AUTOMATED HYBRID-ELECTRIC PROPULSION ARCHITECTURE MODELING FOR CONCEPTUAL AIRCRAFT DESIGN: A NOVEL APPROACH TO INTEGRATING SYSTEM ARCHITECTING IN MDO

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MAHMOUD ESSAM ABDELMONEAM FOUDA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

JANUARY 2024

Approval of the thesis:

AUTOMATED HYBRID-ELECTRIC PROPULSION ARCHITECTURE MODELING FOR CONCEPTUAL AIRCRAFT DESIGN: A NOVEL APPROACH TO INTEGRATING SYSTEM ARCHITECTING IN MDO

submitted by **MAHMOUD ESSAM ABDELMONEAM FOUDA** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Serkan Özgen
Head of Department, **Aerospace Engineering** ______________

Prof. Dr. Dilek Funda Kurtuluş
Supervisor, **Aerospace Engineering, METU** ______________

Dr. Luca Boggero
Co-supervisor, **The German Aerospace Center DLR** ______________

Examining Committee Members:

Prof. Dr. Ozan Tekinalp
Aerospace Engineering, METU ______________

Prof. Dr. Dilek Funda Kurtuluş
Aerospace Engineering, METU ______________

Assoc. Prof. Dr. Nilay Sezer Uzol
Aerospace Engineering, METU ______________

Assist. Prof. Dr. Özge Başkan Perçin
Aerospace Engineering, METU ______________

Prof. Dr. Nevsan Şengil
Aerospace Engineering, THKU ______________

Date:09.01.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Mahmoud Essam Abdelmoneam Fouda

Signature :

ABSTRACT

AUTOMATED HYBRID-ELECTRIC PROPULSION ARCHITECTURE MODELING FOR CONCEPTUAL AIRCRAFT DESIGN: A NOVEL APPROACH TO INTEGRATING SYSTEM ARCHITECTING IN MDO

Fouda, Mahmoud Essam Abdelmoneam
M.S., Department of Aerospace Engineering
Supervisor: Prof. Dr. Dilek Funda Kurtuluş
Co-Supervisor: Dr. Luca Boggero

January 2024, 87 pages

The implementation of electric propulsion systems plays a pivotal role in enhancing the sustainability of air transport, offering novel design possibilities, particularly in the initial design phases to leverage the advantages provided by electric power.

The electrification or hybridization of aircraft presents a multifaceted design problem. To address this matter, the utilization of architecture optimization techniques are employed as a mean to explore an extensive array of designs. The optimization process necessitates the utilization of flexible and automated modeling techniques for propulsion systems. Furthermore, it is crucial to examine the performance of these systems on a holistic aircraft level to establish reasonable objectives for evaluation.

This thesis presents a novel approach for the construction, integration, and evaluation of propulsion architectures in the context of conceptual aircraft design. It aims to integrate the system architecting method, applied to propulsion architecture modeling, into the design optimization process. Utilizing a software toolkit OpenConcept, ADORE and NASA's OpenMDAO, the presented methodology facilitates versatile, autonomous construction and evaluation of propulsion architectures, considering the impact of various architectural decisions on aircraft performance relative to given missions.

This study entails an examination of the integration of electric and hybrid electric propulsion into the commuter aircraft, Beechcraft King Air C90GT. This task entails the automated evaluation of five distinct propulsion systems, conventional, turboelectric, all-electric, series, and parallel hybrid systems. The objective is to minimize fuel consumption and the maximum takeoff weight while taking into account various design ranges and battery energy capacities.

Keywords: System Architecting, Electric Propulsion, Mutlidisiplinary Design Optimization (MDO), Aircraft Design

ÖZ

KAVRAMSAL UÇAK TASARIMI İÇİN OTOMATİK HİBRİT-ELEKTRİKLİ İTKİ MİMARİSİ MODELLEMESİ: SİSTEM MİMARİSİNİ MDO'YA ENTEGRE ETMEK İÇİN YENİ BİR YAKLAŞIM

Fouda, Mahmoud Essam Abdelmoneam
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Dilek Funda Kurtuluş
Ortak Tez Yöneticisi: Dr. Luca Boggero

Ocak 2024 , 87 sayfa

Elektrikli ve hibrit-elektrikli itki sistemlerinin uygulanması, hava taşımacılığı sürdürülebilirliğini artırmada önemli rol oynar. Bu sistemler, elektrik gücünün avantajlarıyla, tasarım sürecinin başında yenilikçi olanaklar sunar.

Hava araçlarının elektrifikasyonu veya hibridleştirilmesi, çeşitli tasarım zorlukları yaratır. Bu sorunları çözmek için mimari optimizasyon teknikleri, geniş tasarım yelpazesini keşfetmekte ve en uygun tasarımları belirlemektedir. Optimizasyon, itki sistemlerinin esnek ve otomatik modellemesini gerektirir; sistemlerin bütünsel hava aracı düzeyinde performansını incelemek esastır.

Tez, kavramsal hava aracı tasarımında itki mimarilerinin oluşturulması ve değerlendirilmesi için yeni bir yaklaşım önerir. Bu yaklaşım, itki mimarilerine sistem mimarisi yöntemini entegre eder. Sunulan metodoloji, çeşitli mimari kararların hava aracı performansına etkisini göz önünde bulundurarak itki mimarilerini otonom olarak inşa etmek ve değerlendirmek için çok yönlü bir teknik kullanır. Araştırmada, OpenCon-

cept, ADORE ve NASA'nın OpenMDAO çerçevesi kullanılmıştır.

Çalışma, elektrikli ve hibrit itki sistemlerinin Beechcraft King Air C90GT banliyö uçağına entegrasyonunu inceler. Beş itki sistemi—geleneksel, turboelektrik, tamamen elektrikli, seri ve paralel hibrit—değerlendirilir. Amaç, tasarım optimizasyonuyla yakıt tüketimi ve maksimum kalkış ağırlığını en aza indirmektir.

Çalışma, elektrikli ve hibrit itki sistemlerinin araştırılması ve geliştirilmesi için yeni beklentiler sunar ve çevre dostu hava taşımacılığının ilerlemesine katkı sağlar.

Anahtar Kelimeler: Sistem Mimarisi, Elektrikli itki, Çok Disiplinli Tasarım Optimizasyonu (MDO), Uçak Tasarımı

To those who persist with unwavering determination, defying every obstacle along the way

ACKNOWLEDGMENTS

This thesis marks the culmination of my journey towards becoming an aerospace engineer. It's time for me to wrap up this chapter and embark on new adventures, both professionally and personally.

I extend my heartfelt thanks to my supervisor, Prof. Dilek Funda Kurtulus, for her unwavering support and assistance during my thesis journey and My Co-Supervisor Dr. Luca Boggero for his enduring kindness and mentorship. I would like to express my deepest gratitude to Jasper Bussemaker from the DLR Institute of Systems Architecture in Aeronautics in Hamburg for his invaluable guidance, patience, and expertise throughout the course of this thesis. His insights and support have been fundamental to the completion of this work. I would also like to thank the DLR Institute of Systems Architecture in Aeronautics and my colleagues there for the memorable and enriching times during my research.

I am also profoundly grateful to my friends, who have provided me with endless encouragement and thoughtful advice. Their companionship during this academic journey has been a source of strength and inspiration.

My heartfelt thanks go to my family for their unwavering love and support. Their sacrifices, understanding, and belief in my abilities have been the pillars of my perseverance and success.

Lastly, a humorous yet heartfelt pat on the back to myself for sticking with me through thick and thin, and never letting me down (even when the coffee ran out); for mastering the art of self-belief and keeping the flame of perseverance burning, even on the rainiest of thesis-research and writing days!

This thesis is not only a reflection of my efforts but also a testament to the continuous support and faith that these remarkable individuals have placed in me.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xix

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation 1

1.1.1 Electric and hybrid-electric Propulsion System Design Complexities 2

1.1.2 Towards Automated Design Space Exploration 3

1.2 Thesis Objective 4

1.3 Research Questions 4

1.4 Requirements 5

1.5 The Outline of the Thesis 6

2 NECESSARY BACKGROUND 7

2.1 System Engineering 7

2.1.1 Document-Based Systems Engineering (DBSE) vs Model-Based Systems Engineering (MBSE) 7

2.1.2 System Architecting process and Architecture Design Space . 8

2.2 Multidisciplinary Design Optimization MDO 8

2.2.1 Conventional vs. Design Optimization Process 9

2.2.2 Optimization Problem formulation 10

2.3 Challenges and Contribution 11

2.3.1 Integrating system Architecting method into MDO 12

2.3.2 MBSE Model Development and Process Integration 13

2.3.3 Scientific Contribution 14

3 PROBLEM DEFINITION AND METHODOLOGY 15

3.1 Problem Definition 15

3.2 Methodology 16

3.2.1 Constructing Propulsion System Architectures 16

3.2.2 Automated System Architecture Builder 18

3.2.3 Design Space Definition and Identifying Possible Architectural Decisions 21

3.2.4 Overall system Level Evaluation 24

3.3 Research Questions Summary 25

4 IMPLEMENTATION 27

4.1 Conceptual Design and mission analysis: OpenConcept 27

4.1.1 Takeoff Analysis 28

4.1.2 Mission Profile 30

4.1.3 Numerical Intergration 31

4.2 Constructing an Automated Numerical Propulsion Architecture model 32

4.3 Multidisciplinary Design Analysis and Optimization (MDAO) 32

4.4 Reference Aircraft Model and Technology Assumptions 33

4.5 Optimization Problem For All Architectures 35

4.6 Research Questions Summary 36

5 RESULTS AND DISCUSSION 39

5.1 MDO Architectures Results and Discussion 39

5.1.1 Conventional and Turbo-Electric Propulsion Architectures . . 40

5.1.2 All Electric propulsion Architecture 43

5.1.3 Series Hybrid propulsion Architecture 45

5.1.4 Parallel Hybrid Propulsion Architecture 48

5.2 Post-Optimality study: Comparison of Architectures 51

5.2.1 Parallel Hybrid vs Series Hybrid Propulsion Architectures . . . 52

5.2.2 Weight Analysis and Components Breakdown 54

5.3 Research Questions Summary 57

6 CONCLUSION 59

REFERENCES 61

APPENDICES

A PROPULSION ARCHITECTURES CONSTRUCTION 65

A.1 Series Hybrid Architecture Powertrain 65

A.2 Parallel Hybrid Architecture Powertrain 67

B ONGOING RESEARCH AT THE DLR IN ARCHITECTURE OPTIMIZATION 69

C SOURCE CODE: OPEN ACCESS 73

C.1 GitHub Repo 73

C.2 Dynamic Propulsion Architecture Class 73

C.3 Dynamic Aircraft Model Class 78

LIST OF TABLES

TABLES

Table 4.1 Powertrain technology assumptions 34

Table 4.2 Optimization problem formulation of this study 36

Table 5.1 MDO Results Summary: Minimum Fuel Burn and MTOW 56

LIST OF FIGURES

FIGURES

Figure 1.1 Principal emissions from aviation (direct and non-direct) and how they impact climate change, reproduced from [1]. 2

Figure 2.1 Knowledge about design vs. Time into design process [2]. 9

Figure 2.2 Conventional vs. Design Optimization Process [3]. 11

Figure 2.3 General Description of Optimization Problem Formulation[3]. . 12

Figure 3.1 Beechcraft KingAir C90 GTi (photo by Joao Carlos Riberio)[4]. 15

Figure 3.2 Conceptual Diagrams of Propulsion System Architectures. . . . 17

Figure 3.3 Logic of the Architecture Builder. 20

Figure 3.4 An Example of the Design Space Model for the propulsion architecture design problem considered in this work. This method allows to explicitly identify possible architectural decisions 22

Figure 3.5 A simple example of a conventional twin propulsion architecture instance using ADORE 23

Figure 3.6 Propeller component details view in ADORE 24

Figure 4.1 Representative Mission Profile [5]. 31

Figure 4.2 The UML diagram depicting the Architecture Builder. 33

Figure 4.3 Reference aircraft model visualization for the KingAir C90 GTi. 34

Figure 5.1 Conventional architecture MDO results. 40

Figure 5.2 Turboelectric architecture MDO results. 41

Figure 5.3 Conventional and Turboelectric Weight breakdown. 42

Figure 5.4 All electric architecture MDO results. 43

Figure 5.5 All Electric Weight breakdown. 45

Figure 5.6 Series hybrid architecture MDO results. 46

Figure 5.7 Series Hybrid Weight breakdown. 47

Figure 5.8 Parallel Hybrid architecture MDO results. 49

Figure 5.9 Parallel Hybrid Weight breakdown. 51

Figure 5.10 The comparative outcomes of parallel hybrid and series hybrid MDO designs. A positive number, specifically denoted by the color red, signifies that the quantity of the parallel hybrid is greater than the quantity of the series hybrid. 52

Figure 5.11 Weight breakdown for 300 nmi and 500 Wh/kg with all architectures. 54

Figure 5.12 Weight breakdown for 500 nmi and 500 Wh/kg with all architectures. 55

Figure A.1 Series hybrid architecture electric power generation elements. . . 66

Figure A.2 Series hybrid architecture mechanical power generation and thrust generation elements. 66

Figure A.3 Parallel hybrid architecture mechanical power generation and thrust generation elements. 68

Figure B.1 Architecture Optimization loop that contains two optimizers. The architecture optimizer manages the suggestion of new architectures and the sizing optimizer manages the performance evaluation of a single architecture. [6] 70

Figure B.2 Design Space Definition including possible architectural choices. [6] 70

Figure B.3 Pareto Front indicating the different Architecture types. [6] . . . 71

Figure B.4 Pareto Front of All Architectures evaluated. [6] 71

LIST OF ABBREVIATIONS

ADORE	Architecture Design Space Reasoning and Optimization Environment
ADSG	Architecture Design Space Graph
AC	Alternating Current
DC	Direct Current
DBSE	Document-Based System Engineering
DoH	Degree of Hybridization
EASA	European Union Aviation Safety Agency
FAA	Federal Aviation Administration
MDO	MutliDisciplinary Design Optimization
MDAO	Multidisciplinary Design Analysis and Optimization
MTOW	Maximum Takeoff Weight
MBSE	Model Based System Engineering
PSFC	Power-Specific Fuel Consumption
SE	System Engineering
SNOPT	Sparse Nonlinear OPTimizer
TLARs	Top-Level Aircraft Requirements

CHAPTER 1

INTRODUCTION

1.1 Motivation

In the contemporary landscape of global emissions, aviation stands out as a significant contributor, responsible for approximately 2.7% of total energy-related CO2 emissions as represented in 1.1. The aviation sector's impact on climate change is effectively doubled when considering the non-CO2 warming effects, primarily from contrail and contrail cirrus cloud formation [1]. This revelation underscores the urgency for transformative policies and technological innovations in the aviation industry [7, 8].

Recognizing this, the European Union, during its climate change conference in Madrid, has expressed a strong commitment to implementing strategies aimed at drastically reducing harmful emissions by the year 2050 [9]. Parallel to these policy initiatives, technological advancements are also making headway. NASA's Subsonic Fixed Wing program, for instance, has set ambitious goals for the next-generation ("N+3") aircraft, anticipated to enter service in the 2030s. These targets include a 55 dB reduction in noise at airport boundaries, a 75% decrease in NOx emissions, and a 70% reduction in fuel consumption compared to 2006 benchmarks [10]. Achieving these objectives necessitates groundbreaking advancements across various sectors, particularly in aviation [11].

Electrification emerges as a particularly promising avenue. Studies funded by NASA have shown that electric propulsion technologies could lead to substantial performance enhancements, aligning with the "N+3" objectives [10]. The rapid progress in electric components and energy storage technologies further increases the feasibil-

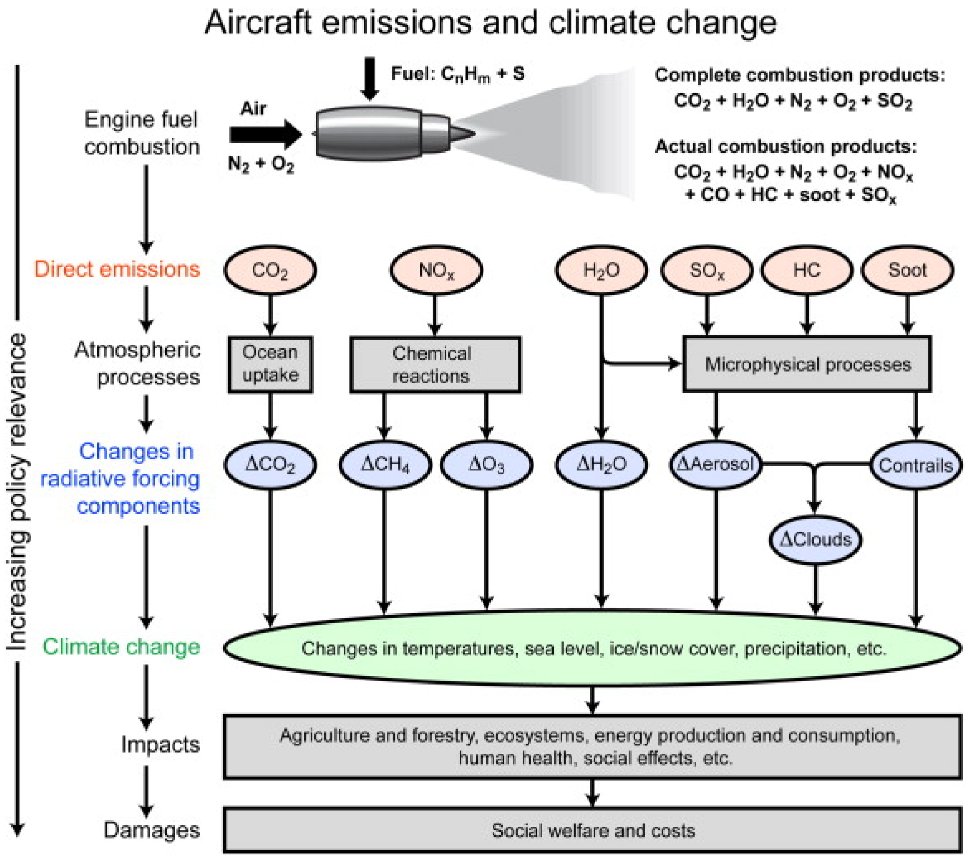


Figure 1.1: Principal emissions from aviation (direct and non-direct) and how they impact climate change, reproduced from [1].

ity of electric propulsion, especially for general and commuter aviation, paving the way towards a greener future for aviation with the possibility of zero-emission flights [12]. Even though the lifecycle assessment of CO2 emissions of partial or all-electric aircraft depends on the power generation strategy, electrification can help eliminate direct pollutants and non-CO2 warming impacts of aviation [1].

1.1.1 Electric and hybrid-electric Propulsion System Design Complexities

The integration of electric propulsion technology into aircraft design introduces a range of complexities. One significant aspect is the expanded design freedom it offers, particularly in terms of the placement and number of propulsors [5]. This flexibility is largely attributed to the scalable nature of electric motors, which differs markedly from traditional aircraft design constraints. Additionally, the advent of novel electric components and their integration strategies necessitates the consideration of innovative architectural designs early in the design process at the conceptual phase [11].

Electric aircraft design also demands a multidisciplinary approach due to the intricate interplay between various engineering disciplines. This interdependence implies that changes in one area can have significant repercussions on others, complicating the parameter definition and analysis of coupled subsystems [13]. Particularly challenging is the integration of aerodynamics, propulsion, and thermal management within electric system designs, posing a significant hurdle for existing analysis frameworks.

The large design space and the imperative for comprehensive subsystem integration present substantial challenges during the conceptual design phase. Designers are tasked with constructing and evaluating numerous architectural possibilities, a process that can be hindered by the limitations in digitalization and automation of the design optimization process. Each unique architecture demands specific attention for its definition and integration, which can be resource-intensive.

Designers often face a dilemma: either narrow the scope of considered designs due to resources and project management constraints, potentially introducing expert bias and subjectivity [14] or allocate extensive resources to manage the repetitive tasks associated with architecture definition and integration. To address these challenges, there is a pressing need for methodologies that enable flexible and rapid descriptions and construction of architectures. Such approaches would facilitate more efficient integration and performance evaluation of the overall vehicle, allowing for more automated exploration of the design space.

1.1.2 Towards Automated Design Space Exploration

The concept of architecture optimization is pivotal in the automated identification of optimal candidate architectures during the early stages of design [15]. Given the high degree of freedom inherent in electric aircraft design, manually pinpointing the best architectures from a range of possible choices is impractical. Architecture optimization entails framing the design challenge as a numerical optimization problem, thereby facilitating automated exploration of the design space.

The effectiveness of architecture optimization is contingent on the ability of the model to quantitatively assess potential architectures [16]. As such, the development of a modular interface that allows for both the automated definition and quantitative evaluation of propulsion architectures is a critical initial step in advancing architecture optimization in electric aircraft design. This approach not only streamlines the design process but also enhances the precision and efficiency of selecting optimal architectures and helps identify potential candidates to be considered for further detailed design phases.

1.2 Thesis Objective

The research objective is: "to advance knowledge and educate stakeholders about system architecture modeling and optimization. This will be achieved by developing a benchmark problem, specifically tailored to electric and hybrid electric propulsion systems architecture design. The approach will integrate the system architecting method into the Multidisciplinary Design Optimization (MDO) process by formulating the design problem as an optimization problem. This initiative aims to enhance understanding and application of complex system architecture in the context of conceptual aircraft design, providing a valuable tool for both researchers and industry professionals."

1.3 Research Questions

The Research Questions (RQ) to be explored in this thesis are outlined below:

- RQ1 How to implement the system architecting method into the conceptual aircraft design process applied to propulsion system design?
  - RQ1.1 What are the requirements to integrate architecture modeling into the design process?
  - RQ1.2 What is the effect of sub-system architecture modeling on the formulation of the design problem and its implementation?

- RQ1.3 What limitation does this approach impose on the design problem?

- RQ2 How to address architectural decisions at the propulsion system level?
  - RQ2.1 Which architectural decisions can be taken into account?
  - RQ2.2 How to address hierarchical architectural decisions in the design problem?
  - RQ2.3 Can the effect of the architecture decisions at the propulsion system level be captured at the aircraft level?

- RQ3 Does this approach enable automated design space exploration?
  - RQ3.1 Can this approach be used to explore a large design space with different decisions at both sub-system and system levels?
  - RQ3.2 What are the objectives and constraints to be used when analyzing such design problems?
  - RQ3.3 What conclusions could be reached using such an automated approach to explore the design space?

1.4 Requirements

To achieve the objective of this research work, a few additional requirements are imposed on the benchmark problem which include:

1. The design variables should include discrete, continuous and categorical variables resulting in a mixed-discrete optimization problem
2. The optimization problem formulation should adapt to design variable hierarchy to account for hierarchical decisions in the design space
3. The performance model is a black box function with no prior knowledge about the behavior of the design space or evaluation function
4. The developed code should be open source to share it with a wider audience in the scientific community and facilitate using it by others

5. The developed framework should be easy to extend to allow further development of the approach such as adding additional disciplines or design considerations.

1.5 The Outline of the Thesis

The thesis is structured as follows: Chapter 2 begins by providing the necessary background to the reader on engineering disciplines relevant to the thesis, particularly focusing on systems engineering (SE), Model-Based Systems Engineering (MBSE), Design Optimization Process and the associated challenges.

Following this, chapter 3 introduces a benchmark problem to advance research and educate stakeholders about system architecture modeling and optimization on the application of electric and hybrid electric propulsion system architecting for conceptual aircraft design. In addition, the methodology used to develop the flexible simulation and evaluation framework is presented. It discusses how propulsion architectures can be constructed using an automated approach, how to define the design space, and which architecture decisions can be included. In addition to remarks regarding the system-level evaluation

Chapter 4 delves into the implementation. It introduces the mission simulation and conceptual aircraft design tool OpenConcept utilized in this research work. It explains the implementation of the approach taken to automatically construct, integrate, and evaluate propulsion architectures. In section 4.5, it concludes with remarks on the optimization problem formulation.

Chapter 5 presents the MDO results for all propulsion architectures in section 5.1 including conventional, turboelectric, all-electric, series, and parallel hybrid. Section 5.2 provides a post-optimality study, where in section 5.2.1 the hybrid configurations. i.e. series and parallel hybrid, are compared in details. Then, a weight analysis is performed to investigate the effect of the different architectures on the total weight of the aircraft at different mission ranges and specific energy assumptions. The thesis concludes with chapter 6, where final thoughts and suggestions are presented.

CHAPTER 2

NECESSARY BACKGROUND

2.1 System Engineering

A system is composed of a set of components and the connections among them, with its overall functionality surpassing that of each of its separate components [17]. This concept gives rise to what is known as emergence, which means that the interaction among the system's components yields a function that exceeds or differs from the functions of its individual components. This system-level function arises when the system operates, and obtaining the desired function is the reason for designing the system [18], i.e. the purpose of the system.

The system engineering process is the undertaking of designing a system considering its purpose by taking into account stakeholder needs and requirements [19]. This process includes requirements definition and analysis, system architecting, and system evaluation.

2.1.1 Document-Based Systems Engineering (DBSE) vs Model-Based Systems Engineering (MBSE)

In Systems Engineering (SE), two predominant methodologies have emerged for the undertaking of SE processes, Document-Based Systems Engineering (DBSE) and Model-Based Systems Engineering (MBSE). DBSE the traditional approach, relies heavily on textual documents to capture, communicate, and analyze system requirements and designs. This approach, while straightforward, often leads to challenges in managing complex information, maintaining consistency across documents, and ensuring effective communication among stakeholders.

In contrast, MBSE relies on digital models as the primary means of information exchange. while both approaches can be used for systems design, there is a transition from DBSE to MBSE as it facilitates a more dynamic and integrated approach, offering enhanced visualization, improved traceability, and a more coherent representation and management of complex systems and related information.

2.1.2 System Architecting process and Architecture Design Space

A System architecture represents a conceptual blueprint that outlines the components of a system and their interconnected relations [17]. The system's function describes what it does and its form describes what the system is. The design engineer or architect is responsible for defining how the system fulfills its function, which entails establishing function-component mapping that describes the complex relations between the system elements [15].

In system design, there are a lot of possibilities to fulfill functions both at the system and subsystem levels. The designer considers the landscape of possible design choices to construct a candidate architecture. Each candidate architecture is a possible candidate to fulfill the system function which is constructed by identifying possible design decisions, selecting among them, and connecting system entities to their functions.

The set of all possible architectural decisions that are considered and can be taken constitutes the architectural design space, consequently, the system architecting process can be viewed as a decision-making process in which possible architectural decisions form a large design space among which candidate architectures can be constructed to form the system architecture.

2.2 Multidisciplinary Design Optimization MDO

Multidisciplinary Design Optimization (MDO) refers to employing optimization techniques to address design problems that involve various disciplines. It is a very effective approach to tackling innovative designs where little experience has been accumulated about the design, as it shortens the time required to design new products and helps gain better design knowledge at the early stage of the design as represented in Figure 2.1.

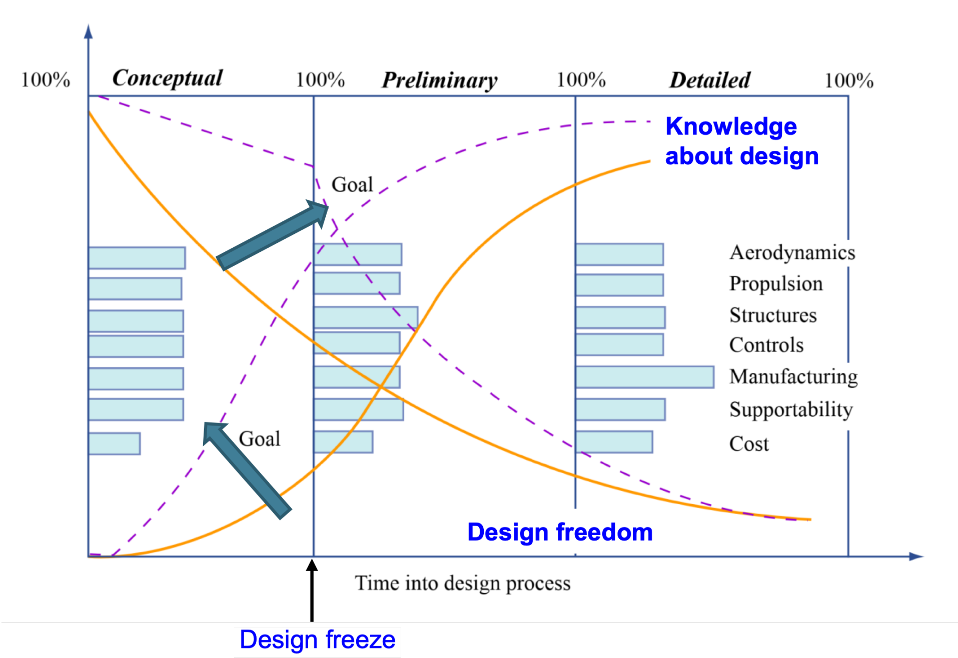


Figure 2.1: Knowledge about design vs. Time into design process [2].

It encompasses the quantitative assessment of the performance of candidate system architectures through the establishment of objectives and constraints that are consistent across all architectures, regardless of their individual structure, functions, and components. MDO is a key enabler for automating the comparison between candidate architectures and integrating the MBSE method into the development process.

2.2.1 Conventional vs. Design Optimization Process

Engineering design is fundamentally an iterative process, essential for developing products that fulfill specific functions. This process becomes increasingly complex for advanced products, such as incorporating electric propulsion in a new aircraft design, necessitating the collaboration of engineering teams across multiple stages, each with its own set of iterative loops.

The design process is typically segmented into distinct phases as shown in Figure 2.1: determining requirements and specifications through market research, customer interviews, and stakeholder needs, generating and considering various system concepts in the conceptual design phase, refining chosen concepts and subsystems in the preliminary design phase, and finalizing every detail in the detailed design phase. Each of these phases involves its own iterations and may require revisiting earlier stages when significant issues arise.

Design optimization emerges as a critical tool in this context, potentially replacing traditional iterative processes to expedite the design cycle and yield superior outcomes. Unlike conventional design, which relies heavily on human intuition and decision-making at various stages, design optimization involves a formal problem formulation, including defining design variables, objectives, and constraints. The evaluation here is strictly numerical, and design changes are algorithmically determined, reducing the need for manual intervention [3]. However, this automated approach still demands significant human expertise, particularly in problem formulation and post-optimization assessment. The two approaches are compared in Figure 2.2.

Engineers play a crucial role in setting objectives, choosing parameters, and enforcing constraints, which significantly influence the optimization outcome. Post-optimality studies are also vital for interpreting results and informing future design improvements.

Incorporating design optimization into various design phases can significantly enhance system performance, reduce costs, and diminish uncertainty more rapidly than conventional methods.

2.2.2 Optimization Problem formulation

The design optimization process necessitates that designers convert their objectives into a mathematical format solvable by optimization algorithms, a step that also enhances their understanding of the design problem. Careful formulation is crucial, as any flaws can lead to the optimizer either failing or converging on a solution that is mathematically sound but practically infeasible or undesirable. Martins et al [3] sug-

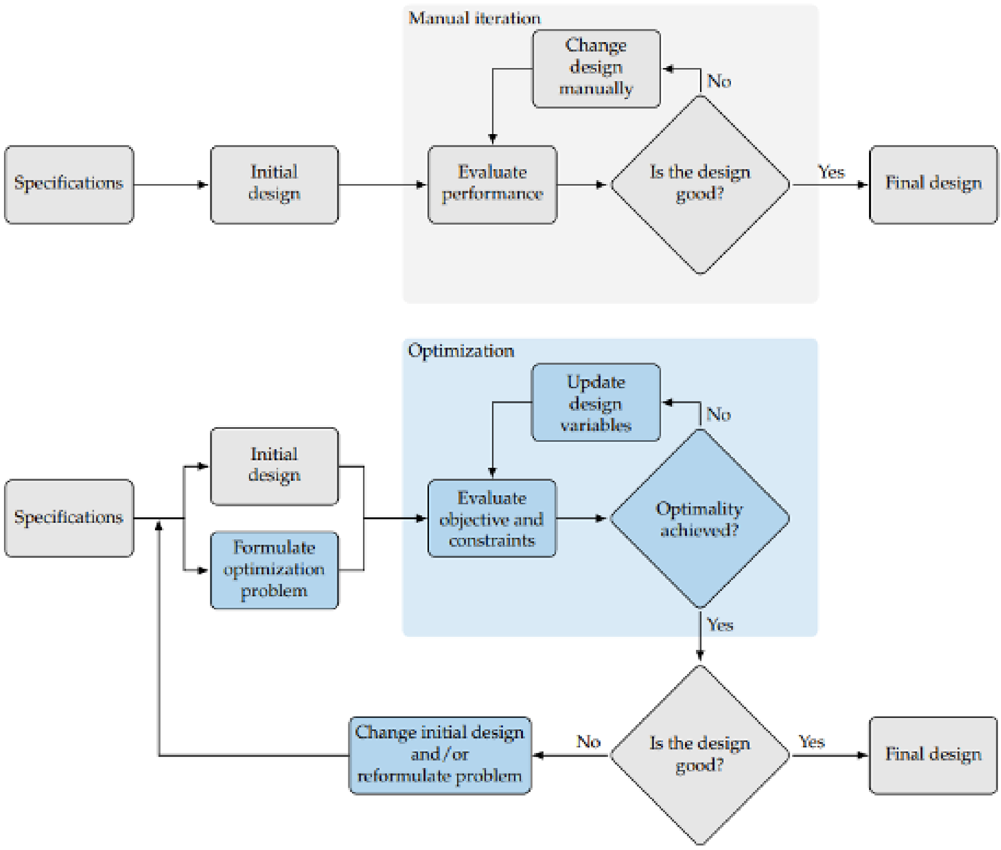


Figure 2.2: Conventional vs. Design Optimization Process [3].

gests the following 5 steps process to formulate the design problem as an optimization problem, Describe the problem, Gather information, Define the design variables, objectives, and constraints. The optimization problem will have the format as shown below in Figure 2.3.

2.3 Challenges and Contribution

Several challenges are highlighted in this section which arise when addressing such an architecting problem. Section 2.3.1 discusses the difficulty of integrating the system architecting method into an optimization process and the motivation for selecting electric and hybrid electric propulsion as a case study. Section 2.3.2 discusses the Model-Based Systems Engineering (MBSE) process and how to connect design requirements, to optimization and evaluation. This section concludes with a description

$$
\begin{array}{lll}
\textit{minimize} & f_m(x) & m = 1, \dots, n_f \\
\\
w.r.t. & \underline{x}_{c,i} \leq x_{c,i} \leq \overline{x}_{c,i} & i = 1, \dots, n_{x,c} \\
 & x_{d,j} \in \{0, \dots, N_j - 1\} & j = 1, \dots, n_{x,d} \\
\\
\textit{subject to} & g_k(x) \leq 0 & k = 1, \dots, n_g \\
 & h_l(x) = 0 & l = 1, \dots, n_h \\
 & c_h(x) = 0 &
\end{array}
$$

Figure 2.3: General Description of Optimization Problem Formulation[3].

of the scientific contribution of this thesis work.

2.3.1 Integrating system Architecting method into MDO

Architectural decisions are usually taken at the early stages of the design without in-depth knowledge regarding their effect on the system's performance. Such decisions are usually taken by industry experts relying on intuition and experience. While it is important to utilize the expertise of the designer to estimate the effect of early decisions on the system's performance, such an approach can sometimes suffer from bias and subjectivity [15]. In addition, in new innovative design problems, such as the electrification of aircraft, the cumulative knowledge regarding the design problem is still progressing, therefore, it is necessary to explore areas in the possible design space to look for new innovative solutions where sometimes limited experience is only available. This requires a methodological approach to explore the design space without excluding potential solutions at the early stage of the design.

The performance model, i.e. evaluation model of a candidate architecture, will generally be an expensive black box function with no previous knowledge about the shape and behavior of the design space. With different types of architectural decisions to consider such as discrete, continuous, categorical, and hierarchical, such architecting characteristics make a challenging class of problems. There is a need to develop an approach that allows the integration of such an architecting process into the conceptual design phase that enables rapid model construction and evaluation of different architecture.

With a detailed review of the literature, it turned out that there is a gap when it comes to including the system architecting phase in the design optimization process. There is a need to develop such an architecting problem to provide assistance in the advancement, assessment, and comparison of approaches utilized in modeling the design space of system architecture.

Electric propulsion aircraft design is a challenging problem characterized by the inherent nature of having a large design space due to the introduction of new components and energy sources into the design problem. Therefore, it would serve as an ideal case study to develop such a design problem to demonstrate the approach for rapid and flexible simulation and evaluation of architectures.

2.3.2 MBSE Model Development and Process Integration

The Model-based Systems Engineering (MBSE) process can be broken down into two main stages: an **upstream phase** encompasses activities such as stakeholder requirement identification, goal establishment, and system architecture design, and a **downstream phase** involves system synthesis, optimization, and performance objective trade-offs.

The primary objective of these two procedures is to ensure that the performance of the developed system aligns with the requirements and needs of the stakeholders. To accomplish this purpose, it is imperative to establish a connection between the upstream and downstream processes. This entails ensuring that the stakeholder requirements identified in the upstream phase can be seamlessly translated into objectives and/or restrictions during the system design phase.

The integration of upstream and downstream processes is anticipated to enhance the inclusivity of the development process for complex systems [20], facilitating the consideration of trade-offs that can eventually enhance system performance.

In this research work, a case study is introduced that allows the description of the design problem as an optimization problem characterized by rapid integration and evaluation of various concepts to satisfy requirements while adhering to design constraints.

2.3.3 Scientific Contribution

Given the inherent difficulties associated with system architecting, it becomes imperative to establish a benchmark problem that may facilitate the advancement and assessment of system architecture design space exploration, in addition to rapid construction and evaluation of candidate architectures.

A case study problem is formulated and refined during the course of the thesis. Its primary objective is to address the existing challenges and also serve as an educational tool for stakeholders to get insights into the system design process. In addition, the benchmark design problem can serve as an initial measure to investigate connecting the earlier and later stages of the Model-Based Systems Engineering (MBSE) process.

CHAPTER 3

PROBLEM DEFINITION AND METHODOLOGY

3.1 Problem Definition

To address the challenges discussed in Chapter 2, a design problem case study is formulated with an application to electric and hybrid electric propulsion system for conceptual aircraft design.

The benchmark problem involves the electrification or hybridization of an existing reference aircraft, the Beechcraft King Air C90GTi presented in Figure 3.1. The commuter aircraft category is selected to allow the investigation of various options of propulsion architectures including the all-electric configuration.

Figure 3.1: Beechcraft KingAir C90 GTi (photo by Joao Carlos Riberio)[4].

The study will focus on the propulsion system architecture as a system of interest. During the process of designing a propulsion system, various architectural decisions are possible. Each of these decisions will affect the performance of the product, i.e. the aircraft in this case. Since the propulsion system is considered a subsystem of the overall aircraft, therefore, to provide a realistic evaluation of a candidate architecture, i.e. a single propulsion architecture, it is necessary to evaluate the performance at the system level, i.e. the aircraft level. The goal is to investigate the effect of architectural decisions at the early stages of the design of the propulsion system on the performance of the overall system, i.e. the aircraft, and compare the performance of various propulsion architectures.

3.2 Methodology

To perform automated design space exploration for the case study of electric and hybrid electric propulsion architecture modeling, a flexible framework that enables automated propulsion architecture construction and evaluation has been developed during this research work.

A propulsion system architecting module is introduced and seamlessly integrated with the established mission analysis toolkit, OpenConcept [5], developed by researchers at Michigan University. This module is responsible for defining and constructing the propulsion system. A reference aircraft model that automatically incorporates the propulsion system into aerodynamic and weight models is dynamically constructed. Utilizing OpenConcept, a mission analysis on this aircraft model is conducted, enabling the calculation of key metrics like fuel consumption and energy usage for the designated propulsion system.

NASA OpenMDAO software [21] is used as the main MDAO framework that combines the different pieces of the design and analysis framework together. This chapter introduces the methodology used to build the framework.

3.2.1 Constructing Propulsion System Architectures

From a system-level perspective, the propulsion system is composed of high-level elements like propellers, batteries, and motors, which work together to generate propulsive power [18]. This system's architecture can be depicted through numerical models of these elements and the information flow between them.

In the approach of this research work, a range of conceptual-level electrical, mechanical, and turbomachinery components are utilized to simulate the propulsion system. These components are interconnected to create various propulsion architecture configurations, such as turboelectric, parallel hybrid, and series hybrid, as illustrated in Figure 3.2. Variations in propulsion system architectures are achieved through the addition, elimination, or reconfiguration of system components.

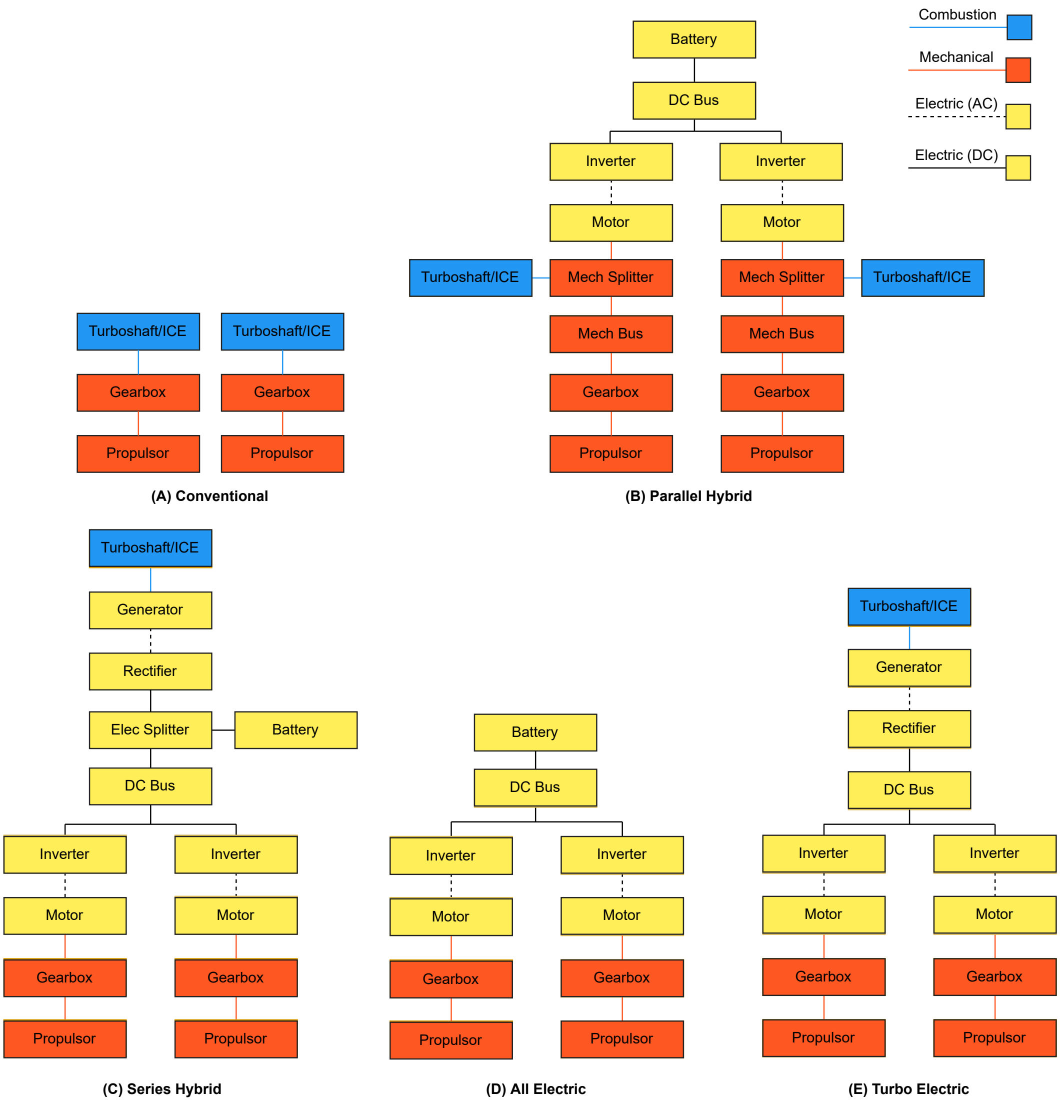


Figure 3.2: Conceptual Diagrams of Propulsion System Architectures.

The propulsion system components considered in this study include:

- **Motor**: produces shaft power by utilizing electric load.
- **Generator**: converts shaft power into electric AC power.
- **Converter**: changes AC to DC power (rectifier) or DC to AC power (inverter).
- **Bus**: manages the distribution of power or loads.
- **Battery**: supplies constant electric DC power with fixed specific energy.
- **Turboshaft**: generates mechanical power from fuel with steady Power-Specific Fuel Consumption (PSFC).
- **Gearbox**: adjusts rotational speed with varying efficiency losses.
- **Propeller**: transforms shaft power into thrust based on an efficiency map.
- **Splitter**: divides power or loads into dual outputs.

Each of these components is characterized by one or more sizing variables. While default values reflecting the current technological state are provided for these inputs, they can be customized by the user.

3.2.2 Automated System Architecture Builder

The architecture constructor assembles various components to form a specific architecture that integrates with an aircraft model. It utilizes a series of classes to define and build the system, categorizing the system into three main types of architectural elements:

- **Elements for Thrust Production**: These elements transform shaft power into thrust.
- **Elements for Generating Mechanical Power**: These components produce mechanical shaft power using either fuel or electric DC power.
- **(Optional) Elements for Generating Electric Power**: These are necessary only if the mechanical power generation requires electric DC power, sourced

either from a battery or an engine sequence (like an engine + generator + rectifier).

These groups collectively offer a comprehensive framework for different propulsion system architectures. The following sections provide a more in-depth exploration of each group.

The thrust generation elements form the foundational aspect of any propulsion architecture. This group includes components like a propeller and, optionally, a gearbox, which are responsible for converting shaft power into thrust.

The mechanical power generation elements are designed to convert fuel or electric DC power into shaft power. There are three subcategories within this group: a standard turboshaft engine, an electric motor, or a hybrid of both. Here, the Degree of Hybridization (DoH) indicates the proportion of shaft power produced by the electric motor (if present) relative to the total shaft power produced. Throttle inputs affect the total power output, varying based on the architecture type; for instance, in a series hybrid, it's applied to the electric motor's rated power, while in a parallel hybrid, it's applied to the combined rated power of both the turboshaft engine and the electric motor.

The electric power generation elements are included when the mechanical power generators require electric DC power. This group has two subcategories, leading to three potential configurations: solely a battery, an engine sequence comprising a generator and rectifier, or a combination of both. In this context, the Degree of Hybridization (DoH) is defined as the ratio of electrical power produced by the battery to the total electrical power generated.

Components within the propulsion system are linked following a functional breakdown of the system's elements. This approach enables users to construct a propulsion system by simply selecting components from each of the three designated groups. The architecture builder then seamlessly assembles these components to create the propulsion system model. The methodology for this interconnection assembly of components is depicted in Figure 3.3. The design of the architecture builder is highly adaptable, enabling the integration of new components to support a broader range of propulsion systems. For instance, introducing a fan into the thrust generation group could serve as an alternative to propellers. Additionally, it's possible to incorporate new types of connection strategies.

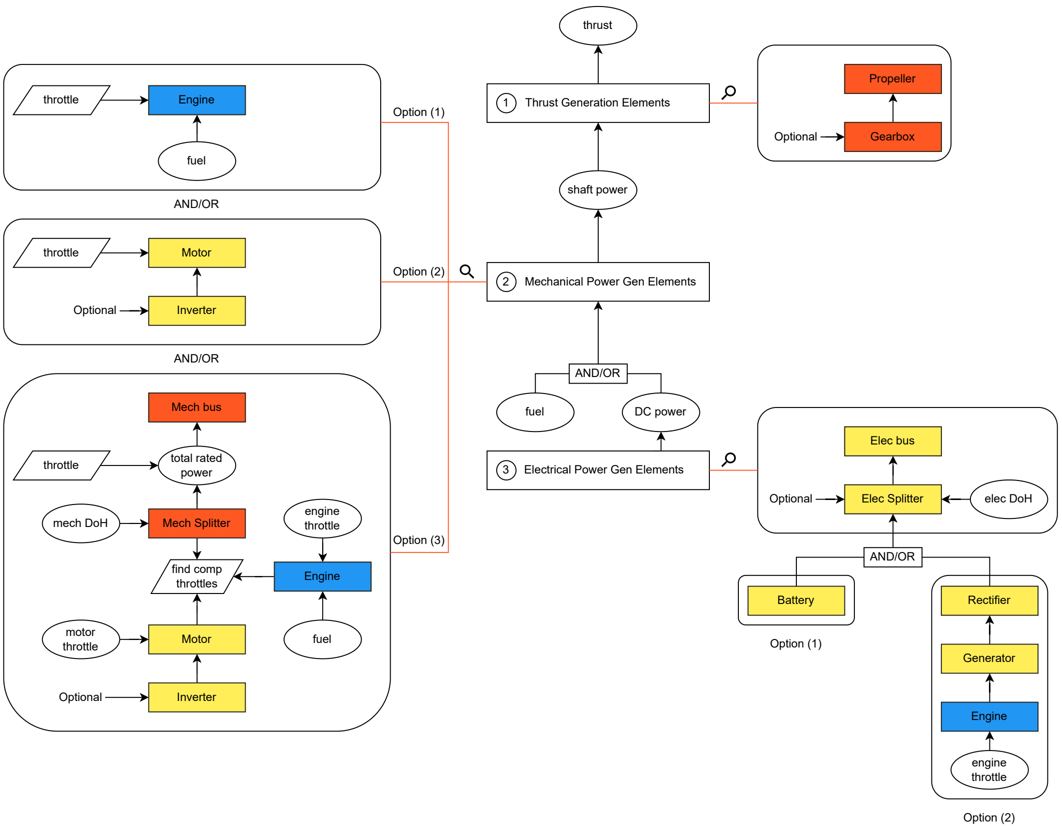


Figure 3.3: Logic of the Architecture Builder.

Presently, the builder is geared towards DC transmission architectures due to their benefits in hybrid electric propulsion [22]. Yet, it's feasible to expand its capabilities to include AC architecture by incorporating a purely electric motor in the mechanical power generation category and a variant of the engine sequence without a rectifier in the electric power category. The existing version of the builder is versatile enough to define architectures with any quantity of propellers, making it suitable for creating various distributed propulsion systems, including all-electric, turboelectric, or series hybrid setups.

3.2.3 Design Space Definition and Identifying Possible Architectural Decisions

The system architecting process can be considered as a decision making process in which the architect identifies possible decisions that can be utilized to fulfill the system function during the product design [16]. Design decisions considered in this study are either continuous, such as the value of the propeller diameter, discrete, such as the number of propellers, categorical, such as the choice of either electric motor, mechanical bus or turboshaft engine to provide shaft power, and hierarchical, which means having a specific choice active because a previous decision is selected, like the need of having an inverter once an electric motor is selected to provide shaft power.

The approach used to model and formalize the design space is presented by Bussemaker et. al. in [15]. The Architecture Design Space Resoning and Optimization Environment (ADORE) tool developed at the German Aerospace Center (DLR) [23] is used to explicitly define the design space and define possible architecture choices. It uses a graph-based formulation to represent the architecture design space and related architectural decisions: The Architecture Design Space Graph (ADSG). Concepts can be represented in the ASDG to describe the design space. Some of these concepts that are used in this paper include:

- **Functions**: specifies what the system should do and can be either solution-neutral or solution-specific;
- **Mapping components to functions**: describes the function a component fulfills and the function it needs (or induces)
- **Component characterization**: describes the number of instances of a component and its attributes
- **Performance Metrics**: specify how architectures can be compared to each other's, can be used as objective or constraint functions in an optimization problem
- **Design variables**: can be associated with functions, components or components' instances to represent sizing parameters

For more details on the ADSG, how it is constructed, and how it is used to create an optimization problem and generate architectures, the reader is referred to [15]. Architecture instances are created from the ADSG by taking architectural decisions.

The design space model starts at the propulsion system boundary function; to provide propulsive power. It contains the different architecting decisions describing explicitly the functions components mapping as presented in Figure 3.4.

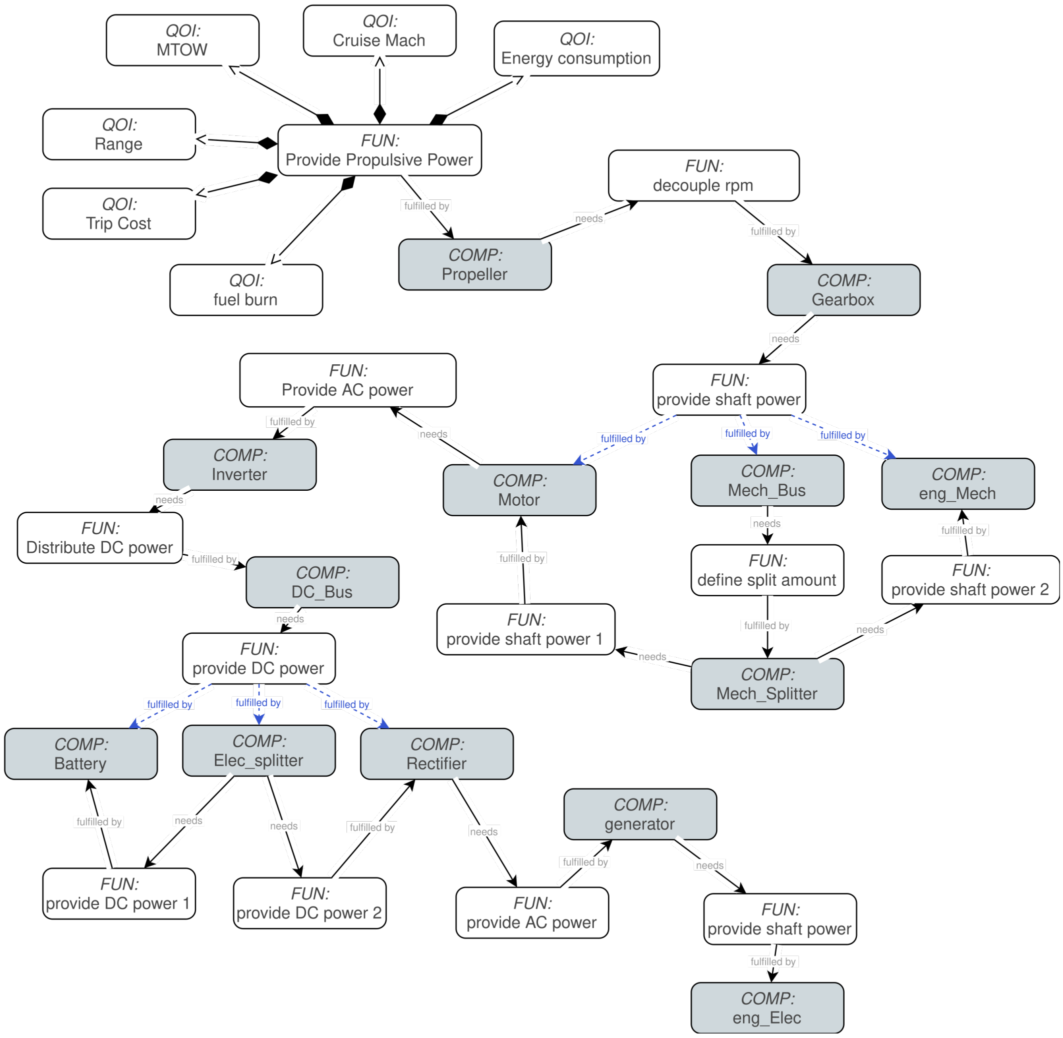


Figure 3.4: An Example of the Design Space Model for the propulsion architecture design problem considered in this work. This method allows to explicitly identify possible architectural decisions

Additionally, the design space model in ADORE provides an intuitive illustration of how the different design decision can lead to different propulsion architecture instances. For example, following the decision-making process of using the propeller component to provide propulsive power, then a gearbox component to decouple the rotational speed, three possible options exist in this design space to provide shaft power to the gearbox component: using a turboshaft engine, an electric motor, or a combination of both components; each of these decisions leads to a different architecture instance. An example of a propulsion architecture instance is provided in Figure 3.5 where the design decisions taken, discrete and continuous, lead to a specific conventional twin propulsion architecture.

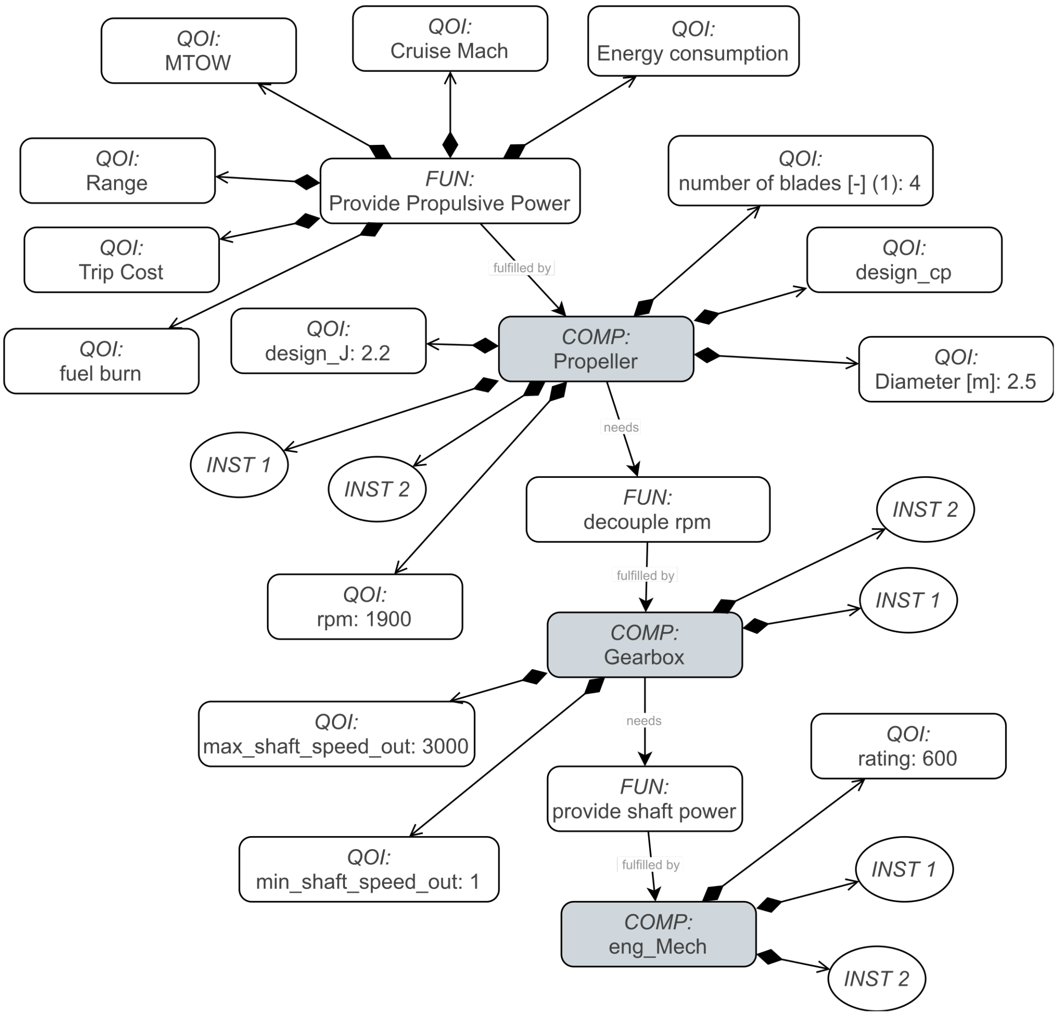


Figure 3.5: A simple example of a conventional twin propulsion architecture instance using ADORE

Component design variables also exist in the architecture design space which describes the attributes and number of instances of specific component, as visualized in

Figure 3.6 for the propeller component.

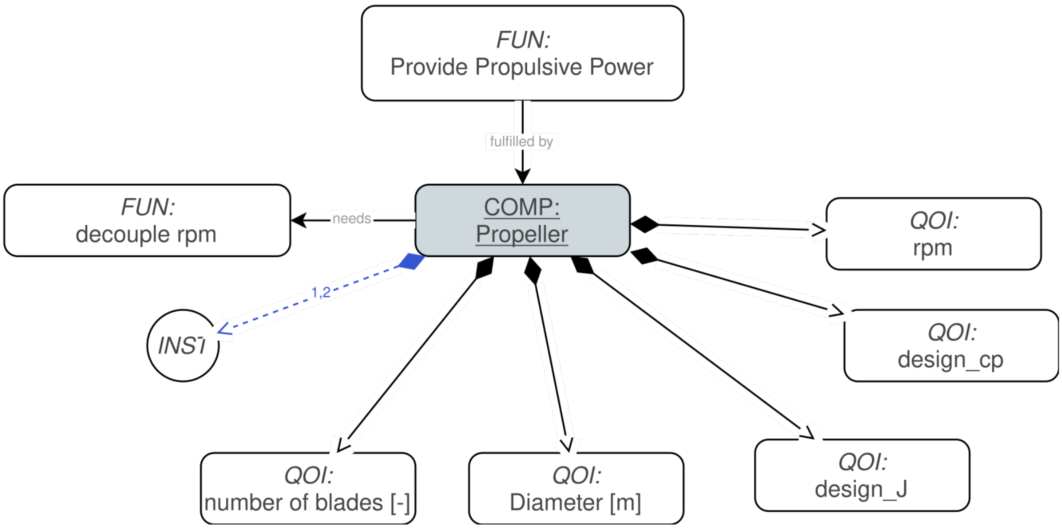


Figure 3.6: Propeller component details view in ADORE

The architecture design space model is constructed by the designer and it provides a generic explicit description of the design space considered which does not necessarily need to include all possible architectural decisions. For example, a designer might decide to only consider decisions related to a turboelectric architecture (a subset of possible architectural choices) in a given design problem, therefore, a different design space model can be constructed to match the designer needs.

After constructing the design space model, function-based architecture instances can be generated as design vectors to start the translation and evaluation processes. In this work, the number of instances of the propeller component is fixed at two to match the reference aircraft model, however, this is only used as a fixed design parameter for this demonstration case. The presented approach allows any number of propellers as long as the evaluation model capabilities can provide a correct estimate of performance outputs to compare different architectures.

3.2.4 Overall system Level Evaluation

Since the propulsion system is only a subsystem of the product intended to be designed, i.e. the aircraft, the performance evaluation must be considered at the overall system level in order to be able to compare the success of different alternative architectures to fulfill the overall system's functions. At the propulsion level, which is the system of interest considered in this research work, the function is to provide thrust, i.e. propulsive power, and the aim is to evaluate the performance of the overall product of interest, which is the aircraft.

Therefore, Conducting a mission analysis at the aircraft level provides an accurate assessment of the performance of a specific propulsion system design. Factors like the efficiency of the propulsion system, the overall weight of the aircraft, and its aerodynamic effectiveness play a crucial role in determining key metrics such as fuel consumption, takeoff weight, and energy expenditure during the mission. The efficiency of the propulsion system directly impacts the weight of the battery and fuel, which, in turn, contributes to the total weight of the aircraft.

These interdependencies create a link between the design of the propulsion system and the primary objectives (like fuel usage, takeoff weight, or energy consumption). A comprehensive mission analysis yields essential data like takeoff weight, fuel usage, and energy consumption, and can be further expanded to calculate additional parameters, such as the cost per nautical mile. These results are instrumental in evaluating and contrasting the efficacy of various propulsion system architectures.

3.3 Research Questions Summary

In this section, the research questions investigated in this chapters are summurized below:

- **RQ2: How to Address Architectural Decisions at the Propulsion System Level?**
  - **RQ2.1: Which Architectural Decisions Can Be Taken into Account?**
    - The architecting problem includes various types of architectural decisions. Design variables include discrete, continuous, and categorical variables. In addition, the architecting process with discrete and categorical variables leads to a decision hierarchy where variables can either be active or inactive based on the decisions taken previ-

ously. This means the evaluation framework needs to ensure that each design vector represents a different architecture, and the numerical model can be interpreted correctly to correspond to the relevant architecture. This is addressed with the implementation of the architecture builder, which can automatically identify the different architecture instances based on user input and construct the corresponding numerical model.

- **RQ2.2: How to Address Hierarchical Architectural Decisions in the Design Problem?**
  - The architecture evaluation framework requires a dynamic method to construct a numerical model for each candidate architecture. The dynamic constructor should include features that allow it to understand the type of architecture being passed and which components are either active or passive in the design vector. In this case, the architecture builder takes the component or architecture description as input and identifies the correct assembly of the components to construct the numerical model of the relevant propulsion system. The system is later integrated into a dynamic aircraft model that is then passed to be evaluated and obtain the performance metric of this specific architecture.

CHAPTER 4

IMPLEMENTATION

NASA's OpenMDAO framework [21] is utilized for the automatic generation of the numerical model and to oversee the information exchange across different models within the aircraft. This software is an open-source, object-oriented platform for multidisciplinary design optimization (MDO), developed in Python. Integrated with this is OpenConcept [4], a toolkit tailored for conceptual aircraft design and optimization. The architecting module, a key part of this integration, is tasked with dynamically creating the propulsion system model that is employed by OpenConcept and constructing the relevant aircraft model as an initial step to perform the mission analysis. This chapter provides a brief description of the tools used and the implementation of the methodology to construct the numerical simulation.

4.1 Conceptual Design and mission analysis: OpenConcept

In this study, OpenConcept is employed to:

- Simulate the propulsion system components and aircraft aerodynamics.
- Offer a numerical base for mission analysis, including numerical integration and mission profiling.

The software toolkit is initially designed for the mission analysis of electric and hybrid-electric fixed-wing aircraft [5], and operates on the OpenMDAO framework [21] developed by NASA. This framework is adept at seamlessly integrating multidisciplinary analysis modules and computing system-level analytic gradients. OpenConcept leverages these analytic derivatives to facilitate swift convergence in problems using gradient-based solvers and to enhance gradient-based optimization efficiency.

An OpenConcept aircraft model inputs design variables (like wing area and propulsion system dimensions), throttle settings, lift coefficients, and flight conditions, and calculates outputs such as thrust, weight, and drag. These design variables typically begin with default settings aligned with the reference aircraft model, including its operating empty weight. The architecture constructor then builds the propulsion system for integration into the aircraft model. Drag calculations are performed using a parabolic drag polar derived from the reference aircraft's parameters.

The `FullMissionAnalysis` mission profile from OpenConcept is utilized in this research. This profile encompasses takeoff, climb, cruise, and descent phases. The takeoff phase includes a balanced field length calculation as detailed by Brelje [5]. The climb, cruise, and descent phases are modeled under steady flight conditions, achieved by adjusting throttle settings and lift coefficients to nullify horizontal and vertical accelerations at each mission integration point. The climb and descent phases follow a specific profile based on predetermined airspeed and vertical speed, while the cruise phase duration is adjusted to ensure the total mission range meets user-defined parameters. A Newton solver is used for system convergence.

4.1.1 Takeoff Analysis

The `takeoff` segment is part of the mission profile considered in this analysis. The module is used to calculate the Balanced Field Length (BFL) and the states of the rated powers of the propulsion architecture. The method to calculate the takeoff distance is described in detail in [24, 5] and a summary of the approach is provided below for convenience. The user provides the control inputs such as throttle input, and the acceleration is calculated using a force balance equation. The takeoff distance is considered in five stages, to understand the stages, let's first define a few velocity terms

- V_0: start speed assumed as 1 m/s
- V_s: stall speed of the aircraft

- V_R: rotation speed assumed as 1.1 V_s
- V_1: takeoff decision speed
- V_2: climb speed assumed as 1.2 V_s

Now given the velocities definition, the takeoff segment is described as follows:

1. Takeoff roll from V_0 to V_1
2. Takeoff roll at One Engine Inoperative (OEI) from V_1 to V_R
3. Rejected takeoff with zero power and max braking from V_1 to V_0
4. Transition in a steady circular arc to the OEI climb-out flight path angle and speed
5. Steady climb at V_2 and OEI power until an obstacle h_0 (Accorinf to certification rules of 14 CFR 23 -> 35 ft) is cleared

The force balance equations for segments 1,2 and 3 is presented in equation 4.1. The stall speed is calcualted as a function of Maximum Takeoff Weight (MTOW) with a seperate model based on reference aircraft, therefore, during the takeoff phase, the velocities, V_0, V_1, and V_R are known. Equation 4.2 can be used to find the distances for different segments depending on the velocities used in the integration. For example, to find the distance for segment 1, i.e. the run-up distance upto decision speed, the start velocity V_0 and takeoff decision speed V_1 are used.

$$\frac{d\vec{V}}{dt} = \vec{T} - \vec{D} - \mu(mg - L). \tag{4.1}$$

$$R_{V1} = \int_{V_0}^{V_1} \frac{dr}{dt}\frac{dt}{dV} dV = \int_{V_0}^{V_1} \frac{V}{a} dV. \tag{4.2}$$

The distance covered during the accelerate-go maneuver is composed of segments 1, 2, 4, and 5, while the distance covered during the accelerate-stop maneuver includes segments 1 and 3. The takeoff module utilizes a Newton solver to modify the designated V1 velocity in order to achieve equal accelerate-go and accelerate-stop distances, or until the accelerate-go distance surpasses the accelerate-stop distance with V_1 equal to V_R. The accelerate-go distance is equivalent to the balanced field length, which can be utilized as an optimization or sizing variable. The takeoff module does not account for the marginal variation in aircraft weight resulting from fuel consumption during takeoff. However, it does monitor the overall fuel (and electricity, if applicable) consumed during the takeoff roll [5].

4.1.2 Mission Profile

The `mission` module serves two primary functions as explained in [5]: establishing control inputs that are determined by conditions to maintain stable flight, and integrating variables such as fuel consumption and energy expenditure. An OpenConcept mission presently comprises three segments, but it can readily be expanded to encompass six segments, thereby incorporating a reserve mission.

- Climb: constant vertical speed and indicated airspeed
- Cruise: constant indicated airspeed and altitude
- Descent: constant indicated airspeed and vertical speed

The airplane is seen as a point mass, which undergoes changes as fuel is consumed, during the flight profile. The residual equation calcualtes the thrust residuals at each flight condition using equation 4.3.

$$\vec{R}_{\text{thrust}} = \vec{T} - \vec{D} - m\vec{g}\sin(\gamma). \tag{4.3}$$

The Newton solver in OpenMDAO achieves convergence by adjusting the major thrust control settings (throttle) until the residuals reach zero. Figure 4.1 provides an example of the simulation for the analysis for a hybrid-electric aircraft profile that demonstrates the flight conditions and states of the vehicle.

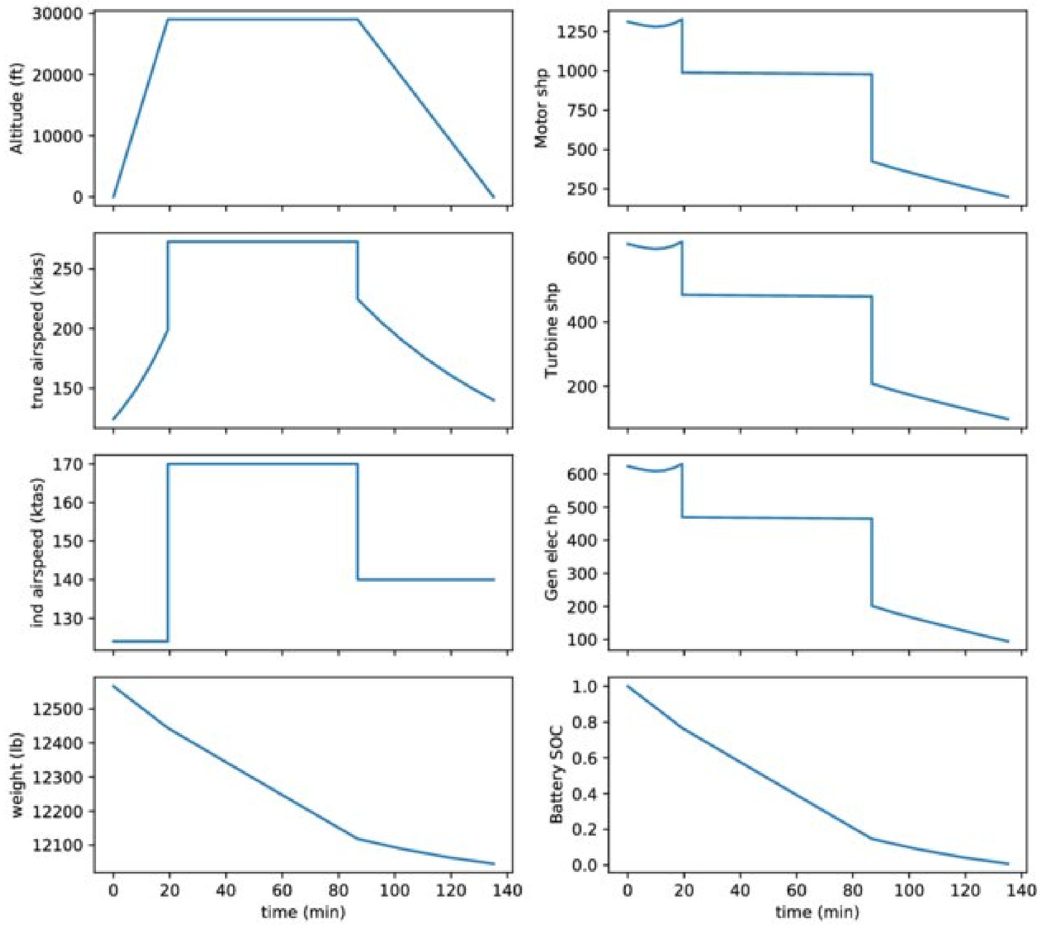


Figure 4.1: Representative Mission Profile [5].

4.1.3 Numerical Intergration

For numerical integration, OpenConcept applies Simpson's Rule in an all-at-once method to integrate state variables as decribed in [5]. The integral is estimated using Simpon's rule as described in equations 4.4 and 4.5 where N is the number of Simpson subintervals, and Δx denotes the uniform spacing between the points.

$$\int_{x_L}^{x_U} f(x)\,dx \approx \frac{\Delta x}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 2f_{2N-2} + 4f_{2N-1} + f_{2N}\right] \tag{4.4}$$

$$\Delta x = \frac{x_U - x_L}{2N} \tag{4.5}$$

OpenConcept employs vectorized calculations in each mission segment for enhanced performance. This approach precludes the use of time-marching ordinary differential equation integration methods, as vectorized calculations require simultaneous computation. The integrator processes essential variables like fuel flow and airspeed to determine mission-critical metrics such as fuel weight and distance traveled. Additionally, the integrator can be combined with other OpenConcept components for advanced analyses, including precise battery temperature modeling and dynamic thermal management system studies [25].

4.2 Constructing an Automated Numerical Propulsion Architecture model

The UML diagram of the architecture builder, which is divided into two separate segments, is illustrated in Figure 4.2. The first part called the `PropSysArch` class, describes the propulsion system, including all of its parts and how they work together. The next section, referred to as the `DynamicPropulsionArchitecture` OpenMDAO group, is responsible for developing the hierarchical framework of the propulsion system model in a way that is consistent with OpenConcept. Within this framework, every part of the propulsion system is represented as a data class, including its inputs and properties, with OpenConcept supplying the basic numerical model. The arrangement described is crucial in facilitating modifications to the numerical model in order to account for different degrees of precision and fidelity.

4.3 Multidisciplinary Design Analysis and Optimization (MDAO)

Finally, several components must be integrated to create an airplane model appropriate for mission analysis in OpenConcept. The first step in this process is to create a baseline airplane, add models for aerodynamics and weight, and use the architecture builder to design a propulsion system. Following that, the propulsion system is incorporated into the overall aircraft model. Afterward, a sizing optimizer is utilized, incorporating the essential design factors and limitations that are tailored to each individual propulsion architecture.

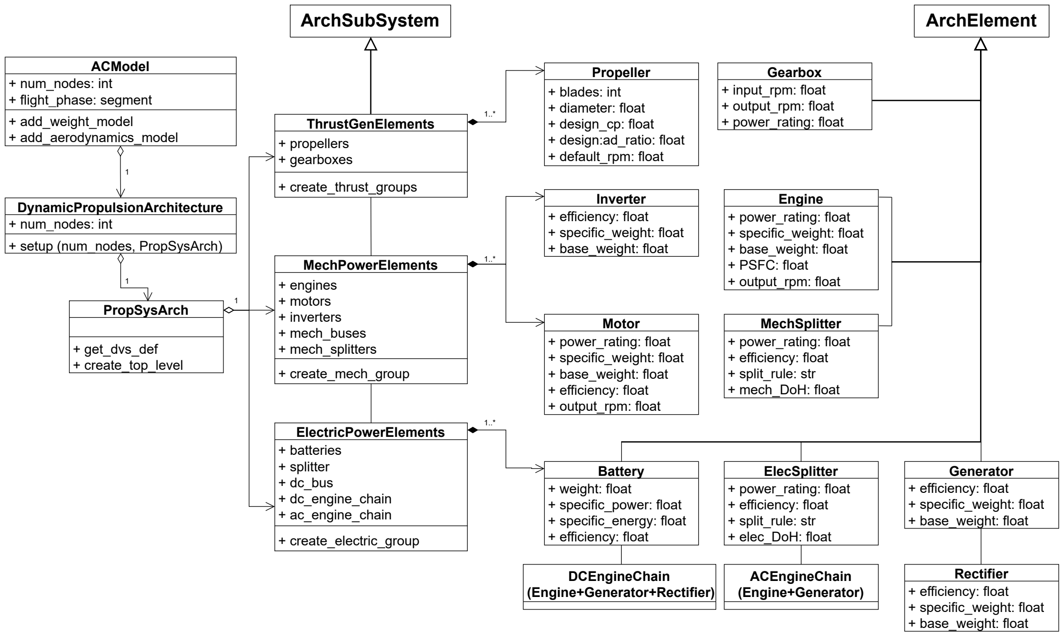


Figure 4.2: The UML diagram depicting the Architecture Builder.

The propulsion system variables, such as the specific power of the electric motor, are assigned default values that correspond to the latest developments in the field. The aerodynamic coefficients, specifically the zero-lift drag coefficient (C_{D0}) and the maximum lift coefficient ($C_{L,\,max}$), are adjusted to match those of the reference aircraft model. The final outcome of this procedure, which is a proficient evaluation of the mission, produces the necessary metrics for evaluating the performance of the selected propulsion system at the aircraft level. These indicators contain important data, including the maximum takeoff weight (MTOW), fuel consumption, and overall energy expenditure.

4.4 Reference Aircraft Model and Technology Assumptions

The Beechcraft King Air C90GT is used as a reference aircraft to test the framework and implement the hybridization or electrification on the base model. A model is initially constructed to test the performance of the reference fuel-based aircraft compared to the real aircraft on a typical mission to validate the initial calculations of the mission analysis toolkit.

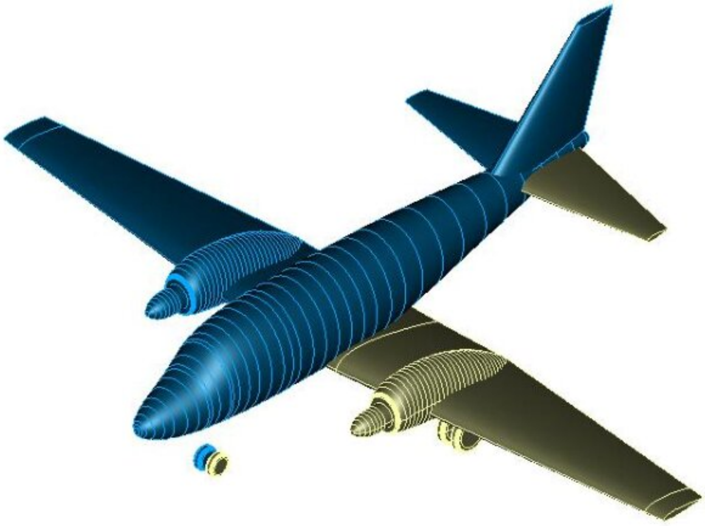


Figure 4.3: Reference aircraft model visualization for the KingAir C90 GTi.

A demonstration of the developed framework and modeling methodology is provided by implementing five unique propulsion architectures. The layouts illustrated in Figure 3.2 contain a range of propulsion systems, including traditional, all-electric, turboelectric, series hybrid, and parallel hybrid twin systems. Similar twin-propeller configurations are incorporated into the design of each system, mirroring those found on the King Air. The performance of each architecture is assessed by considering different mission distances and varied specific energies of the batteries. Technology assumptions incorporated in each system are presented in Table 4.1.

Table 4.1: Powertrain technology assumptions

Component	Specific Power (kW/kg)	Efficiency	PSFC (Ib/hp/hr)
Battery	5.0	97%	-
Motor	5.0	97%	-
Generator	5.0	97%	-
Converter	10.0	97%	-
ElecBus	-	99%	-
Turboshaft	7.15[1]	-	0.6
MechBus	-	95%	-

[1]does not include 104 kg base wt

4.5 Optimization Problem For All Architectures

The optimization procedure utilizes the SNOPT optimizer [26], which is accessed through the pyOptSparse interface provided by OpenMDAO [27]. The design variables considered in this study encompass the rated outputs for both the turboshaft engine and/or electric motor, the maximum takeoff weight (MTOW), the weight of the battery, the diameter of the propeller, and the level of hybridization (DoH) during the cruise phase. In order to adhere to the stall requirements of the reference aircraft, the wing loading is kept constant by modifying the wing area in accordance with variations in maximum takeoff weight (MTOW).

Specific limitations are implemented in order to prevent the turboshaft, electric motor, and battery from exceeding their maximum rated outputs. The specific power of the battery remains constant at 5 kW/kg, however, its specific energy is evaluated within a range of 300–800 Wh/kg. To prevent exceeding the maximum takeoff weight (MTOW) of the reference aircraft model by a substantial margin, a strict upper limit of 5,700 kg has been established. Furthermore, it is worth noting that this restriction is in accordance with the regulations set forth by the European Union Aviation Safety Agency (EASA) and the Federal Aviation Administration (FAA). These regulations specify that pilots must get a type rating in order to operate aircraft with a maximum takeoff weight (MTOW) exceeding 5,700 kg.

The optimization objective is designed to be universally applicable to all architectures, with an initial emphasis on minimizing fuel consumption. In cases when fuel burn is insignificant, the secondary objective is to reduce the maximum takeoff weight (MTOW). The goal function incorporates a calibrated weighting of the maximum takeoff weight (MTOW) to account for the discrepancy in energy density between batteries and aviation fuel. It is noted that aviation fuel possesses an approximate energy density of 12,000 Wh/kg. The general description of this optimization problem is presented in Table 4.2.

The optimization problem is dynamically customized based on the propulsion architecture supplied by the user, resulting in the incorporation or exclusion of certain design variables and constraints. In architectural designs that integrate a battery, the

Table 4.2: Optimization problem formulation of this study

minimize:	fuel burn + 0.01$MTOW$
by varying:	
	$MTOW$
	d_{prop}
	$W_{battery}$
	P_{motor} (rated)
	$P_{turboshaft}$ (rated)
	DoH_{cruise} (degree of hybridization w.r.t power at cruise)
subject to scalar constraints:	
	$R_{TOW} = W_{TO} - W_{fuel} - W_{empty} - W_{payload} - W_{batt} \geq 0$
	$SOC_{batt} \geq 0$ (battery's state of charge at the end of the mission)
	BFL $\leq$ 4,452 ft (no worse than baseline)
and vector constraints:	
	$0 \leq throttle \leq 1$
	$\vec{P}_{motor} \leq P_{motor}$ (rated)
	$\vec{P}_{turboshaft} \leq P_{turboshaft}$ (rated)
	$\vec{P}_{battery} \leq W_{battery} \cdot p_b$

weight of the battery assumes the role of a design variable, with accompanying limits imposed on its energy and power consumption. This methodology enables the adaptable design of the optimization problem, accommodating the distinct demands of various architecture optimization investigations.

4.6 Research Questions Summary

In this section, the research questions investigated in this chapters are summurized below:

- **RQ1 How to implement the system architecting method into the conceptual aircraft design process applied to propulsion system design?**

- **RQ1.1 What are the requirements to integrate architecture modeling into the design process?**
  - To integrate the system architecting method into a conceptual design process, it is necessary to have an automated approach for the generation, construction, integration, and evaluation of system architectures. The focus of this research work is on the construction, integration, and evaluation. The automated construction step is addressed by implementing the architecture builder logic that enables the rapid construction of propulsion systems. The integration is achieved by implementing a dynamic aircraft model that takes the propulsion system as input and provides a model ready for performance evaluation. The evaluation is performed through a mission analysis toolkit to get performance metrics such as fuel and energy usage, in addition to the aircraft's maximum takeoff weight which is obtained through the sizing step.
- **Q1.2 What is the effect of sub-system architecture modeling on the formulation of the design problem and its implementation?**
  - The optimization problem should be dynamically constructed to account for active and inactive design variables resulting from discrete and categorical design decisions' hierarchy. In addition, it should account for all possible design decisions. In addition, the evaluation of a candidate architecture must be performed at the overall system level, i.e. the aircraft level for a realistic evaluation of a subsystem architectural decision.
- **RQ1.3 What limitation does this approach impose on the design problem?**
  - There are limitations in modeling certain architectures imposed by the evaluation framework capabilities, for example, modeling and evaluating a distributed propulsion architecture requires the evaluation model to include the capability to evaluate the aerodynamic behavior of the configuration and related changes in aircraft geometry. Moreover, significant efforts need to be employed to develop the au-

tomation necessary to run such design problems. this requires interfacing with various software tools and employing multidisciplinary knowledge to make sure that the evaluation framework provides a correct evaluation for the system of interest. Lastly, issues with convergence, which is an inherent feature when working with optimization algorithms, remain a significant challenge that needs to be addressed in upcoming research.

- **RQ3 Does this approach facilitate automated design space exploration?**
  - **RQ3.2 What are the objectives and constraints to be used when analyzing such design problems?**
    - The objective used a weighted balance between fuel usage and a fraction of the maximum takeoff weight that corresponds to a factor of the energy content of fuel and battery. The rationale behind this objective selection is to allow the evaluation of the various types of architectures presented in Figure 3.2. The constraints are applied to the rated power of the electric components to make sure that they do not exceed their maximum power. Moreover, the battery state of charge is checked to make sure that there enough enough battery power during the flight. In addition, to make sure that the aircraft performance matches the takeoff characteristics of the reference model, a constraint is applied to the takeoff distance. The throttle input has to always correspond to a realistic value between 0 and 1. The formulation ensures that the design variables and obtained metrics are within reasonable physical bounds.

CHAPTER 5

RESULTS AND DISCUSSION

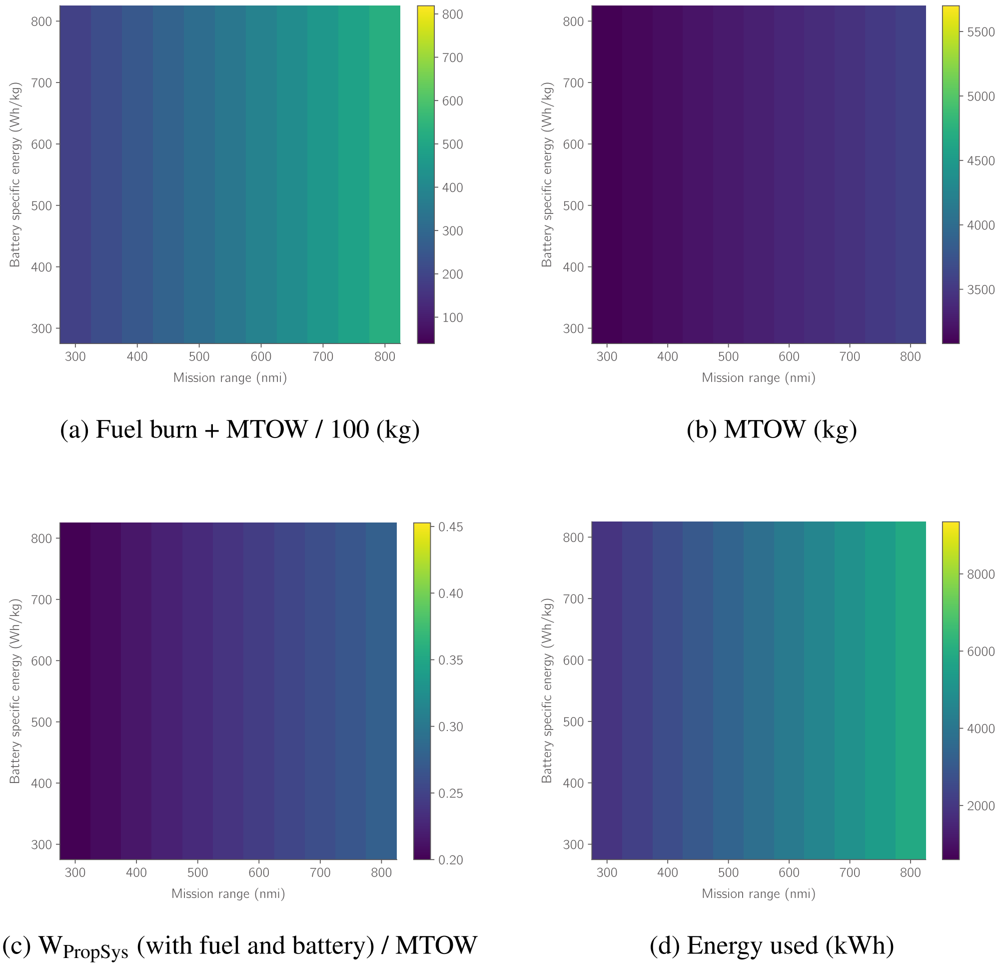
This chapter contains the MDO results of the various propulsion architectures considered in this research as a case study example on a reference model of the King Air C90GT aircraft. The goal is to demonstrate the suggested approach to enable flexible and automated simulations, in addition to integrating the system architecting methods into the conceptual design process. It is crucial to specify that the objective of this research is not to validate or invalidate the viability of incorporating electric propulsion into the redesign of light aircraft. The primary objective is to demonstrate an automated approach that expedites the integration and evaluation of several propulsion systems within a conceptual design and optimization framework.

5.1 MDO Architectures Results and Discussion

The reference aircraft conceptual model is validated, as described in [5]. A comprehensive analysis is conducted for each propulsion architecture, encompassing a matrix consisting of 121 distinct scenarios. These scenarios involve the manipulation of mission range and battery-specific energy within the ranges of 300 to 800 nautical miles and Wh/kg, respectively, with incremental adjustments of 50 units. The optimization of conventional and turboelectric systems is primarily focused on determining the most suitable range of mission distances, without taking into account changes in battery-specific energies. The optimization procedure, which entails the evaluation of a certain architecture, is typically concluded within a duration of approximately two minutes on a standard laptop, with a total of more than 500 optimization cases performed.

5.1.1 Conventional and Turbo-Electric Propulsion Architectures

In the context of conventional and turboelectric systems, the optimization procedure is utilized to determine the optimal rated power for the turboshaft. The conventional architecture's maximum takeoff weight (MTOW) of 3,576 kg is observed at the maximum range of 800 nautical miles, as seen in Figure 5.1b. The aforementioned value exhibits a notable decrease of 26% when compared to the maximum takeoff weight of the turboelectric architecture, which is 4,838 kg, as depicted in Figure 5.2b.



(a) Fuel burn + MTOW / 100 (kg)

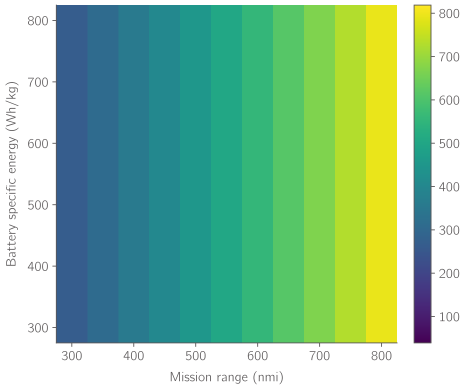
(b) MTOW (kg)

(c) $W_{PropSys}$ (with fuel and battery) / MTOW

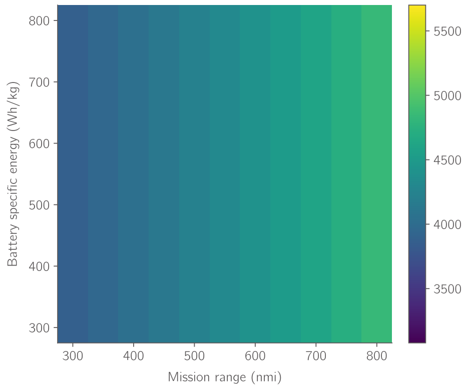
(d) Energy used (kWh)

Figure 5.1: Conventional architecture MDO results.

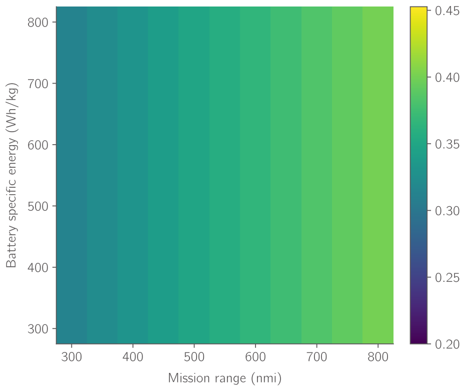
Transitioning from a traditional to a turboelectric system at this particular configuration yields a notable 50% increase in fuel consumption, rising from 491 kg to 743 kg. This corresponds to energy consumption of 5,867 kWh and 8,883 kWh, respectively, as depicted in Figures 5.1d and 5.2d. The observed increase can be attributed to two factors: the additional weight and the reduced efficiency in the propulsion system, which is a result of the supplementary energy conversion stages. The percentage of the overall weight allocated to the propulsion system, encompassing fuel, increases from 30% in the traditional configuration to 40% in the turboelectric scenario.



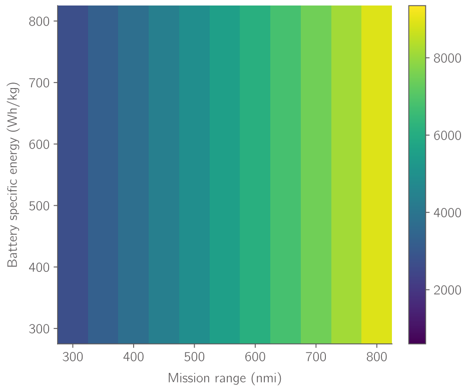
(a) Fuel burn + MTOW / 100 (kg)



(b) MTOW (kg)



(c) $W_{PropSys}$ (with fuel and battery) / MTOW

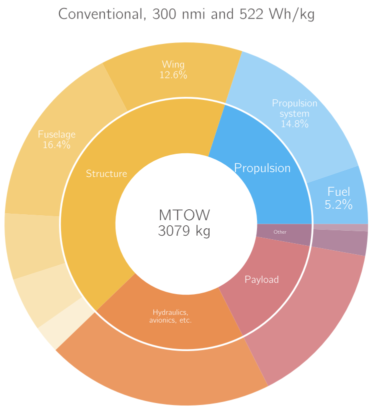


(d) Energy used (kWh)

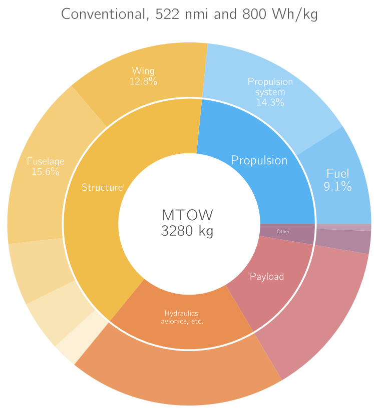
Figure 5.2: Turboelectric architecture MDO results.

The greater weight of the turboelectric arrangement attributed to the incorporation of additional components such as generators, converters, motors, and supplementary structural components, can also be observed in Figure 5.3 where the weight breakdown of each architecture is presented at a specific design point, namely 300 and 522 nmi. It is shown that at 300 nmi, changing the architecture from conventional to turboelectric results in 25% increase in the total weight where the propulsion system contribution to the weight rises from 14.8% to 25.4%.

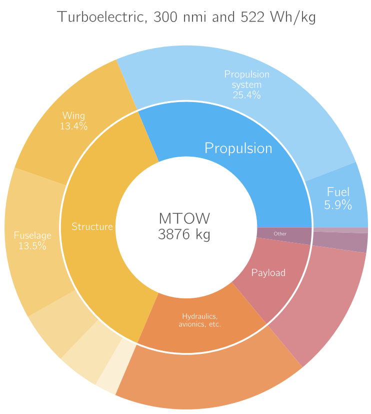
The results of this study are consistent with the anticipated outcome that the turboelectric design exhibits lower efficiency compared to the traditional architecture, particularly in situations where no supplementary benefits in aerodynamic or propulsive efficiency, such as those provided by distributed electric propulsion, are gained. There is a clear and consistent pattern of weight gain and higher energy usage observed when examining Figures 5.1 and 5.2 across all design points.



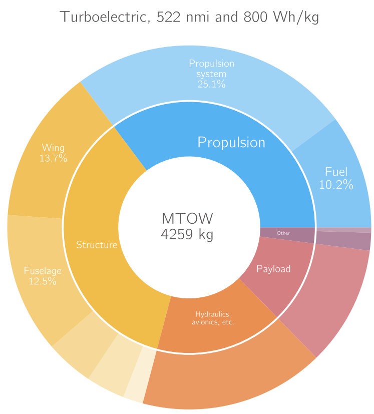
(a) Conventional Weight Breakdown 300 nmi



(b) Conventional Weight Breakdown 522 nmi



(c) Turboelectric Weight Breakdown 300 nmi

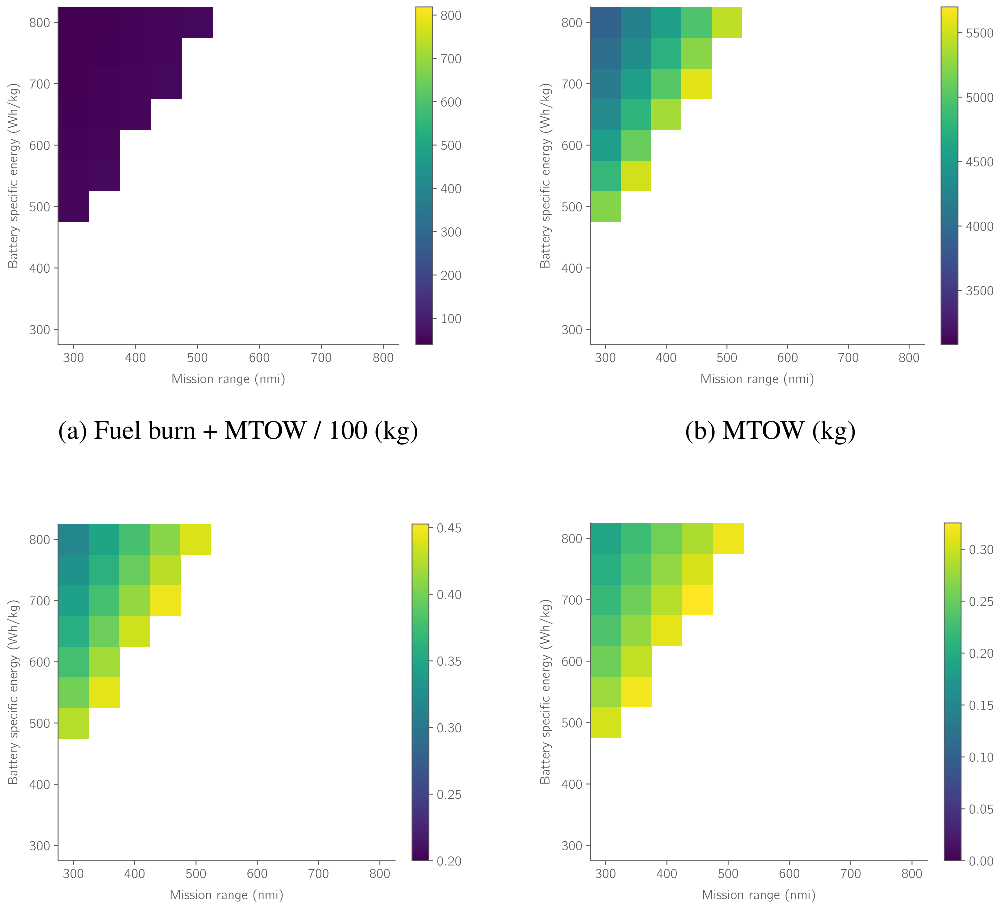


(d) Turboelectric Weight Breakdown 522 nmi

Figure 5.3: Conventional and Turboelectric Weight breakdown.

5.1.2 All Electric propulsion Architecture

The all-electric design demonstrated satisfactory convergence in a mere 21 scenarios. The unfeasible regions, denoted in Figure 5.4 by vacant spaces, are those in which no combination of design variables could permit the aircraft to accomplish the mission without surpassing the component's maximum takeoff weight (MTOW) and power output thresholds. The all-electric architecture MDO results in presented in Figure 5.4.



(a) Fuel burn + MTOW / 100 (kg)

(b) MTOW (kg)

(c) $W_{PropSys}$ (with fuel and battery) / MTOW

(d) Battery weight / MTOW

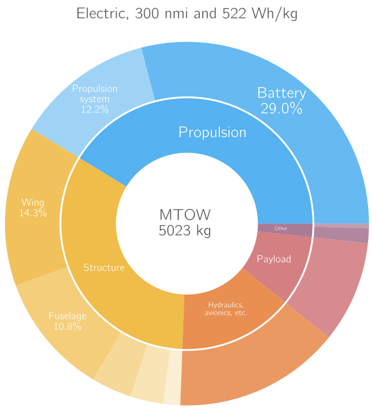
Figure 5.4: All electric architecture MDO results.

With a mission range of 350 nautical miles and a battery specific energy of 550 Wh/kg, the maximum takeoff weight (MTOW) of the aircraft is 5,516 kg, which is only 184 kg below the prescribed upper limit. In order to have a higher range, it is important to improve the specific energy of the battery. The all-electric model represented in Figure 5.4b reaches a maximum takeoff weight (MTOW) of 5,212 kg, enabling a range of 300 nautical miles. This performance is made possible by the utilization of a battery with a specific energy of 500 Wh/kg. The aforementioned figure demonstrates a significant increase in weight of 69% when comparing the current model to the conventional model with equivalent range. According to this model, the propulsion system, including the battery, accounts for more than 40% of the whole takeoff weight.

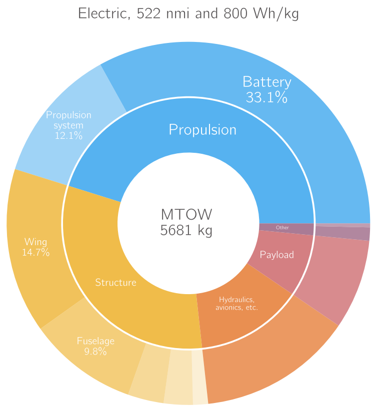
Even when assuming an optimistic scenario of a battery-specific energy of 800 Wh/kg for a range of 300 nautical miles, the all-electric model still exhibits a substantial takeoff weight of 3,908 kg, which is only 14% lighter than the reference aircraft model intended for a range of 1,000 nautical miles.

The dramatic increase in the aircraft weight due to incorporating an all-electric architecture configuration can also be seen in Figure 5.5, where it is shown that the propulsion system accounts for 41.3% of the aircraft weight at 300 nmi with 522 Wh/kg and 45,2% at 522 nmi and 800 Wh/kg. This percentage is significantly higher than both the conventional and turboelectric cases with contributions of 20% and 23.4% for conventional and 31.3% and 35.3 for turboelectric at the same specified ranges.

The selected value of battery-specific energy, denoted as e_b, significantly impacts the design of the airplane model. As an example, when the battery-specific energy is increased from the initial convergence point of 650 Wh/kg at a range of 400 nautical miles, a decrease in maximum takeoff weight (MTOW) is noticed. The recorded values for MTOW are 5,347 kg, 5,014 kg, 4,756 kg, and 4,551 kg. Nevertheless, the decline in question does not exhibit a linear pattern in correspondence with the rise in e_b. The efficacy of weight reduction resulting from an increased e_b gradually reduces as one ascends along the vertical axis depicted in Figure 5.4b. In a similar vein, the weight of the battery, which has considerable importance in determining the overall takeoff weight, exhibits a comparable trend as depicted in Figure 5.4d. The recorded weights for the battery are 1,665 kg, 1,452 kg, 1,288 kg, and 1,158 kg, respectively.



(a) All Electric Weight Breakdown 300 nmi and 522 Wh/kg



(b) All Electric Weight Breakdown 522 nmi and 800 Wh/kg

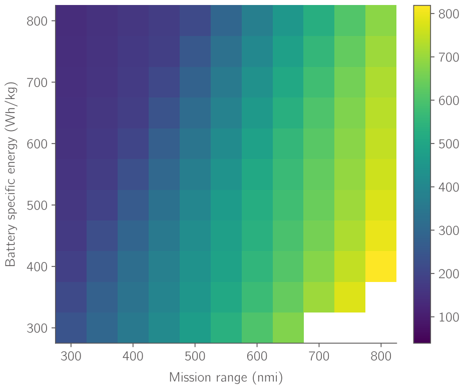
Figure 5.5: All Electric Weight breakdown.

5.1.3 Series Hybrid propulsion Architecture

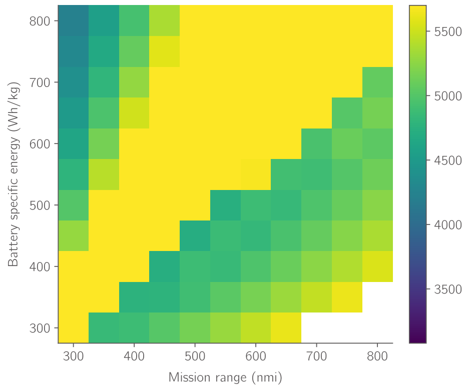
The findings obtained from the optimization of the twin series hybrid model are depicted in Figure 5.6. The optimization approach is successful in determining the proper weights for the battery, electric motor, and turboshaft engine, as well as the optimal degree of hybridization during cruising, in order to minimize the objective function in 97% of cases.

Difficulties develop in situations with greater ranges and lower battery-specific energies, wherein the optimization algorithm encounters challenges in determining a feasible set of design variables that satisfy the limitations, namely the maximum takeoff weight (MTOW) restriction. The present concern arises from the simulation's consistent hybridization rate of 0.4 throughout various mission stages such as takeoff, climb, and descent. This constraint limits the optimizer's capabilities due to the energy requirements associated with these segments. During extended missions characterized by a 0.4 hybridization rate in the aforementioned phases, the weight of the battery reaches a threshold that exceeds the maximum takeoff weight (MTOW) limit.

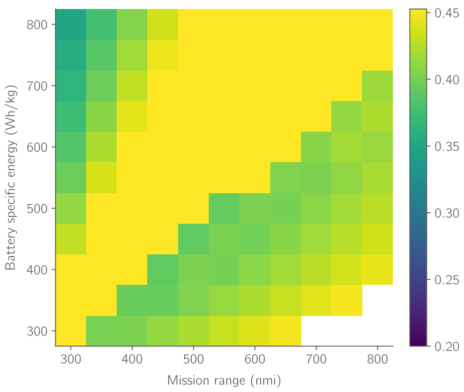
According to the depiction presented in Figure 5.6e, it can be observed that while considering shorter distances and greater specific energies, the optimization process



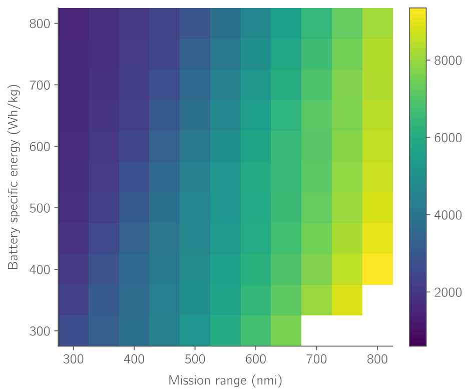
(a) Fuel burn + MTOW / 100 (kg)



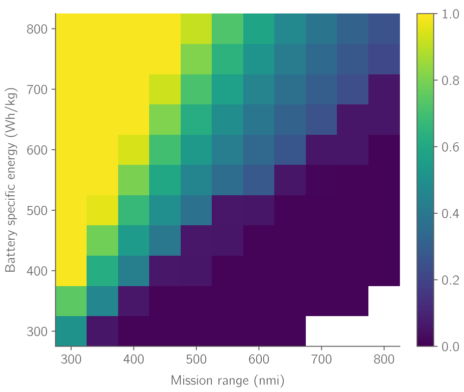
(b) MTOW (kg)



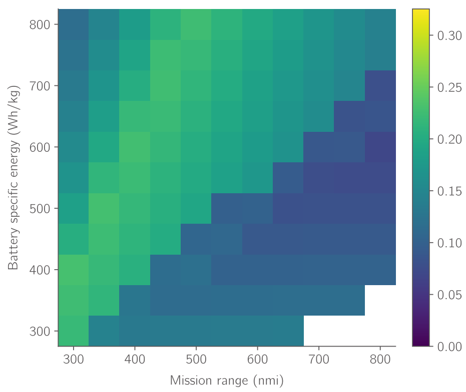
(c) $W_{PropSys}$ (with fuel and battery) / MTOW



(d) Energy used (kWh)



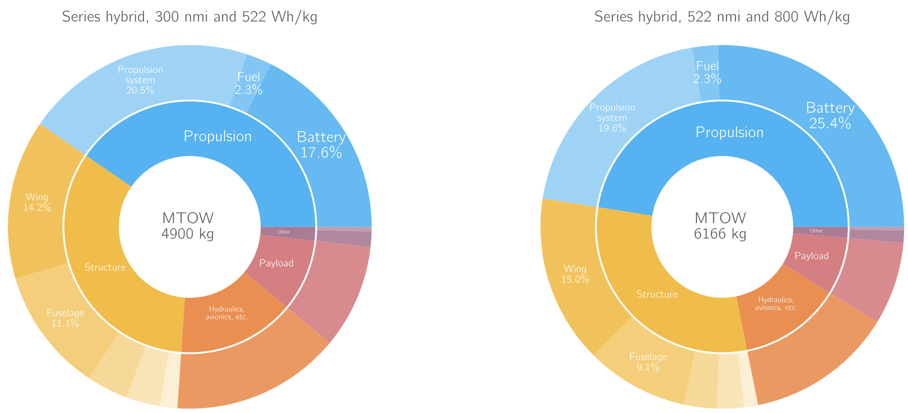
(e) Elec DoH_{Cruise} (0 all fuel, 1 all electric)



(f) Battery weight / MTOW

Figure 5.6: Series hybrid architecture MDO results.

tends to prioritize a flight configuration solely reliant on batteries. Nevertheless, after the maximum takeoff weight (MTOW) threshold of 5,700 kg is attained at an energy density of 500 Wh/kg for a range of 350 nautical miles, the approach is altered to incorporate a fuel quantity that is sufficient to accomplish the mission while adhering to the MTOW restriction. In situations characterized by extended distances and reduced specific energies, as depicted in the lower right quadrant of Figure 5.6e, there is a tendency to prioritize flights that rely solely on fuel. This choice is made in order to decrease the weight of the battery and, subsequently, the maximum takeoff weight (MTOW) component of the objective function.



(a) Series Hybrid Weight Breakdown 300 nmi and 522 Wh/kg

(b) Series Hybrid Weight Breakdown 522 nmi and 800 Wh/kg

Figure 5.7: Series Hybrid Weight breakdown.

The weight of the propulsion system, which encompasses both the battery and fuel components, exhibits variability ranging from 35% to 40% of the overall aircraft weight in fuel-only configurations. In battery-only designs, this weight changes between 35% and 45%, as depicted in Figure 5.6c. The relationship between battery weight and the observed pattern is consistent, as evidenced by the alignment of data points in the top left (battery-only) and bottom right (fuel-only) quadrants of the grid depicted in Figure 5.6f.

The weight breakdown at two different design points, namely 300 nmi with 522 Wh/g and 522 nmi with 800 Wh/kg is presented in Figure 5.7. It is shown that the contribution of the propulsion system weight at these points for the series configuration is at 40.4% and 47.4% which is very close to the all-electric case.

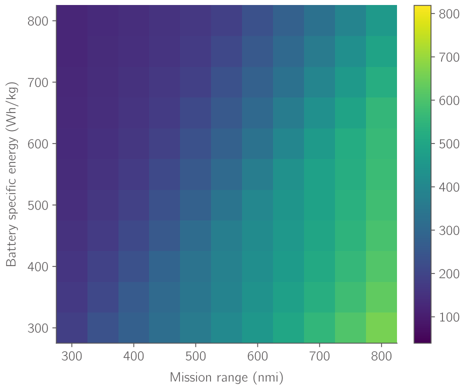
In situations where the aircraft is positioned close to the diagonal of the grid, and the maximum takeoff weight (MTOW) approaches the prescribed limit of 5,700 kg, the optimization algorithm tries to identify the most favorable combination of battery and fuel resources. As an example, while considering a range of 450 nautical miles and a specific energy of 500 Wh/kg, the optimization process aims to achieve a balance between battery usage and fuel consumption in order to prevent going beyond the maximum takeoff weight (MTOW) restriction while simultaneously maximizing the utilization of the battery. The results of this study suggest that employing a hybrid strategy offers significant benefits, particularly in situations where there is a strict maximum takeoff weight (MTOW) constraint. This conclusion aligns with the findings of Brelje's research [5].

5.1.4 Parallel Hybrid Propulsion Architecture

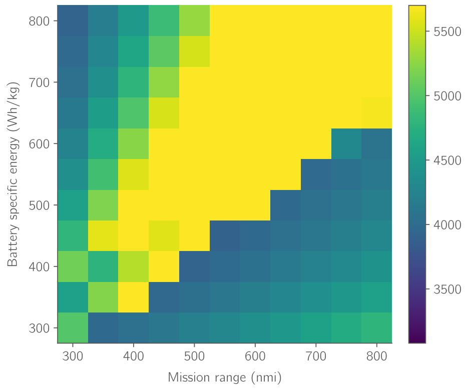
The outputs of the multidisciplinary optimization (MDO) for the parallel hybrid model are depicted in Figure 5.8. The observed trends in the parallel hybrid configuration exhibit similarities to those observed in the series hybrid model. The optimizer favors fuel-only cruising in the lower right quadrant of Figure 5.8e in order to minimize the maximum takeoff weight (MTOW) and comply with design limits. On the contrary, battery-only cruising is favored in the upper left quadrant.

The parallel hybrid vehicle, operating within a range of 400 nautical miles, encounters a maximum takeoff weight (MTOW) constraint. This limitation is reached when the mechanical degree of hybridization (DoH) hits 92% and the battery's specific energy is 500 Wh/kg. The weight of the battery in this configuration is 1,627 kg. The electric motor has been optimized to operate at a power output of 370 kW, while the turboshaft has been tuned to operate at a power output of 539 kW. Moreover, in order to uphold the maximum takeoff weight (MTOW) restrictions, fuel is utilized as a substitute for the battery. At a distance of 450 nautical miles, the mechanical Degree of Hybridization (DoH) experiences a decrease to 78%, as the powers of the electric motor and engine are changed to 311 kW and 453 kW, respectively.

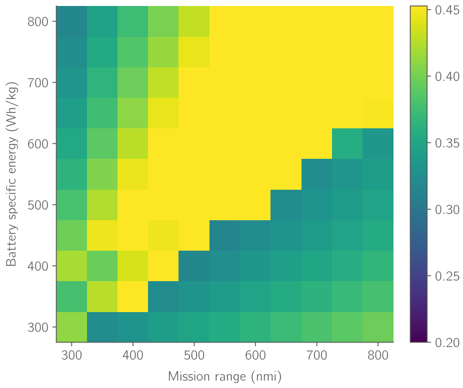
In all scenarios of 400 and 450 nautical miles, with an energy density of 500 Wh/kg, the maximum takeoff weight (MTOW) reaches the upper limit of 5,700 kilograms, as



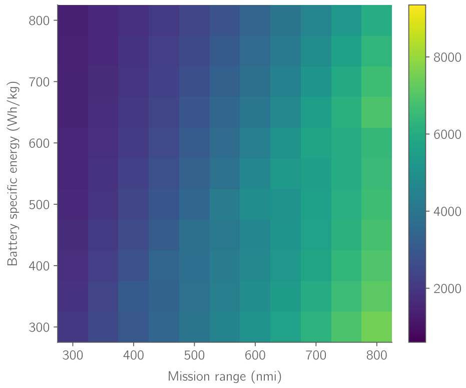
(a) Fuel burn + MTOW / 100 (kg)



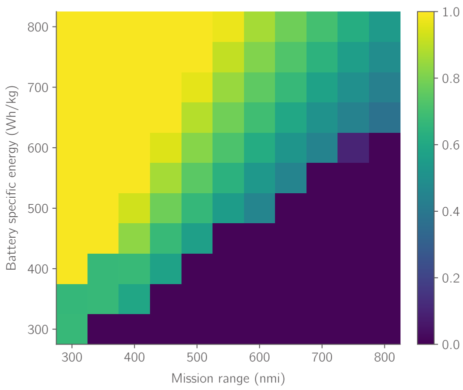
(b) MTOW (kg)



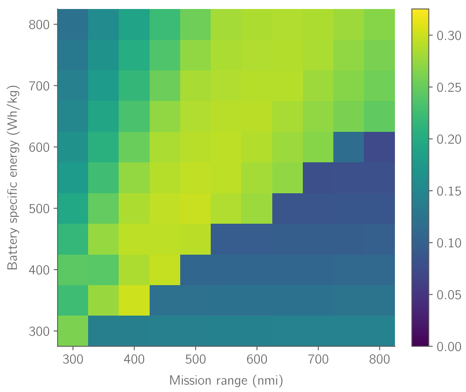
(c) $W_{PropSys}$ (with fuel and battery) / MTOW



(d) Energy used (kWh)



(e) Mech DoH_{Cruise} (0 all fuel, 1 all electric)



(f) Battery weight / MTOW

Figure 5.8: Parallel Hybrid architecture MDO results.

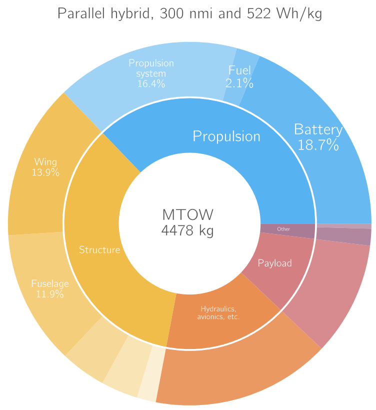
depicted in Figure 5.8b. This observation aligns with the results obtained from series hybrid systems, where the preference for hybrid configurations is determined by the constraints imposed by MTOW limitations.

The sizing method of the parallel hybrid's components is notable. The throttle mechanism regulates the aggregate power output of the electric motor and turboshaft engine. The outcome of this configuration often leads to a decrease in the rated power of the electric motor at identical design points when compared to the series configuration.

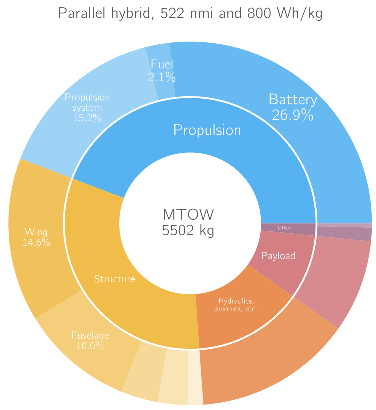
As an example, when considering a distance of 550 nautical miles and an energy density of 550 watt-hours per kilogram, both the series and parallel models demonstrate the ability to attain the maximum takeoff weight (MTOW) limit of 5,700 kg. Nevertheless, the parallel model demonstrates a mechanical Degree of Hybridization (DoH) of 64% while utilizing an electric motor and turboshaft outputs of 268 kW and 389 kW, respectively. In comparison, the current state of the series model exhibits a 36% electric Degree of Hybridization (DoH), wherein the electric motor and turboshaft powers are measured at 794 kW and 791 kW respectively. In the parallel model, the turboshaft is largely utilized to give mechanical shaft power in conjunction with the electric motor. Conversely, in the series model, the turboshaft primarily serves to create electric DC power for the electric motors, which subsequently supply all requisite shaft power.

The series architecture weight breakdown for two design points is presented in Figure 5.9. It is shown that the propulsion system contribution to the total weight is lower than the equivalent series architecture with contributions of 37.2% and 44.2% for the parallel architecture and 40.4% and 47.4% for the series configuration at the same mission ranges of 300 and 522 nmi, and specific energies of 522 and 800 Wh/kg respectively.

The weight of the battery is directly proportional to the selected level of hybridization. The parallel and series models exhibit DoHs of 64% and 36%, respectively, at a distance of 550 nautical miles and an energy density of 550 Wh/kg. These models are associated with battery weights of 1,669 kg and 1,056 kg, which correspond to 29% and 18% of their respective maximum takeoff weights (MTOWs). These findings are depicted in Figures 5.8f and 5.6f. The propulsion systems of the aircraft models, as depicted in Figures 5.8c and 5.6c, exhibit similar overall weights despite a discrepancy of 600 kg in battery weight. Specifically, the total weights of the propulsion



(a) Parallel Hybrid Weight Breakdown 300 nmi and 522 Wh/kg



(b) Parallel Hybrid Weight Breakdown 522 nmi and 800 Wh/kg

Figure 5.9: Parallel Hybrid Weight breakdown.

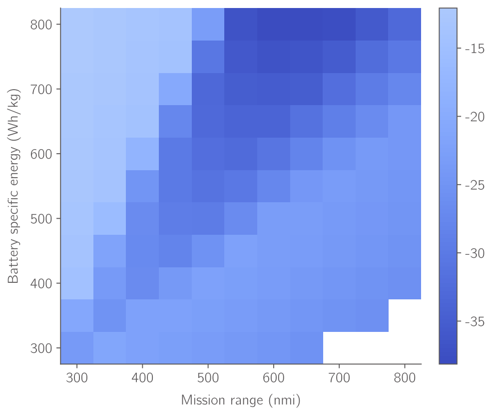
systems are 2331 kg and 2198 kg, which correspond to 41% and 39% of the total weights of the aircraft models, respectively. The observed similarity can be linked to the increased mass of the generator and converters in the series hybrid configuration.

5.2 Post-Optimality study: Comparison of Architectures

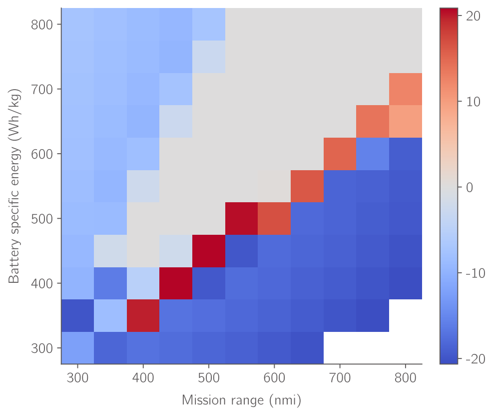
Figure 5.10 depicts the contrasting distinctions between the parallel and series hybrid designs. Furthermore, in order to conduct a comprehensive examination of various designs, two distinct design scenarios have been selected. The first scenario encompasses a range of 300 nautical miles, while the second scenario covers a range of 500 nautical miles. Both scenarios assume a battery-specific energy of 500 Wh/kg. The weight distribution according to the chosen architectures is elaborated upon in Figures 5.11 and 5.12. Table 5.1 systematically presents a comprehensive compilation of information pertaining to all design variables, underlying assumptions, and findings obtained from the optimization process. This table also includes the relevant data associated with the reference aircraft model.

5.2.1 Parallel Hybrid vs Series Hybrid Propulsion Architectures

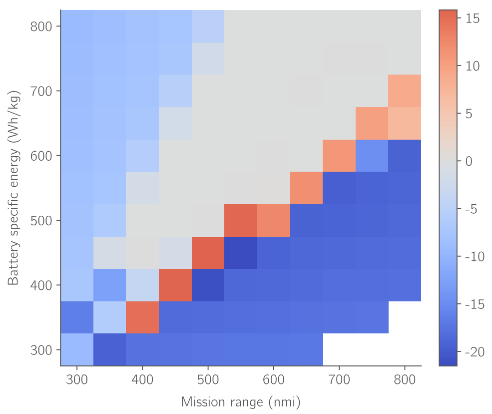
According to the data presented in Figure 5.10b, it can be observed that of the 117 optimization scenarios analyzed, the parallel hybrid configuration exhibits a lower maximum takeoff weight (MTOW) compared to the series hybrid configuration in 107 instances. This outcome is anticipated due to the absence of the generator or rectifier that is necessary for the series architecture.



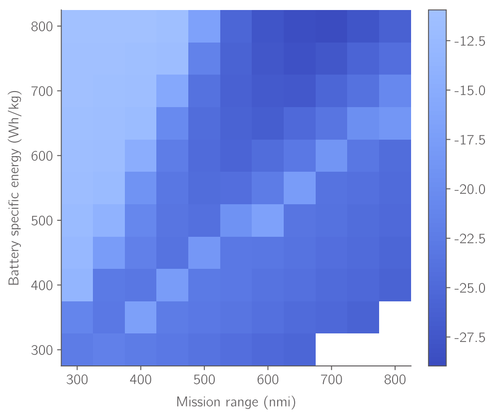
(a) Fuel burn + MTOW / 100 (% difference)



(b) MTOW (% difference)



(c) $W_{PropSys}$ / MTOW (% difference)



(d) Energy used (% difference)

Figure 5.10: The comparative outcomes of parallel hybrid and series hybrid MDO designs. A positive number, specifically denoted by the color red, signifies that the quantity of the parallel hybrid is greater than the quantity of the series hybrid.

There exist instances along the diagonal of the grid when both architectures possess a maximum takeoff weight (MTOW) that is constrained to 5,700 kg (as depicted by the gray region in Figure 5.10b). The instances in which the optimizer transitions from relying solely on battery power to incorporating a sufficient amount of fuel to effectively complete the mission without surpassing the maximum takeoff weight (MTOW) constraint.

The series hybrid arrangement is lighter in 10 specific situations along the diagonal, as indicated by the highlighted red cells. This weight reduction can be attributed to the optimizer's choice of the Degree of Hybridization (DoH). Due to its reduced number of propulsion system components compared to the series hybrid, the parallel hybrid architecture is able to allocate additional weight towards the inclusion of batteries, hence achieving a higher level of hybridization.

As an example, let us examine a scenario involving a distance of 500 nautical miles and an energy density of 450 watt-hours per kilogram. The parallel configuration has a cruise duty cycle of 56%, whereas the series hybrid configuration demonstrates a cruise duty cycle of 5%. The observed variability in the Degree of Hybridization (DoH) necessitates a corresponding requirement for the parallel design to accommodate a battery weight of 1,663 kg. The series architecture necessitates a battery weight of merely 498 kg, as it provides 95% of the energy through the utilization of fuel. This explains the rationale behind the greater significance of parallel design compared to series architecture in a limited number of instances. The propulsion system weight fraction, which encompasses both fuel and battery components, exhibits similar trends, as depicted in Figure 5.10c. In the majority of circumstances, the parallel hybrid arrangement exhibits a lower weight in its propulsion system compared to the series configuration.

Furthermore, the parallel hybrid architecture exhibits a reduced weight and a decreased number of energy conversions within the power train design as compared to the series architecture. The parallel hybrid architecture exhibits reduced energy usage across all design points due to the influence of these two elements, as depicted in Figure 5.10d. This indicates that in the case of an aircraft like the King Air, the series hybrid architecture would need to possess additional aerodynamic efficiency, structure design, or control benefits in order to be utilized within the examined ranges and specific energies, such as distributed electric propulsion. According to the results presented in Figure 5.10a, it can be observed that the utilization of a parallel architecture consistently yields a more optimal design across all design points. This conclusion is based on the evaluation of our goal function, which combines fuel burn and a 0.01 increase in maximum takeoff weight (MTOW).

5.2.2 Weight Analysis and Components Breakdown

Figure 5.11 presents a weight breakdown for the five variants while considering a mission range of 300 nautical miles and a battery specific energy of 500 watt-hours per kilogram. Figure 5.12 depicts an identical scenario, except at a mission range of 500 nautical miles. The conventional architecture has the lowest maximum take-off weight (MTOW) among all designs for the two designated design points, with a battery specific energy of 500 Wh/kg and flight ranges of 300 and 500 nautical miles.

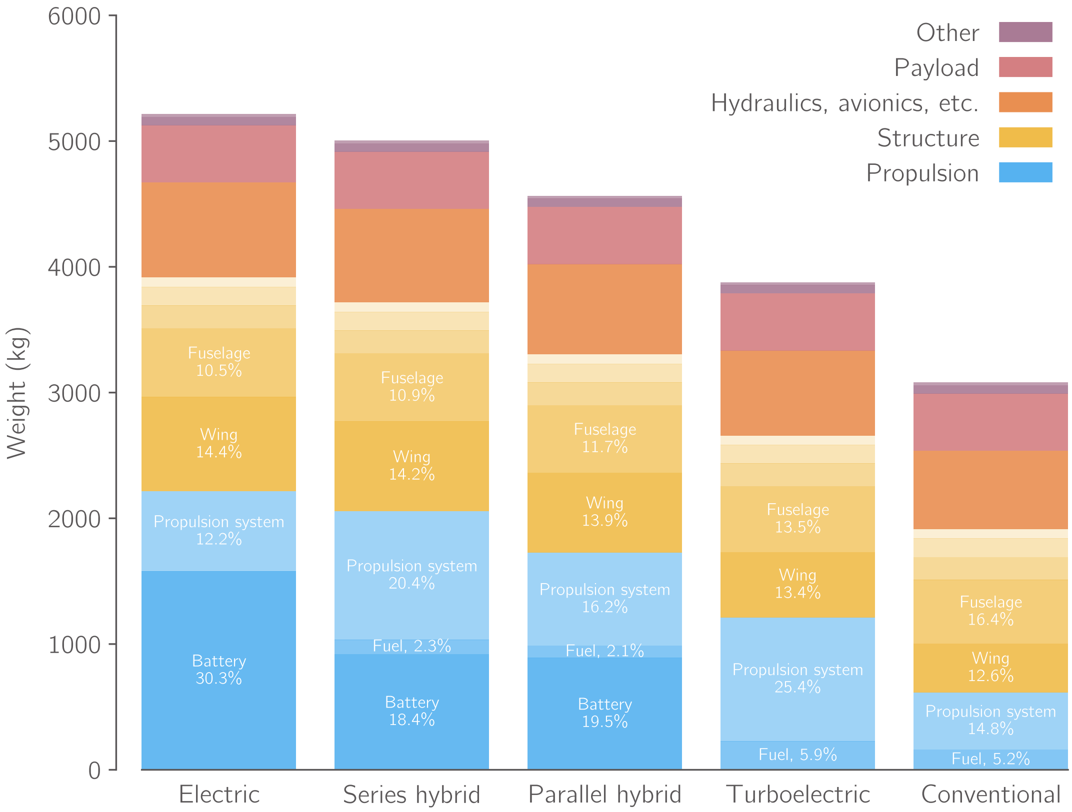


Figure 5.11: Weight breakdown for 300 nmi and 500 Wh/kg with all architectures.

The electric architecture's propulsion system weight fraction, excluding the battery, has the lowest value compared to the other four architectures for the 300 nmi and 500

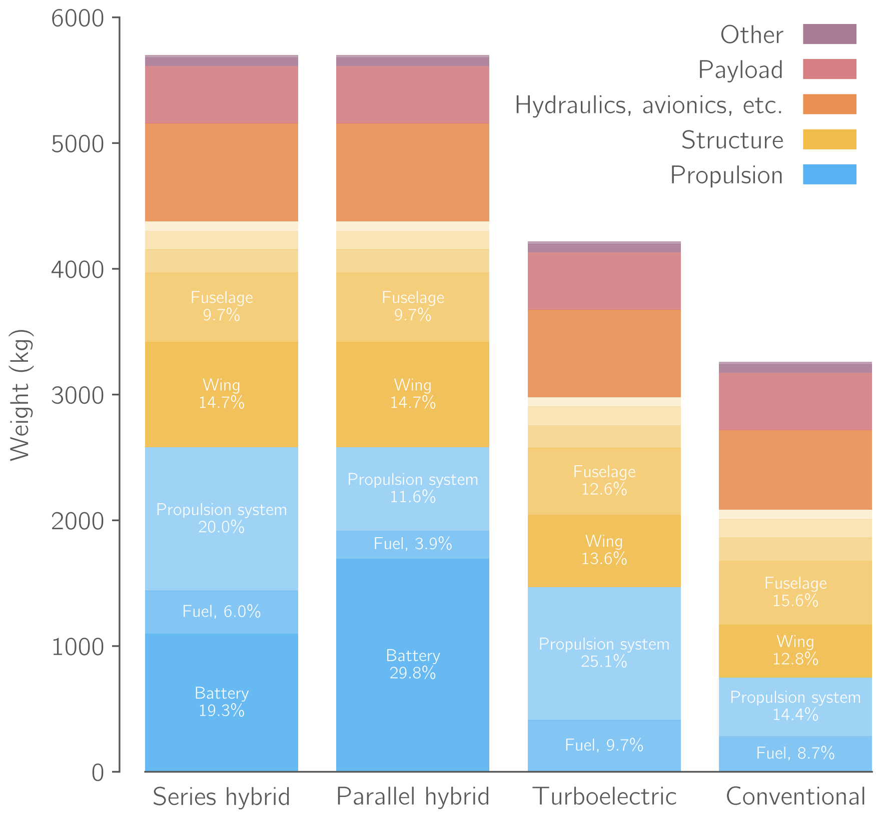


Figure 5.12: Weight breakdown for 500 nmi and 500 Wh/kg with all architectures.

Wh/kg mission. This can be attributed to its superior efficiency and specific power. Nevertheless, because of the significant power demands associated with the electric design, particularly in terms of its maximum takeoff weight (MTOW), the weight of the electric propulsion system (without the battery) remains higher than that of the traditional architecture (excluding fuel). In the case of a 300 nautical mile (nmi) short range, the implementation of electric power in the reference aircraft has resulted in a notable augmentation of the maximum takeoff weight (MTOW) when compared to the traditional configuration. Specifically, the all electric, series hybrid, parallel hybrid, and turboelectric architectures have exhibited MTOW increases of 69%, 62%, 48%, and 25% respectively.

According to the information presented in Table 5.1, the conventional architecture experiences a modest 6% increase in its maximum takeoff weight (MTOW) when the design mission range is extended by 200 nautical miles (nmi). Specifically, the MTOW rises from 3,079 kg to 3,259 kg.

Table 5.1: MDO Results Summary: Minimum Fuel Burn and MTOW

	Parameters	King Air C90GT (Model) [5]	King Air C90GT (Published) [5]	Conventional	Turboelectric	All Electric	Series Hybrid	Parallel Hybrid	Conventional	Turboelectric	Series Hybrid	Parallel Hybrid
Opt Problem Grid	Optimization Rules	Analysis	Analysis	MDO	MDO	MDO	MDO	MDO	MDO	MDO	MDO	MDO
	Specific Energy (Wh/kg)	-	-	-	-	500	500	500	-	-	500	500
	Design Range (nmi)	1,000	894+100	300	300	300	300	300	500	500	500	500
Design Variables	MTOW (kg)	4,581	4,581	3,079	3,876	5,212	5,001	4,563	3,259	4,218	**5,700**	**5,700**
	Mech Turboshaft (kW, for each)	405	405	420	-	-	-	464	444	-	-	389
	Elec Motor (kW, for each)	-	-	-	-	727	698	319	-	590	794	268
	Elec Turboshaft (kW)	-	-	-	902	-	695	-	-	980	791	-
	Prop Diameter (m)	2.29	2.29	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**
	Battery Wt (kg)	-	-	-	-	1,579	921	891	-	-	1,099	1,696
	DoH at cruise	-	-	-	-	-	**99%**	**99%**	-	-	38%	67%
Weights	OEW (kg)	3,255	3,255	2,465	3,195	4,758	4,434	4,013	2,523	3,354	4,907	5,024
	Design Payload (kg)	454	454	454	454	454	454	454	454	454	454	454
	Propulsion System Weight (kg)	-	-	456	986	2,215	1,943	1,632	469	1,059	2,241	2,358
	Propulsion System Fraction (%)	-	-	14.8	25.4	42.5	38.8	35.8	14.4	25.1	39.3	41.4
	Battery Fraction (%)	-	-	-	-	30.3	18.4	19.5	-	-	19.3	30.0
Wing Geometry	Wing Ref Area (m^2)	27.3	27.3	18.4	23.1	31.1	29.8	27.2	19.4	25.1	34.0	34.0
	Wingspan (m)	15.3	15.3	12.6	14.1	16.3	16.0	15.3	12.9	14.7	17.1	17.1
	Aspect Ratio	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58
	Wing Loading (kg/m^2)	167.8	167.8	167.8	167.8	167.8	167.8	167.8	167.8	167.8	167.8	167.8
Aerodynamics	Flaps-Down C_{Lmax}	1.52	-	1.52	1.52	1.52	1.52	1.52	1.52	1.52	1.52	1.52
	Oswald Efficiency	0.80	-	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
	C_{D0} at Cruise	0.0220	-	0.0220	0.0220	0.0220	0.0220	0.0220	0.0220	0.0220	0.0220	0.0220
	C_{D0} at Takeoff	0.0290	-	0.0290	0.0290	0.0290	0.0290	0.0290	0.0290	0.0290	0.0290	0.0290
Performance	Takeoff Rotation Speed (KIAS)	89.8	90	89.8	89.8	89.8	89.8	89.8	89.8	89.8	89.8	89.8
	Climb Rate (ft/min)	1,500	-	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500
	Climb Speed (KIAS)	124	-	124	124	124	124	124	124	124	124	124
	Cruise Speed (KIAS)	170	170	170	170	170	170	170	170	170	170	170
	Cruise Altitude (ft)	29,000	29,000	29,000	29,000	29,000	29,000	29,000	29,000	29,000	29,000	29,000
	Descent Rate (ft/min)	600	-	600	600	600	600	600	600	600	600	600
	Descent Speed (KIAS)	130	-	140	140	140	140	140	140	140	140	140
	BFL,SL,ISA+0 (ft)	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**	**4,452**
Evaluation Outputs	Fuel burn + 0.01MTOW	-	-	191	266	52	164	142	315	452	397	279
	Mission Fuel burn (kg)	728.4	-	160	227	0.0	114	96.3	283	410	340	222
	Total Energy used (kWh)	-	-	1,917	2,715	789	1,819	1,596	3,377	4,903	4,606	3,500

Bold numbers indicate an active constraint, an upper or lower bound.

In the case of alternative architectures, the percentage rise exhibits a more pronounced magnitude. As an example, the turboelectric design exhibits a growth of 9%, the series hybrid demonstrates a growth of 14%, and the parallel hybrid showcases a growth of 25%. This is in spite of the shift towards reduced hybridizations in hybrid designs. The all-electric architecture, while considering a higher range and adhering to the maximum takeoff weight (MTOW) constraint, does not offer a viable alternative and is therefore disregarded.

Based on the discourse presented in Sections 5.1.3 and 5.1.4, it is evident that the optimizer strives to utilize the maximum battery capacity feasible within the confines of the maximum takeoff weight (MTOW) upper limit, particularly on missions characterized by modest ranges and battery-specific energies. In Figure 5.12, it can be observed that for the scenario with a range of 500 nautical miles (nmi) and an energy density of 500 watt-hours per kilogram (Wh/kg), both series and parallel architectures exhibit a weight of 5,700 kilograms (kg). Nevertheless, the parallel hybrid architecture offers the advantage of accommodating a greater battery weight due to its reduced number of propulsion system components. Consequently, the parallel construction exhibits a Degree of Hybridization (DoH) of 67% accompanied by a battery weight of 1,696 kg. In contrast, the series architecture demonstrates a lower DoH of 38% with a comparatively lighter battery weighing 1,099 kg. From the perspective of maximum takeoff weight (MTOW), the conventional architecture provides the most favorable benefit by exhibiting the lowest weight across all design points.

5.3 Research Questions Summary

In this section, the research questions investigated in this chapter are summurized below:

- **RQ2 How to address architectural decisions at the propulsion system level?**
  - **RQ2.3 Can the effect of the architecture decisions at the propulsion system level be captured at the aircraft level**
    - The results presented in section 5.1 clearly show the variations resulting in the performance of the aircraft due to the selection of dif-

ferent architectures. For example, the overall weight of the aircraft increases significantly due to electrification/hybridization almost in all architectures with varying degrees in the all-electric, series, and parallel hybrid cases. The performance evaluation at the aircraft level provides realistic metrics such as maximum takeoff weight, fuel consumption, and energy usage, such as the cases presented in Table 5.1 that reflects the variability in architectural decisions.

- **RQ3 Does this approach enable automated design space exploration?**
  - **RQ3.1 Can this approach be used to explore a large design space with different decisions at both sub-system and system levels?**
    - The approach enables rapid construction and evaluation of various propulsion architectures presented in section 5.1 including conventional and turboelectric architectures in section 5.1.1, all-electric architecture in section 5.1.2, series hybrid in section 5.1.3 and parallel hybrid in section 5.1.4. Sub-system level decisions, for example selecting the source of shaft power as either the electric motor or turboshaft engine, have been implemented. System-level design requirements can also be adjusted such as Balanced Field Length (BFL) for takeoff distance, presented in section 4.5. The approach is flexible, provides an efficient way to evaluate performance of architectures, and can be extended to include additional decisions at both sub-system and system levels.

CHAPTER 6

CONCLUSION

The thesis presents a novel approach to the construction, integration, and quantitative evaluation of propulsion system designs at the aircraft level, utilizing an automated methodology. This process establishes a crucial initial phase in the development of customized simulations that enable comprehensive investigations on design optimization. The methodology outlined utilizes a function-oriented decomposition of propulsion system elements, facilitating efficient and adaptable architectural arrangements. The approach employed in this study involves an examination of the electrification of a King Air C90GTi, a commuter turboprop aircraft.

The analysis encompasses a wide range of battery-specific energies and mission ranges, with the exploration of numerous individual Multidisciplinary Design Optimization (MDO) scenarios. The aforementioned investigations reveal complex patterns within the design space of hybrid architectures. The utilization of hybrid propulsion systems is increasingly perceived as advantageous as compared to both fuel-based and fully electric architectures, however subject to the constraint of a maximum takeoff weight (MTOW) threshold. Based on our research, it has been observed that the conventional architectural approach exhibits the least Maximum Takeoff Weight (MTOW), whilst the all-electric configuration demonstrates the highest level of energy efficiency within the designated design parameters.

Currently, the turboelectric and series hybrid systems are deemed advantageous solely if they yield further benefits in terms of aerodynamics, structure, or control, perhaps facilitated by technologies such as distributed electric propulsion. The complete reliance on electric architecture demonstrates feasibility for design ranges of 500 nautical miles or fewer, contingent upon the availability of batteries with high specific energy. Nevertheless, it is crucial to acknowledge that these particular patterns may not be transferable to alternative contexts, such as novel designs.

An extension of this thesis work is presented in Appendix B, which builds upon the proposed approach by including a complete architecture optimization problem for hybrid electric propulsion. The process entails the utilization of the Architecture Design and Optimization Reasoning Environment (ADORE) to perform design space exploration to generate a Pareto set of architectures derived from the Top-Level Aircraft Requirements (TLARs). The optimization process encompasses nested optimization techniques that aim to optimize both the electrical components and aircraft sizing in conjunction with the architectural design decisions.

This methodology developed in this thesis work enables researchers and designers to efficiently investigate a wide range of architectural options for diverse objectives and assumptions, connecting design space exploration, system architecting and Multidisciplinary Design Optimization (MDO). The code of the framework developed is available at [28] and [29]. Finally, the existing framework has the potential to be seamlessly incorporated into complete aircraft design tools, enabling rapid interdisciplinary conceptual studies.

REFERENCES

[1] A. W. Schäfer, S. R. H. Barrett, K. Doyme, L. M. Dray, A. R. Gnadt, R. Self, A. O'Sullivan, A. P. Synodinos, and A. J. Torija, "Technological, economic and environmental prospects of all-electric aircraft," *Nature Energy*, vol. 4, no. 2, pp. 160–166, 2019.

[2] S. Görtz, "Lecture notes in applied mdo," May 2022.

[3] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge, UK: Cambridge University Press, January 2022.

[4] J. C. RIBEIRO, "Flickr gallery."

[5] B. J. Brelje and J. R. R. A. Martins, "Development of a conceptual design model for aircraft electric propulsion with efficient gradients," in *2018 AIAA/IEEE Electric Aircraft Technologies Symposium*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 2018.

[6] J. Bussemaker, R. G. Sánchez, M. Fouda, L. Boggero, and B. Nagel, "Function-based architecture optimization: An application to hybrid-electric propulsion systems," in *INCOSE International Symposium*, July 2023.

[7] C. S. Dorbian, P. J. Wolfe, and I. A. Waitz, "Estimating the climate and air quality benefits of aviation fuel and emissions reductions," *Atmospheric Environment*, vol. 45, no. 16, pp. 2750–2759, 2011.

[8] G. P. Brasseur, M. Gupta, B. E. Anderson, S. Balasubramanian, S. Barrett, D. Duda, G. Fleming, P. M. Forster, J. Fuglestvedt, A. Gettelman, R. N. Halthore, S. D. Jacob, M. Z. Jacobson, A. Khodayari, K.-N. Liou, M. T. Lund, R. C. Miake-Lye, P. Minnis, S. Olsen, J. E. Penner, R. Prinn, U. Schumann, H. B. Selkirk, A. Sokolov, N. Unger, P. Wolfe, H.-W. Wong, D. W. Wuebbles, B. Yi, P. Yang, and C. Zhou, "Impact of aviation on climate: Faa's aviation

climate change research initiative (accri) phase ii," *Bulletin of the American Meteorological Society*, vol. 97, no. 4, pp. 561–583, 2016.

[9] "UN climate change conference," 2019.

[10] B. J. Brelje and J. R. R. A. Martins, "Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches," *Progress in Aerospace Sciences*, vol. 104, pp. 1–19, 2019.

[11] G. Palaia, D. Zanetti, K. Abu Salem, V. Cipolla, and V. Binante, "Thea-code: a design tool for the conceptual design of hybrid-electric aircraft with conventional or unconventional airframe configurations," *Mechanics & Industry*, vol. 22, p. 19, 2021.

[12] R. A. McDonald, B. J. German, T. Takahashi, C. Bil, W. Anemaat, A. Chaput, R. Vos, and N. Harrison, "Future aircraft concepts and design methods," *The Aeronautical Journal*, vol. 126, no. 1295, pp. 92–124, 2022.

[13] H. L. Silva, G. J. Resende, R. M. C. Neto, A. R. D. Carvalho, A. A. Gil, M. A. A. Cruz, and T. A. M. Guimarães, "A multidisciplinary design optimization for conceptual design of hybrid-electric aircraft," *Structural and Multidisciplinary Optimization*, vol. 64, no. 6, pp. 3505–3526, 2021.

[14] J. H. Bussemaker, P. D. Ciampa, and B. Nagel, "System architecture design space modeling and optimization elements," in *32nd Congress of ICAS*, 2021.

[15] J. H. Bussemaker, P. D. Ciampa, and B. Nagel, "System architecture design space exploration: An approach to modeling and optimization," in *AIAA AVIATION 2020 FORUM*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 2020.

[16] J. H. Bussemaker, T. de Smedt, G. La Rocca, P. D. Ciampa, and B. Nagel, "System architecture optimization: An open source multidisciplinary aircraft jet engine architecting problem," in *AIAA AVIATION 2021 FORUM*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 2021.

[17] E. F. Crawley, B. Cameron, and D. Selva, *System architecture: Strategy and product development for complex systems*. Boston: Pearson, 2016.

[18] G. Esdras and S. Liscouet-Hanke, "Development of core functions for aircraft conceptual design: methodology and results," in *Canadian Aeronautics and Space Institute AERO*, (Montreal, Canada), 2015.

[19] J. Bussemaker, L. Boggero, and B. Nagel, "The agile 4.0 project: Mbse to support cyber-physical collaborative aircraft development," *INCOSE International Symposium*, vol. 33, no. 1, pp. 163–182, 2023.

[20] T. De Smedt, "Aircraft jet engine architecture modeling: Creating a benchmark problem using a system architecting approach with mixed-discrete multi-objective capabilities," 2021. Accessed: 2023-12-06.

[21] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "Openmdao: an open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, vol. 59, no. 4, pp. 1075–1104, 2019.

[22] J. Zamboni, *A method for the conceptual design of hybrid electric aircraft*. M.sc. thesis, 2018.

[23] J. H. Bussemaker, L. Boggero, and P. D. Ciampa, "From system architecting to system design and optimization: A link between mbse and mdao," in *INCOSE International Symposium*, June 2022.

[24] D. P. Raymer, *Aircraft design: A conceptual approach*. AIAA education series, American Institute of Aeronautics and Astronautics, 5. ed. ed., 2012.

[25] E. J. Adler, B. J. Brelje, and J. R. R. A. Martins, "Thermal management system optimization for a parallel hybrid aircraft considering mission fuel burn," *Aerospace*, vol. 9, no. 5, 2022.

[26] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[27] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. Martins, "pyoptsparse: A python framework for large-scale constrained nonlinear optimization of sparse systems," *Journal of Open Source Software*, vol. 5, no. 54, p. 2564, 2020.

[28] M. Fouda, E. J. Adler, J. H. Bussemaker, J. R. Martins, D.F.Kurtulus, L. Boggero, and B. Nagel, “Automated hybrid propulsion model construction forconceptual aircraft design and optimization,” in *33rd Congress of ICAS*, 2022.

[29] M. Fouda, E. J. Adler, and J. H. Bussemaker, “openconcept - a python package for aircraft conceptual design with a focus on propulsion system studies, architecting and integration.” `https://github.com/MahmoudEssamFouda/openconcept/tree/arch-builder/openconcept/architecting`, 2023. Accessed: 2023-12-06.

APPENDIX A

PROPULSION ARCHITECTURES CONSTRUCTION

A.1 Series Hybrid Architecture Powertrain

The code snippet to build a twin series hybrid architecture with one engine, battery, and two motors is

```
proparch = PropSysArch(
    thrust=ThrustGenElements(
        propellers=[Propeller("prop1"),
                    Propeller("prop2")],
        gearboxes=[Gearbox("gearbox1"),
                   Gearbox("gearbox2")],),
    mech=MechPowerElements(
        motors=[Motor("elecmotor", powerrating=2000),
        Motor("elecmotor", powerrating=2000)],
        inverters=Inverter("inverter"),),
    electric=ElectricPowerElements(
        dcbus=DCBus("elecbus"),
        splitter=ElecSplitter("splitter", elecDoH=0.25),
        batteries=Batteries("batpack",
                            weight=4e3,
                            specificenergy=ebatt),
        enginesdc=(
            Engine(name="turboshaft", powerrating=2e3),
            Generator(name="generator"),
            Rectifier(name="rectifier"),),),)
```

In a series hybrid architecture, the implementation of electric power generation, presented in Figure A.1, involves utilizing an electrical Degree of Hybridization (DoH) to distribute the power demand between the engine system and the battery. To align the power levels on either side of this implicit division, an implicit solver is employed to determine the appropriate setting for the engine throttle.

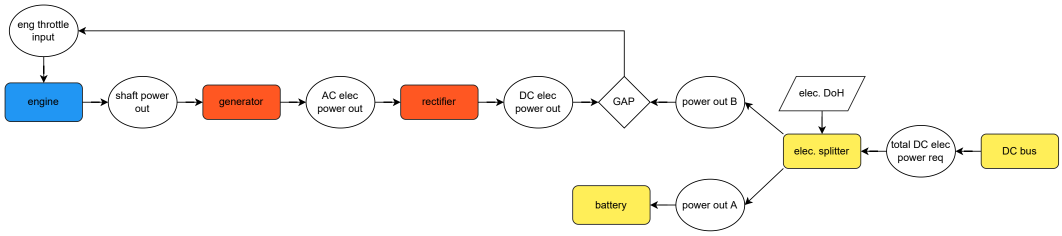


Figure A.1: Series hybrid architecture electric power generation elements.

The generation of mechanical power and thrust in a series hybrid setup, presented in Figure A.2, involves elements where one engine is non-functional (OEI), impacting the secondary electric motor. Under OEI conditions, the throttle input to this second motor is set to zero, treating it as a malfunctioning unit. This implies that the motor produces neither power output nor any electrical load output.

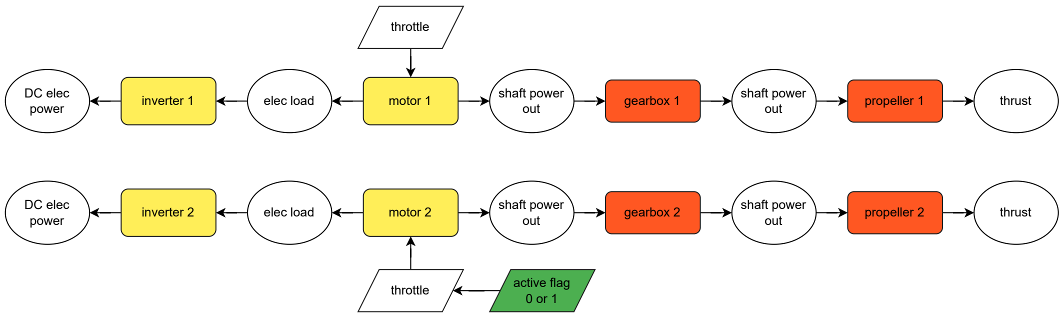


Figure A.2: Series hybrid architecture mechanical power generation and thrust generation elements.

A.2 Parallel Hybrid Architecture Powertrain

The code snippet to build a parallel hybrid architecture

```
proparch = PropSysArch(    parallel hybrid system
    thrust=ThrustGenElements(
        propellers=[Propeller("prop1"),
                    Propeller("prop2")],
        gearboxes=[Gearbox("gearbox1"),
                   Gearbox("gearbox2")],),
    mech=MechPowerElements(
        engines=[Engine("turboshaft",
                 powerrating=600),
                 Engine("turboshaft",
                 powerrating=600)],
        motors=[Motor("motor", powerrating=250),
                Motor("motor", powerrating=250)],
        mechbuses=MechBus("mechbus"),
        mechsplitters=MechSplitter("mechsplitter",
                                   mechDoH=0.25),
        inverters=Inverter("inverter"),),
    electric=ElectricPowerElements(
        dcbus=DCBus("elecbus"),
        batteries=Batteries("batpack",
                            weight=1e3,
                            specificenergy=ebatt)),)
```

In a parallel hybrid setup for generating mechanical power and thrust, presented in Figure A.3, the system is designed to handle scenarios where one engine is inoperative (OEI), particularly affecting the second electric motor unit. This architecture employs a mechanical splitter to divide the shaft power requirements between two paths, A and B, according to a mechanically designed Degree of Hybridization (DoH). The electric motor's power rating is determined to meet the needs of path A throughout the mission, while the engine's power rating is calibrated to provide the required power for path B. In an OEI situation, the power demand for path A drops to zero due to the failure of the second electric motor, which ceases to produce power or carry an electrical load. Consequently, the total available power for this branch is limited to the engine's rated power. A solver is then used to adjust the throttle inputs, ensuring that the power levels are balanced on both sides of this implicit division.

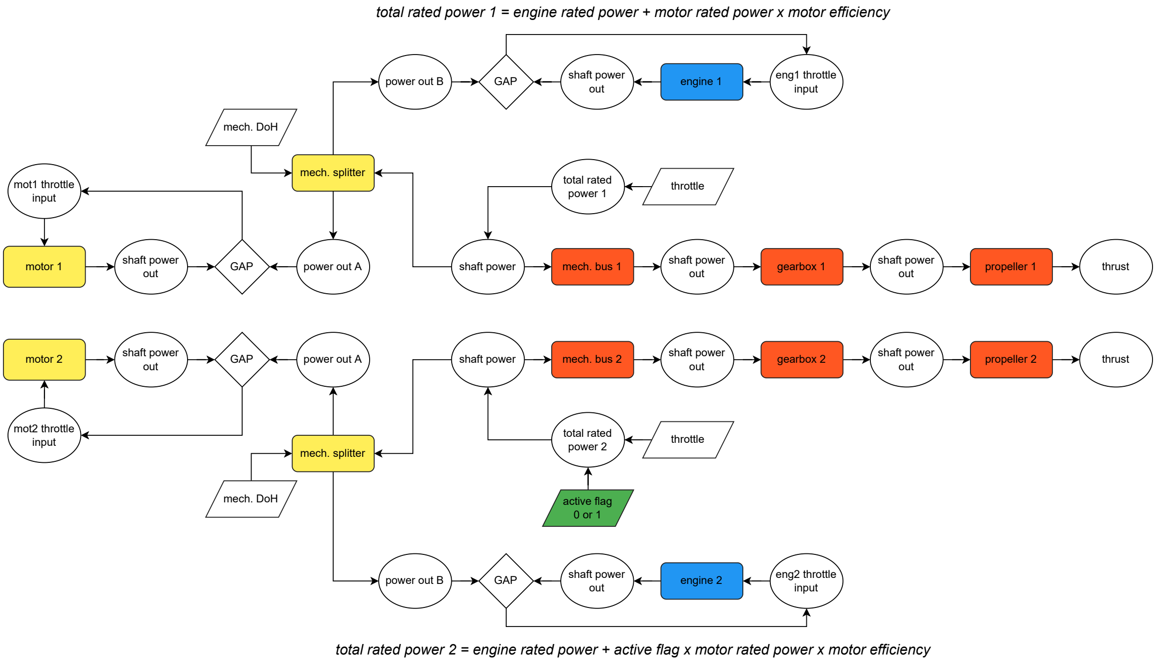


Figure A.3: Parallel hybrid architecture mechanical power generation and thrust generation elements.

APPENDIX B

ONGOING RESEARCH AT THE DLR IN ARCHITECTURE OPTIMIZATION

The German Aerospace Center DLR, Institute of Systems Architecture in Aeronautics in Hamburg, is a leading institute in overall system design, and evaluation research studies. A new study was recently published [6] that relied heavily on extending the work presented in this thesis, to which the author has contributed. The publication utilizes the flexible modeling and simulation capabilities developed to perform a complete architecture optimization loop to demonstrate the framework capabilities to investigate a large design space and obtain a Pareto front of architectures that satisfy the design requirements.

The framework is extended to include an additional architecture optimizer, that manages the suggestion of new architecture to the evaluation framework, developed and presented previously, which serves the aim of a sizing optimizer that evaluates the performance of a single suggested architecture as presented in Figure B.1.

The set of all possible architectural choices is presented using the Architecture Design Space Reasoning Environment (ADORE) developed at the German Aerospace Center (DLR), previously described in section 3.2.3. The tool provides a graph-based method to explicitly represent architectural decisions at the early stages of the conceptual design and connects the definition to existing optimization algorithms [15]. In Figure B.2, a function-based decomposition of system components is presented using the ADORE tool interface to construct a graph-based design space graph, that explicitly defines the available decisions.

The design space definition starts at the system function of providing propulsive

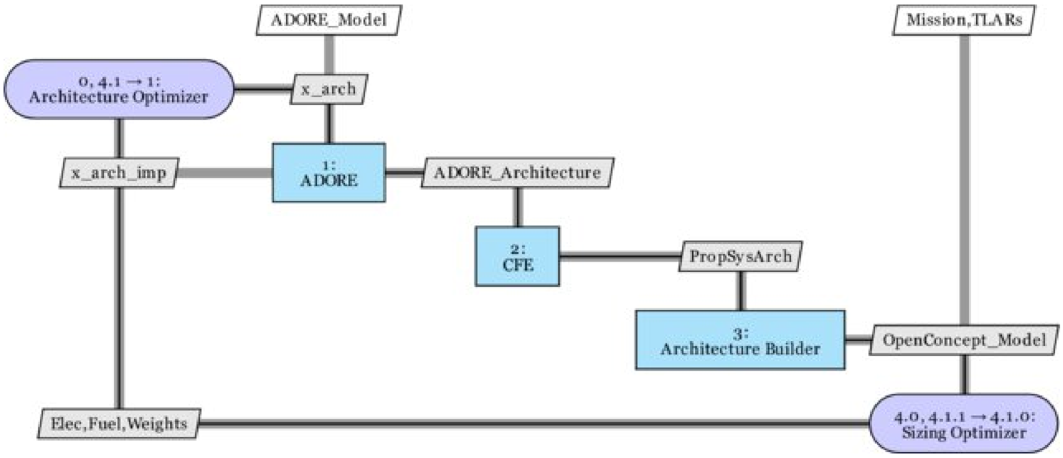


Figure B.1: Architecture Optimization loop that contains two optimizers. The architecture optimizer manages the suggestion of new architectures and the sizing optimizer manages the performance evaluation of a single architecture. [6]

power and expands to include more decisions down the tree. For example, consider the function of generating DC power in Figure B.2, three distinct components can be used to fulfill this function, namely a Rectifier, Electric Hybrid, or Batteries, this represents an architectural choice, and each of these path will lead to a different architecture. The set of all possible architectures that can be taken from all possible decisions in the graph, including the values associated with the attributes of each component, constitute the complete design space definition.

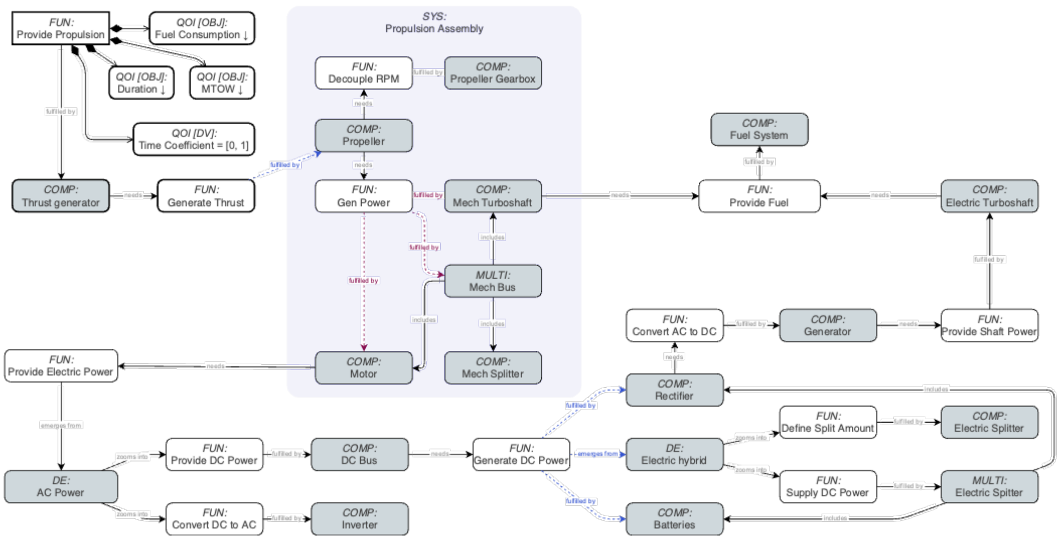


Figure B.2: Design Space Definition including possible architectural choices. [6]

A new optimization problem is ready to be evaluated where the architecture optimizer will suggest new architectures to be evaluated and receive back the evaluation metrics of this architecture. This process is automatically performed by an optimization algorithm that investigates the defined design space, i.e. the set of all possible choices considered. A Pareto front is obtained from this process presented in Figures B.3 and B.4.

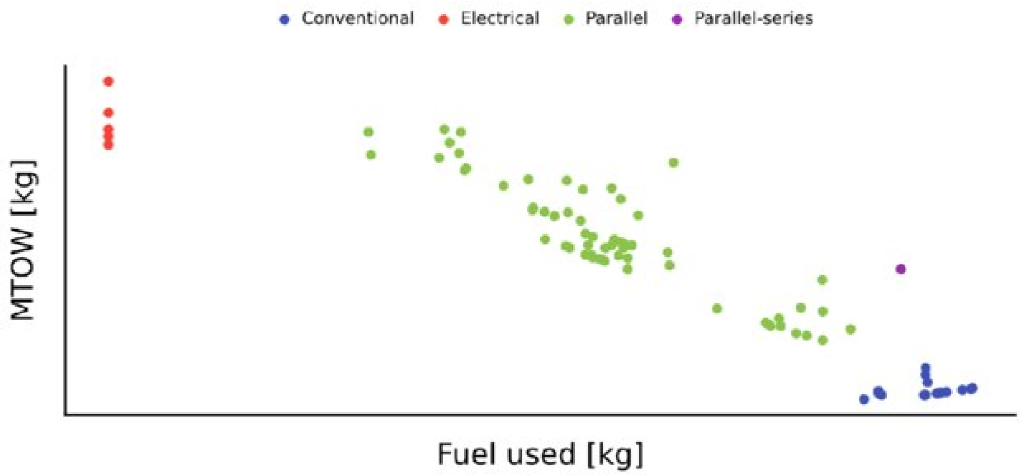


Figure B.3: Pareto Front indicating the different Architecture types. [6]

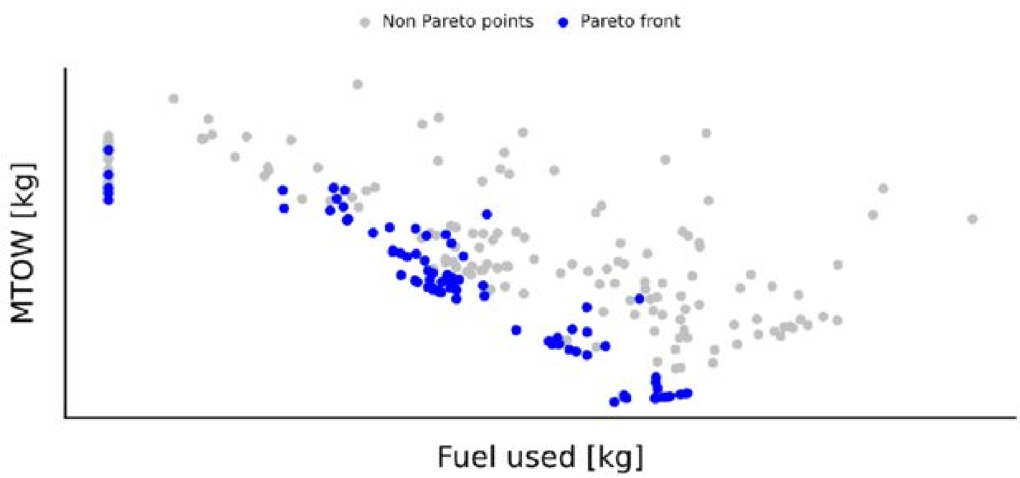


Figure B.4: Pareto Front of All Architectures evaluated. [6]

The Pareto front, shown in blue in Figure B.4, represents the design points that represent the best trade-off according to the design objective function that satisfies the trade-off between two conflicting objectives, in this case, the fuel Consumption and Maximum Takeoff Weight (MTOW). These points show the best architecture candidates to satisfy the design requirement, i.e. objective, specified by the user while simultaneously adhering to the design constraints.

These results demonstrate the feasibility of employing architecture optimization techniques to investigate large design spaces and obtain a Pareto front of architecture candidates in different design problems [6].

APPENDIX C

SOURCE CODE: OPEN ACCESS

C.1 GitHub Repo

The foundational code that underpins the research presented herein is publicly accessible and has been released under an open-source license. It can be retrieved from the GitHub repository as denoted in [29]. For purposes of archival reference and to mitigate the risk of potential future inaccessibility of the link, two principal classes instrumental in the dynamic generation of propulsion architecture and aircraft models are presented.

C.2 Dynamic Propulsion Architecture Class

```
import numpy as np
from typing import *
import openmdao.api as om

from openconcept.architecting.builder.utils import *
from openconcept.architecting.builder.architecture import *

__all__ = ["DynamicPropulsionArchitecture"]

class DynamicPropulsionArchitecture(om.Group):
    """
    A propulsion system architecture analysis group
```

```
built-up from a PropSysArch definition.
Propulsion system inputs (i.e. design variables
and configuration parameters) are defined statically from
ArchElement definitions. These parameters are normally
optimized by the architecture optimizer, and therefore
do not need to be known to the user of this analysis group.

Options
-------
    num_nodes : float
        Number of analysis points to run (default 1)
Inputs
--------
    prop|rpm: float
        propeller rpm (vec, RPM)
    fltcond|rho: float
        air density at the specific flight
        condition (vec, kg/m**3)
    fltcond|Utrue: float
        true airspeed at the flight condition (vec, m/s)
    throttle: float
        throttle input to the engine, fraction
        from 0-1 (vec, '')
    propulsor_active: float (either 0 or 1)
        a flag to indicate on or off for the
        connected propulsor either 1 or 0 (vec, '')
    duration: float
        the amount of time to finish the
        segment in seconds (Scalar, 's')

Outputs
------
    propulsion_system_weight : float
```

```
            The weight of the propulsion
            system (Scalar, 'kg')
            Note: battery weight is included
        fuel_flow: float
            The fuel flow consumed in the
            segment (Vec, 'kg/s')
        soc: float
            State-of-charge along the
            segment (Vec, dimensionless)
        thrust: float
            The total thrust of the propulsion
            system (Vec, 'N')
    """

    def initialize(self):
        self.options.declare("num_nodes",
            default=1, desc="Number of mission
            analysis points to run")
        self.options.declare("architecture",
            types=PropSysArch, desc="The propulsion
            system architecture definition")

    def setup(self):
        nn = self.options["num_nodes"]
        arch: PropSysArch = self.options["architecture"]

        # Define inputs
        main_prop = arch.thrust.propellers[0]
        default_rpm = 2000.0 if main_prop is None
            else main_prop.default_rpm
        input_comp, input_map = collect_inputs(
            self,
        [   (RPM_INPUT, "rpm", np.tile(default_rpm, nn)),
```

```
    (THROTTLE_INPUT, None, np.tile(1.0, nn)),
    (DURATION_INPUT, "s", 1.0),
    (ACTIVE_INPUT, None, np.tile(1.0, nn)),
    (FLTCOND_RHO_INPUT, "kg/m**3", np.tile(1.225, nn)),
    (FLTCOND_TAS_INPUT, "m/s", np.tile(100.0, nn)),],
    name="propmodel_in_collect",)
order = [input_comp.name]

subsys_groups = []
weight_outputs = []
thrust_outputs = []
fuel_flow_outputs = []
soc_outputs = []

# Create thrust generation groups: propellers + gearboxes
thrust_groups = arch.thrust.create_thrust_groups(self, nn)
subsys_groups += thrust_groups.copy()

weight_outputs += [grp.name + "." + WEIGHT_OUTPUT
    for grp in thrust_groups]
thrust_outputs += [grp.name + "." + THRUST_OUTPUT
    for grp in thrust_groups]

# Create mechanical power generation groups:
# motors or engines connected to the propellers
mech_group, electric_power_gen_needed =
    arch.mech.create_mech_group(self, thrust_groups, nn)
subsys_groups += [mech_group]
fuel_flow_outputs += [mech_group.name + "."
    + FUEL_FLOW_OUTPUT]
weight_outputs += [mech_group.name + "."
    + WEIGHT_OUTPUT]
order += [mech_group.name]
```

```
order += [grp.name for grp in thrust_groups]

# If needed, create electrical power generation groups:
# batteries, engines, etc.
if electric_power_gen_needed:
    if arch.electric is None:
        raise RuntimeError("Electrical power
        generation is needed but no
        `ElectricPowerElements` is defined!")

    elec_group = arch.electric.create_electric_group(
        self, mech_group, thrust_groups, nn)
    subsys_groups += [elec_group]

    fuel_flow_outputs += [elec_group.name + "."
        + FUEL_FLOW_OUTPUT]
    weight_outputs += [elec_group.name + "."
        + WEIGHT_OUTPUT]
    soc_outputs += [elec_group.name + "."
        + SOC_OUTPUT]

    order += [elec_group.name]
# Connect inputs
def _connect_input(input_name: str, groups:
    List[om.Group], group_input_name: str = None):
    for group in groups:
        self.connect(input_map[input_name],
        group.name + "." +
        (group_input_name or input_name))

_connect_input(RPM_INPUT, thrust_groups)
_connect_input(THROTTLE_INPUT, [mech_group])
_connect_input(ACTIVE_INPUT, [mech_group])
```

```
        _connect_input(DURATION_INPUT, subsys_groups)
        _connect_input(FLTCOND_RHO_INPUT, subsys_groups)
        _connect_input(FLTCOND_TAS_INPUT, subsys_groups)

        # Create summed outputs
        ff_comp = create_output_sum(self, "fuel_flow",
            fuel_flow_outputs, "kg/s", n=nn)
        wt_comp = create_output_sum(self,
            "propulsion_system_weight", weight_outputs, "kg")
        th_comp = create_output_sum(self, "thrust",
            thrust_outputs, "N", n=nn)
        soc_comp = create_output_sum(self,
            "SOC", soc_outputs, n=nn)
        order += [ff_comp.name, wt_comp.name,
            th_comp.name, soc_comp.name]

        # Define order to reduce feedback connections
        self.set_order(order)
        # code ends
```

C.3 Dynamic Aircraft Model Class

```
import numpy as np
import openmdao.api as om
import openconcept.api as oc
from openconcept.utilities.math.multiply_divide_comp
    import ElementMultiplyDivideComp
from openconcept.analysis.aerodynamics import PolarDrag
from openconcept.utilities.math.integrals import Integrator
from openconcept.architecting.builder.architecture import *
```

```
from openconcept.architecting.builder.arch_group
    import DynamicPropulsionArchitecture

from examples.methods.weights_twin_hybrid import (
    WingWeight_SmallTurboprop,
    EmpennageWeight_SmallTurboprop,
    FuselageWeight_SmallTurboprop,
    NacelleWeight_MultiTurboprop,
    LandingGearWeight_SmallTurboprop,
    FuelSystemWeight_SmallTurboprop,
    EquipmentWeight_SmallTurboprop,)

__all__ = ["DynamicACModel"]


class DynamicACModel(oc.IntegratorGroup):
    """
    OpenConcept-compliant aircraft model. Should be
    created using the DynamicACModel.factory
    function (see below).
    Note: the aircraft weight is calculated by simply
    subtracting the fuel flow from the MTOW over
    the course of the mission. The propulsion architecture
    system weight is added as an output, and should be
    integrated into the OEW external to OpenConcept!

    Usage in the setup function of your main analysis group:
    ```
    arch = PropSysArch(...)

    mission_model = MissionWithReserve(
        num_nodes=nn,
        aircraft_model=DynamicACModel.factory(arch),)
```
```

````
```

Options
-------
    num_nodes : float
        Number of analysis points to run (default 1)
    flight_phase : str|None
        Name of the flight phase (default: None)
    architecture: PropSysArch
        Propulsion system architecture description to use.

Inputs
--------
    fltcond|*: float
        Flight conditions during the mission segment (vec)
            fltcond|rho     Air density      (kg/m**3)
            fltcond|Utrue   True airspeed    (m/s)
            fltcond|CL      Trimmed CL       (-)
            fltcond|q       Dynamic pressure (Pa)
    throttle: float
        Throttle input to the engine, fraction from 0-1 (vec, -)
    propulsor_active: float (either 0 or 1)
        A flag to indicate on or off for the connected
        propulsor either 1 or 0 (vec, -)
    duration: float
        The amount of time to finish the
        segment in seconds (scalar, s)
    ac|*: float
        Aircraft design parameters (scalar)
            ac|aero|polar|CD0_cruise
                # CD0 in cruise       (-)
            ac|aero|polar|CD0_TO
                # CD0 in take-off     (-)
````
````

```
            ac|aero|polar|e
                # Oswald factor       (-)
            ac|aero|wing|S_ref
                # Wing reference area (m**2)
            ac|aero|wing|AR
                # Wing aspect ratio   (-)
            ac|weights|MTOW
                # Max take-off weight (kg)

    Outputs
    ------
        drag: float
            Total drag of the aircraft (Vec, N)
        thrust: float
            Total thrust of the propulsion system (Vec, N)
        weight: float
            Total weight of the aircraft, calculated
            from MTOW and fuel flow (Vec, kg)
        seg_fuel_used: float
            Total fuel used in the mission segment (Scalar, kg)
        propulsion_system_weight : float
            The weight of the propulsion system (Scalar, kg)
    """

    @classmethod
    def factory(cls, architecture: PropSysArch):
        def _factory(num_nodes=1, flight_phase=None):
            return cls(num_nodes=num_nodes,
            flight_phase=flight_phase, architecture=architecture)
        return _factory

    def initialize(self):
        self.options.declare("num_nodes", default=1)
```

```
        self.options.declare("flight_phase", default=None)
        self.options.declare("architecture", types=PropSysArch,
            desc="The propulsion system architecture definition")

    def setup(self):
        nn = self.options["num_nodes"]
        self._add_propulsion_model(nn)
        self._add_drag_model(nn)
        self._add_weight_model(nn)

    def _add_propulsion_model(self, nn):
        self.add_subsystem(
            "propmodel",
            DynamicPropulsionArchitecture(num_nodes=nn,
                architecture=self.options["architecture"]),
            promotes_inputs=["fltcond|*", "throttle",
                "propulsor_active", "duration"],
            promotes_outputs=["fuel_flow", "thrust",
                "propulsion_system_weight"],)

    def _add_drag_model(self, nn):
        # Determine CD0 source based on flight phase
        flight_phase = self.options["flight_phase"]
        if flight_phase not in ["v0v1", "v1v0", "v1vr", "rotate"]:
            cd0_source = "ac|aero|polar|CD0_cruise"
        else:
            cd0_source = "ac|aero|polar|CD0_TO"

        # Add drag model based on simple drag polar model
        self.add_subsystem(
            "drag",
            PolarDrag(num_nodes=nn),
            promotes_inputs=["fltcond|CL", "ac|geom|*",
```

```
                ("CD0", cd0_source), "fltcond|q",
                    ("e", "ac|aero|polar|e")],
            promotes_outputs=["drag"],)

    def _add_weight_model(self, nn):
        # Operating empty weight model
        weight = om.Group()
        const = weight.add_subsystem("const",
            om.IndepVarComp(), promotes_outputs=["*"])
        const.add_output("W_fluids", val=20, units="kg")
        const.add_output("structural_fudge",
            val=1.6, units="m/m")
        weight.add_subsystem(
            "wing",
            WingWeight_SmallTurboprop(),
            promotes_inputs=[
                "ac|weights|MTOW",
                "ac|weights|W_fuel_max",
                "ac|geom|wing|S_ref",
                "ac|geom|wing|AR",
                "ac|geom|wing|c4sweep",
                "ac|geom|wing|taper",
                "ac|geom|wing|toverc",
                "ac|q_cruise",],
            promotes_outputs=["W_wing"],)
        weight.add_subsystem(
            "empennage",
            EmpennageWeight_SmallTurboprop(),
            promotes_inputs=["ac|geom|hstab|S_ref",
                "ac|geom|vstab|S_ref"],
            promotes_outputs=["W_empennage"],)
        weight.add_subsystem(
            "fuselage",
```

```
        FuselageWeight_SmallTurboprop(),
        promotes_inputs=[
            "ac|weights|MTOW",
            "ac|geom|fuselage|length",
            "ac|geom|fuselage|height",
            "ac|geom|fuselage|width",
            "ac|geom|fuselage|S_wet",
            "ac|geom|hstab|c4_to_wing_c4",
            "ac|q_cruise",],
        promotes_outputs=["W_fuselage"],)
    weight.add_subsystem(
        "nacelle", NacelleWeight_MultiTurboprop(),
            promotes_inputs=["P_TO"],
            promotes_outputs=["W_nacelle"])
    weight.add_subsystem(
        "gear",
        LandingGearWeight_SmallTurboprop(),
        promotes_inputs=["ac|weights|MLW",
            "ac|geom|maingear|length",
            "ac|geom|nosegear|length"],
        promotes_outputs=["W_gear"],)
    weight.add_subsystem(
        "fuelsystem",
        FuelSystemWeight_SmallTurboprop(),
        promotes_inputs=["ac|weights|W_fuel_max"],
        promotes_outputs=["W_fuelsystem"],)
    weight.add_subsystem(
        "equipment",
        EquipmentWeight_SmallTurboprop(),
        promotes_inputs=[
            "ac|weights|MTOW",
            "ac|num_passengers_max",
            "ac|geom|fuselage|length",
```

```
            "ac|geom|wing|AR",
            "ac|geom|wing|S_ref",
            "W_fuelsystem",],
        promotes_outputs=["W_equipment"],)
    weight.add_subsystem(
        "structural",
        oc.AddSubtractComp(
            output_name="W_structure",
            input_names=["W_wing", "W_fuselage",
                "W_nacelle", "W_empennage", "W_gear"],
            units="lb",),
        promotes_outputs=["*"],
        promotes_inputs=["*"],)
    weight.add_subsystem(
        "structural_fudge",
        ElementMultiplyDivideComp(
            output_name="W_structure_adjusted",
            input_names=["W_structure", "structural_fudge"],
            input_units=["lb", "m/m"],),
        promotes_inputs=["*"],
        promotes_outputs=["*"], )
    weight.add_subsystem(
        "totalempty",
        oc.AddSubtractComp(
            output_name="ac|weights|OEW",
            input_names=[
                "W_structure_adjusted",
                "W_fuelsystem",
                "W_equipment",
                "W_fluids",
                "propulsion_system_weight",],
            units="lb",),
        promotes_outputs=["*"],
```

```
    promotes_inputs=["*"],)
self.add_subsystem(
    "OEW_calc",
    weight,
    promotes_inputs=["ac|*", "propulsion_system_weight",
        ("P_TO", "ac|propulsion|engine|rating")],
    promotes_outputs=["ac|weights|OEW"],)
# Integrate fuel flow
fuel_int = self.add_subsystem(
    "fuel_int",
    Integrator(num_nodes=nn, method="simpson",
        diff_units="s", time_setup="duration"),
    promotes_inputs=["*"],
    promotes_outputs=["*"],)
fuel_int.add_integrand("fuel_used",
    rate_name="fuel_flow", val=1.0, units="kg")

# Calculate weight by subtracting fuel used from MTOW
# Note: fuel used is accumulated over all mission phases,
# therefore fuel_used here represents the total fuel
# used since the first mission phase
self.add_subsystem(
    "weight",
    oc.AddSubtractComp(
        output_name="weight",
        input_names=["ac|weights|MTOW", "fuel_used"],
        units="kg",
        vec_size=(1, nn),
        scaling_factors=[1, -1],),
    promotes_inputs=["*"],
    promotes_outputs=["weight"],)
# Calculate total fuel used in this mission segment
self.add_subsystem(
```

```
        "seg_fuel_used",
        om.ExecComp(
            ["seg_fuel_used=sum(fuel_used)"],
            seg_fuel_used={"val": 1.0, "units": "kg"},
            fuel_used={"val": np.ones((nn,)),
                "units": "kg"},),
        promotes_inputs=["*"],
        promotes_outputs=["*"],)
    # code ends
```