314, 2007.

[59] Browning, T. R. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Trans. Engineering Management*, 48:292–306, 2001.

[60] Rodgers, J., Korte, J., and Bilardo, V. Development of a genetic algorithm to automate clustering of a dependency structure matrix. Technical report, National Aeronautics and Space Administration (NASA), 2006.

[61] Tang, D., Zheng, L., Li, Z., Li, D., and Zhang, S. Re-engineering of the design process for concurrent engineering. *Computers & Industrial Engineering*, 38(4):479 – 491, 2000.

[62] Browning, T. R. Design Structure Matrix Extensions and Innovations : A Survey and New Opportunities. *IEEE Transactions on Engineering Management*, 63(1):27–52, 2016.

[63] Eppinger, S. D., Joglekar, N. R., Olechowski, A., and Teo, T. Improving the systems engineering process with multilevel analysis of interactions. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 28(04):323–337, sep 2014.

[64] Lambe, A. and Martins, J. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization

processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, 2012.

[65] Hauser, J. and Clausing, D. The house of quality. *Harvard Business Review*, 1988.

[66] Andersson, J. *Multiobjective optimisation in engineering Design*. PhD thesis, Linkping University, 2001.

[67] Gavel, H., Krus, P., and Andersson, J. Quantification of the elements in the relationship matrix, a conceptual study of aircraft fuel system. In *42nd AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2004.

[68] Gavel, H., Olvander, J., and Johansson, B. An Algorithmic Morphology Matrix for Aircraft Fuel System Conceptual Design. *25th International Congress of the Aeronautical Sciences*, 2006.

[69] Olvander, J., Lundén, B., and Gavel, H. A computerized optimization framework for the morphological matrix applied to aircraft conceptual design. *Computer-Aided Design*, 41(3):187–196, 2009.

[70] Frank, C. P., Marlier, R. A., Pinon-Fischer, O. J., and Mavris, D. N. Evolutionary multi-objective multi-architecture design space exploration methodology. *Optimization and Engineering*, 2018.

[71] Suh, N. P. *The Principles of Design (Oxford Series on Advanced Manufacturing)*. Oxford University Press, 1990.

[72] Suh, N. P. *Axiomatic Design: Advances and Applications (MIT-Pappalardo Series in Mechanical Engineering)*. Oxford University Press, 2001.

[73] Saaty, T. L. *Fundamentals of Decision Making and Priority Theory With the Analytic Hierarchy Process (Analytic Hierarchy Process Series, Vol. 6).* RWS Publications, 2000.

[74] Arroyo, P., Tommelein, I. D., and Ballard, G. Comparing AHP and CBA as decision methods to resolve the choosing problem in detailed design. *Journal of Construction Engineering and Management*, 141(1):04014063, 2015.

[75] Crawley, E., Cameron, B., and Selva, D. *System Architecture - Strategy and Product Development for Complex Systems.* Pearson, 2016.

[76] Clark, K. and Fujimoto, T. *Product Development Performance: Strategy, Organization and Management in World Auto Industry.* Harvard Business School Press, 1991.

[77] Anderson, D. M. *Agile Product Development for Mass Customization: How to Develop and Deliver Products for Mass Customization, Niche Markets, JIT, Build-to-order and Flexible Manufacturing.* Irwin Professional Publishing, 1997.

[78] Kidd, M. and Thompson, G. Engineering design change management. *Integrated Manufacturing Systems*, 11(1):74–77, 2000.

[79] Clarkson, P. and Eckert, C. *Design Process Improvement: A review of current practice.* Springer, 2004.

[80] Keller, R. *Predicting Change Propagation: Algorithm, Representations, Software, Tools.* PhD thesis, Cambridge University, 2007.

[81] Hamraz, B., Caldwell, N. H., and Clarkson, P. J. A matrix-calculation-based algorithm for numerical change propagation analysis. *IEEE Transactions on Engineering Management*, 60(1):186–198, 2013.

[82] Cohen, T., Navathe, S., and Fulton, R. C-far, change favorable representation. *Computer-Aided Design*, 32(5):321 – 338, 2000.

[83] Clarkson, P., Simons, C., and Eckert, C. Predicting change propagation in complex design. *Journal of Mechanical Design*, 126(5), 2004.

[84] Suh, E. S., de Weck, O. L., and Chang, D. Flexible product platforms: framework and case study. *Research in Engineering Design*, 18(2):67–89, 2007.

[85] Koh, E. C., Caldwell, N. H., and Clarkson, P. J. A technique to assess the changeability of complex engineering systems. *Journal of Engineering Design*, 4828(March):477–498, 2017.

[86] Nocedal, J. and Wright, S. *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer, 2006.

[87] Ebrahimi, M. and Jahangirian, A. Aerodynamic optimization of airfoils using adaptive parameterization and genetic algorithm. *Journal of Optimization Theory and Applications*, 162(1):257–271, 2014.

[88] Wunderlich, T. Multidisciplinary wing optimization of commercial aircraft with consideration of static aeroelasticity. *CEAS Aeronautical Journal*, 6(3):407–427, 2015.

[89] Oktay, E., Akay, H., and Merttopcuoglu, O. Parallelized structural topology optimization and cfd coupling for design of aircraft wing structures. *Computers and Fluids*, 49:141–145, 2011.

[90] Toropov, V., Jones, R., Willment, T., and Funnell, M. Weight and manufacturability optimization of composite aircraft components based on a genetic algorithm. In *6th World Congresses of Structural and Multidisciplinary Optimization*, 2005.

[91] Sobieszczanski-Sobieski, J. and Haftka, R. T. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*, 14(1):1–23, 1997.

[92] Guenov, M., Fantini, P., Balachandran, L., Maginot, J., Padulo, M., and Nunez, M. Multidisciplinary design optimization framework for the pre design stage. *Journal of Intelligent and Robotic systems*, 59(3):223–240, 2010.

[93] Zingg, D. W., Nemec, M., and Pulliam, T. H. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics*, 17(1-2):103–126, 2008.

[94] Lin, M.-H., Tsai, J.-F., and Yu, C.-S. A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering*, 2012:1–15, 2012.

[95] Jones, D., Mirrazavi, S., and Tamiz, M. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, 2002.

[96] Konak, A., Coit, D. W., and Smith, A. E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[97] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[98] Helbig, M. and Engelbrecht, A. P. Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm and Evolutionary Computation*, 14:31–47, 2014.

[99] Martins, J. and Lambe, A. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013.

[100] Balesdent, M., Bérend, N., Dépincé, P., and Chriette, A. A survey of multidisciplinary design optimization methods in launch vehicle design. *Structural and Multidisciplinary Optimization*, 45(5):619–642, 2011.

[101] Tedford, N. P. and Martins, J. R. R. A. Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11(1):159–183, 2009.

[102] Cramer, E. J., J. E. Dennis, J., Frank, P. D., Lewis, R. M., and Shubin, G. R. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994.

[103] Haftka, R. T. Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103, 1985.

[104] Sobieszczanski-Sobieski, J. Optimization by decomposition: A step from hierarchic to non-hierarchic systems. Technical report, NASA, Langley Research Center, 1988.

[105] Chittick, I. R. and Martins, J. R. R. A. An asymmetric suboptimization approach to aerostructural optimization. *Optimization and Engineering*, 10(1):133–152, 2008.

[106] Hannapel, S. and Vlahopoulos, N. Implementation of set-based design in multidisciplinary design optimization. *Structural and Multidisciplinary Optimization*, 50(1):101–112, 2014.

[107] Veenhuis, C. A set-based particle swarm optimization method. In *Parallel Problem Solving from Nature*, volume 5199, pages 971–980, 2008.

[108] Inselberg, A. *Parallel Coordinates - Visual Multidimensional Geometry and Its Applications.* Springer New York, 2009.

[109] Kipouros, T., Inselberg, A., Parks, G., and Savill, M. Parallel coordinates in computational engineering design. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013.

[110] Hettenhausen, J., Lewis, A., and Kipouros, T. A web-based system for visualisation-driven interactive multi-objective optimisation. *Procedia Computer Science*, 29:1915–1925, 2014.

[111] Georgiades, A., Sharma, S., Kipouros, T., and Savill, M. Predicting and visualizing cost propagation due to engineering design changes. *Proceedings of the 21st International Conference on Engineering Design (ICED 17)*, 2017.

[112] European Aviation Safety Agency. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25. Technical report, EASA, 2016.

[113] Langton, R., Clark, C., Hewitt, M., and Richards, L. *Aircraft Fuel Systems.* Wiley, 2009.

[114] Dean, S., Doherty, J., and Wallace, T. MDO-Based concept modelling and the impact of fuel systems on wing design. *47th AIAA Aerospace Sciences Meeting*, 2009.

[115] Datadvance. pSeven v6.11 User Manual, Available at: https://www.datadvance.net/product/pseven/manual/6.11/index.html, 2017.

[116] Wynn, D., Wyatt, D., Nair, S., and P., C. An introduction to the cambridge advanced modeller. In *Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes*, 2010.

[117] Siemens PLM Software. LMS Imagine.Lab AMESim v15.2, Available at: https://www.plm.automation.siemens.com/en/products/lms/imagine-lab/amesim/platform/index.shtml, 2017.

[118] Macrofocus. High-D v1.0rc13, Available at: https://www.high-d.com/documentation, 2017.

[119] XDAT. XDAT v2.2, Available at: https://www.xdat.org, 2017.

[120] Ardema, M. D., Chambers, M. C., Patron, A. P., Hahn, A. S., Miura, H., and Moore, M. D. Analytical fuselage and wing and weight estimation and of transport and aircraft. Technical memorandum, NASA, 1996.

[121] Jenkinson. *Civil Jet Aircraft Design.* Elsevier Limited, 1999.

[122] Micouin, P. *Model Based Systems Engineering: Fundamentals and Methods (Control, Systems and Industrial Engineering Series).* Wiley-ISTE, 2014.

[123] Marwedel, S., Fischer, N., and Salzwedel, H. Improving the design quality of complex networked systems using a model-based approach. In *3rd International Conference on ModelBased Systems Engineering*, 2010.

[124] Schatz, M. C., Trapnell, C., Delcher, A. L., and Varshney, A. High-throughput sequence alignment using graphics processing units. *BMC Bioinformatics*, 8(1):474, 2007.

[125] Salleh, N. S. M. and Baharim, M. F. Performance comparison of parallel execution using GPU and CPU in SVM training session. In *2015 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*. IEEE, 2015.

[126] Tsotskas, C., Kipouros, T., and Savill, A. M. The design and implementation of a GPU-enabled multi-objective tabu-search intended for real world and high-dimensional applications. *Procedia Computer Science*, 29:2152–2161, 2014.

[127] Park, G.-J., Lee, T.-H., Lee, K. H., and Hwang, K.-H. Robust design: An overview. *AIAA Journal*, 44(1):181–191, 2006.

[128] Garzon, V. E. and Darmofal, D. L. Impact of Geometric Variability on Axial Compressor Performance. *Proceedings of ASME Turbo Expo 2003*, 2003.

[129] Huyse, L., Langley, N., and William, C. Probabilistic Approach to Free-Form Airfoil Shape Optimization Under Uncertainty. *Aiaa Journal*, 40(9), 2002.

[130] Lee, S. H. and Chen, W. A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and Multidisciplinary Optimization*, 37(3):239–253, 2008.

[131] Zadeh, L. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

[132] Castillo, O. and Melin, P. *Type-2 Fuzzy Logic: Theory and Applications (Studies in Fuzziness and Soft Computing)*. Springer, 2007.

[133] Coordinating Research Council Inc. *Handbook of aviation fuel properties*. 2004.

[134] Gray, C. and Shayeson, M. Aircraft Fuel Heat Sink Utilization. Technical report, US Air Force, 1973.

[135] Northcutt, B. and Mudawar, I. Enhanced design of cross-flow microchannel heat exchanger module for high-performance aircraft gas turbine engines. *Journal of Heat Transfer*, 134(June):061801, 2012.

Appendix

Appendix A

Publications

Peer-Reviewed Conference Papers

- Georgiades, A., Sharma, S., Kipouros, T., and Savill, M. Predicting and visualizing cost propagation due to engineering design changes. *Proceedings of the 21st International Conference on Engineering Design (ICED 17)*, 2017.

Submitted Journal Papers (Under Review)

- Georgiades, A., Sharma, S., Kipouros, T., and Savill, M. ADOPT: An augmented set-based design framework with optimisation. *Design Science*, Cambridge University Press, 2018.

Posters and Presentations

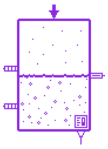
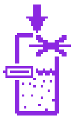
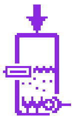
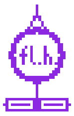
- Georgiades, A. Set-Based design and optimisation of aircraft systems. *International Spring School of Systems Engineering (IS3E)*, 2017.

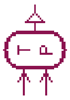
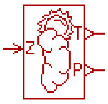
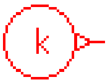
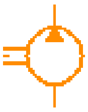
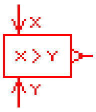
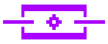
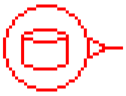
- Georgiades, A. Design Structure Matrices: Overview and applications. *Airbus Functional Analysis Workshop*, 2016.
- Georgiades, A. Synthesis of Systems Engineering methods and MDO tools (Poster). *Airbus PhD Day*, 2016.

Appendix B

AMESim Models

The following table provides a description of the AMESim models used in this work.

AMESim Model	Description
	Aircraft Fuel Tank (defined by tables)
	Tank orifice for gas flow only
	Tank orifice for fluid flow only
	Fluid height sensor

AMESim Model	Description
	Pneumatic pipe
	Pneumatic conversion of signals to Temperature and Pressure
	International Standard Atmosphere table
	Constant signal
	Fixed displacement pump
	Conditional Function (X greater than Y)
	Aircraft attitude
	Gas Mixing pipe
	Input from 1D table

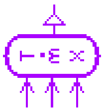
AMESim Model	Description
	Gas Mixture modulated mass flow source
— — —	Direct Connections

Table B.1: Description of AMESim models